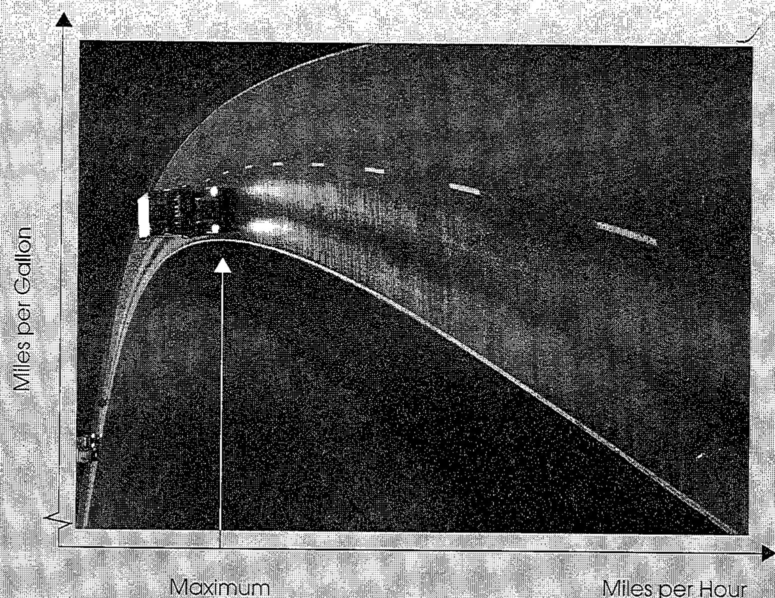
SAS LABORATORY MANUAL

to Accompany

REGRESSION ANALYSIS

Concepts and Applications

3 1/2" DOS disk enclosed



Franklin A. Graybill

Hariharan K. Iyer

Contents

Preface vii

Chapter 1 Review of Basic Statistical Concepts and Matrices 1

Chapter 2 Regression and Prediction 39

Chapter 3 Straight Line Regression 49

Chapter 4 Multiple Linear Regression 77

Chapter 5 Diagnostic Procedures 107

Chapter 6 Applications of Regression I 115

Chapter 7 Applications of Regression II 140

Chapter 8 Alternate Assumptions for Regression 160

Chapter 9 Nonlinear Regression 167

Answers and Solutions 174

Index 291

Chapter 1

Review of Basic Statistical Concepts and Matrices

1.1 Overview

This laboratory manual explains how to use the statistical package SAS to perform calculations for each procedure discussed in the book *Regression Analysis: Concepts and Applications*, which we refer to as the textbook.

General comments

The sections in this manual correspond to sections in the textbook. For example, Section 4.6 in this manual corresponds to Section 4.6 in the textbook, etc. Whenever we refer to a chapter, a section, an equation number, a table, a figure, an exhibit, or a box, these references are to the corresponding chapter, section, equation number, etc., in the textbook. Tables that do not appear in the textbook, but appear in this laboratory manual, are referred to as Table A, Table B, etc. Equation numbers, problems, examples, etc., that begin with the letter S , refer to this manual only. For instance, Problem S2.1.3 refers to Problem 3 in Section 1 in Chapter 2 of this SAS laboratory manual.

What Is SAS?

SAS is a very powerful, general purpose, comprehensive statistical computing package that can perform a wide variety of statistical data analyses and produce many types of plots. SAS may be thought of as a programming language that is especially suited for statistical calculations. Depending on the particular computer system you are using, you can carry out statistical calculations either by typing in appropriate SAS commands, or by choosing the appropriate menu item using either a mouse or the cursor keys. Recent versions of SAS allow you to use an extensive Windows system.

For our discussion, we assume that you are working on a personal computer that has a hard disk drive, usually called the `C` drive, and at least one floppy disk drive, say drive `B`. Then the data disk that accompanies this manual should be inserted into drive `B`. We assume that the subdirectory where the SAS system resides is specified in the path statement in your **autoexec.bat** file. This will enable you to run SAS from any subdirectory you wish. If you have no prior experience with personal computers, you should seek help from your laboratory instructor.

Invoking and exiting SAS

We assume, for the sake of our discussion, that your current working directory is `C:\`, i.e., the *root* directory on the `C` drive, although you may run SAS from any subdirectory you wish. If your current working directory is `C:\`, the computer will display the prompt `C:\` or `C:\>` or something similar. On most computer systems you can start SAS by typing the word `sas` following the prompt `C:\` and pressing the `Enter` key. Try it! If your working directory is a subdirectory, say `C:\work>`, and you want to enter SAS from this subdirectory, then type `sas` following the prompt and press `Enter`. If you have problems at this stage, consult your laboratory instructor.

When you enter the SAS system, the screen on your monitor will typically be split into three sections, called *display manager windows*. If you have a color monitor, each window will be a different color. The windows appear something like the illustration in Figure S1.1.1. The actual positioning of the display manager windows on the monitor may vary from one system to another. These three windows are labeled as follows.

(1) The top window is labeled `OUTPUT`

(2) The middle window is labeled `LOG`

(3) The bottom window is labeled `PROGRAM EDITOR`

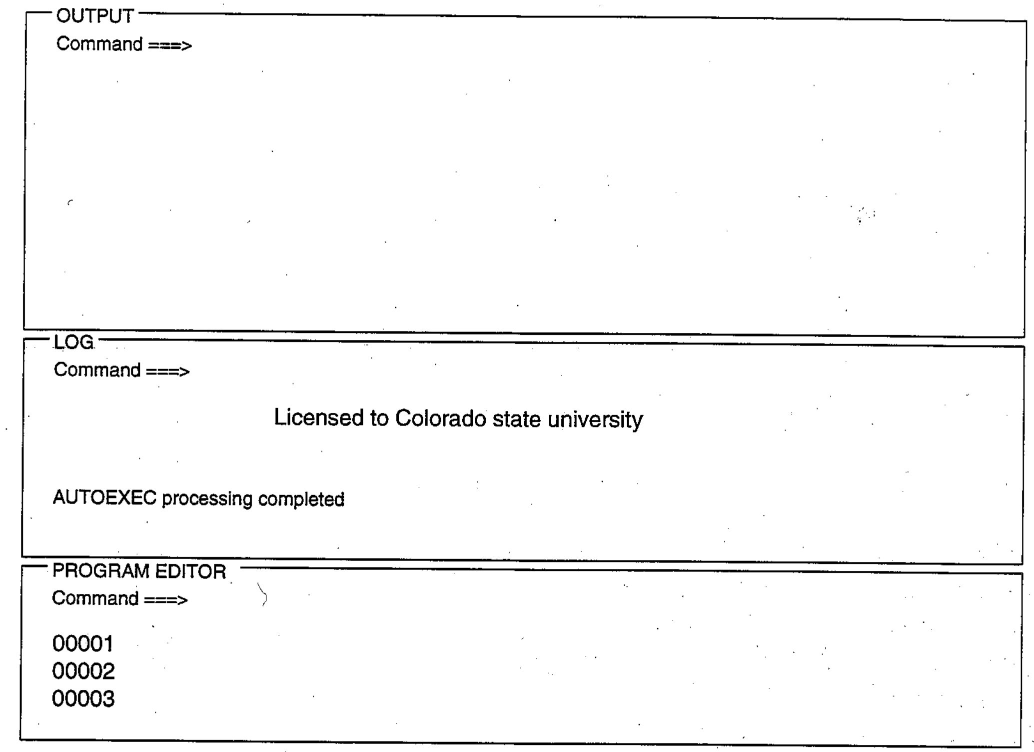


Figure S1.1.1

As a general rule, the output of computations will appear in the `OUTPUT` window. Various messages about your commands and data will appear in the `LOG` window. Error messages, if there are any, are generally given there. The `PROGRAM EDITOR` window is where you will input SAS program commands, enter data, etc. On the first line of each window is the word `Command`, and on this line you will sometimes enter commands for reading in macro files, writing results to files on the disk, exiting SAS, etc. We discuss this later. The numbers `00001`, `00002`, `00003` etc., appear in the `PROGRAM EDITOR` window under the word `Command`, and on these lines you will input SAS commands and enter data.

Your keyboard has a set of ten special keys, called *function keys*, marked `F1` to `F10` (some keyboards have more than 10 *function keys*). Some of these keys have special uses in SAS. For example, to move the cursor from one window to another, press the key F5 several times. Try it! When using any window, you may want it to fill the entire monitor screen (this is called zooming), and you can do this by pressing the key F7. Try it! In the PROGRAM EDITOR window you can toggle back and forth between the Command line and the first numbered line by pressing the Enter key and then pressing the Home key. Try this! To exit SAS you bring the cursor to the Command line of any window, type endsas (or type bye), and press Enter .

Reading Data into SAS

Now that SAS has been invoked, you are ready to get your data into a form that can be read and used by the SAS system. This is done by *creating a* SAS *dataset*. We describe two methods for doing this.

- Enter data via the computer keyboard.
- Read data from a file on the data disk that accompanies this manual.

Creating a Dataset by Entering Data via the Keyboard

Suppose you want to perform calculations using the data in Table A, which consists of five observations on the two variables Y and X.

Table A

Y	X
1.2	6.0
1.8	6.3
2.8	5.8
2.7	5.7
3.5	6.4

This is a very small dataset that we use for illustration, but the commands are the same whether the dataset consists of 1,000 observations on 10 variables, 526 observations on 32 variables, and so on. The process of creating a dataset in SAS is called a SAS Data Step.

To illustrate, we explain the commands to create a dataset containing the data in Table A using the keyboard to enter the data. First you must select a *valid* name for the dataset. A name is *valid* if it consists of a combination of no more than eight letters and numbers, the first of which must be a letter (or an *underscore* character _). For example, st and wxyz1238 are valid names for SAS datasets, but 1238wxyz and s t are not (the name cannot contain blank spaces). It may be helpful to select a name that corresponds to the origin of the data. For example, if the data values are baseball scores you might select the name baseball ; or you might select the name income if the data values are annual incomes of high school teachers. For the dataset in Table A it is natural to select the name tableA .

Next you must select a *valid* name for each variable in the dataset. We give the names Y and X to the variables in Table A. Any name can be given to a variable as long as it satisfies the same conditions as those given above for naming a dataset. As an example, suppose the dataset contains values for three variables. The variables could be named X1, X2, X3 , or they could be named age, weight, height , etc.

Invoke SAS and fill your monitor screen with the PROGRAM EDITOR window where you will see the numbered lines 00001 , 00002 , 00003 , etc. Statements in the following command are to be entered on these lines. Press Enter and the cursor will move to line 00001 , where you will input the first statement.

COMMAND FOR ENTERING DATA VIA THE KEYBOARD AND CREATING A DATASET

```
00001 data tableA;
00002 input Y X;
00003 cards;
00004 1.2 6.0
00005 1.8 6.3
00006 2.8 5.8
00007 2.7 5.7
00008 3.5 6.4
00009 ;
00010 run;
```

We comment briefly on these commands.

(1) The commands are used to put the data from Table A into a dataset which we have named tableA.

(2) The word data in line 00001 is a SAS statement that instructs SAS to create a dataset, and the word tableA tells SAS that you have chosen tableA for the

name of the dataset. Rather than `tableA` , you can use any *valid* name.

(3) The statement in line 00002 is `input Y X;` , and this tells SAS to expect two variables (since two names, `Y X` , are given), to name the first variable `Y` , and to name the second variable `X` . You must give every variable a *valid* name.

(4) The next statement is `cards;` , and this tells SAS that data are to follow. The data are entered by rows with at least one space between any two observations. For example, the two numbers in the first row of Table A are entered as `1.2 6.0` and not as `1.26.0` .

(5) After all the data have been entered, type a semicolon `;` on the line following the last data item. This tells SAS that there are no more data to be read.

(6) The final statement is `run;` , which tells SAS that all the statements for this block of the program have been entered and the commands can now be executed.

It is important to note that each line of a logical SAS statement ends with a semicolon, but the data lines have no punctuation marks. Also, two or more statements can be typed on a line if they are separated by semicolons. When you enter commands in SAS, it makes no difference whether you use uppercase letters, lowercase letters, or a mixture. **To execute a set of commands, press the function key F10 .**

After the commands have been entered correctly and the function key F10 has been pressed, the program will execute. The LOG window will contain information about the execution of the program, or errors in the program if it did not execute. You can now use SAS commands to process the dataset just created. In any SAS session you can create many different datasets and process any one or more of them.

Use the command below to print the dataset just entered so you can examine it for data entry errors. The results will appear in the OUTPUT window. Unless we state otherwise, all commands are entered on the numbered lines 00001 , 00002 , 00003 , etc., in the PROGRAM EDITOR window, but for simplicity we will often omit these numbers.

PRINT COMMAND

```
proc print data=tableA;
run;
```

This command is a SAS procedure command (only the first four letters, `proc` , are used) and instructs SAS to print the dataset `tableA` . If the command is


```
proc print data=income;
run;
```

this instructs SAS to print the dataset `income` . Of course, the dataset `income` must have been created in the current SAS session.

When you press the function key F10 , the three windows appear while the program is executing. Some information will appear in the LOG window, so go there by pressing the key F5 twice. Then press the key F7 to fill the screen. If there is more than one page, use the PgUp (page up) and PgDn (page down) keys to scroll through the pages. You can scroll down one line at a time by pressing the Enter key.

Since the data are displayed in the OUTPUT window, press the F5 key to move to that window. There you will see your data, the data in Table A. You should check it carefully to be sure there are no data entry errors. The SAS response in the OUTPUT window is

```
--------------------------------------------------------------------
                    SAS  0:00 Saturday, January 1, 1994    1

          OBS      Y      X

           1      1.2    6.0
           2      1.8    6.3
           3      2.8    5.8
           4      2.7    5.7
           5      3.5    6.4
--------------------------------------------------------------------
```

The first line of the output gives the date that the results were printed (of course your output will have a different date than what is shown above) and the number of the page, which is 1. Sometimes the output will require several pages and it may be helpful to have them numbered. However, we will henceforth not explicitly display this first line when listing SAS outputs in this manual.

In summary, the command `proc print data=tableA;` asks SAS to print the data in the dataset `tableA`. The data are printed in the OUTPUT window, and they appear in columns labeled by the corresponding variable names, with an additional column (the first one) labeled OBS (observations). This column can be quite useful for locating specific observations in large datasets. For example, the value of the variable Y for observation 3 (i.e., OBS 3) is 2.8.

The dataset we have just created is a **temporary SAS dataset and will be erased when you exit the SAS system.** Hence you may want to save this dataset so you can use it during another SAS session. If you don't save it, you will not only have to re-enter the data into the computer (which is a huge task if there are several observations and many variables), but you will also have to print and examine them to be sure there are no data entry errors. Later we show you how to save a dataset so you can use it in a future SAS session without having to enter it again via the keyboard.

As stated, entering a large dataset via the keyboard can take a significant amount of time and effort, but this is generally not necessary for working the problems in the textbook or this laboratory manual because all of the datasets used are stored in files on the disk (we refer to it as the data disk) that accompanies this manual. We now show you how to create a dataset, not by entering it via the keyboard, but rather by transferring (reading) data from a file on the data disk.

Creating a Dataset by Reading Data from a File

For convenience each set of data that appears in the textbook is stored in two formats on the data disk that accompanies this manual. The first format is an ASCII (American Standard Code for Information Interchange) data file that contains data. Most statistical software packages are equipped to read data from ASCII files. The names for ASCII data files that are on the data disk have the extension **dat**. The second format is a SAS data file that contains data and additional information such as the names of the variables in the file and the number of observations, and this file can be used only with the SAS computing system, and only on the same type of computer that is used to create this SAS data file. The names for SAS data files that are on the data disk have the extension **ssd**, and these were created for use with personal computers running under DOS or WINDOWS. Thus the file name **table161.dat** refers to an ASCII file that contains data, whereas the file name **table161.ssd** contains the same dataset (along with names of variables and other information) stored as a SAS data file.

Creating a Temporary Dataset from an ASCII File

We now show you how to read in an ASCII data file, name the variables in the file, and create a temporary SAS dataset. For illustration we use the ASCII data file **table161.dat** on the data disk, which we assume has been inserted in drive `B`. This file contains data for a single variable, which we wish to name mpg. The command is

COMMAND TO READ AN ASCII FILE FROM THE DATA DISK

```
data table161;
infile 'b:\table161.dat';
input mpg;
run;
```

In the preceding command, the first statement

```
data table161;
```

informs the SAS system that a temporary dataset is to be created, and it is to be named `table161` (any *valid* name can be used). The next statement

```
infile 'b:\table161.dat';
```

tells the system that you want to bring in the file **table161.dat**, which is in directory `B:\` . The next statement,

```
input mpg;
```

informs the system that there is one column of numbers in the file **table161.dat** and it is to be named `mpg` . You can use any *valid* name for the variable. This command is similar to the command to create a dataset by entering the data via the keyboard, except that the `cards;` statement followed by the data, is replaced by the `infile` statement. Press the `F10` key to execute the commands. You have now created a temporary dataset by reading the data from the ASCII data file **table161.dat** on the data disk.

In the same manner, you can read in any ASCII data file that is on the data disk by replacing `table161.dat` in the preceding command with the name of the file you wish to read, and replacing the statement

```
input mpg;
```

with the statement

```
input name1 name2 ...  namek;
```

where the ASCII data file in question consists of k columns of numbers (corresponding to k variables), and you wish to name these variables `name1`, `name2`, ..., etc. Of course, the chosen names must be valid names in SAS.

For each command we describe, you should invoke SAS, type the command statements, and press the F10 key to execute them. In future we sometimes omit these instructions. You should try out each command discussed, not just read about it.

Instructions for Using a SAS Data File on the Data Disk

As stated earlier, an ASCII data file on the data disk (one with the extension **dat**) contains only the data, but a SAS data file (one with the extension **ssd**) contains the data, the name and number of variables, and other information that may be useful for examining the contents of a dataset without printing the entire dataset.

To use SAS to process data that are in a SAS data file on the data disk, you do not need to create a temporary dataset, but *you can give SAS the name of the directory where the SAS data files are located, and use these files directly.* You can think of the data disk as a library that contains SAS data files (files with extension **ssd**), and give SAS the location of these files with a `libname` (`library name`) statement as follows.

LIBNAME COMMAND

```
libname my 'b:\';
run;
```

SAS requires you to give a *nickname* for the directory where the SAS data files are located. We have chosen the nickname `my` to represent the directory `b:\`, but any name can be used as long as it is a combination of no more than eight letters and numbers, the first of which must be a letter. To execute the preceding command, press F10. You can now use SAS `proc` statements to process data in any SAS data file on the data disk. For example, if you want to examine the contents of the SAS data file **table161.ssd** on the data disk without actually printing out the data, use the following command.

COMMAND TO EXAMINE THE CONTENTS OF A SAS DATA FILE

```
libname my 'b:\';
proc contents data=my.table161;
run;
```

The SAS response is

```
--------------------------------------------------------------------------------
                              CONTENTS PROCEDURE

Data Set Name:  MY.TABLE161                       Type:
Observations:   10                                Record Len: 12
Variables:      1
Label:
          -----Alphabetic List of Variables and Attributes-----

#  Variable  Type  Len  Pos  Label
1  MPG       Num     8    4
--------------------------------------------------------------------------------
```

From this you can see that the SAS data file **table161.ssd** contains ten observations of one variable labeled `MPG`. To print the observations in this file, use the following command.

COMMAND TO PRINT A SAS DATA FILE

```
libname my 'b:\';
proc print data=my.table161;
run;
```

The first statement declares the nickname (`my`) of the directory (`B:\`) where SAS data files are stored. This statement needs to be given only once during a SAS session, but it must be given before the prefix `my` is used in any command. The second statement tells SAS to print the data that are in the file **table161.ssd** in the directory my which is the nickname for the directory b:\ . The preceding statements provide a convenient way to use SAS proc (procedure) commands to process data in SAS data files without creating a temporary dataset.

The output from the preceding print command is

```
-----------------------------------------------------------------------
                         OBS      MPG

                           1     25.72
                           2     25.24
                           3     25.19
                           4     25.88
                           5     26.42
                           6     24.48
                           7     25.11
                           8     24.29
                           9     25.06
                          10     25.63
-----------------------------------------------------------------------
```

Thus, in a SAS procedure statement (a statement to do computing, printing, plotting, etc.), just give the appropriate proc command followed by the instruction data=my.filename; , which tells SAS the location (my) and the name (filename) of the SAS data file you want to use. Try this by printing several SAS data files that are on the data disk!

Next we describe a command for computing various summary statistics such as the minimum, the maximum, the mean, the standard deviation, the variance, the median, etc., for a given dataset. The command is

```
proc univariate;
```

as given below for the data in the SAS data file **table161.ssd**.

PROC UNIVARIATE COMMAND

```
libname my 'b:\';
proc univariate data=my.table161;
run;
```

The output from this command is

```
---------------------------------------------------------------------------
                         UNIVARIATE PROCEDURE

Variable=MPG

                               Moments

              N              10  Sum Wgts        10
              Mean       25.302  Sum         253.02
              Std Dev  0.639267  Variance  0.408662
              Skewness  0.04471  Kurtosis  -0.12279
              USS       6405.59  CSS        3.67796
              CV       2.526547  Std Mean  0.202154
              T:Mean=0  125.162  Prob>|T|    0.0001
              Sgn Rank     27.5  Prob>|S|    0.0020
              Num ^= 0       10

                         UNIVARIATE PROCEDURE

Variable=MPG

                           Quantiles(Def=5)

               100% Max      26.42        99%     26.42
                75% Q3       25.72        95%     26.42
                50% Med     25.215        90%     26.15
                25% Q1       25.06        10%    24.385
                 0% Min      24.29         5%     24.29
                                           1%     24.29
```

```
                 Range          2.13
                 Q3-Q1          0.66
                 Mode          24.29

                       UNIVARIATE PROCEDURE

Variable=MPG

                             Extremes

                   Lowest    Obs      Highest    Obs
                    24.29(     8)      25.24(      2)
                    24.48(     6)      25.63(     10)
                    25.06(     9)      25.72(      1)
                    25.11(     7)      25.88(      4)
                    25.19(     3)      26.42(      5)
---------------------------------------------------------------------------
```

The proc univariate command results in a three part output for each variable in a dataset. These are labeled Moments , Quantiles , and Extremes , respectively. The quantities which are of principal interest to us at the present time are listed under the heading Moments . They are as follows.

(1) N is the number of observations.

(2) Mean is the mean of the observations.

(3) Std Dev is the standard deviation of the observations. This is computed by using the formula given in (1.6.2) in the textbook. This is appropriate when working with sample data. In particular, this is the appropriate calculation for the present example. However, when working with population data, the correct formula to calculate the standard deviation is given in (1.4.3) in the textbook. SAS will use this formula if requested to do so. This can be done by using the option vardef = n in the proc univariate statement. We give an example of this later. Read the SAS/PROCEDURES guide for details.

(4) USS is the uncorrected sum of squares of the observations, viz., $\sum y_i^2$ (here y_i represents the value of mpg for car i).

(5) Sum is the sum of the observations.

(6) Variance is the variance of the observations. SAS calculates this by squaring the sample standard deviation as given in (1.6.2) in the textbook. When working with population data you should use the formula in (1.4.4). You can request SAS to use this formula by specifying the option vardef = n as part of the proc univariate statement.

(7) Std Mean is the standard error of the mean of the observations.

Other quantities in the preceding output are discussed as and when we need them.

If a dataset consists of several variables, the output from the proc univariate command would be several pages long. If you are not interested in all of the summary quantities listed in the output from the proc univariate command, but only in a selected subset of them, you can use the command proc means . This will compute

- N Obs, the number of observations
- N, the number of nonmissing observations
- Minimum, the minimum value of the observations
- Maximum, the maximum value of the observations
- Mean, the mean of the observations
- Std Dev, the standard deviation of the observations

To illustrate, we use the SAS data file **table161.ssd**.

PROC MEANS COMMAND

```
libname my 'b:\';
proc means data=my.table161;
run;
```

The response in the OUTPUT window is

```
----------------------------------------------------------------------
     Analysis Variable : MPG

     N Obs   N      Minimum       Maximum          Mean       Std Dev
     -----------------------------------------------------------------
        10  10   24.2900000    26.4200000    25.3020000     0.6392669
     -----------------------------------------------------------------
----------------------------------------------------------------------
```

If you are interested in only the mean and the standard deviation, use the command

```
proc means data=my.table161 mean std;
```

The SAS response is

```
----------------------------------------------------------------------
                Analysis Variable : MPG

                N Obs          Mean       Std Dev
                ---------------------------------
                   10    25.3020000     0.6392669
                ---------------------------------
----------------------------------------------------------------------
```

Note: As mentioned earlier, the command `libname my 'b:\';` needs to be given only once in each SAS session, but it must be given before the prefix `my` is used in any command. Even if we do not explicitly list this command when explaining other commands, you should make sure that this command has already been issued. Furthermore, note that SAS uses the formulas given in (1.6.1) and (1.6.2) to calculate the mean and the standard deviation, which are the appropriate formulas when working with sample data. For population data, the correct formulas are in (1.4.2) and (1.4.3), respectively. You can instruct SAS to use these formulas by specifying the option `vardef=n` in the `proc means` statement. We give an example of this later.

Example S1.1.1

To illustrate the commands we have discussed, we use the data in the SAS data file gpa.ssd. This dataset contains five variables. We give the commands to examine

the contents of this file, to print the data, and to compute summary statistics. The commands and the output follow.

SOME COMMANDS FOR EXAMINING AND SUMMARIZING DATA IN A SAS DATA FILE

```
libname my 'b:\';
proc contents data=my.gpa;
proc print data=my.gpa;
proc means data=my.gpa;
run;
```

```
----------------------------------------------------------------------
                        CONTENTS PROCEDURE

Data Set Name:  MY.GPA                    Type:
Observations:   20                        Record Len: 44
Variables:      5
Label:

            -----Alphabetic List of Variables and Attributes-----

#  Variable  Type  Len  Pos  Label
1  GPA       Num     8    4
5  HSENGL    Num     8   36
4  HSMATH    Num     8   28
2  SATMATH   Num     8   12
3  SATVERB   Num     8   20

         OBS    GPA   SATMATH   SATVERB   HSMATH   HSENGL

           1   1.97      321       247      2.30     2.63
           2   2.74      718       436      3.80     3.57
           3   2.19      358       578      2.98     2.57
           4   2.60      403       447      3.58     2.21
           5   2.98      640       563      3.38     3.48
           6   1.65      237       342      1.48     2.14
           7   1.89      270       472      1.67     2.64
           8   2.38      418       356      3.73     2.52
           9   2.66      443       327      3.09     3.20
          10   1.96      359       385      1.54     3.46
          11   3.14      669       664      3.21     3.37
```

```
12   1.96   409   518   2.77   2.60
13   2.20   582   364   1.47   2.90
14   3.90   750   632   3.14   3.49
15   2.02   451   435   1.54   3.20
16   3.61   645   704   3.50   3.74
17   3.07   791   341   3.20   2.93
18   2.63   521   483   3.59   3.32
19   3.11   594   665   3.42   2.70
20   3.20   653   606   3.69   3.52

N Obs  Variable  N      Minimum        Maximum          Mean        Std Dev
---------------------------------------------------------------------------
   20  GPA       20      1.6500000      3.9000000      2.5930000     0.6217894
       SATMATH   20    237.0000000    791.0000000    511.6000000   166.2003863
       SATVERB   20    247.0000000    704.0000000    478.2500000   132.6165327
       HSMATH    20      1.4700000      3.8000000      2.8540000     0.8527503
       HSENGL    20      2.1400000      3.7400000      3.0095000     0.4841104
---------------------------------------------------------------------------

-----------------------------------------------------------------------------
```

Using the five statements in the preceding command you can obtain a great deal of information about the data in the SAS data file **gpa.ssd**. Try these commands on other SAS data files on the data disk.

Problems

For all problems, give the appropriate SAS commands and the answer when required. Problems S1.1.1 to S1.1.3 refer to the following data.

Y	X	Z
1.5	600	34.5
1.9	590	43.9
1.2	710	30.3
2.1	560	31.7
1.6	610	42.1
1.7	700	39.0

S1.1.1 Enter the data via the keyboard, create a temporary dataset, and name it `prob111`. Name the variables `Y`, `X`, and `Z`, respectively.

S1.1.2 Print the dataset in Problem S1.1.1.

S1.1.3 Use a suitable SAS command to find the sample mean of X, the sample mean of Y, and the sample mean of Z.

S1.1.4 Use appropriate SAS commands to examine the contents of the SAS data file **table164.ssd** on the data disk. How many variables are there?

S1.1.5 In Problem S1.1.4 find the mean and the standard deviation of the sample observations.

S1.1.6 The data disk contains the SAS data file **agebp.ssd**. Use SAS commands to examine its contents without printing the data.

S1.1.7 In Problem S1.1.6, find the maximum of each variable.

S1.1.8 In Problem S1.1.6, find the mean and the standard deviation of the sample values of each variable.

S1.1.9 In Problem S1.1.6, print the data.

S1.1.10 The data disk contains the SAS data file **chol.ssd**. Use appropriate SAS commands to examine what is in this file. How many variables are there? How many observations? What are the names of the variables?

S1.1.11 In Problem S1.1.10, print the data.

S1.1.12 In Problem S1.1.10, find the mean and the standard deviation of the sample values of each variable in the dataset.

S1.1.13 In Problem S1.1.10, find the minimum and the maximum values of each variable.

1.2 Basic Ingredients for Statistical Inference

There are no calculations in this section that require SAS.

1.3 Populations

There are no calculations in this section that require SAS.

1.4 Model

There are no calculations in this section that require SAS.

1.5 Parameters (Summary Numbers)

There are no calculations in this section that require SAS.

1.6 Samples and Inference

In this section we introduce several SAS commands that can be used to compute quantities discussed in Section 1.6 in the textbook. First we show you how to perform some simple arithmetic calculations. Consider the data in Table C below.

Table C

X	Y	Z
12	2	32
21	4	16
31	1	35
52	5	25
37	3	27
35	6	24

We show you how to add, subtract, multiply, and divide any two columns of data (element by element), where the columns are variables in a dataset and contain the same number of elements. To illustrate, we first create a temporary dataset named `tableC` using the data in Table C. We name the variables in this dataset `X`, `Y`, and `Z`, respectively. The SAS statements for creating this dataset are as follows.

```
data tableC;
input X Y Z;
cards;
```



```
12 2 32
21 4 16
31 1 35
52 5 25
37 3 27
35 6 24
;
run;
```

Refer to Section 1.1 of this manual to review the SAS commands for creating temporary SAS datasets.

The following SAS statements illustrate the basic arithmetic operations available in SAS.

SAS COMMANDS TO ADD, SUBTRACT, MULTIPLY, AND DIVIDE COLUMNS OF DATA

```
data new;
set tableC;
U=X+4*Y;
V=3*X-Z;
W=(X+2*Y)/U;
keep U V W;
run;
proc print data=new;
run;
```

We explain each statement in the preceding command.

(1) The first statement, `data new;` , tells SAS to create a temporary dataset and name it `new`.

(2) The second statement, `set tableC;` , tells SAS that all of the data contained in the dataset `tableC`, created earlier, should be copied into the temporary dataset `new`. Thus the data values for the variables `X` , `Y` , and `Z` (which are in the dataset `tableC`), are copied into the dataset `new`. In addition, this dataset will also contain variables to be computed using `X` , `Y` , and `Z` . You might think of

the first two statements as telling SAS to create a temporary dataset called new using the variables in the dataset tableC plus possibly some other variables.

(3) The third statement, `U=X+4*Y;` , performs arithmetic operations on columns X, Y, Z and the result is a new variable named U which will be put in the temporary dataset new. Note that the symbol * is used for multiplication. The operations are performed element by element in each column. For example, the statement `U = X + 4*Y;` produces a column U where $U_i = X_i + 4Y_i$, etc.

(4) The fourth statement, `V=3*X-Z;` , performs another arithmetic operation on the columns of new and the result is a new variable named V, which will be put in the temporary dataset new.

(5) The fifth statement, `W=(X+2*Y)/U;` performs still another arithmetic operation on the columns X , Y , and Z of new, and in addition this arithmetic operation uses U , a variable just computed. The new variable is named W and it will be put in the temporary dataset new.

(6) The sixth statement, `keep U V W;` , tells SAS to keep only the variables U , V , and W in the dataset new. If you don't tell SAS which variables to keep, all variables will be kept in the dataset new, including X , Y , and Z , the variables that were copied from the original dataset tableC, into the dataset new, using the set command.

(7) The seventh statement is `run;` . When the F10 key is pressed, SAS will execute the preceding statements and create the temporary dataset new. As explained above, this dataset will contain the variables U , V , and W .

(8) The eighth statement, `proc print data=new;` , tells SAS to print the temporary dataset new just created.

(9) The ninth and final statement is a `run;` statement.

The output from the preceding set of commands is

```
---------------------------------------------------------------------------
                    OBS     U      V        W

                     1     20      4     0.80000
                     2     37     47     0.78378
                     3     35     58     0.94286
                     4     72    131     0.86111
                     5     49     84     0.87755
                     6     59     81     0.79661

---------------------------------------------------------------------------
```


If the divisor is zero in any computation, a message in the LOG window will tell you that an error has been committed. You should perform some of the arithmetic operations by hand to help you understand the commands and the results just discussed.

Computing Confidence Intervals and Test Statistics

For a one-variable population $\{Y\}$, there is no simple built-in command in SAS for computing general confidence intervals or tests for μ_Y or σ_Y. You can use the proc means command for computing $\hat{\mu}_Y$, $\hat{\sigma}_Y$, and $SE(\hat{\mu}_Y)$, which are the ingredients used in Table 1.6.2 for computing confidence intervals for μ_Y and σ_Y, and in Boxes 1.6.1 and 1.6.2 for performing tests about μ_Y and σ_Y. For illustration we use Example 1.6.1. The data for this example are in Table 1.6.1 and in the SAS data file **table161.ssd** on the data disk. The command is

COMMAND TO OBTAIN THE ESTIMATE OF THE MEAN, THE ESTIMATE OF THE STANDARD DEVIATION, AND THE STANDARD ERROR OF THE MEAN

```
libname my 'b:\';
proc means data=my.table161 mean std stderr;
run;
```

The output from this command is

```
--------------------------------------------------------------------
            Analysis Variable : MPG


            N Obs        Mean         Std Dev      Std Error
            ---------------------------------------------------
               10     25.3020000     0.6392669     0.2021540
            ---------------------------------------------------

--------------------------------------------------------------------
```

From this we get $\hat{\mu}_Y = 25.302$, $\hat{\sigma}_Y = 0.6392669$, and $SE(\hat{\mu}_Y) = 0.2021540$.

Problems

Problems S1.6.1–S1.6.6 refer to the dataset in Table 1.6.4, which is a simple random sample of size 30 from a Gaussian population with mean μ_Y and standard deviation σ_Y. This dataset is also stored in the files **table164.dat** and **table164.ssd** on the data disk. For each problem, give the appropriate SAS commands in addition to the answer.

S1.6.1 Print the contents of the file. How many variables are in the file and what are the names of the variables? Create a temporary SAS dataset from this file and give it the name `tab164`.

S1.6.2 Find the estimate of μ_Y.

S1.6.3 Find a 90% upper confidence bound for μ_Y.

S1.6.4 Find t_C, the computed t-value for testing

$$\text{NH: } \mu_Y = 4.5 \text{ against AH: } \mu_Y \neq 4.5$$

S1.6.5 In Problem S1.6.4 find the P-value for the test.

S1.6.6 Find t_C, the computed t-value for testing

$$\text{NH: } \mu_Y \leq 5.0 \text{ against AH: } \mu_Y > 5.0$$

1.7 Functional Notation

There are no calculations in this section that require SAS.

1.8 Matrices and Vectors

While SAS is primarily a system for data analysis, SAS/IML is a module in SAS that can be used for matrix operations. IML stands for **I**nteractive **M**atrix **L**anguage. This module is an extremely powerful tool that can be used for all sorts of calculations involving matrices as well as for statistical simulation studies. For our purposes we will need only the simplest of the matrix commands available in SAS/IML.

In this section we describe some of the commands in SAS/IML for matrix calculations that are useful for performing many of the computations in the textbook. First you must invoke SAS. Once you are in the SAS system, give the following command to get into IML, where, as usual, the command statements should be typed on the lines numbered `00001` , `00002` , etc., in the `PROGRAM EDITOR` window.

INVOKING IML

```
proc iml;
reset nolog;
```

The first statement in the preceding command invokes IML, and the second statement routes all printed output to the `OUTPUT` window. If you don't include this statement, the output will be intertwined with messages in the `LOG` window. Press `F10` and IML is ready for use to process matrices. To exit IML but remain in SAS, enter the following command at a numbered statement line, then press `F10`.

COMMAND TO EXIT IML BUT REMAIN IN SAS

```
quit;
```

We explain three ways to enter matrices into SAS so that the SAS/IML system can process them.

- Entering matrices via the keyboard
- Loading matrices from a file where they were stored in a previous SAS session.
- Creating matrices from data in an ASCII or SAS file on the data disk.

Entering matrices via the keyboard

Suppose you wish to enter (via the keyboard) the two matrices A and B given by

$$A = \begin{bmatrix} 12 & 32 & 31 & 27 \\ 38 & 54 & 19 & 10 \\ 65 & 76 & 23 & 24 \\ 24 & 12 & 26 & 52 \end{bmatrix} \qquad B = \begin{bmatrix} 24 & 17 & 27 & 32 \\ 39 & 13 & 15 & 37 \\ 16 & 42 & 26 & 33 \\ 36 & 37 & 23 & 41 \end{bmatrix}$$

The command to enter the 4 × 4 matrix A into the computer via the keyboard is given below. The commands are entered in the PROGRAM EDITOR window on the numbered lines 00001, 00002, ... , etc. Make sure you have invoked IML before you enter the following commands.

COMMAND TO ENTER THE MATRIX A INTO THE COMPUTER VIA THE KEYBOARD

```
A={12 32 31 27,
38 54 19 10,
65 76 23 24,
24 12 26 52};
```

Note that the elements of the matrix are enclosed in braces { } and are entered by rows, with a comma at the end of each row except the last. After the last row is entered, type a brace and a semicolon. Note also that there is a space between any two elements. Press enter after each line. Several rows of numbers can be entered on a single line provided that the rows are separated by commas. For instance, the above command could be typed in as

```
A={12 32 31 27, 38 54 19 10, 65 76 23 24, 24 12 26 52};
```

Next we enter the 4 × 4 matrix B into the computer via the keyboard.

COMMAND TO ENTER THE MATRIX B INTO THE COMPUTER VIA THE KEYBOARD

```
B={24 17 27 32,
39 13 15 37,
16 42 26 33,
36 37 23 41};
```

As usual, press the F10 key to execute the statements. Next we give the command to print the matrices so you can view them and check them to be sure you have entered them correctly.

COMMAND TO PRINT MATRICES

```
print A B;
```


You should notice three things here.

(1) The print statement in SAS/IML is not proc print , but merely print followed by the name of the matrices you want printed.

(2) A semicolon is required at the end of each SAS/IML statement.

(3) There is no run; statement. To execute a set of SAS/IML statements, press the F10 key.

The SAS response to the preceding print command appears in the OUTPUT window and is

```
----------------------------------------------------------------
                A
               12        32        31        27
               38        54        19        10
               65        76        23        24
               24        12        26        52

                B
               24        17        27        32
               39        13        15        37
               16        42        26        33
               36        37        23        41
----------------------------------------------------------------
```

If you want to save the matrices A and B (which were entered via the keyboard) so you can use them in a future SAS session, the command is

STORING MATRICES

```
libname save 'c:\work';
reset storage='save.matrix';
store A B;
```

The first statement libname save 'c:\work'; gives the nickname save to the directory where we wish to store matrices. This directory is c:\work in the present situation. The second statement, `reset storage='save.matrix';` , states that the nickname of the directory is `save` and the filename where matrices will be stored is `matrix`. Upon execution of this command, matrices A and B will be stored in the file **matrix.sct** in the directory `c:\work` (SAS adds the extension `sct`). If you want to store the matrices in another directory, say the directory `c:\workload\tuesday` , then the `libname` statement is

```
libname save 'c:\workload\tuesday';
```

Loading Matrices from a File Where They Were Stored in a Previous SAS/IML Session

If you want to load the matrices that are stored in a file (perhaps during a previous SAS/IML session), you must know the name of the file and the directory where it is located. We assume that the matrices are in the storage file **matrix.sct** in the directory `c:\work` . To load them during a SAS/IML session, the command is

COMMAND TO LOAD MATRICES THAT ARE IN A STORAGE FILE

```
libname save 'c:\work';
reset storage='save.matrix';
load A B;
```

When you press the `F10` key, the matrices A and B are loaded and you can process them. To examine the contents of the storage file **matrix.sct**, which is in the directory `c:\work` , the command is

COMMAND TO EXAMINE THE CONTENTS OF A STORAGE FILE

```
libname save 'c:\work';
reset storage='save.matrix';
show storage;
```

Creating matrices from a data file

Often it is necessary to create a matrix using the observations in a SAS or ASCII

variables, `bp` and `age`, in the ASCII data file **agebp.dat**, we create a 20×2 matrix which we will name `q`. The command statements are (do not invoke IML before giving this command)

SAS COMMAND TO CREATE A MATRIX q FROM A SET OF OBSERVATIONS IN AN ASCII DATA FILE

```
data agebp;
infile 'b:\agebp.dat';
input bp age;
run;
proc print data=agebp;
run;

proc iml;
reset nolog;

use agebp;
read all into q;
print q;
```

The first group of (six) statements in the preceding command creates a temporary SAS data file called `agebp` from the ASCII file **agebp.dat**, and prints the data. These statements have been discussed in Section 1.1 of this manual. The next group of (two) statements invokes IML and directs the results of computations to the `OUTPUT` window, rather than the `LOG` window, as explained previously. The last group of (three) statements tells SAS to `use agebp`, the dataset just created, read all variables into columns of a matrix which is to be named `q` and, finally, print the matrix `q`. This matrix is now ready to be processed using SAS/IML commands. You should check the output and make sure that the matrix `q` does indeed consist of the columns of the dataset `agebp`.

Next we give several commands to demonstrate how to do simple matrix arithmetic using SAS/IML.

Matrix Arithmetic

To explain some of the matrix calculations that can be carried out in SAS/IML, we use the matrices A and B, which we created earlier and stored in the file **matrix.sct** in the directory c:\work . First load the matrices A and B into SAS/IML. Then use the following SAS statements.

COMMANDS TO PERFORM MATRIX ARITHMETIC

```
C=A+B;
D=A-B;
E=A*B;
F=A`;
G=inv(A);
```

The first statement adds A and B and puts the result in C. The second statement subtracts B from A and puts the result in D. The third statement multiplies A and B, with B on the right, and puts the result in E (note that the symbol * is used for multiplication). The fourth statement computes the transpose of A and puts the result in F. The symbol ` , which is the "open quote" symbol, not an apostrophe, is used to transpose a matrix. The fifth and last statement computes the inverse of A and puts the result in G.

If matrices are not of the proper size for a particular arithmetic operation, an error message will appear in the OUTPUT or LOG window. As usual, after each command press F10 to execute it. If you give the command

```
print A B C D E F G;
```

all matrices just created will be printed in the OUTPUT window. If you do not want to print all of the matrices, give the command print followed by the name of those matrices you want printed. For example, if you want to print only C and G, use the command print C G; . The SAS response to the command

```
print A B C D E F G;
```

is given below.

```
A
12    32    31    27
38    54    19    10
65    76    23    24
24    12    26    52
```



```
B
24    17    27    32
39    13    15    37
16    42    26    33
36    37    23    41

C
36    49    58    59
77    67    34    47
81   118    49    57
60    49    49    93

D
-12    15     4    -5
 -1    41     4   -27
 49    34    -3    -9
-12   -25     3    11

E
3004  2921  2231  3698
3682  2516  2560  4251
5756  3947  4045  6635
3332  3580  2700  4202

F
12    38    65    24
32    54    76    12
31    19    23    26
27    10    24    52

G
-0.162615 0.3695665 -0.211651 0.1110496
0.1508801 -0.369222 0.2298826 -0.113437
-0.169321 0.5534367 -0.344091 0.1402976
0.1248952 -0.362082 0.2166806 -0.075994
```

Problems

S1.8.1 Problems (a)–(m) refer to the matrix X and the vector y defined below. For each problem, give the appropriate SAS (or SAS/IML) command and the answer if requested.

$$X = \begin{bmatrix} 12 & 28 & 21 \\ 14 & 31 & 46 \\ 20 & 21 & 31 \\ 11 & 19 & 21 \\ 16 & 13 & 34 \\ 39 & 26 & 30 \\ 25 & 37 & 15 \end{bmatrix} \qquad y = \begin{bmatrix} 9 \\ 13 \\ 28 \\ 6 \\ 32 \\ 16 \\ 24 \end{bmatrix}$$

(a) Read the two matrices X and y into the computer via the keyboard, and print them to be sure there are no data entry errors.

(b) Compute X^T and X^TX.

(c) Compute X^Ty.

(d) Compute $(X^TX)^{-1}$.

(e) Compute $(X^TX)^{-1}X^Ty$.

(f) Compute y^Ty.

(g) Compute $y^T[I - X(X^TX)^{-1}X^T]y$, where I is the 7 by 7 identity matrix. *You can create the k by k identity matrix I in SAS/IML by using the command* `I = i(k);` *where k is any positive integer.*

(h) Compute E where $E = I - (\frac{1}{7})J$, where I is the 7 by 7 identity matrix, and J is a 7 by 7 matrix with each element equal to 1. *You can create an r by c matrix J whose elements are all equal to* g *by using the SAS/IML statement* `J=j(r,c,g);` .

(i) Show that $EE = E$.

(j) Show that $y^T[(\frac{1}{7})J]y = 7\bar{y}^2$.

(k) Show that $\bar{y} = (\frac{1}{7})\mathbf{1}^Ty$ where $\mathbf{1}$ is a 7 by 1 vector with each element equal to 1.

(l) Show that $y^TEy = \sum_{i=1}^{7}(y_i - \bar{y})^2 = SSY$.

(m) Show that $EJ = 0$.

S1.8.2 In Problem S1.8.1, store the two matrices X and y in the file **Xy.sct** in the root directory of drive `C`.

S1.8.3 After working Problem S1.8.2, exit SAS. Now invoke SAS and IML and load the matrices X and y of Problem S1.8.2 from the file **Xy.sct** in the root directory of drive `C`.

1.9 Multivariate Gaussian Populations

To examine large datasets, it is sometimes useful to plot them so you can study them visually. In this section we discuss SAS commands that can be used to plot histograms of single columns of data. We also explain the command to plot two-variable data. For illustrations, we use the data in the two SAS data files **bivgauss.ssd** and **bivngaus.ssd** on the data disk.

Histograms

SAS has commands to construct either vertical histograms or horizontal histograms, which are labeled `vbar charts` and `hbar charts` , respectively. The command to construct and display a vertical histogram of the data for the variable X_1 in the SAS data file **bivgauss.ssd** is given below.

VBAR CHART (VERTICAL HISTOGRAM) COMMAND

```
options center linesize=75 pagesize=35;
libname my 'b:\';
proc chart data=my.bivgauss;
vbar X1;
run;
```

SAS responds with:

```
---------------------------------------------------------------------------

                          FREQUENCY OF X1

FREQUENCY

    |                                          **
150 +                                          **
    |                                     **   **   **
    |                                     **   **   **
    |                                **   **   **   **
    |                                **   **   **   **
100 +                                **   **   **   **   **
    |                                **   **   **   **   **
    |                           **   **   **   **   **   **   **
    |                           **   **   **   **   **   **   **
    |                           **   **   **   **   **   **   **
 50 +                      **   **   **   **   **   **   **   **
    |                      **   **   **   **   **   **   **   **   **
    |                      **   **   **   **   **   **   **   **   **   **
    |                 **   **   **   **   **   **   **   **   **   **   **
    |       **   **   **   **   **   **   **   **   **   **   **   **   **   **
    --------------------------------------------------------------------------------
       -    -    -    -    -    -    -                                  1    1    1
       9    8    6    5    3    2    0    0    2    3    5    6    8    9    1    2    4
       .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .    .
       7    2    7    2    7    2    7    7    2    7    2    7    2    7    2    7    2
       5    5    5    5    5    5    5    5    5    5    5    5    5    5    5    5    5

                                    X1 MIDPOINT
---------------------------------------------------------------------------
```

The first statement is an `options` statement that tells SAS the length of the lines (75 characters) and the number of lines per page (35 lines) to use. It also asks SAS to `center` the output on the page. If the values for **linesize** and **pagesize** are not set, then SAS will use the default values for these. The size of the histogram displayed in the SAS output will depend on the values specified in the `options` statement. You should try different linesize and pagesize values to see how they affect the histogram that is displayed. To learn more about the `options` statement, consult the SAS reference manuals. The second statement is the usual **libname** statement. The third statement tells SAS to build a chart using the data in the SAS data file **bivgauss.ssd** which is stored in the directory `b:\` . The fourth statement tells SAS that the chart is to be a vertical one using the variable X1.

The `hbar` chart statement in SAS, besides constructing a horizontal histogram (hbar chart), gives you some additional information. The command and the resulting output are given below.

HBAR CHART (HORIZONTAL HISTOGRAM) COMMAND

```
proc chart data=my.bivgauss;
hbar X1;
run;
```

```
---------------------------------------------------------------------------

                          FREQUENCY OF X1

   X1                                                  CUM             CUM
MIDPOINT                                       FREQ   FREQ  PERCENT  PERCENT
           |
 -9.75     |*                                     3      3     0.30     0.30
 -8.25     |*                                     7     10     0.70     1.00
 -6.75     |***                                  14     24     1.40     2.40
 -5.25     |****                                 22     46     2.20     4.60
 -3.75     |**********                           49     95     4.90     9.50
 -2.25     |*****************                    83    178     8.30    17.80
 -0.75     |************************            118    296    11.80    29.60
  0.75     |****************************        141    437    14.10    43.70
  2.25     |********************************    163    600    16.30    60.00
  3.75     |****************************        141    741    14.10    74.10
  5.25     |*********************               104    845    10.40    84.50
  6.75     |****************                     80    925     8.00    92.50
  8.25     |*******                              36    961     3.60    96.10
  9.75     |******                               28    989     2.80    98.90
 11.25     |**                                    8    997     0.80    99.70
 12.75     |                                      2    999     0.20    99.90
 14.25     |                                      1   1000     0.10   100.00
           --------+-------+-------+-------+-
                  40      80     120     160

                         FREQUENCY
---------------------------------------------------------------------------
```

You should try different linesize and pagesize options to see how the shape of the horizontal histogram is affected by them. From these histograms you can see that the one-variable Gaussian population $\{X_1\}$ appears to be symmetric with a mean close to 2.25.

There are several options you can use with the `proc chart` command, and if you are interested, you should consult the SAS reference manuals.

From the horizontal chart (histogram) you can obtain a great deal of information about the data. For example, you can determine the frequency and the percent of the observations in the dataset that lie in each interval or class of the histogram. You can also obtain the cumulative frequencies and cumulative percents of the observations in the dataset that lie below the upper limit of any of the histogram intervals. The `MIDPOINT` of each interval is used to identify the interval.

Plotting One Variable Against Another

To illustrate the command for plotting one variable against another variable, we use the data from the SAS data file **bivngaus.ssd**. The command to plot X_2 against X_1 is given below.

COMMAND TO PLOT ONE VARIABLE AGAINST ANOTHER VARIABLE

```
options linesize=75 pagesize=35;
proc plot data=my.bivngaus;
plot X2*X1='*';
run;
```

The first statement specifies the `linesize` and the `pagesize` for the output. The second statement tells SAS that a two-dimensional scatter-plot is desired and that the data are to be read from the file **bivngaus.ssd** which is stored in a directory whose nickname is `my` . The second statement tells SAS to plot X_2 (the first variable in `X2*X1`) on the vertical axis and X_1 (the second variable in `X2*X1`) on the horizontal axis. If the names of the two variables in the dataset are (say) `height` and `weight` , the second statement would be `plot height*weight='*'` , which would plot *height* on the vertical axis and *weight* on the horizontal axis. The portion of this statement given by `='*';` instructs SAS to use the symbol `*` as the plotting symbol.

SAS responds with:

```
---------------------------------------------------------------------------

                     Plot of X2*X1.  Symbol used is '*'.

   40 +                              *  *
      |
      |                     *   * *    **
      |               *            ***  *
      |                 *     **  * **  **
      |                      * *  ****** *
   20 +              *      **************
      |                 *  ***************
      |              *   ** **************
   X2 |               *  * * *************
      |                  *****************
      |          *     *   ***************
    0 +           *          *    ********************* ** **
      |                                  **************** ** *
      |                                  ************** **  *
      |                                  *****************
      |                                   *************
      |                                   ********** ***
  -20 +                                   *********  ***
      |                                    *** ****            *
      |                                   *  **  **  *
      |                                   **     *
      |                                        *
      |
  -40 +
      ---+--------+--------+--------+--------+--------+--------+--------+--
        -15      -10      -5        0        5       10       15       20

                                        X1
NOTE: 737 obs hidden.
---------------------------------------------------------------------------
```

The statement `NOTE: 737 obs hidden` in the last line of the output means that 737 of the points to be plotted are so close to points that are already plotted that they cannot all be individually displayed. This is due to the limited resolution of the printing device. However, this plot resembles the plot in Figure 1.9.6 in the textbook. To obtain high resolution plots, you can use the command `proc gplot` in place of the command `proc plot` . SAS might ask you to type in the name of the graphics output device. Consult the SAS/GRAPH manuals for details.

If the second statement in the preceding command is `plot X2*X1;` instead of `plot X2*X1='*';` , SAS uses the letter `A` as the plotting symbol. If there are two (respectively, three, four, etc.) points so close together that distinct symbols `A` cannot be printed, then the letters `B` (respectively, `C`, `D` , etc.) are used. The symbol `B` stands for two overlapping points, `C` stands for three overlapping points, and so on. If you want another plotting symbol, say `o` , then replace the second statement in the preceding command with `plot X2*X1='o';` , etc.

To learn more about SAS commands for plotting, consult the SAS/GRAPH reference manuals.

Problems

Give the SAS commands required to answer each problem.

S1.9.1. Examine the contents of the file **bivgauss.ssd** and the file **bivngaus.ssd** stored on the data disk.

S1.9.2. For the data in the file **bivgauss.ssd**, obtain a vertical histogram for the variable X_2.

S1.9.3. For the data in the file **bivgauss.ssd**, obtain a horizontal histogram for the variable X_2.

S1.9.4. For the data in the file **bivngaus.ssd**, obtain vertical and horizontal histograms for the variable X_1.

S1.9.5. For the data in the file **bivngaus.ssd**, plot X_1 against X_2 using the symbol `+` as the plotting symbol.

S1.9.6. For the data in the file **bivgauss.ssd**, plot X_2 against X_1 using the symbol `o` as the plotting symbol.

S1.9.7. For the data in the file **bivgauss.ssd**, plot X_1 against X_2.

Chapter 2

Regression and Prediction

2.1 Overview

There are no calculations in this section that require SAS.

2.2 Prediction

There are no calculations in this section that require SAS.

2.3 Regression Analysis

In Chapter 1 of this manual we introduced some basic SAS commands and illustrated their use. Our discussion there was mainly about a one-variable dataset. In this section we introduce some SAS commands for processing datasets that contain more than one variable. We use Example 2.3.1 and Task 2.3.1 to illustrate the commands. These require the use of the data in Table D-1 in Appendix D, which are also stored in the ASCII data file **car.dat** and in the SAS data file **car.ssd**. As usual, we encourage you to work along on the computer, try out each command, and verify the results.

We begin by examining the contents of the SAS data file **car.ssd** using the command `proc contents` .

```
libname my 'b:\';
proc contents data=my.car;
run;
```

The SAS response in the OUTPUT window is

```
--------------------------------------------------------------------------

                        CONTENTS PROCEDURE

 Data Set Name:  MY.CAR                 Type:
 Observations:   1242                   Record Len: 36
 Variables:      4
 Label:

          -----Alphabetic List of Variables and Attributes-----

 # Variable  Type  Len  Pos  Label
 1 CARNO     Num     8    4
 4 MILES     Num     8   28
 2 MTCOST    Num     8   12
 3 PRICE     Num     8   20
--------------------------------------------------------------------------
```

From this output we observe that the file **car.ssd** contains 1,242 observations and four variables named carno, mtcost, price, and **miles**. The name **carno** is not the name of an actual variable, but it is a label for an identification number associated with each car.

At this stage we may want to display the data in the OUTPUT window, and we do this with the following statements.

```
libname my 'b:\';
proc print data=my.car;
run;
```

Remember, it is only necessary to give the command statement `libname my 'b:\';` once during a SAS session, at any time before the name `my` is first referenced. We will generally assume that this command has already been given, and will not explicitly include it in each command discussed hereafter.

The SAS response is

```
--------------------------------------------------------------------------

            OBS    CARNO    MTCOST    PRICE    MILES

              1        1       551    36400    12400
              2        2       661    15200    15400
              3        3       679    14100    16000
              4        4       561    22500    12100
              5        5       497    20600    11200
            ...
            ...

           1238     1238       381    23600     8000
           1239     1239       464    30700     7300
           1240     1240       563    21100    12000
           1241     1241       602    22300    14000
           1242     1242       582    14600    13100
--------------------------------------------------------------------------
```

To save space we have reproduced only the first five lines and the last five lines of data.

Example 2.3.1 in the Textbook

Here we illustrate SAS commands that can be used to perform the computations required in Example 2.3.1. First we obtain Table 2.3.1, which is a subset of the data in the SAS data file **car.ssd** that is on the data disk. We want the maintenance costs of cars that were driven 14,000 miles the first year. The following command will extract the desired data values and put them in a temporary SAS dataset named subpop.

COMMAND FOR SELECTING A SUBPOPULATION

```
data subpop;
set my.car;
if miles=14000;
proc print data=subpop;
run;
```

The first statement tells SAS to create a temporary dataset named subpop. The second statement asks SAS to copy the contents of the file **car.ssd**, which is in the directory `b:\` (whose nickname is `my`), to this temporary dataset. The third statement specifies that only those observations for which the value of `miles` equals 14,000 should be retained. The fourth statement requests SAS to print the dataset subpop just created. The fifth statement is the usual `run` statement. The result after execution of this command is shown below.

OBS	CARNO	MTCOST	PRICE	MILES
1	78	656	16100	14000
2	209	633	18400	14000
3	382	637	12900	14000
4	402	612	22000	14000
5	626	624	21900	14000
6	641	620	18100	14000
7	777	605	17000	14000
8	888	607	13300	14000
9	891	654	25600	14000
10	928	620	17300	14000
11	1029	622	16500	14000
12	1030	645	9500	14000
13	1040	567	15300	14000
14	1093	596	23700	14000
15	1199	639	13600	14000
16	1241	602	22300	14000

Notice that this subpopulation contains 16 cars, each of which was driven 14,000 miles the first year after purchase. The variables `mtcost` and `miles` make up Table 2.3.1, and

you can selectively print only the data in that table with the following statements.

```
proc print data=subpop;
var mtcost miles;
run;
```

vardef=n ⇒ population data variance degree of freedom = n.

The mean and the standard deviation of `mtcost`, the first-year maintenance costs of cars in the subpopulation that were driven 14,000 miles the first year, are obtained using the `proc means` command as follows.

```
proc means data=subpop vardef=n;
id carno;
run;
```

Note that we have used the option `vardef=n` in the `proc means` statement, because we are working with (sub)population data. The use of this option instructs SAS to use the formula (1.4.3) in the textbook, rather than formula (1.6.2). Also, since we do not want the mean and the standard deviation for the (identification number) variable `carno`, we use the statement `id carno;` to tell SAS to bypass it. The SAS response to the preceding command is

N Obs	Variable	N	Minimum	Maximum	Mean	Std Dev
16	MTCOST	16	567.0000000	656.0000000	621.1875000	22.4449848
	PRICE	16	9500.00	25600.00	17718.75	4280.80
	MILES	16	14000.00	14000.00	14000.00	0

If we let Y denote maintenance cost, X_1 denote price, and X_2 denote miles driven during the first year, the preceding output gives $\mu_Y(14,000) = \$621.19$ and $\sigma_Y(14,000) = \$22.44$. If we do not use the option `vardef=n` then SAS would use (1.6.2) to calculate the standard deviations. In particular, this would yield the value \$23.18, rather than the correct value of \$22.44, for the standard deviation of this subpopulation.

Next we demonstrate SAS commands that can be used to obtain some of the quantites in Task 2.3.1.

Task 2.3.1 in the Textbook

The data in this task are also in Table D-1 in Appendix D and in the files **car.ssd** and **car.dat** on the data disk. In parts 1(a) and 1(b), we need a histogram, as well as the mean and the standard deviation, of the first-year maintenance costs (Y) of all cars in the population. To construct a histogram (horizontal bar chart) of the values of the variable Y = mtcost, use the following statements.

```
proc chart data=my.car;
hbar mtcost;
run;
```

SAS responds with

```
--------------------------------------------------------------------------------

                         FREQUENCY OF MTCOST

 MTCOST                                           CUM              CUM
MIDPOINT                                  FREQ   FREQ  PERCENT   PERCENT
         |
   360   |*****                             38     38     3.06      3.06
   400   |*****************                128    166    10.31     13.37
   440   |******************************** 233    399    18.76     32.13
   480   |*****************************    214    613    17.23     49.36
   520   |*********************            155    768    12.48     61.84
   560   |*******************              139    907    11.19     73.03
   600   |************                      91    998     7.33     80.35
   640   |************                      88   1086     7.09     87.44
   680   |********                          61   1147     4.91     92.35
   720   |*****                             39   1186     3.14     95.49
   760   |****                              29   1215     2.33     97.83
   800   |**                                14   1229     1.13     98.95
   840   |*                                  9   1238     0.72     99.68
   880   |                                   3   1241     0.24     99.92
   920   |                                   1   1242     0.08    100.00
         ---------+-------+-------+-------
                 60     120     180

                     FREQUENCY
--------------------------------------------------------------------------------
```

From this output you can obtain information about the population such as

(1) the number (frequency) of the population values that are in each histogram interval,

(2) the cumulative frequency of the population values that are less than the upper endpoint of each histogram interval,

(3) the percent of the population values that are in each interval of the histogram, and

(4) the cumulative percent of the population values that are less than the upper endpoint of each histogram interval.

Also you can see that the distribution of mtcost is not symmetric. The preceding chart corresponds to the histogram (turned sideways) in Figure 2.3.3 in the textbook.

In part 2(a) of Task 2.3.1, we want the first-year maintenance cost of car number 354. We can obtain this value from Table D-1, but here we show you how to obtain it using a SAS command.

SAS COMMAND TO OBTAIN A SINGLE OBSERVATION FROM A DATA SET

```
data oneobs;
set my.car;
if carno=354;
proc print data=oneobs;
run;
```

The result of the preceding command is

```
--------------------------------------------------------------------

          OBS    CARNO    MTCOST    PRICE    MILES

           1      354      483      17700     9600
--------------------------------------------------------------------
```

Thus, the first-year maintenance cost for car number 354 is $483.00.

In part 2(b) of Task 2.3.1, we want the mean of the first-year maintenance costs of cars in the entire population. We can obtain this information, and much more, by computing various summary statistics for each variable. As we discussed in Section 1.1 of this manual, we can use the proc univariate command, but proc means command is sufficient here. This command has been discussed previously. The required statements are

```
proc means data=my.car vardef=n;
id carno;
run;
```

SAS responds with:

N Obs	Variable	N	Minimum	Maximum	Mean	Std Dev
1242	MTCOST	1242	352.0000000	925.0000000	526.1417069	105.9232892
	PRICE	1242	7200.00	38300.00	19647.75	5835.83
	MILES	1242	1600.00	18500.00	11114.49	3083.15

From this output you can obtain several statistics for each of the variables. For example, we see that

(1) The mean of the variable `mtcost` is \$526.14, and the standard deviation is \$105.92.

(2) The mean of the variable `price` is \$19,647.75, and the standard deviation is \$5,835.83.

(3) The mean of the variable `miles` is 11,114.49 miles, and the standard deviation is 3,083.15 miles

Observe that we have used the `vardef = n` option in the `proc means` command because we are working with population data here.

In part 3 of Task 3.2.1 we want a plot of `mtcost` against `miles`, and we can obtain this using the following statements.

```
proc plot data=my.car;
plot mtcost*miles='*'/hpos=50 vpos=15;
run;
```

The response from SAS is shown below. Observe that the output resembles the plot in Figure 2.3.4, which was obtained with a different statistical package.


```
              Plot of MTCOST*MILES.  Symbol used is '*'.

  1000 +
       |                                           **
       |                                         ****
MTCOST |                                      ******
       |                                   *******
       |                               ********
       |                          **********
   500 +                  **************
       |       **********************
       |    *   **************
       |
       |
       |
       |
     0 +
       -+-----------+-----------+-----------+-----------+-
        0         5000       10000       15000       20000

                             MILES

NOTE: 1154 obs hidden.
```

Note that 1,154 of the 1,242 total observations are hidden! Repeat this `plot` command with `vpos = 20` instead of `vpos = 15`.

Problems

Problems S2.3.1–S2.3.15 refer to the population data given in Table D-1 in Appendix D that are also stored in the files **car.dat** and **car.ssd** on the data disk. For each problem, give the appropriate SAS command and the answer.

S2.3.1 Create a temporary dataset from the ASCII file **car.dat** and name it auto. Name the variables `id`, `Y`, `X1`, and `X2`, where `id` = `carno`, `Y` = `mtcost`, `X1` = `price`, and `X2` = `miles`.

S2.3.2 Compute the mean and the standard deviation of `Y` and `X1`.

S2.3.3 What are the minimum and maximum values of the variable X1? Of the variable X2?

S2.3.4 Print the values of the variable Y.

S2.3.5 Construct a horizontal histogram for the variable Y.

S2.3.6 Construct a vertical histogram for the variable X1 and also for the variable X2.

S2.3.7 Use SAS/IML and compute SSY.

S2.3.8 Use SAS/IML and compute $\sum_{i=1}^{1242} Y_i^2$.

S2.3.9 Find the standard deviation of X2.

S2.3.10 What is the mean and the standard deviation of the variable U defined by U = X1 + 3 X2?

S2.3.11 Use the SAS data file **car.ssd** and plot `price` against `mtcost`.

S2.3.12 In Problem S2.3.11, plot the values of `mtcost` against `miles`.

S2.3.13 What was the first-year maintenance cost for car number 792 in the population?

S2.3.14 Consider the subpopulation of cars that sold for $12,500.

(a) How many cars are in this subpopulation?

(b) Which cars sold for $12,500 (give their item numbers)?

(c) Explicitly list the first-year maintenance costs associated with these cars.

(d) Calculate the mean and the standard deviation of the maintenance costs for these cars.

S2.3.15 Give the answers to Problem S2.3.14 for the subpopulation of cars that sold for $9,600.

Chapter 3

Straight Line Regression

3.1 Overview

In this chapter we show how SAS can be used to compute many of the quantities needed in straight line regression. Not all of the computations can be done directly using the built in commands in the present version of SAS, and for these we have supplied SAS programs (on the data disk) that we refer to as **macros**. As usual, sections in this laboratory manual discuss SAS computing procedures needed in the corresponding sections of the textbook.

3.2 An Example

All of the computations required in Section 3.2 can be carried out using the SAS procedures `proc contents` , `proc plot` , `proc chart` , `proc univariate` , and `proc means` , that were discussed in Chapters 1 and 2 of this manual. You should refer those chapters for information about these commands.

3.3 Straight Line Regression Model–Assumptions (A) and (B)

There are no calculations in this section that require SAS.

3.4 Point Estimation

In this section we show how SAS can be used to compute point estimates of parameters in straight line regression. We refer to Task 3.4.1, where an investigator is studying the relationship of Y, the weight of crystals, to X, the number of hours the crystals are required to grow. The data are given in Table 3.4.2 and are also stored in the SAS data file **crystal.ssd** and the ASCII data file **crystal.dat**. Assumptions (A) are presumed to be valid and the data were obtained by sampling with preselected X values. The calculations required to estimate β_0, β_1, and σ, may be conveniently carried out using the SAS command `proc reg`, to be discussed shortly, but first you should examine the contents of the SAS data file **crystal.ssd**, print and plot the data, and examine them for abnormalities or obvious violations of assumptions. SAS responses to `proc contents`, `proc plot`, and `proc print` are given below.

```
--------------------------------------------------------------------------------

                         CONTENTS PROCEDURE

Data Set Name:  MY.CRYSTAL                Type:
Observations:   14                        Record Len: 20
Variables:      2
Label:

         -----Alphabetic List of Variables and Attributes-----

#  Variable  Type  Len  Pos  Label
2  TIME      Num     8   12
1  WEIGHT    Num     8    4
```



```
           Plot of WEIGHT*TIME.  Legend: A = 1 obs, B = 2 obs, etc.

       15 +                                                           A
          |
          |                                                       A
          |                                           A    A
       10 +                                                    A
          |                                       A    A
   WEIGHT |                             A
          |
        5 +           A    A    A          A
          |
          |
          |    A
        0 +  A
          ---+----+----+----+----+----+----+----+----+----+----+----+----+----+--
             2    4    6    8   10   12   14   16   18   20   22   24   26   28

                                      TIME

                          OBS    WEIGHT    TIME

                            1      0.08      2
                            2      1.12      4
                            3      4.43      6
                            4      4.98      8
                            5      4.92     10
                            6      7.18     12
                            7      5.57     14
                            8      8.40     16
                            9      8.81     18
                           10     10.81     20
                           11     11.16     22
                           12     10.12     24
                           13     13.12     26
                           14     15.04     28
--------------------------------------------------------------------------------
```

From the preceding output we see that the file **crystal.ssd** contains two variables, viz., the response variable `weight` and the predictor variable `time`, and a straight line model appears to be reasonable.

Regression

The SAS command for computing regression quantities under the straight line model

$$\mu_Y(x) = \beta_0 + \beta_1 x$$

is the command `proc reg` . To compute a straight line regression for the data in the file **crystal.ssd**, use the following statements.

REGRESSION COMMAND

```
proc reg data=my.crystal;
model weight = time;
run;
```

We are assuming that you have already given the command `libname my b:\` , giving the nickname `my` to the directory `b:\` that contains the SAS data file **crystal.ssd**. As mentioned previously, this statement needs to be given only once in each SAS session. The first and second statements tell SAS to perform a straight line regression analysis of `weight` on `time` using the data in the SAS data file **crystal.ssd**. Thus the model is

$$\texttt{weight} = \beta_0 + \beta_1 \texttt{time}$$

Execute the command by pressing the `F10` key. The result in the `OUTPUT` window is given below (usually, we reproduce only those lines of output that are of immediate interest to us; the actual output may be more detailed than what is shown here). SAS may split up the output into several pages depending on the value of the `pagesize` option used, but we generally do not display the page numbers.


```
--------------------------------------------------------------------------------
Model: MODEL1
Dependent Variable: WEIGHT

                        Analysis of Variance

                              Sum of         Mean
      Source        DF       Squares       Square      F Value       Prob>F

      Model          1     230.63070    230.63070      204.578       0.0001
      Error         12      13.52819      1.12735
      C Total       13     244.15889

          Root MSE       1.06177     R-square       0.9446
          Dep Mean       7.55286     Adj R-sq       0.9400
          C.V.          14.05782

                        Parameter Estimates

                     Parameter      Standard     T for H0:
    Variable  DF      Estimate         Error   Parameter=0     Prob > |T|

    INTERCEP   1      0.001429    0.59938725         0.002         0.9981
    TIME       1      0.503429    0.03519723        14.303         0.0001
--------------------------------------------------------------------------------
```

The estimates of the regression coefficients are given under the column labeled `Parameter Estimate` . From this we obtain

$$\hat{\beta}_0 = 0.001429 \text{ and } \hat{\beta}_1 = 0.503429$$

From the column labeled `Standard Error` we get

$$SE(\hat{\beta}_0) = 0.59938725 \text{ and } SE(\hat{\beta}_1) = 0.03519723$$

The quantity labeled `Root MSE` is $\hat{\sigma}$, so we have $\hat{\sigma} = 1.06177$. These values are of course the same (perhaps after rounding) as those obtained in Task 3.4.1. The second line in the output, viz.,

```
Dependent Variable:  WEIGHT
```

indicates that the response variable (SAS uses the term `Dependent Variable` to mean Response Variable) is `weight`. We discuss the remaining quantities in the output as we encounter them in the textbook.

In its complete form, the `proc reg` command is capable of processing additional optional arguments. We discuss these as and when they are needed. If you are curious, you should consult the SAS/STAT guide for details.

Problems

For each problem, give the appropriate SAS command and the answer if required.

S3.4.1 Print the data of Problem 3.2.1 in the textbook. They are given in Table 3.2.3 and also stored in the SAS data file **table323.ssd** on the data disk.

S3.4.2 Plot `score` against `hours` for the data in Problem S3.4.1.

S3.4.3 For the data in Problem S3.4.1, use SAS commands to compute estimates of β_0, β_1, $\mu_Y(x)$, and σ.

S3.4.4 Repeat Problems S3.4.1–S3.4.3 using the data of Problem 3.2.5 in the textbook. These are given in Table 3.2.4 and also stored in the SAS data file **table324.ssd**.

S3.4.5 Consider the data in Table 3.4.3 in the textbook. These are also stored in the file **arsenic.ssd** on the data disk. Print the data and plot the `measured` values against the `true` values.

S3.4.6 In Problem S3.4.5, compute the estimates of β_0, β_1, $\mu_Y(x)$, and σ.

3.5 Checking Assumptions

In this section we explain how SAS commands can be used to perform many of the calculations for regresssion diagnostics discussed in Section 3.5 of the textbook. In particular, we demonstrate the commands to compute the following:

(1) Fitted values $\hat{\mu}_Y(x_i)$ (sometimes called **fits** or **predicted values**).

(2) Residuals $\hat{e}_i$.

(3) Hat values $h_{i,i}$.

(4) Standardized residuals r_i.

(5) Gaussian scores (nscores) $z_i^{(n)}$.

For illustration, we use the crystal data and exhibit the SAS commands that are used to obtain the results in Example 3.5.1. If any SAS commands have already been discussed in previous sections we do not repeat them here. Before proceeding, you should examine the contents of the file **crystal.ssd** and print the data contained in it.

The following SAS command can be used to create a dataset, which we name `diagnstc`, containing several diagnostic statistics for straight line regression. In particular, the dataset will contain the residuals $\hat{e}_i$, the fitted values $\hat{\mu}_Y(x_i)$, the hat values $h_{i,i}$, and the standardized residuals r_i.

DIAGNOSTICS COMMAND

```
proc reg data=my.crystal;
model weight=time;
output out=diagnstc p=fits r=residual student=stdresid h=hatvals;
proc print data=diagnstc;
run;
```

We explain each statment in the above command.

(1) The first statement is the `prog reg` command, which states that we want to perform a regression analysis using the data in the SAS data file **crystal.ssd** which is located in the directory `b:\` (whose nickname is `my`).

(2) The second statement tells SAS that the model to use is $\mu_Y(x) = \beta_0 + \beta_1 x$, where Y = weight of crystals and X = time (number of hours) the crystals grow.

(3) The third statement tells SAS to create a temporary dataset named `diagnstc`, and to store the computed diagnostic statistics in that dataset. The phrase `output out=` in that statement is a SAS command and must be written as indicated. However, rather than the name `diagnstc`, you can give the dataset any *valid* name you choose. The expression `p=` is a SAS expression (the letter `p` stands for predicted values) and must be written as indicated. The name on the right hand side of the expression `p=` tells SAS the name to use for the predicted values, i.e., the fitted values $\mu_Y(x_i)$. We have chosen the name `fits` for this

variable, but you can use any *valid* name. Likewise, the expression r=residual asks SAS to store the residuals in a variable named residual, the expression student=stdresid asks SAS to store the standardized residuals in a variable named stdresid (SAS uses the term studentized residuals for what we call standardized residuals), and the expression h=hatvals tells SAS to store the hatvalues in a variable named hatvals. The names we have chosen for the diagnostic statistics are indicative of the quantities they represent. You can, however, use any valid name for a variable in place of the name we have chosen. For instance, you can use the name standres instead of the name stdresid for the standardized residuals. The quantities to the left of the equal sign, viz., p , r , student , and h , must be typed in exactly as indicated.

(4) The fourth statement instructs SAS to print the dataset diagnstc.

(5) The fifth and final statement is the usual run statement that tells SAS to execute the statements preceding it when the F10 key is pressed.

When the above command is executed, SAS displays the usual regression computations in the OUTPUT window, stores the requested diagnostic quantities – fits, residuals, standardized residuals, and hat values – in a temporary dataset named diagnstc, and prints the contents of this dataset. The SAS response is displayed below.

```
Model: MODEL1
Dependent Variable: WEIGHT

                         Analysis of Variance

                          Sum of         Mean
Source           DF      Squares       Square      F Value       Prob>F

Model             1    230.63070    230.63070      204.578       0.0001
Error            12     13.52819      1.12735
C Total          13    244.15889

     Root MSE       1.06177     R-square       0.9446
     Dep Mean       7.55286     Adj R-sq       0.9400
     C.V.          14.05782
```



```
                          Parameter Estimates

                     Parameter      Standard     T for H0:
Variable  DF          Estimate         Error   Parameter=0     Prob > |T|

INTERCEP   1          0.001429    0.59938725         0.002         0.9981
TIME       1          0.503429    0.03519723        14.303         0.0001

OBS    WEIGHT    TIME      FITS     RESIDUAL    STDRESID    HATVALS

  1     0.08       2     1.0083    -0.92829    -1.01438    0.25714
  2     1.12       4     2.0151    -0.89514    -0.94518    0.20440
  3     4.43       6     3.0220     1.40800     1.44726    0.16044
  4     4.98       8     4.0289     0.95114     0.95781    0.12527
  5     4.92      10     5.0357    -0.11571    -0.11481    0.09890
  6     7.18      12     6.0426     1.13743     1.11767    0.08132
  7     5.57      14     7.0494    -1.47943    -1.44682    0.07253
  8     8.40      16     8.0563     0.34371     0.33614    0.07253
  9     8.81      18     9.0631    -0.25314    -0.24874    0.08132
 10    10.81      20    10.0700     0.74000     0.73420    0.09890
 11    11.16      22    11.0769     0.08314     0.08373    0.12527
 12    10.12      24    12.0837    -1.96371    -2.01847    0.16044
 13    13.12      26    13.0906     0.02943     0.03107    0.20440
 14    15.04      28    14.0974     0.94257     1.02999    0.25714
```

Verify that the values of the diagnostic statistics listed in this SAS output agree with the values given in Table 3.5.1 in the textbook.

Later, we will discuss other diagnostic quantities that can be computed using SAS commands. Of course, these computations may be performed for any dataset you wish. You then need to use the appropriate SAS data file in place of **crystal.ssd**.

Next we obtain the Gaussian scores (nscores) of the standardized residuals for the crystal data and plot them.

Gaussian Scores (Nscores)

In a regression problem, to help determine if assumptions (**A**) or (**B**) are satisfied, we use a rankit plot of the standardized residuals (r_i). First we create a temporary file, which we call newdata, that contains the variables stdresid (standardized residuals) and the nscores (Gaussian scores) of the standardized residuals. We assume that the variable stdresid has been computed as indicated in the previous command (DIAGNOSTICS COMMAND), and is stored in the temporary SAS dataset diagnstc. In the following command we use this file to create the file newdata, which contains stdresid and nscores.

COMMAND TO COMPUTE NSCORES

```
proc rank normal=blom data=diagnstc out=newdata;
var stdresid;
ranks nscores;
run;
```

We now explain each statement in the preceding command.

(1) The first part of statement one, namely `proc rank normal=blom`, tells SAS to compute nscores using a formula derived by G. Blom. The remainder of this statement, namely, `data=diagnstc out=newdata;` , tells SAS that the data to use to compute nscores are in the temporary dataset diagnstc, and the temporary dataset to store the computed nscores is to be named newdata.

(2) The second statement, `var stdresid;` , tells SAS to compute nscores for the variable stdresid (this is the name we supplied for standardized residuals when creating the temporary dataset diagnstc, containing the diagnostic statistics of interest, in the DIAGNOSTICS COMMAND).

(3) The third statement tells SAS to use the name nscores for the Gaussian scores just computed. You could use any other valid name you wish. Nscores are a special case of a class of statistics called *rank scores.* This is the reason for the word rank being used in the third statement. This is also the reason that nscores are computed using `proc rank` . The dataset newdata contains the computed nscores and all the variables in the dataset diagnstc, which include the response variable, the predictor variables, and the variable stdresid.

(4) The fourth statement is the usual `run` statement.

After you execute the preceding command, the output dataset newdata will contain the variable stdresid (which is in the dataset diagnstc) and the nscores for stdresid. We illustrate the above command by using the crystal data of Example 3.5.1 in the textbook, where we want to compute nscores for the standardized residuals in the regression of Y (weight) on X (time). The DIAGNOSTICS COMMAND discussed earlier is used

to compute the regression of Y on X and to obtain the standardized residuals. The complete command is as follows.

```
proc reg data=my.crystal;
model weight=time;
output out=diagnstc student=stdresid;
proc rank normal=blom data=diagnstc out=newdata;
var stdresid;
ranks nscores;
run;
```

The first three statements in the preceding command instruct SAS to use the data in the file **crystal.ssd** and compute the regression of weight on time, then to compute the standardized residuals (stdresid) and to save them in the temporary dataset diagnstc. The next four statements instruct SAS to compute the Gaussian scores of the standardized residuals, give the name nscores to the resulting variable, and to store the values of these two variables (nscores and stdresid) in the temporary SAS dataset newdata. Next we plot the stdresid against the nscores using the `plot command` .

```
proc plot data=newdata;
plot stdresid*nscores='o';
run;
```

```
---------------------------------------------------------------------------------

            Plot of STDRESID*NSCORES.  Symbol used is 'o'.

S  2 +
t    |
u    |                                                             o
d    |                                                    o  o   o
e    |                                                 o
n    |                                              o
t  0 +                                     o  o  o
i    |                                  o
z    |
e    |                          o  o
d    |                     o
     |
R -2 +            o
e    ---+--------+--------+--------+--------+--------+--------+--------+--------+--
s     -2.0     -1.5     -1.0     -0.5      0.0      0.5      1.0      1.5      2.0
i
                             RANK FOR VARIABLE STDRESID
```

The plot is similar to that in Figure 3.5.21 except the scale is different. Now we print the file newdata, which contains weight, time, the standardized residuals stdresid, and the nscores, using the print command.

```
proc print data=newdata;
run;
```

SAS responds with:

OBS	WEIGHT	TIME	STDRESID	NSCORES
1	0.08	2	-1.01438	-0.89943
2	1.12	4	-0.94518	-0.66075
3	4.43	6	1.44726	1.70755
4	4.98	8	0.95781	0.66075
5	4.92	10	-0.11481	-0.26699
6	7.18	12	1.11767	1.20534
7	5.57	14	-1.44682	-1.20534
8	8.40	16	0.33614	0.26699
9	8.81	18	-0.24874	-0.45498
10	10.81	20	0.73420	0.45498
11	11.16	22	0.08373	0.08807
12	10.12	24	-2.01847	-1.70755
13	13.12	26	0.03107	-0.08807
14	15.04	28	1.02999	0.89943

Problems

Problems S3.5.1 through S3.5.9 refer to Example 3.5.2. The data for this example are given in Table 3.5.2 and are also stored in the files **car20.ssd** and **car20.dat**. For each problem, give the appropriate SAS commmand and the answer if required.

S3.5.1 Examine the contents of the SAS data file **car20.ssd** on the data disk.

S3.5.2 Plot Y (mtcost) against X (miles).

S3.5.3 Regress Y (mtcost) on X (miles) and obtain the standardized residuals r_i (name them standres), the residuals $\hat{e}_i$ (name them resd), and the fitted values $\hat{\mu}_Y(x_i)$ (name them fitval).

S3.5.4 Obtain estimates of β_1, β_0, $\mu_Y(x)$, and σ.

S3.5.5 Calculate $\hat{\mu}_Y(9400)$.

S3.5.6 Print the values of mtcost, miles, the residuals $\hat{e}_i$, the fits $\hat{\mu}_Y(x_i)$, and the standardized residuals r_i, in one table.

S3.5.7 Obtain a plot of the standardized residuals r_i against the x_i values (miles).

S3.5.8 Compute the nscores of the standardized residuals.

S3.5.9 In Problem S3.5.8, plot the standardized residuals against the nscores.

3.6 Confidence Intervals

The output of the proc reg command gives the values of $\hat{\beta}_0$, $\hat{\beta}_1$, and the standard errors of these quantities. You can use these in (3.6.1) to compute confidence intervals for β_0 and β_1. The output from this command also gives $\hat{\sigma}$ and you can use this in (3.6.8) to compute confidence intervals for σ. However, there are no built-in SAS commands that will compute $1-\alpha$ confidence intervals for regression parameters except for special values of $1-\alpha$. In particular, there are no built-in SAS commands for computing general $1-\alpha$ confidence intervals for the following.

(a) σ_Y

(b) σ

(c) $\mu_Y(x)$ (for user specified values of x)

(d) β_0

(e) β_1

(f) $\theta = a_0\beta_0 + a_1\beta_1$ (for user specified values of a_0 and a_1).

(g) $Y(x)$ (for user specified values of x)

It is worth noting that the SAS procedure GLM does offer a facility for obtaining point estimates and their standard errors for user specified linear combinations of the regression parameters. Using this information one could compute confidence intervals having the desired confidence levels. However, we have written three SAS programs, called **macros**, that you can use to compute confidence intervals for the quantities in **(a)**–**(g)** above. We discuss these macros in this section and show you how to use them. The macros, which are on the data disk, are

- **sgmaconf**, which can be used to compute confidence intervals for σ_Y and σ in **(a)** and **(b)** above.
- **citheta**, which can be used to compute confidence intervals for the general linear combination θ in **(f)**. This macro can also be used to compute confidence intervals for $\mu_Y(x)$, β_0, and β_1 in **(c)**, **(d)**, and **(e)** above, by choosing appropriate values for a_0 and a_1.
- **predy**, which can be used to compute prediction intervals for $Y(x)$ in **(g)**.

Each macro has associated with it two files, each with the same name as the macro, one with the extension **mac** (for **macro** file), and the other with the extension **sas**, that contain the necessary SAS commands to implement the macro. The **mac** file serves as the *front end* for the macro and it automatically calls the **sas** file which contains the SAS statements for performing the required computations. For example, to compute confidence intervals for σ and σ_Y, the two files **sgmaconf.mac** and **sgmaconf.sas** are used. The macros and the data files are on the data disk. To use them, you must insert the disk in one of the floppy drives of the personal computer you are using. We assume that the disk is inserted in drive `B`. You will need to use only the files with the extension **mac**, but these in turn will invoke the files with the extension **sas** during the execution of the macros.

First we show you how to use the macro **sgmaconf** to compute confidence intervals for σ and/or σ_Y.

Sgmaconf Macro

To use any macro, you must read the contents of the corresponding **mac** file into the `PROGRAM EDITOR` window, and enter the requested information on designated lines. Accordingly, to use the macro **sgmaconf**, invoke SAS, and on the `Command` line of the `PROGRAM EDITOR` window type `include 'b:\macro\sgmaconf.mac'` and press

`Enter`. This command brings the file `sgmaconf.mac` into the `PROGRAM EDITOR` window. We display it below.

```
--------------------------------------------------------------------------------
00001 Title 'Confidence interval for sigma';
00002 proc iml;
00003
00004 ****** On line 00006 enter the confidence coefficient;
00005 cc=
00006                                  0.95
00007
00008 ;
00009 ****** On line 00011 enter the estimate of sigma;
00010 s=
00011                                  1.000
00012
00013 ;
00014 ****** On line 00016 enter the degrees of freedom;
00015 df=
00016                                  25
00017
00018 ;
00019
00020 %include 'b:\macro\sgmaconf.sas';
--------------------------------------------------------------------------------
```

You must enter relevant information in order for the macro to carry out the computations. In the above display, the lines that begin with `******` are instructions telling you what data to enter, and where (which line) to enter them. For example, on line 00006 you enter the confidence coefficient (it will replace 0.95 that is on that line). On line 00011 you enter the estimate of sigma (it will replace 1.000 that is on that line). On 00016 you enter the degrees of freedom used to estimate sigma (it will replace 25 that is on that line).

As an illustration, we compute a 90% two-sided confidence interval for σ in the blood pressure problem of Task 3.6.2. From that task we obtain $\hat{\sigma} = 2.8356$, and the degrees of freedom $df = n - 2 = 22$. After you invoke SAS and bring the file **sgmaconf.mac** into the `PROGRAM EDITOR` window, you input these values on appropriate lines. At this point, the `PROGRAM EDITOR` window will have the following statements.

```
00001 Title 'Confidence interval for sigma';
00002 proc iml;
00003
00004 ****** On line 00006 enter the confidence coefficient;
00005 cc=
00006                           0.90
00007
00008 ;
00009 ****** On line 00011 enter the estimate of sigma;
00010 s=
00011                           2.8356
00012
00013 ;
00014 ****** On line 00016 enter the degrees of freedom;
00015 df=
00016                           22
00017
00018 ;
00019
00020 %include 'b:\macro\sgmaconf.sas';
```

Press the F10 key and the program will execute. The following result will be displayed in the OUTPUT window.

```
                  Confidence interval for sigma


      For a two-sided 90% confidence interval for sigma

      the lower confidence bound is   2.2835 and

      the upper confidence bound is   3.7865
```


Thus the confidence statement is

$$C[2.2835 \leq \sigma \leq 3.7865] = 0.90$$

For another illustration, we compute a 90% two-sided confidence interval for σ_Y for the blood pressure data in Task 3.6.2. We need $\hat{\sigma}_Y$, which can be obtained from Task 3.6.2, and it is equal to 20.3391 with associated degrees of freedom $df = n - 1 = 23$. To start the program, invoke SAS, bring the file **sgmaconf.mac** into the PROGRAM EDITOR window by typing `include 'b:\macro\sgmaconf.mac'` on the command line and pressing Enter . Input the required quantities, and press the F10 key. The output is shown below.

```
                  Confidence interval for sigma


      For a two-sided 90% confidence interval for sigma

      the lower confidence bound is  16.4473 and

      the upper confidence bound is  26.9598
```

Thus the confidence statement is

$$C[16.4473 \leq \sigma_Y \leq 26.9598] = 0.90$$

Confidence Interval for $\theta = a_0\beta_0 + a_1\beta_1$

Here we discuss the macro **citheta** which can be used for computing a confidence interval for the linear combination $\theta = a_0\beta_0 + a_1\beta_1$. Remember, this macro can be used to obtain confidence intervals for the following

(1) β_0, by setting $a_0 = 1$ and $a_1 = 0$.

(2) β_1, by setting $a_0 = 0$ and $a_1 = 1$.

(3) $\mu_Y(x) = \beta_0 + \beta_1 x$, by setting $a_0 = 1$ and $a_1 = x$ for any specified value x.

(4) $\mu_Y(x_1) - \mu_Y(x_2)$, by setting $a_0 = 0$ and $a_1 = x_1 - x_2$.

The SAS statements for this macro are in the files **citheta.mac** and **citheta.sas**, both of which are on the data disk, which we assume is in drive `B` . The name **citheta** stands for **c**onfidence **i**nterval for **theta**. To start the macro, invoke SAS, and on the command line of the `PROGRAM EDITOR` window type `include 'b:\macro\citheta.mac'` . Press `Enter` and this will bring the following SAS statements to the screen.

```
-------------------------------------------------------------------------

00001 Title 'Confidence interval for theta';
00002 libname my 'b:\';proc iml; reset nolog;
00003
00004 ****** On line 00007 enter the name of the SAS data file
00005 ****** you want to use;
00006 use
00007                              my.filename
00008 ;
00009 ****** On line 00013 enter the name of the response variable
00010 ****** exactly as it appears in the data file;
00011
00012 read all var{
00013                              response variable
00014 } into yvar;
00015
00016 ****** On line 00020 enter the name of the predictor variable
00017 ****** exactly as it appears in the data file;
00018
00019 read all var{
00020                              predictor variable
00021 } into xvar;
00022
00023 ****** On line 00025 enter the desired confidence coefficient;
00024 cc=
00025                              0.95
00026 ;
00027 ****** On line 00029 enter the vector a;
00028 a={
00029                              0   1
00030
00031 };%include 'b:\macro\citheta.sas';
-------------------------------------------------------------------------
```


You must input the quantities explained on the lines that begin with `******` . They are described below.

- On line 00007 enter the name of the SAS data file you want to use. This must include the prefix `my` , and will replace the expression `my.filename` . For example, if you wish to use the data in the file **crystal.ssd**, the expression on line 00007 will be `my.crystal` , etc.
- On line 00013 enter the name of the response variable, exactly as it appears in the data file. This will replace the words `response variable` on that line.
- On line 00020 enter the name of the predictor variable, exactly as it appears in the data file. This will replace the words `predictor variable` on that line.
- On line 00025 enter confidence coefficient you want to use. Your value will replace `0.95` unless, of course, you want to use 0.95.
- On line 00029 enter the values of a_0 and a_1 you want to use. The value of a_0 should be entered first, and the value of a_1 second.

Press the `F10` key to execute the macro. The result will be displayed in the `OUTPUT` window.

To illustrate, we use part (2) of Task 3.6.1, where an investigator wants to obtain a 95% confidence interval for β_1 for the arsenic data. These data are given in Table 3.6.1 and are also stored in the SAS data file **arsenic.ssd**. Hence you should enter `my.arsenic` on line 00007 of the preceding command. On line 00013, you should enter `measured` to replace the words `response variable` , because the name of the response variable, as it appears in the data file, is `measured` . You should enter `true` on line 00020 to replace the words `predictor variable` , because the name of the predictor variable for this problem, as it appears in the data file **arsenic.ssd**, is `true` . If you are not sure what the exact names of the response variable and the predictor variable are in the data file, then you should use `proc contents` first to determine this. Finally, enter `0.95` on line 00025, and `0 1` on line 00029 (since $a_0 = 0$ and $a_1 = 1$ here). Press `F10` to execute the program. The following results will be displayed in the `OUTPUT` window.

```
-----------------------------------------------------------------------
                    Confidence interval for theta

        The point estimate of theta is   0.9877

        For a two-sided  95%   confidence interval for theta

        the lower confidence bound is   0.9582   and

        the upper confidence bound is   1.0172
-----------------------------------------------------------------------
```

So $\hat{\theta} = \hat{\beta}_1 = 0.9877$, and the confidence statement is

$$C[0.9582 \leq \beta_1 \leq 1.0172] = 0.95$$

You should compare these results with those obtained in Task 3.6.1 to verify that they are the same (within rounding error).

Prediction Interval for $Y(x)$

SAS has no built-in command to compute general $1-\alpha$ prediction intervals, but we can use the macro **predy** to do this. The SAS commands for this macro are stored in the two files **predy.mac** and **predy.sas**. Invoke SAS, and on the Command line in the `PROGRAM EDITOR` window type

```
include 'b:\macro\predy.mac'
```

and press `Enter` . This will bring the following statements from the file **predy.mac** to the screen.

```
-----------------------------------------------------------------------
00001 Title 'Predicted values and prediction intervals';
00002 libname my 'b:\';proc iml; reset nolog;
00003
00004 ****** On line 00007 enter the name of the SAS data file
00005 ****** you want to use;
00006 use
00007                             my.filename
00008 ;
00009 ****** On line 00013 enter the name of the response variable
00010 ****** exactly as it appears in the data file;
00011
00012 read all var{
00013                             response variable
00014 } into yvar;
00015
00016 ****** On line 00020 enter the name of the predictor variable
00017 ****** exactly as it appears in the data file;
00018
00019 read all var{
00020                             predictor variable
00021 } into xvar;
00022
00023 ****** On line 00025 enter the desired confidence coefficient;
00024 cc=
00025                             0.95
00026 ;
00027 ****** On line 00029 enter the value of x;
00028 x=
00029                             100
00030
00031 ;%include 'b:\macro\predy.sas';
-----------------------------------------------------------------------
```


To use this macro, you enter the appropriate information on lines 00007, 00013, 00020, 00025, and 00029, as requested in the statements beginning with `******` , and press the `F10` key.

We illustrate the use of this macro by computing a 95% two-sided prediction interval for $Y(60)$, for the age-blood pressure data in Task 3.6.2. The data are stored in the SAS data file **agebp.ssd** on the data disk. Blood pressure is the response variable and age is the predictor variable. You must input the following quantities.

- On line 00007 enter the name (along with the prefix `my`) of the SAS data file that contains the data you want to use. For this problem, enter `my.agebp` .
- On line 00013 enter the name of the response variable exactly as it appears in the data file. For this problem, enter `bp` to replace the words `response variable` .

- On line 00020 enter the name of the predictor variable. For this problem, you should enter `age` to replace the words `predictor variable`.
- On line 00025 enter the desired confidence coefficient. For this problem, the desired confidence coefficient is 0.95, which in this case is already there.
- On line 00029 enter the value of X, say x, you want to use to predict Y. For this problem you will enter 60, which will replace the value 100 that is present initially.

After the requested quantities are entered, press the F10 key to execute the program. The output from the preceding macro is displayed in the OUTPUT window, and we reproduce it below.

```
----------------------------------------------------------------------
              Predicted values and prediction intervals


        The point estimate of Y(x) for x =  60.00 is 163.3200

        For a two-sided 95.0% prediction interval for Y(x)

        the lower bound is  157.1401  and

        the upper bound is  169.4999
----------------------------------------------------------------------
```

From this we get $\hat{Y}(60) = 163.32$, and the confidence statement is

$$C[157.1401 \leq Y(60) \leq 169.4999] = 0.95$$

Problems

Problems S3.6.1–S3.6.10 refer to the arsenic data of Task 3.6.1, which are given in Table 3.6.1, and are also stored in the files **arsenic.ssd** and **arsenic.dat**. We use Y to denote the measured value, and X to denote the true value. Exhibit the SAS commands that can be used to compute the needed quantities, and where appropriate, give the answers.

S3.6.1 Calculate $\hat{\beta}_0$, $\hat{\beta}_1$, $\hat{\mu}_Y(x)$, $\hat{\sigma}$, $SE(\hat{\beta}_0)$, and $SE(\hat{\beta}_1)$.

S3.6.2 Plot Y against X.

S3.6.3 Compute the residuals $\hat{e}_i$, the fitted values $\hat{\mu}_Y(x_i)$, and the standardized residuals r_i.

S3.6.4 Plot the standardized residuals r_i against y_i, against x_i, and against $\hat{\mu}_Y(x_i)$.

S3.6.5 Compute a 90% confidence interval for $\mu_Y(0)$, i.e., for β_0.

S3.6.6 Compute a 90% confidence interval for β_1.

S3.6.7 Find $\hat{\mu}_Y(3)$.

S3.6.8 Compute a 95% confidence interval for $\mu_Y(3)$.

S3.6.9 Calculate $\hat{Y}(3)$.

S3.6.10 Compute a 95% prediction interval for $Y(3)$.

3.7 Tests

The `proc reg` command will compute the P-value for testing the following pairs of hypotheses.

(1) NH: $\beta_0 = 0$ versus AH: $\beta_0 \neq 0$

(2) NH: $\beta_1 = 0$ versus AH: $\beta_1 \neq 0$

The P-value for these tests are given in the column labeled `Prob > |T|` in the output from the `proc reg` command. See Section 3.4 of this manual for a sample output. Moreover, as part of the `proc reg` command, SAS offers an optional command called `test`, which can be used to calculate the P-values for two-sided tests concerning linear combinations $\theta = a_0\beta_0 + a_1\beta_1$, where the user must specify the coefficients a_0 and a_1. Although one could deduce the appropriate P-value for a one-sided test from the P-value for the corresponding two-sided test, you will find it more convenient to use the macro **test** that we have supplied on the data disk. This macro will perform the computations for testing the following pairs of hypotheses.

(a) NH: $\theta = q$ versus $\theta \neq q$

(b) NH: $\theta \leq q$ versus $\theta > q$

(c) NH: $\theta \geq q$ versus $\theta < q$

where $\theta = a_0\beta_0 + a_1\beta_1$ is a linear combination of the regression coefficients β_0 and β_1 in the straight line regression model $\mu_Y(x) = \beta_0 + \beta_1 x$. Since β_0, β_1, and $\mu_Y(x)$ can be obtained as special cases of θ (by selecting the appropriate values for a_0 and a_1) this macro can be used to perform tests about any of these quantities. The procedure for conducting these tests is explained in Box 3.7.4 in the textbook. To use the macro, invoke SAS, go to the Command line of the PROGRAM EDITOR window, and type `include 'b:\macro\test.mac'` . Press `Enter` and the following statements will appear in the PROGRAM EDITOR window.

```
--------------------------------------------------------------------------

00001 Title 'Test for theta';
00002 libname my 'b:\';proc iml; reset nolog;
00003
00004 ****** On line 00007 enter the name of the SAS data file
00005 ****** you want to use;
00006 use
00007                                my.filename
00008 ;
00009 ****** On line 00013 enter the name of the response variable
00010 ****** exactly as it appears in the data file;
00011
00012 read all var{
00013                                response variable
00014 } into yvar;
00015
00016 ****** On line 00020 enter the name of the predictor variable
00017 ****** exactly as it appears in the data file;
00018
00019 read all var{
00020                                predictor variable
00021 } into xvar;
00022
```



```
00023 ****** On line 00025 enter the value of q;
00024 q=
00025                                0
00026 ;
00027 ****** On line 00029 enter the vector a;
00028 a={
00029                                0   1
00030
00031 };%include 'b:\macro\test.sas';

--------------------------------------------------------------------------
```

To use the macro, you must input the following quantities.

(1) On line 00007 enter the name of the SAS data file that contains the data you want to use. This file name will replace the expression `my.filename;` which is present initially. Remember to use the prefix `my` .

(2) On line 00013 enter the name of the response variable as it appears in the data file.

(3) On line 00020 enter the name of the predictor variable as it appears in the data file.

(4) Input the value of q on line 00025 to replace 0.

(5) Input the values of a_0 and a_1 on line 00029 to replace the expression `0   1` .

To illustrate the use of this macro we refer to Task 3.7.1, where an investigator is interested in using the sample arsenic data, in the SAS data file **arsenic.ssd**, to help decide whether $\beta_0 = 0$. In this context one may wish to test

$$\text{NH: } \beta_0 = 0 \text{ versus AH: } \beta_0 \neq 0$$

To use the macro **test** for this problem, enter `my.arsenic` on line 00007; enter **`measured`** for **`response variable`** on line 00013; enter **`true`** for **`predictor variable`** on line 00020; enter 0 for q on line 00025; enter `1   0` on line 00029 for a_0 and a_1, respectively. After the proper quantities are entered, press the `F10` key to execute the program. The following output is displayed in the OUTPUT window.

```
---------------------------------------------------------------------------
                              Test for theta

For NH: theta     =     0.0000 vs AH: theta not =    0.0000, P value = 0.094

For NH: theta < or =    0.0000 vs AH: theta     >    0.0000, P value = 0.047

For NH: theta > or =    0.0000 vs AH: theta     <    0.0000, P value = 0.953
---------------------------------------------------------------------------
```

The test of interest here yields a P-value equal to 0.094. If you use $\alpha = 0.05$ for this test, then NH is not rejected.

Problems

S3.7.1 This problem is discussed in Task 3.7.2 where an investigator is interested in testing

$$\text{NH: } 6\beta_0 + 264\beta_1 \leq 50 \quad \text{against} \quad \text{AH: } 6\beta_0 + 264\beta_1 > 50$$

Use the macro **test** to perform this test and determine the P-value. The data are in the SAS data file **crystal.ssd** on the data disk.

S3.7.2 This problem refers to Problem 3.7.1 in the textbook. The data are in the SAS data file **shelflif.ssd** on the data disk. Use the macro **test** to determine the P-value for the following tests.

$$\text{(a) NH: } \beta_1 = 0 \quad \text{versus} \quad \text{AH: } \beta_1 \neq 0$$

$$\text{(b) NH: } \mu_Y(13) \leq 650 \quad \text{versus} \quad \text{AH: } \mu_Y(13) > 650$$

3.8 Analysis of Variance

In Section 3.8 of the textbook we discuss *Analysis of Variance* and present an Analysis of Variance (ANOVA) table. The `proc reg` command will produce an analysis of variance table as part of the output. For a sample output, you can refer to the output of the `proc reg` command in Section 3.4 of this manual.

Problems

S3.8.1 For the shelf life data in Table 3.7.1, use SAS commands to produce an Analysis of Variance table. The data are in the SAS data file **shelflif.ssd** on the data disk.

S3.8.2 For the age and blood pressure data in Table 3.6.2, compute and display an ANOVA table. The data are in the SAS data file **agebp.ssd** on the data disk.

S3.8.3 For the grades26 data in Table 3.2.2, compute and display an ANOVA table. These data are in the SAS data file **grades26.ssd** on the data disk.

3.9 Coefficient of Determination and Coefficient of Correlation

The output of the `proc reg` command discussed in Section 3.4 of this manual contains a quantity labeled `R-square` which is a point estimate of $\rho^2_{Y,X}$ provided that the sample data are obtained by simple random sampling. The square root of `R-square` , using the sign of the estimate of β_1, results in an estimate of $\rho_{Y,X}$, the simple correlation coefficient of Y and X. From the output of the `proc reg` command for the data in the SAS data file **crystal.ssd**, we see that $\hat{\rho}^2_{Y,X} = 0.9446$. There is no built-in SAS command for computing confidence intervals for $\rho_{Y,X}$, but we discuss a procedure for this in Box 3.9.2 of the textbook. For straight line regression recall that $\rho_{Y,X}$ and σ_Y/σ are related by

$$\rho^2_{Y,X} = \frac{\sigma^2_Y - \sigma^2}{\sigma^2_Y} = 1 - \frac{1}{(\sigma_Y/\sigma)^2}$$

So from a confidence interval for $\rho_{Y,X}$ you can obtain a confidence interval for σ_Y/σ as explained in Box 3.9.3.

3.10 Regression Analysis When There Are Measurement Errors

All computations required in this section have been explained in previous sections.

3.11 Regression through the Origin

To perform the calculations for straight line regression through the origin (i.e., when β_0 is known to be 0) you can use the command given below. We assume that the data are in the SAS data file `filename` (in the macro, the expression `filename` must be replaced by the actual name of the SAS data file), that the response variable is named Y, and the predictor variable is named X. If names other than Y and X are used, then appropriate substitutions must be made in the following statements.

REGRESSION COMMAND FOR MODEL WITH NO INTERCEPT

```
proc reg data=my.filename;
model Y=X/noint;
run;
```

The expression `noint` means no intercept, and hence the command will perform calculations for the regression model

$$\mu_Y(x) = \beta_1 x$$

All other SAS commands proceed as discussed in previous sections.

Problems

S3.11.1 For the gravity data in Table 3.11.1, use appropriate SAS commands to compute $\hat{\beta}_1$ and $\hat{\sigma}$. These data are in the SAS data file **gravity.ssd** on the data disk.

Chapter 4

Multiple Linear Regression

4.1 Overview

No computing instructions are needed in this section.

4.2 Notations and Definitions

All computing needed in this section has been discussed in the preceding Chapters.

4.3 Assumptions for Multiple Linear Regression

No computations are required for this section.

4.4 Point Estimation

In this section we describe two ways of obtaining point estimates for β_0, β_1, $\ldots$, β_k, $\mu_Y(x_1, \ldots, x_k)$, $Y(x_1, \cdots, x_k)$, and $\sigma_{Y|X_1,\cdots,X_k} = \sigma$, using SAS.

(1) By using matrix commands in SAS/IML and the formulas in (4.4.8), (4.4.9), (4.4.10), and (4.4.16).

(2) By using the `proc reg` command in SAS.

We use the GPA data of Example 4.4.2 to illustrate these two methods. In that example the value of k is 4, but you should have no trouble doing the computations for any value of k.

Regression Computations Using Matrices

Consider Example 4.4.2 where the data are given in Table 4.4.3 and are also stored in the files **gpa.dat** and **gpa.ssd** on the data disk. The population regression function, given in (4.4.24), is

$$\mu_Y(x_1, x_2, x_3, x_4) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4$$

where Y = GPA at the end of one year, X_1 = SATmath, X_2 = SATverb, X_3 = HSmath, and X_4 = HSengl. We first use SAS/IML commands to show how to use matrices to obtain the point estimates of the quantities of interest. As usual, you should invoke SAS and examine the contents of the file **gpa.ssd** using the `proc contents` command.

First we create a 20 by 1 vector $\boldsymbol{y}$ whose elements are the values of GPA, and then create a 20 by 5 matrix $\boldsymbol{X}$, whose first column contains 20 ones and whose last four columns contain the values of SATmath, SATverb, HSmath, and HSengl, respectively (see (4.4.7)). The following command is used to create these matrices.

COMMAND TO CREATE THE VECTOR $\boldsymbol{y}$ AND THE MATRIX $\boldsymbol{X}$ FROM THE SAS DATA FILE GPA.SSD

```
libname my 'b:\';

proc iml;
reset nolog;

use my.gpa;
read all var{gpa} into y;
read all var{satmath satverb hsmath hsengl} into q;

ones=j(20,1,1);
x=ones||q;

print x;
print y;
```

To explain the preceding command we have broken it into five groups of statements. Notice that these five groups of statements are separated from one another by blank lines. This is a commonly used practice in SAS programming. SAS will ignore these blank lines, but they help in organizing a long SAS program into meaningful groups.

(1) The first group consists of a single statement which is the usual `libname` statement.

(2) The second group of (two) statements invokes IML and routes the results to the `OUTPUT` window instead of the `LOG` window.

(3) The third group of (three) statements tells SAS to use the permanent SAS data file **gpa.ssd** from the directory `b:\` , read all the observations for the variable gpa into a vector we name `y`, and read all the observations for the variables `SATmath`, `SATverb`, `HSmath`, and `HSengl` into a matrix we name `q`. These commands are explained in Section 1.8 of this manual.

(4) The fourth group consists of two statements. The first of these two statements creates a 20 by 1 matrix named ones; this is actually a vector with each element equal to +1. This vector will be used as the first column of $\boldsymbol{X}$. The general command is `k=j(r,c,a);` , which asks SAS to create a matrix `k` with `r` rows and `c` columns, with each element of the matrix having the value `a`. The second statement in this group, namely `x=ones||q;` , instructs SAS to create a matrix

x with the vector ones as the first column, and the matrix q as the remaining four columns. More generally, the command F||G forms a matrix by joining (concatenating) the columns of F followed by the columms of G. To use this command, the matrices F and G must have the same number of rows. Note that, in the preceding command we have used lower case letters for both vectors and matrices. Since SAS does not distinguish between lower and upper case letters, we will *generally* use lower case letters throughout. In fact, we generally use names, rather than just single letters (such as xmatrix , ones , etc.), for matrices and vectors.

(5) The last group of two statements asks SAS to print X, and then print y (which are denoted by x and y, respectively, in the above command).

SAS responds with:

```
------------------------------------------------------------------------

              X
              1      321      247      2.3     2.63
              1      718      436      3.8     3.57
              1      358      578     2.98     2.57
              1      403      447     3.58     2.21
              1      640      563     3.38     3.48
              1      237      342     1.48     2.14
              1      270      472     1.67     2.64
              1      418      356     3.73     2.52
              1      443      327     3.09      3.2
              1      359      385     1.54     3.46
              1      669      664     3.21     3.37
              1      409      518     2.77      2.6
              1      582      364     1.47      2.9
              1      750      632     3.14     3.49
              1      451      435     1.54      3.2
              1      645      704      3.5     3.74
              1      791      341      3.2     2.93
              1      521      483     3.59     3.32
              1      594      665     3.42      2.7
              1      653      606     3.69     3.52
```



```
                 Y
              1.97
              2.74
              2.19
               2.6
              2.98
              1.65
              1.89
              2.38
              2.66
              1.96
              3.14
              1.96
               2.2
               3.9
              2.02
              3.61
              3.07
              2.63
              3.11
               3.2

------------------------------------------------------------------------
```

You should check the entries of these matrices to convince yourself that they are indeed correct.

Next, we demonstrate the SAS/IML matrix commands to compute various parameter estimates. We use the matrices x and y that were created as a result of the preceding command. In particular, we are still in IML. We first list the necessary commands and then explain their meanings. Recall, we use x for the matrix X and y for the vector y.

```
betahat = inv(x'*x)*x'*y;
e = y-x*betahat;
sigmahat = sqrt(e'*e/15);
```

The first statement computes $\hat{\beta}$ using the formula in (4.4.8). The second statement computes the vector $\hat{e}$. The final statement asks SAS to calculate $\hat{\sigma}$ using (4.4.19), (4.4.17), and (4.4.16). Note that the number 15 appearing on the right hand side of the last statement is the value of $n-k-1$, because $n=20$ and $k=4$ in this example. Use these commands and verify that

$$\hat{\beta} = [0.1615496,\ 0.0020102,\ 0.0012522,\ 0.1894402,\ 0.0875637]^T$$

and that $\hat{\sigma} = 0.2685143$.

Although the above calculations were explained to illustrate the use of the matrix formulas presented in Chapter 4, in practice there is no need to carry out these calculations since SAS has certain built-in commands that automatically perform the necessary matrix computations for multiple linear regression. We discuss these next.

The PROC REG Command

The primary SAS command for multiple regression computations is the `proc reg` command. Recall that this command was also used in Chapter 3 for straight line regression. For an illustration, suppose we wish to obtain the estimated regression function of Y on X_1, X_2, X_3, and X_4. Suppose that the data are in a SAS data file named **filename.ssd** and that the model is given by

$$\mu_Y(x_1, x_2, x_3, x_4) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 \qquad \text{(S4.4.1)}$$

The command for performing the necessary computations is as follows.

THE PROC REG COMMAND IN SAS

```
proc reg data=my.filename;
model Y=X1 X2 X3 X4;
run;
```

Remember to execute the usual `libname my 'b:\'` command before using the *nickname* `my`. The first statement above asks SAS to run a regression using the data in the file **filename.ssd** in the directory `b:\` whose *nickname* is `my`. The second statement tells SAS what the model is. Note that the model statement in the above command implies that the regression function is the one given in (S4.4.1). Note also that, in this statement, you must use variable names that are exactly the same as the names of the variables in the SAS data file **filename.ssd**. Press the `F10` key to execute program. The results appear in the `OUTPUT` window.

We illustrate the preceding command by running a regression of GPA on SATmath, SATverb, HSmath, and HSengl, for the gpa data in the SAS data file **gpa.ssd** on the data disk. The command and the output are as follows.

SAS COMMAND FOR REGRESSION OF GPA DATA

```
proc reg data=my.gpa;
model gpa=satmath satverb hsmath hsengl;
run;
```

```
--------------------------------------------------------------------------------
Model: MODEL1
Dependent Variable: GPA

                        Analysis of Variance

                          Sum of         Mean
Source          DF       Squares       Square      F Value        Prob>F

Model            4       6.26432      1.56608       21.721        0.0001
Error           15       1.08150      0.07210
C Total         19       7.34582

        Root MSE       0.26851     R-square       0.8528
        Dep Mean       2.59300     Adj R-sq       0.8135
        C.V.          10.35535

                         Parameter Estimates

                   Parameter       Standard     T for H0:
Variable  DF        Estimate          Error    Parameter=0     Prob > |T|

INTERCEP   1        0.161550     0.43753205          0.369         0.7171
SATMATH    1        0.002010     0.00058444          3.439         0.0036
SATVERB    1        0.001252     0.00055152          2.270         0.0383
HSMATH     1        0.189440     0.09186804          2.062         0.0570
HSENGL     1        0.087564     0.17649628          0.496         0.6270
--------------------------------------------------------------------------------
```

The only quantities in the above output that concern us at the present time are the point estimates of β_0, β_1, β_2, β_3, and β_4. These are given in the output under the label `Parameter Estimate` in the section titled `Parameter Estimates`. The estimate corresponding to the row labeled `INTERCEP` is the estimate of β_0. The estimates for $\beta_1, \ldots, \beta_4$ are listed along rows labeled by the corresponding predictor variable names. For instance, the estimate of β_2 is listed along the row labeled `SATVERB` because the predictor variable X_2 is named SATVERB , etc. Thus, we get

$\hat{\beta}_0 = 0.161550$ (INTERCEPT)
$\hat{\beta}_1 = 0.002010$ (estimated coefficient of SATMATH)
$\hat{\beta}_2 = 0.001252$ (estimated coefficient of SATVERB)
$\hat{\beta}_3 = 0.189440$ (estimated coefficient of HSMATH)
$\hat{\beta}_4 = 0.087564$ (estimated coefficient of HSENGL)

It follows that the estimated regression function of Y on X_1, X_2, X_3, and X_4 is

$$\hat{\mu}_Y(x_1, x_2, x_3, x_4) = 0.161550 + 0.002010x_1 + 0.001252x_2 + 0.189440x_3 + 0.087564x_4$$

where GPA $= Y$, SATmath $= X_1$, SATverb $= X_2$, HSmath $= X_3$, and HSengl $= X_4$. The standard errors of the parameter estimates $\hat{\beta}_i$ are in the column labeled Standard Error . They are

$$SE(\hat{\beta}_0) = 0.43753205$$
$$SE(\hat{\beta}_1) = 0.00058444$$
$$SE(\hat{\beta}_2) = 0.00055152$$
$$SE(\hat{\beta}_3) = 0.09186804$$
$$SE(\hat{\beta}_4) = 0.17649628$$

These along with the point estimates of the β_i can be used to obtain confidence intervals.

The estimate of σ in the output is indicated by the label Root MSE . Thus, we have $\hat{\sigma} = 0.26851$.

Problems

Problems S4.4.1 – S4.4.4 refer to Task 4.4.1. The data are given in Table 4.4.4 and are also stored in the file **table444.ssd** on the data disk. The regression function is

$$\mu_Y(x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

where Y = strength, X_1 = temp, and X_2 = pressure.

S4.4.1 Examine the contents of the data file and print the data it contains.

S4.4.2

(a) Give the SAS/IML commands to obtain each of the following matrices: $X, y, X^TX, (X^TX)^{-1}, X^Ty$.

(b) Give the SAS/IML commands to compute $\hat{\beta}$ and exhibit the result.

(c) Give the SAS/IML commands to compute $\hat{e} = y - X\hat{\beta}$ and print it.

(d) Give the SAS/IML commands to compute $\hat{\sigma}$ and exhibit its value.

(e) Use the proc reg command to obtain $\hat{\beta}$ and $\hat{\sigma}$.

S4.4.3 Suppose the regression function of Y on X_1 is given by

$$\mu_Y^{(A)}(x_1) = \beta_0^A + \beta_1^A x_1.$$

Find $\hat{\beta}_0^A$, $\hat{\beta}_1^A$, and $\hat{\sigma}_{Y|X_1}$ using matrix commands.

S4.4.4 In Problem S4.4.3, compute the required quantities using the proc reg command.

4.5 Residual Analysis

In this section we explain how SAS can be used to perform the calculations needed for residual analysis discussed in Section 4.5. Specifically, we consider SAS commands that can be used to compute residuals, fits, standardized residuals, hat values, and nscores for multiple linear regression. We use the electric bill data of Example 4.5.1 to illustrate the commands. The data are given in Table 4.5.1 and are also stored in the SAS data file **electric.ssd** on the data disk. As usual, you should first examine the contents of the data and confirm that it contains the response variable Y = bill, and the predictor variables X_1 = income, X_2 = persons, and X_3 = area, respectively.

An extended version of the proc reg command can be used to obtain the fitted values $\hat{\mu}_Y(x_1, x_2, x_3)$, the residuals $\hat{e}_i$, the standardized residuals r_i, and the hat values $h_{i,i}$, in addition to the point estimates of β_i and σ. The command is

DIAGNOSTICS COMMAND

```
proc reg data=my.electric;
model bill=income persons area;
output out=diagnstc p=fits r=residual student=stdresid h=hatvals;

proc rank normal=blom data=diagnstc out=newdata;
var stdresid;
ranks nscores;
run;
```

The first set of (three) statements is the same as that in the DIAGNOSTICS COMMAND in Section 3.5, except the dataset here (i.e., electric) has three predictor variables. The second set of (four) statements is the same as the command to compute nscores, explained in Section 3.5. Print the dataset newdata and verify that the results agree (within rounding error) with the corresponding results in Exhibit 4.5.1 (in Exhibit 4.5.1 the hat values were not printed). Also, using the above results you can obtain the plots in Example 4.5.1.

Checking Gaussian Assumptions

Next we exhibit SAS commands that can be used to help determine if a k-variable population is Gaussian. To illustrate, we again use the data in the SAS data file **electric.ssd**. These data were obtained by simple random sampling from a 4-variable population, where the variables are Y = bill, X_1 = income, X_2 = persons, and X_4 = area. To help determine if assumptions **(B)** apply, we examine four different linear combinations of these variables. You should try others, including Y, X_1, X_2, and X_3 themselves! For each of the four linear combinations, we construct Gaussian rankit-plots. The command and the output are as follows.

COMMAND TO OBTAIN LINEAR COMBINATIONS OF VARIABLES AND COMPUTE NSCORES OF THE RESULTS

```
data linear;
set my.electric;
w1 = 5*bill + income + 1500*persons + 2*area;
w2 = 5*bill + income - 1500*persons + 2*area;
w3 = 5*bill + income - 1500*persons - 2*area;
w4 = -5*bill + income - 1500*persons - 2*area;
keep w1 w2 w3 w4;

proc rank normal=blom data=linear out=newdata;
var w1 w2 w3 w4;
ranks nscorew1 nscorew2 nscorew3 nscorew4;
run;
```

The first group of (seven) statements instructs SAS to form a temporary dataset called linear which is to contain w1, w2, w3, and w4, the four linear combinations to be constructed. The statement keep w1 w2 w3 w4 asks SAS to keep only the variables w1, w2, w3, and w4 in the dataset linear. The coefficients which make up the linear combinations being examined are chosen so that no single variable dominates the value of the linear combination. This is especially important when the different variables forming the linear combinations take on values that are not commensurate with each other as is the case here – the sample values of Y (bill) range between \$96.00 and \$1,272.00 whereas the sample values of X_2 (persons) range between 1 and 7.

The second group of (four) statements tells SAS to create a temporary dataset called **newdata**, which will consist of the variables w1, w2, w3, w4, and their corresponding nscores. The names nscorew1, nscorew2, nscorew3, and nscorew4 are the names we have given to the variables containing the nscores of w1, w2, w3, and w4, respectively. We print the dataset newdata and get

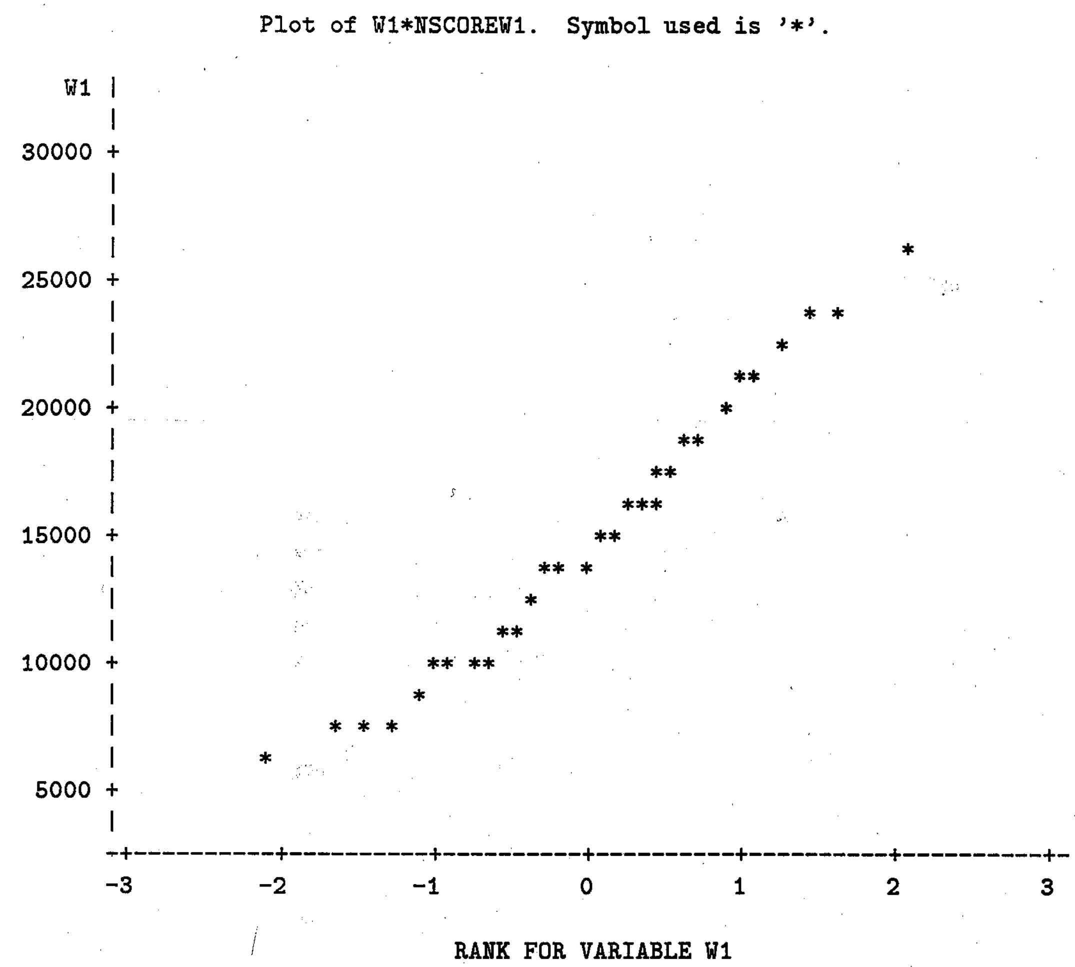
OBS	W1	W2	W3	W4	NSCOREW1	NSCOREW2	NSCOREW3	NSCOREW4
1	9680	3680	-960	-3240	-0.76335	-1.10289	-0.33553	0.97721
2	7190	4190	-130	-1690	-1.67015	-0.76335	-0.11000	1.67015
3	13300	7300	420	-6060	-0.25902	0.25902	0.25902	0.03660
4	11760	8760	1400	-3880	-0.41406	0.49523	0.97721	0.57981
5	16250	7250	-1710	-7230	0.33553	0.18400	-0.49523	-0.25902
6	17550	5550	-3210	-9570	0.57981	-0.25902	-1.10289	-0.86532

7	7810	4810	1490	-2950	-1.24896	-0.49523	1.10289	1.10289
8	7590	4590	-10	-1450	-1.42802	-0.57981	0.03660	2.09135
9	13610	7610	1330	-6110	0.03660	0.33553	0.66876	-0.03660
10	23130	8130	-2510	-13550	1.42802	0.41406	-0.97721	-1.24896
11	6810	3810	210	-1830	-2.09135	-0.86532	0.11000	1.42802
12	13560	4560	-2160	-6360	-0.03660	-0.66876	-0.76335	-0.18400
13	16350	13350	3150	-5610	0.41406	1.42802	2.09135	0.18400
14	21420	420	-6660	-15060	0.97721	-1.67015	-2.09135	-2.09135
15	19210	13210	1370	-7390	0.76335	1.24896	0.86532	-0.33553
16	9780	3780	-980	-3740	-0.66876	-0.97721	-0.41406	0.71605
17	22860	13860	1340	-11020	1.24896	2.09135	0.76335	-1.10289
18	11500	5500	-740	-4460	-0.57981	-0.33553	-0.18400	0.49523
19	9620	6620	580	-2180	-0.86532	-0.03660	0.49523	1.24896
20	13420	10420	1660	-3740	-0.14700	0.76335	1.24896	0.71605
21	24160	6160	-4320	-14760	1.67015	-0.11000	-1.67015	-1.42802
22	11730	5730	330	-5190	-0.49523	-0.18400	0.18400	0.25902
23	15180	9180	1220	-6340	0.18400	0.66876	0.57981	-0.11000
24	14800	8800	520	-5840	0.11000	0.57981	0.41406	0.11000
25	17060	5060	-2340	-9420	0.49523	-0.41406	-0.86532	-0.76335
26	18940	12940	2140	-7460	0.66876	1.10289	1.42802	-0.41406
27	21560	12560	440	-10360	1.10289	0.97721	0.33553	-0.97721
28	12750	6750	-50	-4850	-0.33553	0.03660	-0.03660	0.33553
29	9050	50	-3510	-4470	-1.10289	-2.09135	-1.24896	0.41406
30	25980	10980	-2100	-14820	2.09135	0.86532	-0.66876	-1.67015
31	19420	13420	2780	-7780	0.86532	1.67015	1.67015	-0.57981
32	9600	3600	-1720	-3280	-0.97721	-1.24896	-0.57981	0.86532
33	13420	1420	-3700	-7660	-0.14700	-1.42802	-1.42802	-0.49523
34	16020	7020	-780	-8460	0.25902	0.11000	-0.25902	-0.66876

Next we obtain the rankit plots of the above linear combinations by plotting the values of w1, w2, w3, and w4, against the corresponding nscores. The command and output for the rankit-plot of w1 are given below.

COMMAND FOR RANKIT PLOTS

```
options linesize=75 pagesize=35;
proc plot data=newdata;
plot w1*nscorew1='*';
run;
```

NOTE: 3 obs hidden.

You can try different linesize and pagesize options to get the scale of the graph to your liking. Alternatively, you can experiment with the hpos and the vpos options of the plot command. We leave it to you to obtain the rankit plots for w2, w3, w4, and perhaps a few more linear combinations. These plots should help you evaluate the validity of assumptions (B) for the electric data.

Problems

S4.5.1 For Problem 4.5.1 in the textbook, use SAS commands discussed in this section to work parts (a) through (f) below. The model is

$$\mu_Y(x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

where Y = `strength`, X_1 = `temp`, X_2 = `pressure`, and the data are stored in the SAS data file **table444.ssd**.

(a) Regress Y on X_1, X_2.

(b) Compute the fits, $\hat{\mu}_Y(x_{i,1}, x_{i,2}, x_{i,3}) = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \beta_3 x_{i,3}$.

(c) Compute the residuals, $\hat{e}_i$.

(d) Compute the standardized residuals, r_i.

(e) Compute the nscores $z_i^{(n)}$ of the standardized residuals.

(f) Plot the standardized residuals against the fits, against the Y values, against the nscores, against X_1, and against X_2.

S4.5.2 For Problem 4.5.2 in the textbook, use SAS commands discussed in this section to work parts (a) through (f) below. The model is

$$\mu_Y(x_1, x_2, x_3, x_4) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta 4 x_4$$

where Y = GPA, X_1 = SATmath, X_2 = SATverb, X_3 = HSmath, and X_4 = HSengl.

(a) Regress Y on X_1, X_2, X_3, and X_4.

(b) Compute the fits, $\hat{\mu}_Y(x_{i,1}, x_{i,2}, x_{i,3}, x_{i,4})$.

(c) Compute the residuals, $\hat{e}_i$.

(d) Compute the standardized residuals, r_i.

(e) Compute the nscores $z_i^{(n)}$ of the standardized residuals.

(f) Plot the standarized residuals against the fits, against the Y values, against the nscores, and against each of X_1, X_2, X_3, and X_4.

S4.5.3 In Problem S4.5.2, obtain rankit plots of several linear combinations of the variables Y, X_1, X_2, X_3, and X_4.

4.6 Confidence Intervals

Formulas for computing point estimates and the corresponding standard errors for the regression coefficients $\beta_0, \ldots, \beta_k$, which are the ingredients needed to compute confidence intervals, are given in Sections 4.4 and 4.6 of the textbook. In Section 4.4 of this manual we showed how these quantities can be obtained using the SAS command `proc reg` . The present version of SAS does not have a built-in command that will calculate general confidence intervals for all regression parameters for user specified values of $1 - \alpha$. In Section 4.6 of the textbook, you learned how these computations can be done using matrices, but this requires a significant amount of tedious work. To make it easier to obtain point estimates, their standard errors, and confidence intervals for general linear combinations

$$\theta = a_0\beta_0 + a_1\beta_1 + \cdots + a_k\beta_k$$

for values of a_i that you specify, we have supplied, on the data disk, a macro named **cilinear**, which stands for **c**onfidence **i**ntervals for **linear** combinations of the β_i. In this section we show how to use this macro. The SAS statements for this macro are stored in the two files **cilinear.mac** and **cilinear.sas**, respectively, on the data disk.

To illustrate the use of this macro, we compute a 90% two-sided confidence interval for β_3 as required in part 1 of Task 4.6.1. The model is

$$\mu_Y(x_1, x_2, x_3, x_4) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4$$

where Y = GPA, X_1 = SATmath, X_2 = SATverb, X_3 = HSmath, and X_4 = HSengl, and assumptions (**B**) are presumed to apply.

To start the macro, invoke SAS, and on the `Command` line of the `PROGRAM EDITOR` window type

```
include 'b:\macro\cilinear.mac'
```

and press `Enter` . This brings the following SAS statements to the screen.

```
-----------------------------------------------------------------------

00001 Title 'Confidence interval for theta';
00002 libname my 'b:\';proc iml; reset nolog;
00003
00004 ****** On line  00007 enter the name of the SAS data file
00005 ****** that contains the data you want to use:
```

```
00006 use
00007                   my.filename
00008 ;
00009
00010 ****** On line 00013 enter the name of the response variable
00011 ****** exactly as it appears in the data file;
00012 read all var {
00013                   response variable
00014 } into yvar;
00015
00016 ****** Use lines 00022 to 00024 to enter the names of the predictor
00017 ****** variables exactly as they appear in the data file. You can
00018 ****** enter as many variable names on a line as will fit.
00019 ****** Leave at least one space between variable names.
00020 ****** Do not use any punctuation marks;
00021 read all var {
00022                   predictor1  predictor2  predictor3
00023                   predictor4  ... etc.
00024
00025 } into xvar;
00026
00027 ****** On line  00029 enter the confidence coefficient;
00028 cc=
00029                   0.95
00030 ;
00031 ****** On line 00038 enter the vector a. The first element of the
00032 ****** vector  a  must correspond to the intercept (which is
00033 ****** assumed to be present in the model). The order of the
00034 ****** remaining coefficients in the vector  a  must correspond
00035 ****** to the order in which you entered the names of the predictor
00036 ****** variables on lines 00022--00024;
00037 a={
00038                   0  0  0  1  0
00039
00040 };%include 'b:\macro\cilinear.sas';
------------------------------------------------------------------------
```

You must enter the following information on appropriate lines in the PROGRAM EDITOR window.

(1) On line 00007 enter the name of the file where the data are located. For this illustration, my.gpa replaces my.filename.

(2) On line 00013 enter the name of the response variable as it appears in the data

file. If you are not sure, then use proc contents to find out what the name is for the response variable. For this illustration, the name gpa should replace the words response variable.

(3) On lines 00022, 00023, and 00024 you must replace the words

```
00022                predictor1  predictor2  predictor3
00023                predictor4  ... etc.
00024
```

with the names of the predictor variables as given in the data file. If you are not sure, then use proc contents to find out what their names are. For this illustration, the names are satmath, satverb, hsmath, and hsengl. You can use one, two, or all three lines 00022, 00023, and 00024 to enter these names. Leave at least one space between the variable names and do not use any punctuation marks. One possible way to enter these names is given below.

```
00022                satmath     satverb     hsmath
00023                hsengl
00024
```

Another way is as follows.

```
00022                satmath     satverb
00023                hsmath
00024                hsengl
```

(4) On line 00029 enter the confidence level you want to use. For this illustration, 0.90 replaces 0.95.

(5) On line 00038 enter the elements in the vector a (i.e., the coefficients a_i). Leave at least one space between each element of a. For this illustration, the correct numbers are 0 0 0 1 0 .

Press the F10 key and the following results will appear in the OUTPUT window.

```
---------------------------------------------------------------------------
                    Confidence interval for theta


     The point estimate of theta is        0.1894

     The standard error of this estimate is       0.0919


     For a two-sided  90% confidence interval for theta

     the lower confidence bound is      0.0284    and

     the upper confidence bound is      0.3505
---------------------------------------------------------------------------
```

Thus $\hat{\beta}_3 = 0.1894$, $SE(\hat{\beta}_3) = .0919$, and the confidence statement is

$$C[0.0284 \leq \beta_3 \leq 0.3505] = 0.90$$

Verify that these results are the same as those obtained in Task 4.6.1.

In some problems a confidence interval for σ may be of interest. This can be obtained by using the macro **sgmaconf** discussed in Section 3.6 of this manual.

Problems

S4.6.1 Use SAS commands to obtain the results in Exhibit 4.6.2 in the textbook, and also work Problems 4.6.6 through 4.6.8. Use macros when SAS commands are not available.

4.7 Tests

In Section 3.7 of this manual we explained how to use the macro **test** to do the computing needed for statistical tests in straight line regression discussed in Boxes 3.7.1 to 3.7.4 in the textbook. In this section we describe a macro, named **testmult**, that can be used to perform the tests described in Box 4.7.1 for multiple regression. This macro

computes the test statistic t_C and the P-value for tests explained in Box 4.7.1 for linear combinations

$$\theta = a_0\beta_0 + a_1\beta_1 + \cdots + a_k\beta_k$$

The Macro TESTMULT for Testing θ

The SAS statements for the macro **testmult** are stored in the files **testmult.mac** and **testmult.sas** on the data disk. We illustrate how to use this macro by applying it to the GPA data in Example 4.7.1. These data are also in the SAS data file **gpa.ssd** on the data disk. We use these data to test

$$\text{NH: } \mu_Y(594, 665, 3.42, 2.70) \leq 2.5 \quad \text{versus} \quad \text{AH: } \mu_Y(594, 665, 3.42, 2.70) > 2.5$$

Note that

$$\mu_Y(594, 665, 3.42, 2.70) = \beta_0 + 594\beta_1 + 665\beta_2 + 3.42\beta_3 + 2.70\beta_4 = \theta$$

So $a_0 = 1$, $a_1 = 594$, $a_2 = 665$, $a_3 = 3.42$, $a_4 = 2.70$, and $q = 2.5$. The test is

$$\text{NH: } \theta \leq 2.5 \quad \text{versus} \quad \text{AH: } \theta > 2.5$$

To use the macro, invoke SAS, and on the *Command* line of the `PROGRAM EDITOR` window type `include 'b:\macro\testmult.mac'` and press **Enter**. This brings the following statements to the `PROGRAM EDITOR` window.

```
---------------------------------------------------------------------------

00001 Title 'Test for theta';
00002 libname my 'b:\';proc iml; reset nolog; option nodate;
00003
00004 ****** On line  00007 enter the name of the SAS data file
00005 ****** that contains the data you want to use;
00006 use
00007                    my.filename
00008 ;
00009
00010 ****** On line 00013 enter the name of the response variable
00011 ****** exactly as it appears in the data file;
00012 read all var {
00013                    response variable
00014 } into yvar;
00015
00016 ****** Use lines 00022 to 00024 to enter the names of the predictor
```

```
00017 ****** variables exactly as they appear in the data file. You can
00018 ****** enter as many variable names on a line as will fit.
00019 ****** Leave at least one space between variable names.
00020 ****** Do not use any punctuation marks;
00021 read all var {
00022                     predictor1  predictor2  predictor3
00023                     predictor4  ... etc.
00024
00025 } into xvar;
00026
00027 ****** On line  00029 enter the value of q;
00028 q=
00029                     0
00030 ;
00031 ****** On line 00038 enter the vector a. The first element of the
00032 ****** vector  a  must correspond to the intercept (which is
00033 ****** assumed to be present in the model). The order of the
00034 ****** remaining coefficients in the vector  a  must correspond
00035 ****** to the order in which you entered the names of the predictor
00036 ****** variables on lines 00022--00024;
00037 a={
00038                     1  594  665  3.42  2.70
00039
00040 };%include 'b:\macro\testmult.sas';
-----------------------------------------------------------------------------
```

You must enter the following quantities on the specified lines of the `PROGRAM EDITOR` window.

(1) On line 00007 enter the name of the file that contains the data. The file is assumed to be a SAS data file that is on the data disk which is in drive `B` . Thus replace `my.filename` by `my.gpa`.

(2) On line 00013 enter the name of the response variable exactly as it appears in the data file. For this illustration, the name is `gpa` , so replace the words `response variable` with `gpa` .

(3) Use lines 00022, 00023, and 00024 to enter the names of the predictor variables exactly as given in the data file. You may use one, two, or all three of these lines depending on how much space you need to enter the required variable names. For this illustration the predictor variable names are `satmath`, `satverb`, `hsmath`, and `hsengl`, respectively, so replace the following lines


```
00022                 predictor1  predictor2  predictor3
00023                 predictor4  ... etc.
00024
```

with

```
00022                 satmath     satverb     hsmath
00023                 hsengl
00024
```

There must be at least one blank space between variable names. Do not use any punctuation marks. Another way to enter these names is as follows.

```
00022                 satmath     satverb
00023                 hsmath
00024                 hsengl
```

(4) On line 00029 enter the value of q. For this illustration the value of q is 2.5, so replace the number 0 on this line with the number 2.5.

(5) On line 00038 enter the elements of the vector $\boldsymbol{a}$. Make sure that these coefficients correspond to the order in which you entered the names for the predictor variables on lines 00022–00024. For this illustration, the required coefficients are 1 594 665 3.42 2.70 . These values are already present on line 00038 so no change is required here for this problem.

After these quantities have been entered and checked, press the `F10` key to execute the macro. The following result will be displayed in the `OUTPUT` window.

```
-----------------------------------------------------------------------------
                              Test for theta

For NH: theta      =   2.500 vs AH: theta not =   2.500, P value = 0.0011

For NH: theta < or =   2.500 vs AH: theta     >   2.500, P value = 0.0006

For NH: theta > or =   2.500 vs AH: theta     <   2.500, P value = 0.9994
-----------------------------------------------------------------------------
```

Since we are testing NH: $\theta \leq 2.5$ versus AH: $\theta > 2.5$, the P-value is 0.0006. Hence NH would be rejected at any of the usual α levels. You should check the above results against the calculations shown in Example 4.7.1.

Problems

S4.7.1 For the GPA data in the SAS data file **gpa.ssd** on the data disk, test the following using $\alpha = .05$. State your conclusion for each.

(a) NH: $\beta_1 = 0.003$ versus AH: $\beta_1 \neq 0.003$.

(b) NH: $\beta_2 \leq 0.001$ versus AH: $\beta_2 > 0.001$.

(c) NH: $\mu_Y(500, 615, 3.10, 2.90) \leq 2.5$ versus AH: $\mu_Y(500, 615, 3.10, 2.90) > 2.5$.

S4.7.2 Work part (b) of Problem 4.7.1 in the textbook using the macro discussed in this section.

4.8 Analysis of Variance

In Section 4.8 of the textbook we discussed the quantities displayed in an analysis of variance table and how these quantities can be used for making inferences in regression. An ANOVA table can be computed with the SAS `proc reg` command. We demonstrate this by using Example 4.8.1. The data are given in Table 4.4.3 and are also stored in the SAS data file **gpa.ssd** on the data disk. The command and the corresponding output are as follows.

PROC REG COMMAND

```
proc reg data=my.gpa;
model gpa=satmath satverb hsmath hsengl;
run;
```

```
Model: MODEL1
Dependent Variable: GPA
```

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	4	6.26432	1.56608	21.721	0.0001
Error	15	1.08150	0.07210		
C Total	19	7.34582			

Root MSE	0.26851	R-square	0.8528
Dep Mean	2.59300	Adj R-sq	0.8135
C.V.	10.35535		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
INTERCEP	1	0.161550	0.43753205	0.369	0.7171
SATMATH	1	0.002010	0.00058444	3.439	0.0036
SATVERB	1	0.001252	0.00055152	2.270	0.0383
HSMATH	1	0.189440	0.09186804	2.062	0.0570
HSENGL	1	0.087564	0.17649628	0.496	0.6270

The analysis of variance in the preceding output is the same as the one in Table 4.8.2 in the textbook, except the one above contains an additional column `Prob>F` which gives the P-value for the analysis of variance F-test.

Problems

S4.8.1 In Problem 4.8.1 in the textbook, exhibit an Analysis of Variance.

S4.8.2 For the data in the SAS data file **grocery.ssd**, compute the ANOVA table given in Problem 4.8.2 of the textbook.

S4.8.3 Calculate the ANOVA table in Problem 4.8.3 in the textbook.

4.9 Comparison of Two Regression Functions (Nested Case)

Since SAS does not directly compute confidence intervals for ratios of standard deviations, we have supplied a macro on the data disk for this purpose. This macro is called **ratiosgm**, which stands for **ratio** of **sigmas**. The SAS statements for this macro are stored in the two files **ratiosgm.mac** and **ratiosgm.sas**.

Suppose we have two models, model-A and model-B, given by

$$\text{model-A:}\quad \mu_Y^{(A)}(x_1,\ldots,x_k) = \beta_0^A + \beta_1^A x_1 + \cdots + \beta_k^A x_k$$

with standard deviation σ_A, and

$$\text{model-B:}\quad \mu_Y^{(B)}(x_1,\ldots,x_m) = \beta_0^B + \beta_1^B x_1 + \cdots + \beta_m^B x_m$$

with standard deviation σ_B. Thus, model-A is the *full model* and model-B is a *submodel*. To compute a confidence interval for σ_A and/or σ_B you can use the macro **sgmaconf** discussed in Section 3.6 of this manual. However, the macro **ratiosgm** will compute the following.

- A confidence interval for σ_A, the subpopulation standard deviation for model-A, with confidence coefficient $1-\alpha_A$ specified by the investigator.
- A confidence interval for σ_B, the subpopulation standard deviation for model-B, with confidence coefficient $1-\alpha_B$ specified by the investigator.
- A confidence interval for σ_B/σ_A, with confidence coefficient greater than or equal to $1-\alpha_A-\alpha_B$ (this uses the Bonferroni method).

You must input the following information.

(1) The estimate of σ_A for model-A, the degrees of freedom associated with this estimate (these can be obtained from an appropriate ANOVA table), and the confidence coefficient $1-\alpha_A$ for σ_A.

(2) The estimate of σ_B for model-B, the degrees of freedom associated with this estimate (these can be obtained from an appropriate ANOVA table), and the confidence coefficient $1-\alpha_B$ for σ_B.

To illustrate the use of this macro we refer to Example 4.9.4. In that example, we want to determine how good model-A is for predicting Y (for this, we compute a confidence interval for σ_A), how good model-B is for predicting Y (for this, we compute a confidence interval for σ_B), and how much better model-A is than model-B for predicting Y (for this, we compute a confidence interval for σ_B/σ_A). The two models are

$$\text{model-}A:\quad \mu_Y^{(A)}(x_1,x_2,x_3,x_4) = \beta_0^A + \beta_1^A x_1 + \beta_2^A x_2 + \beta_3^A x_3 + \beta_4^A x_4$$

and

$$\text{model-}B:\quad \mu_Y^{(B)}(x_3,x_4) = \beta_0^B + \beta_3^B x_3 + \beta_4^B x_4$$

The following quantities, which are needed as input to the macro, are given in Exhibit 4.9.1:

(1) $\hat\sigma_A = 0.2685$ with degrees of freedom 15

(2) $\hat\sigma_B = 0.3771$ with degrees of freedom 17.

As you know, these quantities can be obtained from appropriate ANOVA tables.

Suppose we want 90% confidence intervals for σ_A and σ_B, i.e., $\alpha_A = 0.10$ and $\alpha_B = 0.10$. This will lead to a confidence interval for σ_B/σ_A with confidence coefficient greater than or equal to $1-\alpha_A-\alpha_B = 0.80$. To use the macro, invoke SAS, and on the `Command` line of the `PROGRAM EDITOR` window type `include 'b:\macro\ratiosgm.mac'` . The following SAS statements will appear in that window.

```
------------------------------------------------------------------------

00001 Title 'Confidence intervals for sigma(A), sigma(B), sigma(B)/sigma(A)';
00002 proc iml;
00003
00004 ****** On line 00007 enter the confidence
00005 ****** coefficient for sigma(A);
00006 ca=
00007                    0.95
00008 ;
00009 ****** On line 00012 enter the confidence
00010 ****** coefficient for sigma(B);
00011 cb=
00012                    0.95
00013 ;
00014 ****** On line 00016 enter the estimate of sigma(A);
00015 sa=
00016                    10.00
00017 ;
00018 ****** On line 00020 enter the degrees of freedom for sigma(A);
00019 dfa=
00020                    15
00021 ;
00022 ****** On line 00024 enter the estimate of sigma(B);
00023 sb=
```

```
00024                       30.00
00025 ;
00026 ****** On line 00028 enter the degrees of freedom for sigma(B);
00027 dfb=
00028                       25
00029
00030 ;%include 'b:\macro\ratiosgm.sas';
```

Enter the following information on the indicated lines to replace the quantities there. On line 00007 enter 0.90; on line 00012 enter 0.90; on line 00016 enter 0.2685; on line 00020 enter 15; on line 00024 enter 0.3771; on line 00028 enter 17. Press the F10 key to execute the macro. The results displayed in the OUTPUT window are given below.

```
        Confidence intervals for sigma(A), sigma(B), sigma(B)/sigma(A)


         For a two-sided  90.0% confidence interval for sigma(A)

         the lower confidence bound is    0.2080 and

         the upper confidence bound is    0.3859


         For a two-sided  90.0% confidence interval for sigma(B)

         the lower confidence bound is    0.2960 and

         the upper confidence bound is    0.5280


         For a two-sided confidence interval for sigma(B)/sigma(A)
         with confidence coefficient greater than or equal to  80%

         the lower confidence bound is    0.7671 and

         the upper confidence bound is    2.5385
```

Thus we obtain the following.

$$\hat{\sigma}_A = 0.2685, \quad C[0.2080 \le \sigma_A \le 0.3859] = 0.90$$

$$\hat{\sigma}_B = 0.3771, \quad C[0.2960 \le \sigma_B \le 0.5280] = 0.90,$$

and

$$C[0.7671 \le \sigma_B/\sigma_A \le 2.5385] \ge 0.80$$

Problems

S4.9.1 Work parts (a) and (b) of Problem 4.9.1 in the textbook using the macro discussed in this section.

S4.9.2 In Problem 4.9.1 in the textbook, compute a two-sided confidence interval for σ_B/σ_A with confidence coefficient greater than or equal to 95%.

4.10 Comparison of Two Multiple Regression Models (Non-nested Case)

To compute confidence intervals for σ_A, σ_B, and σ_B/σ_A for the non-nested case, you can use the macro **ratiosgm** discussed in Section 4.9 of this manual.

4.11 Lack-of-Fit

As explained in Section 4.11 of the textbook, a computer is a practical necessity for obtaining confidence intervals for the lack-of-fit constants θ_i since tedious matrix calculations are involved. We have written a macro named **lackfit** for this purpose. Besides computing confidence intervals for the lack-of-fit constants, this macro will also output the P-value for a traditional lack-of-fit test. The macro commands are stored in the files **lackfit.mac** and **lackfit.sas** on the data disk.

To illustrate the use of this macro, we refer to Example 4.11.3 where an investigator wants to examine the relationship between blood pressure (Y) and age (X) to determine if the model $P_Y(x) = \beta_0 + \beta_1 x$ is close enough to the unknown regression function $\mu_Y(x)$

to be useful for predicting blood pressure using age. The data are in the file **bp.ssd** on the data disk and are also exhibited in Table 4.11.2.

To use the macro, invoke SAS, and on the `Command` line of the `PROGRAM EDITOR` window type `include 'b:\macro\lackfit.mac'` . This brings the following statements to the `PROGRAM EDITOR` window.

```
--------------------------------------------------------------------------
00001 Title 'Lack-of-fit Analyses';
00002 libname my 'b:\';data rawdata(keep= yvar xvar);
00003
00004 ****** On line 00007 enter the name of the SAS data file
00005 ****** that contains the data you want to use;
00006 set
00007                            my.filename
00008 ;
00009 ****** On line 00012 enter the name of the response variable, and
00010 ****** on line 00014 enter the name of the predictor variable;
00011 rename
00012                       response variable
00013 = yvar
00014                       predictor variable
00015
00016 = xvar;proc iml;
00017
00018 ****** On line 00020 enter the confidence coefficient;
00019 cc=
00020                            0.95
00021
00022 ;%include 'b:\macro\lackfit.sas';
--------------------------------------------------------------------------
```

Enter the following information on the specified numbered lines in the `PROGRAM EDITOR` window.

(1) On line `00007` you must input the name of the file that contains the data. You will note that the `libname` is `my` and so you will use `my.bp`.

(2) On line `00012` enter the name of the response variable. In the present example, you must replace the expression `response variable` by the actual name, `bp` , of the response variable, exactly as it appears in the SAS data file. If you are unsure about the name of the response variable, use `proc contents` to examine the names of the variables stored in the SAS data file under consideration.

(3) On line `00014` enter the name of the predictor variable. In the present example, you must replace the expression `predictor variable` by the actual name, `age` , of the predictor variable, exactly as it appears in the SAS data file. If you are unsure about the name of the predictor variable, use `proc contents` to examine the names of the variables stored in the SAS data file under consideration.

(4) On line `00020` enter the desired confidence coefficient. For this example the confidence coefficient is `0.95`. This value is already present and so does not need to be changed for this example.

After these quantities have been entered and checked press the `F10` key to execute the macro. The following results appear in the `OUTPUT` window.

```
--------------------------------------------------------------------------
                          Lack-of-fit Analyses

   The estimate of beta(0) is   63.0433
   The estimate of beta(1) is    1.7453

   The estimate of sigma (pure error) is    3.7657

   The estimate of the theta(1) is    1.5733
   The estimate of the theta(2) is   -1.0033
   The estimate of the theta(3) is   -2.2967
   The estimate of the theta(4) is    1.3100
   The estimate of the theta(5) is    0.4167

   The standard error of the estimate of theta(1) is    1.1213
   The standard error of the estimate of theta(2) is    1.4405
   The standard error of the estimate of theta(3) is    1.4157
   The standard error of the estimate of theta(4) is    1.3595
   The standard error of the estimate of theta(5) is    1.0871

   The confidence interval for theta(1) is   -1.6172  to    4.7639
   The confidence interval for theta(2) is   -5.1021  to    3.0955
```

```
The confidence interval for theta(3) is   -6.3248  to    1.7315
The confidence interval for theta(4) is   -2.5582  to    5.1782
The confidence interval for theta(5) is   -2.6764  to    3.5098

The sum of squares for lackfit is   56.8936 with df=   3

The sum of squares for pure error is  283.6167 with df=  20

The computed F value for the lack-of-fit test is    1.3373

The P-value for the lack-of-fit test is  0.290
-----------------------------------------------------------------------
```

Of course, these results are the same (except possibly for rounding errors) as those in Example 4.11.3 in the textbook.

Problems

S4.11.1 In part (b) of Exercise 4.12.2, use the macro **lackfit** and find $\hat{\theta}_i$, $SE(\hat{\theta}_i)$, and simultaneous confidence intervals for θ_i with confidence coefficient ≥ 0.95. Check your results against your answers obtained without using the macro.

Chapter 5

Diagnostic Procedures

5.1 Overview

There are no calculations in this section that require SAS.

5.2 Outliers

To examine a set of data for outliers as explained in Section 5.2 of the textbook, it is useful to examine the following:

(1) The fitted values

$$\hat{\mu}_Y(x_{i,1}, \ldots, x_{i,k})$$

(2) The residuals

$$\hat{e}_i = y_i - \hat{\mu}_Y(x_{i,1}, \ldots, x_{i,k})$$

(3) The standardized residuals

$$r_i = \frac{y_i - \hat{Y}(x_{i,1}, \ldots, x_{i,k})}{\hat{\sigma}\sqrt{1 - h_{i,i}}}$$

(4) The studentized deleted residuals

$$T_i = \frac{y_i - \hat{Y}_{-i}(x_{i,1}, \ldots, x_{i,k})}{\hat{\sigma}_{(-i)}/\sqrt{1 - h_{i,i}}}$$

As explained in Sections 3.5 and 4.5 of this manual, these quantities can be obtained using various optional commands available with `proc reg` . We refer to Example 5.2.1 to explain the use of these commands. In that example, an investigator is studying the relationship of insurance premiums (Y) with the ages (X_1) of cars and their prices (X_2), respectively. The data are given in Table 5.2.1 and are also stored in the files **premiums.ssd** and **premiums.dat** on the data disk. As usual, you should use the `proc contents` command to see what the file contains. The following commands are used to compute the four diagnostic statistics referred to above.

SAS COMMAND TO COMPUTE SOME REGRESSION DIAGNOSTICS

```
libname my 'b:\';
proc reg data=my.premiums;
model premium=age price;
output out=diagnstc
       p=fits
       r=residual
       student=stdresid
       rstudent=tresid;
proc print data=diagnstc;
run;
```

The only new command is `rstudent=tresid;` where `rstudent` is a SAS keyword that asks SAS to compute the studentized deleted residuals. Instead of the name `tresid` , you can use any *valid* name for the studentized deleted residuals. It is advisable to choose a name that helps you remember what has been computed. You should also note that the `output` statement on the fourth line of the above command actually extends all the way to the end of line eight. This is a long statement, and to make it easier to read the program, it has been broken up into several lines. However, the semicolon appears only at the end of the statement, and not at the end of each line.

The result of the `proc print` command appears in the OUTPUT window and is

----

Model: MODEL1
Dependent Variable: PREMIUM

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	2	2492087.0202	1246043.5101	708.495	0.0001
Error	33	58037.72979	1758.71908		
C Total	35	2550124.7500			

Root MSE	41.93708	R-square	0.9772
Dep Mean	485.58333	Adj R-sq	0.9759
C.V.	8.63643		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
INTERCEP	1	6.896788	24.51817037	0.281	0.7802
AGE	1	-5.099627	0.38940052	-13.096	0.0001
PRICE	1	0.039533	0.00120946	32.686	0.0001

OBS	PREMIUM	AGE	PRICE	FITS	RESIDUAL	STDRESID	TRESID
1	221	57	11804	182.87	38.1338	0.95962	0.95844
2	448	8	12926	477.10	-29.1040	-0.72149	-0.71615
3	515	6	14054	531.90	-16.8966	-0.41918	-0.41388
4	632	12	17486	636.98	-4.9762	-0.12178	-0.11995
5	48	47	8700	111.15	-63.1519	-1.57678	-1.61473
6	189	30	8570	192.71	-3.7062	-0.09163	-0.09025
7	581	34	18982	583.93	-2.9259	-0.07124	-0.07016
8	102	39	9198	171.64	-69.6364	-1.71939	-1.77448
9	404	33	14986	431.05	-27.0514	-0.65491	-0.64914
10	83	59	8473	40.98	42.0176	1.07656	1.07924
11	280	56	13891	270.47	9.5287	0.23852	0.23508
12	565	13	16127	578.15	-13.1512	-0.32131	-0.31690
13	1105	10	29480	1121.33	-16.3349	-0.43316	-0.42776
14	388	46	15868	399.62	-11.6244	-0.28516	-0.28115
15	435	2	10782	422.94	12.0571	0.30633	0.30208
16	309	11	8645	292.56	16.4359	0.41454	0.40927
17	322	17	9086	279.40	42.5996	1.06031	1.06237
18	741	32	22559	735.53	5.4651	0.13511	0.13309

19	500	34	14969	425.28	74.7203	1.80976	1.87775
20	626	1	14861	589.30	36.7021	0.91975	0.91754
21	1051	34	29733	1008.95	42.0543	1.12130	1.12584
22	845	4	22893	891.53	-46.5285	-1.17327	-1.18023
23	278	59	15198	306.84	-28.8421	-0.72822	-0.72294
24	333	56	16696	381.36	-48.3615	-1.21368	-1.22275
25	650	34	20411	640.42	9.5814	0.23453	0.23114
26	772	27	23128	783.53	-11.5273	-0.28539	-0.28138
27	477	19	16507	562.58	-85.5760	-2.07728	-2.19403
28	443	37	13704	359.97	83.0285	2.01678	2.12100
29	692	3	16472	642.79	49.2136	1.22453	1.23420
30	618	36	18422	551.59	66.4119	1.61684	1.65923
31	1050	7	27110	1042.94	7.0595	0.18290	0.18020
32	643	45	22968	685.41	-42.4087	-1.06941	-1.07182
33	116	46	9177	135.11	-19.1088	-0.47524	-0.46959
34	269	9	8977	315.89	-46.8884	-1.18467	-1.19221
35	259	38	10514	228.761	30.2385	0.74147	0.73631
36	491	16	13739	468.447	22.5526	0.55065	0.54476

You should compare these results with the entries in Exhibit 5.2.1.

Sometimes it may be advantageous to print the variables in a dataset in a different order than the order in which they occur in the dataset. This can be done with the `proc print` command as follows.

PROC PRINT COMMAND ARRANGING THE VARIABLES IN A SPECIFIED ORDER

```
proc print data=diagnstc;
var age price premium fits residual stdresid tresid;
run;
```

You can print the variables in any order you want by using a `var` statement with the variables listed in the order you want them to appear in the output. The output from the preceding command will have the `premium` column next to the column of `fits` so you can visually subtract the two columns and get the next column, the column of `residuals`. If you don't want to print all of the variables, just list those you want printed.

Problems

S5.2.1 Use the appropriate SAS commands to obtain the table of residuals, fits, etc., displayed in Exhibit 5.2.2. Note that Exhibit 5.2.2 uses the data in Table 5.2.1 (stored also in the file **premiums.ssd**), with the premium value 491 for the last observation changed to 1491. The following SAS statements may be used to change this value and create a modified dataset.

COMMAND TO CHANGE A VALUE IN A DATASET

```
libname my 'b:\';
data modified;
set my.premiums;
if _n_ = 36 then premium=1491;
run;
```

These statements ask SAS to create a temporary SAS dataset named `modified`, which is to contain a copy of the contents of the file **premiums.ssd** (this is done by the `set` statement), but changing the value 491 to 1491 for observation 36 (this is done by the `if` statement). You should print this dataset and examine the value of `premium` for observation 36. You can then use the dataset `modified` in the `proc reg` command to obtain the required diagnostic statistics.

5.3 Leverages or Hat-values

As discussed in Section 5.3, hat-values can be used as a measure of how typical or atypical the predictor values are (i.e., how typical or atypical the $X_1, X_2, \ldots, X_k$ values are), and they can be computed by specifying `h = hatvals` as part of the `output` statement within `proc reg`. Refer to Section 4.5 of this manual for illustrations.

5.4 Influential Observations – Cook's Distance and DFFITS

Recall that Cook's distance and/or DFFITS can be helpful in determining which (if any) values in a data set are influential observations. Values of Cook's distance and DFFITS can be computed using the appropriate optional SAS statements within `proc` reg . For illustration, we use the artificial data in Table 5.4.1, which are stored in the files **table541.ssd** and **table541.dat** on the data disk.

SAS COMMANDS FOR COMPUTING COOK'S DISTANCE AND DFFITS

```
proc reg data=my.table541;
model y=x;
output out=diagnstc cookd=cooksd dffits=dffits;
run;
```

The first two statements are the usual commands to obtain a regression analysis for the data in the file **table541.ssd**. The third statement tells SAS to create a temporary dataset with the name diagnstc which is to contain Cook's distances and DFFITS. The keywords cookd and dffits on the left of the = signs are SAS keywords and must appear exactly as indicated. However, the names on the right hand side of the = signs can be any *valid* name for variables. We have chosen the name cooksd for the variable whose values are Cook's distances for the observations, and the name dffits (same as the keyword!) for the name of the variable whose values are DFFITS for the observations. You can use other valid names if you wish. If you print the data set diagnstc just created, you will get the values of Cook's distance and DFFITS as given in Exhibit 5.4.1.

Problems

S5.4.1 Use the GPA data in the file **gpa.ssd** and exhibit the appropriate SAS commands and the answer for each problem.

(a) $h_{4,4}$. (b) $DFFITS_2$.
(c) Cook's distance c_9. (d) r_6.
(e) $\hat{e}_2$. (f) Studentized deleted residual T_7.

5.5 Ill-conditioning and Multicollinearity

As discussed in Section 5.5, if the columns of the X matrix are (approximately) linearly related, then multicollinearity exists and it may be very difficult to obtain reliable estimates of the β_i, etc. Several diagnostic measures have been suggested for detecting the

presence of approximate linear relationships among the predictor variables. One of the measures is the so called **variance inflation factor** (VIF). In SAS, this quantity can be computed for each predictor variable X_j using an option of the model statement, which is part of the proc reg command. We illustrate this option using insurance data in Table 5.2.1. These data are stored in the files **premiums.ssd** and **premiums.dat** on the data disk. The relevant SAS command is given below.

COMMAND FOR COMPUTING VARIANCE INFLATION FACTORS

```
proc reg data=my.premiums;
model premium=age price /vif;
run;
```

In the second line above, we have used the keyword vif at the end of the model statement. This tells SAS to compute the variance inflation factor for each predictor. The output includes the usual regression estimates along with the estimates of the variance inflation factors for each predictor variable. The SAS response follows.

```
---------------------------------------------------------------------------

Model: MODEL1
Dependent Variable: PREMIUM

                        Analysis of Variance

                           Sum of          Mean
Source          DF        Squares        Square      F Value       Prob>F

Model            2   2492087.0202  1246043.5101      708.495       0.0001
Error           33     58037.72979   1758.71908
C Total         35   2550124.7500

    Root MSE       41.93708     R-square       0.9772
    Dep Mean      485.58333     Adj R-sq       0.9759
    C.V.            8.63643
```

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
INTERCEP	1	6.896788	24.51817037	0.281	0.7802
AGE	1	-5.099627	0.38940052	-13.096	0.0001
PRICE	1	0.039533	0.00120946	32.686	0.0001

Variable	DF	Variance Inflation
INTERCEP	1	0.00000000
AGE	1	1.02726269
PRICE	1	1.02726269

The variance inflation factor for each predictor variable is in the column with the heading `Variance Inflation`. For this example, you see that the variance inflation factors are quite small and there is no indication of multicollinearity.

Chapter 6

Applications of Regression I

6.1 Overview

There are no calculations in this section that require SAS.

6.2 Prediction Intervals

In this section we explain how to use the macro **pred** that we have supplied on the data disk, for calculating predicted values and prediction intervals for the mean of h future Y values. The macro statements are in the files **pred.mac** and **pred.sas** on the data disk. Predicted values and prediction intervals for the sum of h future Y values can be obtained from the results for the mean of h future Y values by multiplying the results for the mean by h, and for a single future Y value they can be obtained by taking $h = 1$.

To explain this macro, we use Task 6.2.1 where an agency that evaluates the performance of used cars wants to obtain a 95% two-sided prediction interval for

$$Y_1(6.0, 24, 48.9) + Y_2(15.0, 21, 32.1),$$

which is the *total* first-year maintenance cost for two cars, where car 1 will be driven 6,000 miles the first year after it is purchased, is 24 months old, and has 48,900 miles registered on its odometer, whereas car 2 will be driven 15,000 miles the first year after it is purchased, is 21 months old, and has 32,100 miles registered on its odometer. The data are given in Table 6.2.1 and are stored in the files **usedcars.dat** and **usedcars.ssd** on the data disk. To execute the macro, invoke SAS, and on the Command line of the `PROGRAM EDITOR` window type

```
include 'b:\macro\pred.mac'
```

which brings the following statements to the screen.

```
------------------------------------------------------------------------------

00001 Title 'Predicted value and prediction interval for YA';
00002 libname my 'b:\';proc iml;reset nolog; option nodate;
00003
00004 ****** On line 00007 enter the name of the SAS data file
00005 ****** that contains the data you want to use;
00006 use
00007                         my.filename
00008 ;
00009 ****** On line 00013 enter the name of the response variable
00010 ****** exactly as it appears in the data file;
00011
00012 read all var{
00013                          response variable
00014 } into yvar;
00015
00016 ****** On lines 00022 through 00024 enter the names of the
00017 ****** predictor variables exactly as they are in your data
00018 ****** file. You can type in as many names as will fit on a
00019 ****** line. Leave at least one space between variable names.
00020 ****** Do not use any punctuation marks;
00021 read all var{
00022                          predictor1   predictor2
00023                          predictor3   predictor4
00024                          ... etc.
00025 } into xvar;
00026
00027 ****** Beginning on line 00040 enter the vectors x1 x2 ... xh,
00028 ****** for which predictions are required.  The order in which
00029 ****** you enter the coefficients must correspond to the order
00030 ****** in which the predictor variables names are entered above.
00031 ****** Enter one vector per line. End each line (except the last)
00032 ****** with a comma. There is no punctuation mark after the last
00033 ****** vector. For example, if h=2 and the number of predictor
00034 ****** variables is 4, the two vectors x1 and x2 could be
00035 ****** as follows:
00036 ****** 1  5.7  12.0  8.4  11.5,
00037 ****** 1  8.1  3.9   5.3  13.1
```



```
00038 ; g={
00039
00040                            1  5.0 10.0 20.0,
00041                            1 15.7 25.4 35.8
00042
00043
00044
00045
00046
00047 };
00048 ****** On line 00050 enter the desired confidence coefficient;
00049 cc =
00050                            0.95
00051
00052 ;%include 'b:\macro\pred.sas';

------------------------------------------------------------------------------
```

Following the instructions on the lines which begin with `******` , you must enter the following data on the indicated lines.

(1) On line 00007 enter the name of the file that contains the data you want to use. As usual the prefix is `my`. For this example we need to enter `my.usedcars` which will replace `my.filename` .

(2) On line 00013 enter the name of the response variable as it appears in the data file. For this example, replace the words `response variable` with `mtcost` .

(3) Use lines 00022 through 00024 to enter the names of the predictor variables exactly as they are in the data file. For this example the names are `miles` , `age` , and `odometer` . After you enter the required information on these lines, they should look something like this,

```
00022                  miles
00023                  age        odometer
00024
```

or, like this,

```
00022                  miles
00023                  age
00024                  odometer
```

etc.

(4) Beginning on line 00040 enter the vectors x_1, x_2, etc. You enter them as row vectors. Don't forget the leading element 1 if the regression model has an intercept (i.e., if the model includes the term β_0). Enter one vector per line and enter a comma after each line (vector) except the last. No punctuation mark is entered after the last vector. For the present example, enter `1 6.0 24.0 48.9,` on line 00040 to replace the values `1 5.0 10.0 20.0,` that are listed there; on line 00041 enter `1 15.0 21.0 32.1` to replace `1 15.7 25.4 35.8`.

(5) On line 00050 enter the confidence coefficient you want to use to replace `0.95`, unless, of course, you want to use the value 0.95 (we do use the value 0.95 for this example).

To execute the macro commands, press the `F10` key. The result given below will appear in the `OUTPUT` window.

```
---------------------------------------------------------------------------

        Predicted value and prediction interval for YA


    The estimate of YA is          YAhat =     207.8350
    The value of SE(YAhat) is                   43.6808

       A  95% prediction interval for YA is
           119.4078 to     296.2623

---------------------------------------------------------------------------
```

Thus, the predicted average first-year maintenance cost of these two cars is $\hat{Y}_A = \$207.84$. The standard error of $\hat{Y}_A$ is \$43.6808. A 95% prediction interval for Y_A is given by

$$C[\$119.41 \leq Y_A \leq \$296.26] = 0.95$$

The point estimate of the sum, Y_S, of the two ($h = 2$) future Y values is obtained by multiplying $\hat{Y}_A$ by 2. So we get $\hat{Y}_S = \$415.67$. To get a 95% prediction interval for Y_S we multiply the bounds for Y_A by 2 and get

$$C[\$238.82 \leq Y_S \leq \$592.52] = 0.95$$

These results are the same as in part (2) of Task 6.2.1 (within rounding error).

Problems

S6.2.1 For the car data in Task 6.2.1, which are also stored in the file **usedcars.ssd**, find a point estimate of the first-year maintenance cost of each of three cars which were chosen at random from the following subpopulations.

(a) Car 1 will be driven 10,000 miles, is 20 months old, and has 30,200 miles showing on its odometer.

(b) Car 2 will be driven 8,500 miles, is 15 months old, and has 15,000 miles showing on its odometer.

(c) Car 3 will be driven 6,500 miles, is 24 months old, and has 28,000 miles on its odometer.

S6.2.2 In S6.2.1, find a 90% prediction interval for the first-year maintenance cost of Car 1.

S6.2.3 In S6.2.1, find a 90% prediction interval for the total first-year maintenance cost of the three cars.

6.3 Tolerance Intervals

In this section we explain how to use the macro **toleranc** that we have supplied on the data disk, for computing point estimates and confidence intervals for tolerance points discussed in Section 6.3. The commands for the macro are in the files **toleranc.mac** and **toleranc.sas** on the data disk. To illustrate, we consider Example 6.3.3 where it is required to compute a point estimate and a 95% confidence interval for $\lambda_{0.80}(3)$, a number such that 80% of the values in the subpopulation with $X = 3$ are below it. The data are given in Table 6.3.1, and are also stored in the files **table631.dat** and **table631.ssd**.

To use the macro, invoke SAS, and on the Command line in the `PROGRAM EDITOR` window type

```
include 'b:\macro\toleranc.mac'
```

This brings the following SAS statements to the screen.

```
------------------------------------------------------------------------
00001 Title 'Estimates and Confidence Intervals for Tolerance Points';
00002 libname my 'b:\';proc iml;reset nolog; option nodate;
00003
00004 ****** On line 00007 enter the name of the SAS data file
00005 ****** that contains the data you want to use;
00006 use
00007                               my.filename
00008 ;
00009 ****** On line 00013 enter the name of the response variable
00010 ****** exactly as it appears in the data file;
00011
00012 read all var{
00013                               response variable
00014 } into yvar;
00015
00016 ****** On lines 00022 through 00024 enter the names of the
00017 ****** predictor variables exactly as they are in your data
00018 ****** file. You can type in as many names as will fit on a
00019 ****** line. Leave at least one space between variable names.
00020 ****** Do not use any punctuation marks;
00021 read all var{
00022                               predictor1   predictor2
00023                               predictor3   predictor4
00024                               ... etc.
00025 } into xvar;
00026
00027 ****** On line 00029 enter the value of p;
00028 p=
00029                               0.80
00030 ;
00031 ****** On line 00033 enter the confidence coefficient;
00032 cc=
00033                               0.95
00034 ;
00035 ****** On line 00043 enter the vector  x  defined in (6.3.6);
00036 ****** The order of the numbers in the vector  x  must correspond
00037 ****** to the order in which the predictor variable names are
00038 ****** entered above, with the first number being  1  since we
00039 ****** have assumed that an intercept is present in the model.
00040 ****** The number of elements in the  x  vector must equal the
00041 ****** number of parameters in the model;
00042 x={
00043                               1  2.3  4.5  3.5 ... etc
00044
00045 };%include 'b:\macro\toleranc.sas';
------------------------------------------------------------------------
```


Enter the appropriate file name on line 00007, the name of the response variable, exactly as it is in the data file, on line 00013, use lines 00022–00024 to enter the names of the predictor variables, enter the value of p on line 00029, the value of $1-\alpha$ on line 00033, and the elements of the vector x on line 00043, respectively. For the present example the following information must be input on the indicated lines.

```
00007                    my.table631
00013                    y
00022                    x
00023
00024
00029                    0.80
00033                    0.95
00043                    1   3.0
```

Note that there is only one predictor variable for this example. Other situations could have several predictor variables. After you enter these on the appropriate lines (to replace the quantities there, if necessary), press the F10 key to execute the macro commands. The following results will appear in the OUTPUT window.

```
------------------------------------------------------------------------
        Estimates and Confidence Intervals for Tolerance Points


    The estimate of lambda, the number such that  80% of the
    subpopulation Y values are below it, is       0.6624

    A 95% confidence interval for lambda is
            0.1178 to      1.4655
------------------------------------------------------------------------
```

Thus a point estimate of $\lambda_{.80}(3.0)$ is 0.6624, which of course is the same as what was obtained in Example 6.3.3 (within rounding error). Also, a 95% confidence statement for $\lambda_{.80}(3.0)$ is

$$C[0.1178 \leq \lambda_{.80}(3.0) \leq 1.4655] = 0.95$$

which is also the same as in Example 6.3.3 (within rounding error).

Problems

S6.3.1 In Example 6.3.3 in the textbook, find a point estimate and a 95% two-sided confidence interval for $\lambda_{0.20}(3.0)$, the number such that 20% of the Y values in the subpopulation with $X = 3.0$ are below it. Use the macro **toleranc** discussed in this section.

S6.3.2 Work Exercise 6.9.2 in the textbook using the macro **toleranc**.

6.4 Calibration and Regulation for Straight Line Regression

There are no built-in commands in SAS for computing point estimates and confidence intervals for parameters in Calibration and Regulation problems, so we have supplied macros that can be used for this purpose. The macro **calib** may be used to compute point estimates and confidence intervals in calibration problems. The SAS statements for this macro are stored in the files **calib.mac** and **calib.sas**. Likewise, the macro **regul** may be used to compute point estimates and confidence intervals in regulation problems. The SAS commands for this macro are stored in the files **regul.mac** and **regul.sas**. We discuss calibration first and regulation next.

To use the macro **calib**, invoke SAS, and on the `Command` line of the `PROGRAM EDITOR` window, type

```
include 'b:\macro\calib.mac'
```

and press Enter . This brings the following SAS statements to the screen.

```
--------------------------------------------------------------------------
00001 Title 'Calibration';
00002 libname my 'b:\';proc iml;reset nolog;option nodate;
00003
```



```
00004 ****** On line 00007 enter the name of the SAS data file
00005 ****** that contains the data you want to use;
00006 use
00007                          my.filename
00008 ;
00009 ****** On line 00012 enter the name of the response variable
00010 ****** exactly as it is in the data file;
00011 read all var{
00012                          response variable
00013 } into yvar;
00014
00015 ****** On line 00018 enter the name of the predictor variable
00016 ****** exactly as it is in the data file;
00017 read all var{
00018                          predictor variable
00019 } into xvar;
00020
00021 ****** On line 00023 enter the value of y0;
00022 y0=
00023                          100
00024 ;
00025 ****** On line 00027 enter the confidence coefficient;
00026 cc=
00027                          0.95
00028
00029 ;%include 'b:\macro\calib.sas';
--------------------------------------------------------------------------
```

Follow the instructions given on lines beginning with `******` , and enter the appropriate quantities on the specified lines. Then press the `F10` key to execute the macro commands.

To illustrate, we use Example 6.4.3 in the textbook where we are interested in calibrating a thermometer. The data are given in Table 6.4.1, and are also stored in the files **thermom.ssd** and **thermom.dat** on the data disk. For this example, you must enter the following information on the indicated lines, replacing the quantities already there if necessary.

```
00007                  my.thermom
00012                  reading
```

```
00018                knowntmp
00023                104
00027                0.95
```

On pressing the `F10` key the program runs, and the following results appear in the `OUTPUT` window.

```
----------------------------------------------------------------------

                          Calibration


        The point estimate of x0 is    103.9949

        A finite width  95% confidence interval for x0 exists.

        The lower bound is      103.4041

        The upper bound is      104.5895
----------------------------------------------------------------------
```

If the confidence region is not an interval, the macro will tell you so. Thus we see that $\hat{x}_0 = 103.995$ and the 95% confidence interval is given by

$$C[103.40 \leq x_0 \leq 104.59] = 0.95$$

Next we demonstrate the macro **regul** which is useful for obtaining a point estimate and a confidence interval for x_0 in a regulation problem. The SAS statements for this macro are in the files **regul.mac** and **regula.sas**. The following example illustrates the use of this macro.

Example S6.4.1

An investigator is studying the relationship of Y, the compression strength of cement blocks, and X, the amount of sand added to cement. An experiment is conducted by adding specified amounts of sand to the cement mixture and measuring the strength of the blocks. We suppose that assumptions **(A)** are satisfied and the data are obtained by sampling with preselected X values. The data are given below and also stored in the files **cement.ssd** and **cement.dat** on the data disk. The investigator wants to determine x_0, the amount of sand that must be used so that the *average* strength of the

resulting population of blocks is 10,000 pounds per square inch. Since we are interested in the average strength, we use the macro **regul** to compute a point estimate and a 90% confidence interval for x_0.

Cement Strength Data

Y strength (in thousands of lbs)	X amount of sand (in percent)
8.8	5
9.2	7
9.8	8
11.1	9
11.5	10
11.6	11
13.1	12
12.8	13
14.7	15
16.1	20

To use the macro, invoke SAS, and on the `Command` line of the `PROGRAM EDITOR` window type

```
include 'b:\macro\regul.mac'
```

This brings the following SAS statements to the window.

```
----------------------------------------------------------------------

00001 Title 'Regulation';
00002 libname my 'b:\';proc iml;reset nolog;option nodate;
00003
00004 ****** On line 00007 enter the name of the SAS data file
00005 ****** that contains the data you want to use;
00006 use
00007                          my.filename
00008 ;
00009 ****** On line 00012 enter the name of the response variable
00010 ****** exactly as it is in the data file;
```

```
00011 read all var{
00012                          response variable
00013 } into yvar;
00014
00015 ****** On line 00018 enter the name of the predictor variable
00016 ****** exactly as it is in the data file;
00017 read all var{
00018                          predictor variable
00019 } into xvar;
00020
00021 ****** On line 00023 enter the value of m0;
00022 m0=
00023                          100
00024 ;
00025 ****** On line 00027 enter the confidence coefficient;
00026 cc=
00027                          0.95
00028
00029 ;%include 'b:\macro\regul.sas';
```

For the example under discussion, you must enter the following information on the indicated lines replacing the quantities already there.

```
00007            my.cement
00012            y
00018            x
00023            10
00027            0.90
```

After you enter these quantities and check them, press the F10 key to execute the macro commands. The following result will appear in the OUTPUT window.

```
---------------------------------------------------------------------

                              Regulation


        The point estimate of x0 is        7.4977

        A finite width  90% confidence interval for x0 exists.

        The lower bound is        6.6934

        The upper bound is        8.1713

---------------------------------------------------------------------
```


Thus we see that $\hat{x}_0 = 7.4977$ and the 90% confidence statement is

$$C[6.6934 \leq x_0 \leq 8.1713] = 0.90$$

If the confidence region is not a finite width interval, the macro will tell you so.

Problems

S6.4.1 Work Problems 6.4.1 and 6.4.2 in the textbook using the macros discussed in this section.

S6.4.2 Work Problems 6.4.3 and 6.4.4 in the textbook using the macros discussed in this section.

S6.4.3 Work Problems 6.4.5 and 6.4.6 in the textbook using the macros discussed in this section.

6.5 Comparison of Several Straight Line Regressions – Identical, Parallel, and Intersecting Lines

In this section we discuss a macro we have written and supplied on the data disk that can be used to perform the computations for comparing several regression functions. This macro is called **compare**, and it will calculate point estimates and simultaneous confidence intervals for m linear combinations of α_i and β_j with confidence coefficient greater than or equal to $1 - \alpha$. See (6.5.15). The macro statements are in the files **compare.mac** and **compare.sas**. We explain this macro by using Example 6.5.3, where the data are given in Table 6.5.2 and are also stored in the files **eggshell.ssd** and **eggshell.dat** on the data disk. In that example, we need 95% simultaneous confidence intervals for the following $m = 6$ linear combinations θ_i.

$$\theta_1 = \alpha_1 - \alpha_2 \qquad \theta_2 = \alpha_1 - \alpha_3 \qquad \theta_3 = \alpha_2 - \alpha_3$$

$$\theta_4 = \beta_1 - \beta_2 \qquad \theta_5 = \beta_1 - \beta_3 \qquad \theta_6 = \beta_2 - \beta_3$$

To execute the macro, invoke SAS, and on the Command line in the PROGRAM EDITOR window type

```
include 'b:\macro\compare.mac'
```

This brings the following statements to the screen.

```
------------------------------------------------------------------------
00001 Title 'Comparison of Regression Lines';
00002 libname my 'b:\';proc iml;reset nolog; option nodate;
00003
00004 ****** On line 00007 enter the name of the SAS data file
00005 ****** that contains the data you want to use;
00006 use
00007                       my.filename
00008 ;
00009
00010 ****** On lines 00016 through 00021 enter the names of the
00011 ****** response variable and the predictor variable for each
00012 ****** straight line. The variable names should be typed exactly
00013 ****** as they appear in the data file. Use at least one space
00014 ****** between names. Do not use any punctuation marks;
00015 read all var{
00016                       response variable1    predictor variable1
00017                       response variable2    predictor variable2
00018                       response variable3    predictor variable3
00019                       ... etc.
00020
00021
00022 } into data;
00023
00024 ****** On line 00026 enter the confidence coefficient;
00025 cc =
00026                       0.95
00027 ;
00028 ****** Beginning on line 00036 enter the vectors d(i) in (6.5.15).
00029 ****** Put one vector per line with a comma at the end of each line
00030 ****** except the last line. The last line has no punctuation mark.
00031 ****** The numbers in each vector must follow the following order.
00032 ******
00033 ****** alpha1 beta1 alpha2 beta2 alpha3 beta3 alpha4 beta4...etc.
00034 ******;
00035 d={
00036                       1  0 -1  0  0  0,
00037                       1  0  0  0 -1  0,
```



```
00038                       0  0  1  0 -1  0,
00039                       0  1  0 -1  0  0,
00040                       0  1  0  0  0 -1,
00041                       0  0  0  1  0 -1
00042
00043
00044
00045
00046
00047
00048 };%include 'b:\macro\compare.sas';
------------------------------------------------------------------------
```

As usual, follow the instructions given on the lines beginning with ****** . For this example, you must enter the following information on the indicated lines replacing the quantities already there.

```
00007                 my.eggshell
00016                 y1 x1
00017                 y2 x2
00018                 y3 x3
00019
00020
00021
00026                 0.95
00036                 1  0 -1  0  0  0,
00037                 1  0  0  0 -1  0,
00038                 0  0  1  0 -1  0,
00039                 0  1  0 -1  0  0,
00040                 0  1  0  0  0 -1,
00041                 0  0  0  1  0 -1
00042
00043
00044
00045
00046
00047
```

Notice that many of the entries that appear on the screen are already correct for the current example, so no changes are required on the corresponding lines. Press the

F10 key to execute the macro commands. The following results will appear in the OUTPUT window.

```
---------------------------------------------------------------------------
                    Comparison of Regression Lines

        The point estimates and simultaneous confidence intervals
        for the thetas with confidence coefficient greater than
        or equal 95% are given below

          THETA     ESTIMATE        LOWER        UPPER
              1      -0.5012      -4.2849       3.2826
              2       0.9436      -2.5718       4.4591
              3       1.4448      -2.3793       5.2689
              4       1.9546       1.4522       2.4570
              5       2.7860       2.4409       3.1311
              6       0.8314       0.3708       1.2919
---------------------------------------------------------------------------
```

Thus the required point estimates and confidence bounds are obtained very easily using this macro.

Problems

S6.5.1 In Example 6.5.3 in the textbook, use the macro **compare** to obtain confidence intervals for

$$\theta_1 = \alpha_1 - \alpha_2, \quad \theta_2 = \alpha_1 - \alpha_3, \quad \theta_3 = \alpha_2 - \alpha_3$$

so that you have confidence of at least 90% that all intervals are simultaneously correct.

6.6 Intersection of Two Straight Line Regression Functions

In this section we discuss a macro we have written and supplied on the data disk, for computing a point estimate and a confidence interval for x_0, the point where two straight

line regression functions intersect. This macro is called **inter**, and the macro statements are in the files **inter.mac** and **inter.sas** on the data disk. We use Example 6.6.2 to illustrate this macro. In that example, an investigator wants to compare the hardness of eggshells for breeds 2 and 3 for values of the food supplement in the range from 2 to 20 units. To help make this comparison, we want to determine x_0, the X value at the point where the regression lines for breed 2 and breed 3 intersect. We find a point estimate of x_0 and a 95% confidence region for x_0. The data are given in Table 6.6.1 and are also stored in the files **eggshell.ssd** and **eggshell.dat** on the data disk. The data, which we reproduce for your convenience, are as follows.

y1	x1	y2	x2	y3	x3
8.42	1	9.86	3	6.52	2
14.68	3	9.54	3	5.11	5
21.42	5	11.96	4	7.75	7
25.45	6	12.46	5	6.84	8
27.14	7	11.38	6	7.65	10
30.53	8	14.69	8	9.49	15
34.51	9	16.48	9	7.03	16
34.52	9	20.11	12	9.41	18
33.24	10			12.01	20
39.63	11				
43.98	12				
47.77	14				

Observe that there are actually three breeds represented in the sample data. But, for this example, we are only interested in determining where the straight line regression functions for breed 2 and breed 3 intersect.

To execute the macro, type

```
include 'b:\macro\inter.mac'
```

on the Command line in the PROGRAM EDITOR window, and press Enter . The following statements appear on the screen.

```
---------------------------------------------------------------------------
00001 Title 'Intersection of two straight line regression functions';
00002 libname my 'b:\';
00003 data rawdata(keep = yline1 xline1 yline2 xline2);
00004
```

```
00005 ****** On line 00008 enter the name of the SAS data file
00006 ****** that contains the data you want to use;
00007 set
00008                my.filename
00009 ;
00010
00011
00012 ****** On line 00021 enter the name of the response variable
00013 ****** for the first straight line;
00014 ****** On line 00023 enter the name of the predictor variable
00015 ****** for the first straight line;
00016 ****** On line 00025 enter the name of the response variable
00017 ****** for the second straight line;
00018 ****** On line 00027 enter the name of the predictor variable
00019 ****** for the second straight line;
00020 rename
00021                response variable for the first straight line
00022 =yline1
00023                predictor variable for the first straight line
00024 =xline1
00025                response variable for the second straight line
00026 =yline2
00027                predictor variable for the second straight line
00028 =xline2
00029
00030 ;proc iml;reset nolog;
00031
00032 ****** On line 00034 enter the confidence coefficient;
00033 c=
00034                            0.95
00035 ;
00036 %include 'b:\macro\inter.sas';
```

Follow the instructions on the lines beginning with ****** . For this example, you must enter the following information on the indicated lines replacing the quantities already there.

```
00008          my.eggshell
00021          y2
00023          x2
00025          y3
00027          x3
00034          0.95
```


After you enter the appropriate values and check them, press the F10 key to execute the macro commands. The following result will appear in the OUTPUT window.

```
----------------------------------------------------------------------

         Intersection of two straight line regression functions


            The point estimate of x0 is       -1.7378

        A finite width  95% confidence interval for x0 exists
        and it is given by

        the interval from      -7.6205 to       1.0876

----------------------------------------------------------------------
```

Thus the required confidence statement is

$$C[-7.6205 \le x_0 \le 1.0876] = 0.95.$$

Thus, using this confidence interval, we would perhaps conclude that the two population regression lines do not intersect in the range of interest, viz., $2 \le X \le 20$. Furthermore, since $\hat{\alpha}_2 > \hat{\alpha}_3$, we might also conclude that the average hardness of eggshells will be greater for breed 2 than for breed 3, for all values of food supplement in the range 2 to 20.

Problems

S6.6.1 For the eggshell data in Example 6.5.3, use the macro **inter** and find a point estimate and a 90% confidence region for x_0, the point where the straight line regression functions for breeds 1 and 3 intersect.

S6.6.2 For the eggshell data in Example 6.5.3, use the macro **inter** and find a point estimate and a 90% confidence region for x_0, the point where the straight line regression functions for breeds 1 and 2 intersect.

6.7 Maximum or Minimum of a Quadratic Regression Model

In this section we describe a macro named **quadr**, supplied by us on the data disk, that can be used to compute a point estimate and a confidence interval for x_0, the X value where a quadratic regression function attains its maximum (or minimum) value. The macro commands are stored in the files **quadr.mac** and **quadr.sas** on the data disk.

To illustrate how this macro works, we use Example 6.7.3 where it is desired to determine the temperature x_0 for obtaining the maximum rate of production of sulfuric acid. The data are given in Table 6.7.2 and are also stored in the files **sulfuric.dat** and **sulfuric.ssd** on the data disk. We use the macro **quadr** to obtain a point estimate and a 95% confidence region for x_0.

Invoke SAS, and on the `Command` line of the `PROGRAM EDITOR` window type

```
include 'b:\macro\quadr.mac'
```

and press `Enter`. The following statements will appear on the screen.

```
------------------------------------------------------------------------
00001 Title 'Maximum or minimum of a quadratic regression model';
00002 libname my 'b:\'; data rawdata(keep= yvar xvar);
00003
00004 ****** On line 00007 enter the name of the SAS data file
00005 ****** that contains the data you want to use;
00006 set
00007                              my.filename
00008 ;
00009 ****** On line 16 enter the name of the response variable as it
00010 ****** appears in the data file.
00011 ****** On line 18 enter the name of the predictor variable as it
00012 ****** appears in the data file;
00013
00014 rename
00015
00016                              response variable
00017 = yvar
00018                              predictor variable
```



```
00019 = xvar
00020
00021
00022 ;proc iml;
00023
00024 ****** On line 00026 enter the confidence coefficient;
00025 c=
00026                              0.95
00027
00028
00029 ;%include 'b:\macro\quadr.sas';
------------------------------------------------------------------------
```

For our example, replace `my.filename` on line `00007` with `my.sulfuric`. On line `00016` replace the words `response variable` with `tons`, which is the name of the response variable as it appears in the data file. Likewise, on line `00018` replace the words `predictor variable` with `temp`, which is the name of the predictor variable as it appears in the data file. Finally, replace `0.95` on line `00026` by the desired value of the confidence coefficient. For the present example the desired confidence coefficient is 0.95 and so we do not need to change the entry on line `00026`.

After entering the appropriate values and checking them, press the `F10` key to execute the macro. The following result appears in the `OUTPUT` window.

```
------------------------------------------------------------------------
          Maximum or minimum of a quadratic regression model

               The point estimate of x0 is  272.2905


 A finite width  90% confidence interval for x0 exists and is given by

 the interval from  257.4097 to  293.5311
------------------------------------------------------------------------
```

Thus the maximum yield is estimated to occur at 272.29 °C. A 95% confidence statement for x_0, the temperature at which the maximum yield occurs, is

$$C[257.41 \le x_0 \le 293.53] = 0.95$$

Problems

S6.7.1 For Problem 6.7.1 in the textbook, use the macro **quadr** and find a 90% confidence region for x_0, the amount of sand to use to maximize the average crushing strength of the cement. The data are stored in the files **concrete.ssd** and **concrete.dat**. They are also displayed in Table 6.7.3 in the textbook.

S6.7.2 Plot the data in Table 6.7.3 (these data are in the file **concrete.ssd**) using the plotting symbol $*$. Does the maximum of the data appear to be close to the estimate of x_0 computed in Problem S6.7.1?

6.8 Linear Splines

In this section we explain how to use the macro **spline**, that we have supplied on the data disk, to calculate point estimates and confidence intervals for spline regression functions. The macro commands are in the files **spline.mac** and **spline.sas**. To illustrate, we work Example 6.8.3 where the data are given in Table 6.8.1 and are also stored in the files **sales.dat** and **sales.ssd** on the data disk. We obtain a point estimate and a 95% confidence interval for $\theta = \beta_1$.

Recall that, in this example, the population (spline) regression function is given by

$$\mu_Y(x) = \begin{cases} \mu_Y^{(1)}(x) = \alpha_1 + \beta_1 x & \text{for } 0 \le x \le 50 \\ \mu_Y^{(2)}(x) = \alpha_2 + \beta_2 x & \text{for } 50 \le x \le 100 \end{cases}$$

You can use the macro to compute point estimates and one-at-a-time $1 - \alpha$ confidence intervals for specified linear functions (you select the a_i and b_j)

$$a_1\alpha_1 + b_1\beta_1 + a_2\alpha_2 + b_2\beta_2$$

To use the macro, first invoke SAS, and on the `Command` line in the `PROGRAM EDITOR` window, type

```
include 'b:\macro\spline.mac'
```

This brings the following statements to the screen.


```
--------------------------------------------------------------------------
00001 Title 'Spline regression';
00002 libname my 'b:\';data rawdata(keep= yvar xvar);
00003
00004 ****** On line 00007 enter the name of the SAS data file
00005 ****** that contains the data you want to use;
00006 set
00007                       my.filename
00008 ;
00009 ****** On line  00014  enter the name of the response variable
00010 ******    as it appears in the data file;
00011 ****** On line  00016  enter the name of the predictor variable
00012 ******    as it appears in the data file;
00013 rename
00014                 response variable
00015 =yvar
00016                 predictor variable
00017 =xvar
00018
00019 ;proc iml;
00020
00021 ****** On line 00023 enter the value of q;
00022 q=
00023                       100
00024
00025 ;
00026 ****** On line 00028 enter the confidence coefficient;
00027 c=
00028                       0.95
00029 ;
00030
00031 ****** On line 00036 enter the coefficients of the linear comb-
00032 ****** ination you want to use.  Enter them in the following
00033 ****** order:--
00034 ******          a(1)    b(1)    a(2)    b(2);
00035 d={
00036                  0       1       0       1
00037
00038 };%include 'b:\macro\spline.sas';
--------------------------------------------------------------------------
```

For Example 6.8.3, enter the following information on the specified lines.

(1) On line 00007 enter the name of the SAS data file that contains the data you want to use. For this problem, you will enter `my.sales` and this will replace `my.filename` .

(2) On line 00014 enter the name of the response variable as it appears in the data file. For this example, enter `sales` to replace the words `response variable` .

(3) On line 00016 enter the name of the predictor variable as it appears in the data file. For this example, enter `advbudgt` to replace the words `predictor variable` .

(4) On line 00023 enter the value of the knot-point q. For this example $q = 50$, so replace `100` on line `00023` by 50.

(5) On line 00028 enter the desired confidence coefficient. This is 0.95 for the present example, so no change is required on this line.

(6) On line 00036 enter the values of a_1, b_1, a_2, b_2. For the example, our interest is in $\theta = \beta_1$ so we enter `0 1 0 0` to replace `0 1 0 1`.

After the appropriate quantities have been entered and checked, press the F10 key to execute the macro. The following results appear in the OUTPUT window.

```
---------------------------------------------------------------------------

                        Spline regression


     The point estimates of alpha1, beta1, alpha2, and beta2,
     respectively, are

          201.4454,     5.0218,    404.2462,      0.9658

     The point estimate of sigma is       11.0488

     The point estimate of theta is        5.0218

     A  95% confidence interval for theta is given by
     the interval from       4.4153   to       5.6283
---------------------------------------------------------------------------
```


Thus we see that the point estimate for β_1 is 5.0218, and the confidence statement is

$$C[4.42 \le \beta_1 \le 5.63] = 0.95$$

If you want to compute point estimates and/or confidence intervals for $\mu_Y(x)$ for a specified value of x,

enter `1 x 0 0` on line 00036 if $x \le q$, or

enter `0 0 1 x` on line 00036 if $x > q$.

Problems

S6.8.1 This problem refers to the data and the model discussed in Example 6.8.3 in the textbook, where Y is sales and X is money spent on advertising. The data are given in Table 6.8.1 and are also stored in the files **sales.ssd** and **sales.dat** on the data disk. In this problem $q = 50$.

(a) Find the point estimate of α_1.

(b) Find a 90% confidence interval for α_1.

(c) Plot the estimated spline regression function.

(d) Compute a point estimate and a 90% confidence interval for $\mu_Y(75)$, the average sales (in thousands of dollars) a company expects if it plans to spend 75 thousand dollars on advertising.

(e) The vice president of a company wants to determine how much more the average sales would be if the company spent 80 thousand dollars rather than 60 thousand dollars on advertising next year. Estimate this quantity and obtain a 90% confidence interval for it.

Chapter 7

Applications of Regression II

7.1 Overview

No computing instructions are needed in this section.

7.2 Subset Analysis and Variable Selection

No computing instructions are needed in this section.

7.3 All Subsets Regression

In this section and the next, we illustrate SAS commands that can be used for subset analysis and variable selection.

The SAS command proc reg offers a facility for examining all of the possible subset models as long as the number of predictors is less than or equal to 10. The models are evaluated and ordered according to their C_p values, or their adjusted R-square values, or their R-square values, where you select which criterion is to be used. When there are eleven or more predictors, the number of possible subset models is very large and SAS

will not print the results for all of these models. However, you can specify how many of the subset models (at most equal to the number of predictors in the full model) you wish to examine, and SAS will print the criterion values for the specified number of best subset models.

To illustrate the relevant SAS commands, we use the GPA data of Example 4.4.2, which are given in Table 4.4.3 and are also stored in the files **gpa.dat** and **gpa.ssd**. The response variable Y is named GPA and the predictor variables X_1, X_2, X_3, and X_4 are named SATmath, SATverb, HSmath, and HSengl, respectively. The following SAS statements ask SAS to compute the C_p values, along with $adj\text{-}R^2$ and s (RMSE), for all of the possible subset models, and order them according to increasing values of C_p.

COMMAND FOR BEST SUBSETS REGRESSION ORDERED BY C_p

```
00001 libname my 'b:\';
00002 proc reg data=my.gpa;
00003 model gpa=satmath satverb hsmath hsengl/selection=cp
00004           adjrsq rmse;
00005 run;
```

The first statement above is the familiar libname statement. The second statement invokes the reg procedure and specifies that the data are in the file **gpa.ssd**. Line 00003 specifies the full model, and the option selection=cp specifies that the subset models should be ordered from best to worst according to the C_p criterion. If you want them ordered according to another criterion, say the rsquare criterion, then the keyword rsquare replaces the keyword cp . The SAS statement beginning on line 00003 is too long to fit on a single line and so we have split it into two lines. Thus, lines 00003 and 00004 together constitute a single SAS statement. You can tell this is so by observing that the semicolon does not appear at the end of line 00003, but does appear at the end of line 00004. The keywords rmse and adjrsq on line 00004 specify that the values of s (RMSE) and $adj\text{-}R^2$ are to be displayed for each subset model included in the output. C_p values will also be displayed because the command asks SAS to order the subset models according to the values of C_p. The value of R^2 is automatically included for each subset model even though it is not explicitly requested.

To execute the preceding commands, enter them in the PROGRAM EDITOR window and press the F10 key. The response from SAS is as follows.

```
------------------------------------------------------------------------------
N = 20     Regression Models for Dependent Variable: GPA

      C(p)   R-square      Adjusted       Root Variables in Model
                       In  R-square        MSE

   3.24614 0.85035772  3 0.82229979 0.26211225 SATMATH SATVERB HSMATH
   5.00000 0.85277358  4 0.81351320 0.26851428 SATMATH SATVERB HSMATH HSENGL
   5.25639 0.81099674  2 0.78876106 0.28577901 SATMATH SATVERB
   7.19530 0.79196616  2 0.76749159 0.29982143 SATMATH HSMATH
   7.25222 0.81103768  3 0.77560725 0.29454235 SATMATH SATVERB HSENGL
   8.15492 0.80217758  3 0.76508588 0.30136853 SATMATH HSMATH HSENGL
  12.35334 0.72170925  1 0.70624865 0.33700262 SATMATH
  14.01469 0.72503316  2 0.69268412 0.34469568 SATMATH HSENGL
  14.82993 0.73666167  3 0.68728573 0.34771001 SATVERB HSMATH HSENGL
  19.53723 0.67082890  2 0.63210288 0.37714342 HSMATH HSENGL
  23.18701 0.63500597  2 0.59206550 0.39713536 SATVERB HSMATH
  30.63181 0.56193459  2 0.51039748 0.43507604 SATVERB HSENGL
  36.70997 0.48264669  1 0.45390483 0.45949153 HSMATH
  42.40237 0.42677523  1 0.39492941 0.48366690 SATVERB
  48.40914 0.36781820  1 0.33269699 0.50793119 HSENGL
------------------------------------------------------------------------------
```

There are $k = 4$ predictor variables and hence $2^k - 1 = 15$ possible subset models (excluding the model β_0). The computer output contains the following information. The 15 subset models are ordered, from best to worst, according to the C_p criterion (in column 1). For each subset model, the values of C_p, R^2, Adj$-R^2$, and s (under the label `Root MSE`) are printed. The number of predictor variables included in each subset model is also displayed under the label `In`. The names of the variables in each subset model are displayed under the label `Variables in Model`. The output may be split over two or more pages depending on its length, but we have suppressed the page numbers.

Since we did not specify the number of 'best' subset models we wanted, SAS, by default, prints out summary information for all the subset models, because the number of predictors is less than or equal to 10 (actually the number of predictors is 4 in this problem). From the output above we see that the best 1-variable model (look in the column labeled `In` for the first occurrence of the number `1`) uses SATmath and has a C_p value of 12.35334; the second-best 1-variable model (look for the second occurrence of the `1` in the column labeled `In`) uses HSmath and has a C_p value of 36.70997.

The best 2-variable model (look for the first occurrence of 2 in the column labeled `In`) uses SATmath and SATverb, and has a C_p value of 5.25639; the second-best model with 2 predictors uses SATmath and HSmath and has a C_p value of 7.19530.

The best 3-variable model uses the predictors SATmath, SATverb, and HSmath, and has C_p equal to 3.24614 (this is also the best of *all* the possible subset models according to the C_p criterion, since this is the smallest overall C_p value). The second best 3-variable model contains SATmath, SATverb, and HSengl, and has C_p equal to 7.25222.

The best (and only) 4-variable model contains SATmath, SATverb, HSmath, and HSengl and has C_p equal to 5.0.

If you only want summary information for the best few subset models rather than all of them, you can specify the number of subset models you wish to examine by including the option `best = m` in the model statement. This will instruct SAS to print the results for only the best `m` subset models. For example, the commands for obtaining the *8 best subset models* in the GPA problem are shown below.

SAS COMMAND FOR 8 BEST SUBSET MODELS

```
00001 libname my 'b:\';
00002 proc reg data=my.gpa;
00003 model gpa = satmath satverb hsmath hsengl/selection=cp rmse
00004            adjrsq best=8;
00005 run;
```

You should observe that lines 00003 and 00004 together constitute a single SAS statement, but because the command is too long to fit on one line it has been split into two lines. You can tell that these two lines together form a single statement by the fact that there is no semicolon at the end of line 00003, but there is one at the end of line 00004.

The SAS response in the `OUTPUT` window is as follows.

```
--------------------------------------------------------------------------------
N = 20     Regression Models for Dependent Variable: GPA

     C(p)   R-square       Adjusted       Root Variables in Model
                     In    R-square        MSE

   3.24614 0.85035772  3 0.82229979 0.26211225 SATMATH SATVERB HSMATH
   5.00000 0.85277358  4 0.81351320 0.26851428 SATMATH SATVERB HSMATH HSENGL
   5.25639 0.81099674  2 0.78876106 0.28577901 SATMATH SATVERB
   7.19530 0.79196616  2 0.76749159 0.29982143 SATMATH HSMATH
   7.25222 0.81103768  3 0.77560725 0.29454235 SATMATH SATVERB HSENGL
   8.15492 0.80217758  3 0.76508588 0.30136853 SATMATH HSMATH HSENGL
  12.35334 0.72170925  1 0.70624865 0.33700262 SATMATH
  14.01469 0.72503316  2 0.69268412 0.34469568 SATMATH HSENGL
--------------------------------------------------------------------------------
```

If you ask SAS to order the subset models according to the `rsquare` criterion, and if in addition you use the `best=m` option by specifying the option command

$$/\texttt{selection} = \texttt{rsquare}\quad \texttt{best} = \texttt{m};,$$

the output you get will be somewhat different from the `best=m` optional statement discussed previously for the C_p criterion. In this case SAS will give you

(1) the `best m` models using one predictor variable,

(2) the `best m` models when two predictors are used,

(3) The `best m` models when three predictors are used,

(4) and so forth.

Ordering the models from best to worst by the `rmse` criterion (i.e. by the s criterion) is the same as ordering the models by the adjusted R-square (`adjrsq`) criterion. So, if you use the option `/selection=rmse` the command will not execute.

Problems

S7.3.1 Give the SAS commands for obtaining Exhibit 7.3.4 in the textbook.

S7.3.2 In Problem 7.3.2 in the textbook, give the SAS commands for obtaining the eight best subset models of each subset size.

7.4 Alternative Methods for Subset Selection

In this section we discuss SAS commands that can be used to do the computations for backward, forward, and stepwise regression. We begin with stepwise regression since the backward and forward procedures are obtained as special cases of the stepwise procedure.

Stepwise regression

The SAS procedure `proc reg` can be used to perform a stepwise regression by choosing the option `selection = stepwise` in the `model` statement. The criteria for entering or removing predictors from a model are named `SLENTRY` (or `SLE` for short) for *F-in* and `SLSTAY` (or `SLS` for short) for *F-out*. The letters `SL` in `SLENTRY` and `SLSTAY` stand for **S**ignificance **L**evel, viz., the *P*-value. In the forward mode of the stepwise procedure, a variable will be added to the current model provided that the *P*-value for the test comparing the current model with the candidate model is less than or equal to `SLE`. Likewise, in the backward mode of the stepwise procedure, a variable will be deleted from the current model provided that the *P*-value for the test comparing the current model with the candidate model is greater than `SLS`. It is important that the value of `SLE` be smaller or equal to the value of `SLS`; otherwise, infinite looping could occur where the same predictor variable is repeatedly added and deleted.

For the sake of discussion, assume that the response variable is named `Y` and the predictor variables are named `X1`, `X2`, `X3`, `X4`, and `X5`, respectively. In particular, we have assumed that the number of predictor variables is 5 for illustrative purposes. Suppose the data are stored in a SAS data file named **data.ssd**. Under these circumstances, the basic SAS statements for stepwise regression, with β_0 as the *initial model*, `SLE = 0.10`, and `SLS = 0.15` are as follows.

STEPWISE REGRESSION USING PROC REG

```
libname my 'b:\';
proc reg data=my.data;
model y = x1 x2 x3 x4 x5 /selection=stepwise
            sle=0.10 sls=0.15;
run;
```

We illustrate the procedure using Example 7.4.3. The data are given in Table 4.4.3 and are also stored in the file **gpa.ssd** on the data disk. The response variable is `GPA` and the predictor variables are `SATmath`, `SATverb`, `HSmath`, and `HSengl`. We now ask SAS to perform a stepwise regression analysis using `SLE = 0.06` and `SLS = 0.10`. Note that, in SAS, we are unable to specify criterion values for entering and removing variables in terms of F-values (i.e., $F-in$ and $F-out$ values of 2, 3, 4, etc.) as we discussed in the textbook. But an F-value of 4 corresponds, roughly, to a P-value of 0.06, if the numerator degrees of freedom is 1 and the denominator degrees of freedom is in the range from 15 to 20. Likewise, an F-value of 3 corresponds, roughly, to a P-value of 0.10, if the numerator degrees of freedom is 1 and the denominator degrees of freedom is close to 15. So, we use `SLE = 0.06` and `SLS = 0.10`, and these will be approximately equivalent to specifying $F-in = 4.0$ and $F-out = 3.0$, which were the values used in Example 7.4.3 in the textbook. Generally, you might use `SLE = 0.05` but, to illustrate the procedure, we want this example to correspond as closely as possible to the example in the textbook. The default values (i.e., values that SAS will use if you do not specify them yourself) are automatically set at 0.15 (i.e., P-value = 0.15). The relevant SAS statements for the current example are as follows.

STEPWISE REGRESSION USING GPA DATA

```
libname my 'b:\';
proc reg data=my.gpa;
model gpa=satmath satverb hsmath hsengl/selection=stepwise
        sle=0.06 sls=0.10;
run;
```

Press the `F10` key and the following result appears in the `OUTPUT` window.

Stepwise Procedure for Dependent Variable GPA

Step 1 Variable SATMATH Entered R-square = 0.72170925 C(p) = 12.35334

	DF	Sum of Squares	Mean Square	F	Prob>F
Regression	1	5.30154623	5.30154623	46.68	0.0001
Error	18	2.04427377	0.11357076		
Total	19	7.34582000			

Variable	Parameter Estimate	Standard Error	Type II Sum of Squares	F	Prob>F
INTERCEP	0.96699086	0.24963334	1.70414326	15.01	0.0011
SATMATH	0.00317828	0.00046518	5.30154623	46.68	0.0001

Bounds on condition number: 1, 1

Step 2 Variable SATVERB Entered R-square = 0.81099674 C(p) = 5.25638

	DF	Sum of Squares	Mean Square	F	Prob>F
Regression	2	5.95743607	2.97871804	36.47	0.0001
Error	17	1.38838393	0.08166964		
Total	19	7.34582000			

Variable	Parameter Estimate	Standard Error	Type II Sum of Squares	F	Prob>F
INTERCEP	0.50714165	0.26672665	0.29524762	3.62	0.0743
SATMATH	0.00260559	0.00044323	2.82242207	34.56	0.0001
SATVERB	0.00157415	0.00055547	0.65588984	8.03	0.0115

Bounds on condition number: 1.262436, 5.049743

Step 3 Variable HSMATH Entered R-square = 0.85035772 C(p) = 3.24613

	DF	Sum of Squares	Mean Square	F	Prob>F
Regression	3	6.24657473	2.08219158	30.31	0.0001
Error	16	1.09924527	0.06870283		
Total	19	7.34582000			

Variable	Parameter Estimate	Standard Error	Type II Sum of Squares	F	Prob>F
INTERCEP	0.33424978	0.25874739	0.11464759	1.67	0.2148
SATMATH	0.00218487	0.00045532	1.58193515	23.03	0.0002
SATVERB	0.00131233	0.00052521	0.42893384	6.24	0.0237
HSMATH	0.17987024	0.08767859	0.28913865	4.21	0.0570

Bounds on condition number: 1.583729, 13.41417

```
All variables in the model are significant at the 0.1000 level.
No other variable met the 0.0600 significance level for entry into the model

          Summary of Stepwise Procedure for Dependent Variable GPA

        Variable          Number   Partial    Model
Step    Entered Removed      In       R**2     R**2       C(p)         F    Prob>F

   1    SATMATH               1    0.7217   0.7217    12.3533   46.6806    0.0001
   2    SATVERB               2    0.0893   0.8110     5.2564    8.0310    0.0115
   3    HSMATH                3    0.0394   0.8504     3.2461    4.2085    0.0570
---------------------------------------------------------------------------------
```

SAS prints out an analysis of variance table and other useful information for each step of the stepwise procedure. In this problem there are three steps. In `Step 1` the variable `SATMATH` is entered in the model. In `Step 2`, the variable `SATVERB` is entered and the two variables, `SATMATH` and `SATVERB`, are now in the model. In `Step 3`, the variable `HSMATH` is entered and now there are three variables, `SATMATH`, `SATVERB`, and `HSMATH`, in the model. No other variables enter or leave. The `Summary` section of the output tells you what variables were entered and which variables were removed during each step.

Note that the final model obtained above is the same model as the final model in Example 7.4.3, which is given in (7.4.61). For a detailed discussion of the quantities printed by SAS for stepwise regression, and for other options available with this procedure, you should refer to the SAS/STAT guide.

The START Option in PROC REG

In Example 7.4.4, we again perform a stepwise regression using the GPA data, with the same values of `SLE` and `SLS`, but with a different initial model. The initial model this time is

$$\beta_0 + \beta_3 x_3 + \beta_4 x_4$$

Thus, the initial model contains variables X_3 = HSmath and X_4 = HSenglish. There is an option in the `model` statment in `proc reg` for specifying an initial model for stepwise regression. This option is specified using the statement

$$\texttt{start} = \texttt{s}$$

where the word `start` is a SAS keyword and the argument `s` means that the initial model uses the first `s` predictor variables specified in the `model` statement. So, when you type in the statement `model y =` , be sure that the first `s` predictor variables after the `=` are the variables you want in your initial model.

Since the example under consideration uses the model containing `HSmath` and `HSengl` as the initial model, and `SLE = 0.06` and `SLS = 0.10`, the following statements are appropriate.

SAS COMMANDS FOR STEPWISE REGRESSION WITH USER SPECIFIED INITIAL MODEL

```
00001 libname my 'b:\';
00002 proc reg data=my.gpa;
00003 model gpa=hsmath hsengl satmath satverb/selection=stepwise
00004          sle=0.06 sls=0.10 start=2;
00005 run;
```

The model statement is too long to fit on one line and so we have split it into two lines. The nonoccurrence of a semicolon at the end of the first line of the model statement tells SAS that the next line (which does have a semicolon at the end) is a continuation of this line. Notice that the argument of the keyword `start =` is `2`. This tells SAS to use an initial model that includes the first two predictor variables following the `=` sign in the model statement (i.e., to use **hsmath** and **hsengl** as the two variables in the initial model). When you press the `F10` key, SAS responds as follows.

```
--------------------------------------------------------------------------------

              Stepwise Procedure for Dependent Variable GPA

Step 0 The First  2 Vars Entered  R-square = 0.67082890  C(p) = 19.53723134

                DF        Sum of Squares     Mean Square           F   Prob>F

Regression       2            4.92778832      2.46389416       17.32   0.0001
Error           17            2.41803168      0.14223716
Total           19            7.34582000

                Parameter       Standard         Type II
Variable         Estimate          Error  Sum of Squares           F   Prob>F

INTERCEP      -0.34001347     0.56441815      0.05161826        0.36   0.5548
HSMATH         0.41711592     0.10544213      2.22586203       15.65   0.0010
HSENGL         0.57902131     0.18573410      1.38235265        9.72   0.0063

Bounds on condition number:       1.079973,       4.31989
```

```
Step 1  Variable SATMATH Entered  R-square = 0.80217758  C(p) =  8.15491674

                DF        Sum of Squares      Mean Square          F   Prob>F

Regression       3            5.89265213       1.96421738      21.63   0.0001
Error           16            1.45316787       0.09082299
Total           19            7.34582000

               Parameter        Standard          Type II
Variable        Estimate           Error   Sum of Squares          F   Prob>F

INTERCEP      0.26897930      0.48818658       0.02757159       0.30   0.5893
HSMATH        0.24740837      0.09904667       0.56668902       6.24   0.0238
HSENGL        0.17562383      0.19324941       0.07501124       0.83   0.3769
SATMATH       0.00212935      0.00065330       0.96486381      10.62   0.0049

Bounds on condition number:     2.466307,     17.36901

Step 2  Variable HSENGL Removed    R-square = 0.79196616  C(p) =  7.19529572

                DF        Sum of Squares      Mean Square          F   Prob>F

Regression       2            5.81764088       2.90882044      32.36   0.0001
Error           17            1.52817912       0.08989289
Total           19            7.34582000

               Parameter        Standard          Type II
Variable        Estimate           Error   Sum of Squares          F   Prob>F

INTERCEP      0.64380589      0.25984109       0.55184813       6.14   0.0240
HSMATH        0.23310652      0.09728646       0.51609465       5.74   0.0284
SATMATH       0.00250959      0.00049916       2.27220521      25.28   0.0001

Bounds on condition number:     1.454712,     5.818846

Step 3  Variable SATVERB Entered  R-square = 0.85035772  C(p) =  3.24613734

                DF        Sum of Squares      Mean Square          F   Prob>F

Regression       3            6.24657473       2.08219158      30.31   0.0001
Error           16            1.09924527       0.06870283
Total           19            7.34582000
```



```
               Parameter        Standard          Type II
Variable        Estimate           Error   Sum of Squares          F   Prob>F

INTERCEP      0.33424978      0.25874739       0.11464759       1.67   0.2148
HSMATH        0.17987024      0.08767859       0.28913865       4.21   0.0570
SATMATH       0.00218487      0.00045532       1.58193515      23.03   0.0002
SATVERB       0.00131233      0.00052521       0.42893384       6.24   0.0237

Bounds on condition number:     1.583729,     13.41417

All variables in the model are significant at the 0.10 level.
No other variable met the 0.05 significance level for entry into the model.


        Summary of Stepwise Procedure for Dependent Variable GPA

      Variable         Number  Partial    Model
Step  Entered Removed      In     R**2     R**2        C(p)          F   Prob>F

   1  SATMATH               3   0.1313   0.8022      8.1549    10.6236   0.0049
   2          HSENGL        2   0.0102   0.7920      7.1953     0.8259   0.3769
   3  SATVERB               3   0.0584   0.8504      3.2461     6.2433   0.0237

------------------------------------------------------------------------------
```

Note that in Step 0 the (initial) model contains HSMATH and HSENGL. In Step 1 the variable SATMATH enters. In Step 2 the variable HSENGL is removed, and in Step 3 the variable SATVERB enters. No other variables enter or leave so the final model contains SATMATH, SATVERB, HSMATH, which is the same result as in (7.4.75).

Next we explain how to carry out a forward selection analysis or a backward elimination analysis using proc reg .

Forward Selection

For the sake of discussion, assume that the response variable is named Y, that there are 5 predictor variables, named X1, X2, X3, X4, and X5, respectively, and that the data are in the file **data.ssd** on the data disk. The SAS commands for the forward selection procedure using SLE = 0.05 are as follows.

SAS COMMAND FOR FORWARD SELECTION PROCEDURE

```
libname my 'b:\';
proc reg data=my.data;
model y=x1 x2 x3 x4 x5/selection=forward sle=0.05;
run;
```

For the problem you wish to solve, you must substitute the correct data file name, variable names, and the value of SLE in the appropriate places.

Backward elimination

For the sake of discussion suppose the response variable is named Y and that there are 5 predictor variables, named X1, X2, X3, X4, and X5, respectively. Suppose also that the data are in the file **data.ssd**. For this scenario, SAS commands for the backward elimination procedure are given below. In the command we use `SLS = 0.10` .

SAS COMMAND FOR BACKWARD ELIMINATION PROCEDURE

```
libname my 'b:\';
proc reg data = my.data;
model y = x1 x2 x3 x4 x5/selection=backward sls=0.10;
run;
```

For the problem you wish to solve, you must substitute the correct data file name, variable names, and the value of `SLS` in the appropriate places.

Problems

S7.4.1 Use the SAS commands and options discussed in this section to work Example 7.4.5. Use `sle = 0.15` and `sls = 0.15` in place of *F-in* = 3.0 and *F-out* = 3.0, respectively.

7.5 Growth Curves

In this section we discuss a macro we have supplied on the data disk that will enable you to do the computations necessary for growth curves as discussed in Section 7.5. This macro is named **growth** and it will compute point estimates and confidence intervals for any specified linear combination θ of the model parameters, where

$$\theta = a^T\beta = a_0\alpha + a_1\beta + a_2\gamma + \cdots$$

The SAS commands for this macro are stored in the files **growth.mac** and **growth.sas**. Only polynomial growth curve models can be fitted using this macro.

To use this macro, the Y data must be organized in columns as in Table S7.5.1 below. Also see Table S7.5.2 and Table S7.5.3 below (they are the same as Table 7.5.2 and Table 7.5.3, respectively, in the textbook).

Table S7.5.1

A schematic representation of the sample data for a growth curve study

Item	Response at time t_1	$\cdots$	Response at time t_j	$\cdots$	Response at time t_k
1	$y_{1,1}$	$\cdots$	$y_{1,j}$	$\cdots$	$y_{1,k}$
2	$y_{2,1}$	$\cdots$	$y_{2,j}$	$\cdots$	$y_{2,k}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
i	$y_{i,1}$	$\cdots$	$y_{i,j}$	$\cdots$	$y_{i,k}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
m	$y_{m,1}$	$\cdots$	$y_{m,j}$	$\cdots$	$y_{m,k}$

Table S7.5.2
Drug Concentration Data (in milligrams/liter)

Subject	t_1 1 hour	t_2 2 hours	t_3 3 hours	t_4 4 hours
1	10.55	4.11	2.00	1.02
2	10.47	4.30	2.15	1.11
3	9.46	3.81	1.78	0.94
4	9.27	3.72	1.92	0.95
5	9.37	3.75	1.95	0.97
6	9.67	4.28	1.96	1.04
7	10.58	3.95	2.30	1.08
8	9.96	3.73	1.86	1.01
9	9.84	3.92	2.00	1.05
10	10.20	4.20	1.96	1.03
11	9.45	4.18	2.18	1.02
12	9.64	4.04	2.08	0.96
13	10.03	4.01	2.08	1.04
14	9.81	3.65	1.97	0.97
15	10.74	4.41	2.07	1.03
16	10.08	3.80	1.86	0.99
17	10.00	3.84	2.07	0.95
18	9.73	3.94	1.93	0.96
19	9.64	4.24	2.11	1.06
20	10.40	4.11	2.07	1.01
21	10.34	4.20	2.21	1.14
22	10.09	4.35	1.91	1.07
23	9.51	3.74	1.87	0.99
24	9.63	3.77	1.96	1.01

Table S7.5.3
Ramus Height of 20 Boys

Boy	t_1 age 8	t_2 age $8\frac{1}{2}$	t_3 age 9	t_4 age $9\frac{1}{2}$
1	47.8	48.8	49.0	49.7
2	46.4	47.3	47.7	48.4
3	46.3	46.8	47.8	48.5
4	45.1	45.3	46.1	47.2
5	47.6	48.5	48.9	49.3
6	52.5	53.2	53.3	53.7
7	51.2	53.0	54.3	54.5
8	49.8	50.0	50.3	52.7
9	48.1	50.8	52.3	54.4
10	45.0	47.0	47.3	48.3
11	51.2	51.4	51.6	51.9
12	48.5	49.2	53.0	55.5
13	52.1	52.8	53.7	55.0
14	48.2	48.9	49.3	49.8
15	49.6	50.4	51.2	51.8
16	50.7	51.7	52.7	53.3
17	47.2	47.7	48.4	49.5
18	53.3	54.6	55.1	55.3
19	46.2	47.5	48.1	48.4
20	46.3	47.6	51.3	51.8

The following points should be noted.

(1) The $y_{i,j}$ data must be in consecutive columns as in Tables S7.5.1–S7.5.3, starting with the Y values corresponding to the first time point and ending with the Y values for the last time point. The number of columns is denoted by k, and it is equal to the number of time points $t_1, t_2, \ldots, t_k$, at which each item is observed. Note that $k = 4$ in Tables S7.5.2 and S7.5.3.

(2) The sample size is denoted by m. The value of m is 24 for Table S7.5.2 and m is 20 for Table S7.5.3.

(3) The number of unknown parameters in the growth curve model is denoted by p. For the model in (7.5.7) the value of p is 3. The degree of the polynomial growth curve is $p - 1$, so for the model in (7.5.7) the degree is 2. In Example 7.5.1, the growth curve model is given by $\mu_Y(t) = \alpha + \beta t$, which is a polynomial in t of degree

1; in Example 7.5.2, the growth curve is a polynomial of degree 2 (i.e., quadratic) in t, given by $\alpha + \beta t + \gamma t^2$.

(4) The X matrix has size k by p where p is the number of unknown parameters in the growth model. The first column of X is a column of 1's (in the SAS macro we have assumed that an intercept is present). For a 3^{rd} degree polynomial model, the X matrix has 4 columns, the first column being the column of ones, the second column has elements t_i, the third column has elements t_i^2, and the fourth column has elements t_i^3.

(5) You must input the values of $t_1, t_2, \ldots, t_k$ and the number of unknown parameters in the growth curve model.

(6) You must also input the vector $a = [\, a_0 \;\; a_1 \;\; \cdots \;\; a_{p-1} \,]^T$ consisting of the coefficients in the linear function

$$\theta = a^T\beta = a_0\alpha + a_1\beta + \ldots + a_{p-1}\delta$$

We illustrate the macro by using it to perform the required calculations in Example 7.5.6, where an investigator wants to establish a growth curve for the ramus bone in young boys. A simple random sample of 20 boys was selected and the ramus height for each boy was measured (in millimeters) at ages 8.0, 8.5, 9.0, and 9.5 years. The data are in Table S7.5.3 above and are also in the files **ramus.dat** and **ramus.ssd** on the data disk. We compute a 95% two-sided confidence interval for β, the *average population growth rate* of the ramus bone, where we assume that the *population growth curve* is

$$\mu_Y(t) = \alpha + \beta t$$

Note that for this example $p = 2$, $m = 20$, and $k = 4$. Also the X matrix is

$$X = \begin{bmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_k \end{bmatrix} = \begin{bmatrix} 1 & 8.0 \\ 1 & 8.5 \\ 1 & 9.0 \\ 1 & 9.5 \end{bmatrix} \qquad \text{(S7.5.1)}$$

Further note that $\mathbf{a}^T = [\, 0 \;\; 1 \,]$ and $\theta = \mathbf{a}^T\beta = \beta$.

To use the macro, invoke SAS, and on the `Command Line` of the `PROGRAM EDITOR` window type

```
include 'b:\macro\growth.mac'
```

and press `Enter`. This brings the following statements to the screen.


```
--------------------------------------------------------------------------------
00001 Title 'Growth curve analysis';
00002 libname my 'b:\'; data temp;
00003
00004 ****** On line 00007 enter the name of the SAS data file
00005 ****** that contains the data you want to use;
00006 set
00007                       my.filename
00008
00009 ;proc iml;
00010 ****** On line 00012 enter the number of time points k;
00011 k=
00012                       5
00013 ;
00014 ****** On line 00016 enter the values of t1  t2  t3 ... tk;
00015 t={
00016                       2  4  6  8  10
00017 };
00018
00019 ****** On line 00022 enter  p  the number of unknown parameters
00020 ****** in the polynomial growth curve model;
00021 p=
00022                       3
00023 ;
00024 ****** On line 00027 enter the coefficients of the vector  a .
00025 ****** Enter them in the order   a0  a1  a2  a3 ...     ;
00026 a={
00027                       1  0  0  0
00028 };
00029 ****** On line 00031 enter the confidence coefficient;
00030 c=
00031                       0.95
00032
00033
00034 ;%include 'b:\macro\growth.sas';
--------------------------------------------------------------------------------
```

For Example 7.5.6, you must enter the following data on the indicated lines.

(1) On line `00007` enter `my.ramus` to replace `my.filename`.

(2) On line 00012 enter the number `4` to replace `5` since there are 4 time points in this example.

(3) On line 00016 enter `8.0 8.5 9.0 9.5` to replace `2 4 6 8 10`.

(4) On line 00022 enter `2` to replace `3` since we are using a first degree polynomial and hence the number of parameters is 2.

(5) On line 00027 enter `0 1` to replace `1 0 0 0`.

(6) On line 00031 enter the confidence coefficient you want to use to replace `0.95` unless you want to use `0.95` itself. So, in the present example leave the value `0.95` as is.

After entering the appropriate values and checking them, press the `F10` key and SAS will execute the commands contained in the macro. The following results will be displayed in the `OUTPUT` window.

```
----------------------------------------------------------------------
                         Growth curve analysis

 The estimated beta coefficients are

                   33.7475
                     1.866


 The estimated value of theta is     1.866
 and its standard error is       0.2605989


 For a two-sided confidence interval for theta with
 confidence coefficient equal to  95%

 the lower confidence bound is  1.3205602  and
 the upper confidence bound is  2.4114398
----------------------------------------------------------------------
```

From this we get $\hat{\beta} = 1.866$ millimeters, and the 95% confidence statement for β is

$$C[1.32 \leq \beta \leq 2.41] = 0.95$$

Problems

S7.5.1 Work Problem 7.5.1 using the macro described in this section.

S7.5.2 Work Exercise 7.6.2 using the macro described in this section.

Chapter 8

Alternate Assumptions for Regression

8.1 Overview

No computing instructions are needed in this section.

8.2 Straight Line Regression with Unequal Subpopulation Standard Deviations

In this section we demonstrate how SAS can be used to compute weighted regression calculations for a straight line model. We illustrate by using the data of Example 8.2.1 which are given in Table 8.2.1, and are also stored in the files **carbmon.dat** and **carbmon.ssd** on the data disk. The response variable Y is named `CO` (carbmon monoxide) and the predictor variable X is named **cars**.

In this example, it is known that the subpopulation standard deviations $\sigma_Y(X)$ are not all the same, but the investigator expects the weighted regression assumptions in Box 8.2.1 to hold with $\sigma_Y(x) = \sigma_0 g(x)$ where σ_0 is an unknown constant and $g(x) = \sqrt{x}$. We explain the SAS commands for estimating β_0 and β_1, and for computing standard errors of these estimates, using a weighted least squares regression analysis where the

user supplies the 'weights'. In fact, you must first create a dataset which contains the response variable, the predictor variable, and the weights. For the present example, this is done using the following SAS statements. We have also included the command to print the dataset created so that we can check the numbers in it.

```
libname my 'b:\';
data tempcarb;
set my.carbmon;
wts=1/cars;
proc print data=tempcarb;
run;
```

The first statement is a `libname` statement which has been discussed earlier. The second statement asks SAS to create a temporary data set and give it the name `tempcarb` (**temp**orary **carb**on monoxide). Statement three asks SAS to copy the contents of the file **carbmon.ssd** into the data set `tempcarb`. Statement four instructs SAS to compute a new variable named `wts` (short for *weights*) and it is to be equal to `1/cars` (i.e., $1/[g(X)]^2 = 1/X =$ `1/cars`). This new variable will be part of the (temporary) data set `tempcarb` that is being created. Statement five requests SAS to print the contents of this data set.

After entering these commands in the `PROGRAM EDITOR` window and pressing the `F10` key, SAS responds with the following in the `OUTPUT` window.

OBS	CO	CARS	WTS
1	5817	873	.0011455
2	1063	109	.0091743
3	2616	398	.0025126
4	2018	353	.0028329
5	3147	506	.0019763
6	7210	1026	.0009747
7	4339	862	.0011601
8	5153	742	.0013477
9	4450	786	.0012723
10	5591	896	.0011161
11	2747	377	.0026525
12	3712	720	.0013889
13	2354	655	.0015267

Next we give the SAS commands for performing a weighted regression of $Y =$ CO on $X =$ cars, using the weights in wts .

COMMAND FOR WEIGHTED REGRESSION

```
00001 proc reg data=tempcarb;
00002 model co=cars/i;
00003 weight wts;
00004 run;
```

Note that we use the same SAS procedure, viz., proc reg , for weighted regression as we did for ordinary regression. In the model statement we specify that the model to be fitted is

$$\mu_Y(x) = \beta_0 + \beta_1 x$$

where Y is co (it is immaterial whether we use lower case or upper case letters for the variable names) and X is cars. We have also asked SAS to print out the matrix $C^{(w)}$, the weighted C matrix. This is done by using the keyword i as an option following the *slash* (/) in the model statement. It is the weight statement in line 00003 that tells SAS to perform a weighted regression using the variable named wts which contains the weights.

Note that these commands will not execute if they are not used in the same SAS session as the one in which the temporary data set tempcarb was created since the latter data set will be lost when you exit SAS. In that case you must create tempcarb again as explained above. After entering the above commands in the PROGRAM EDITOR window and pressing the F10 key, SAS responds as follows.

```
Model: MODEL1

          X'X Inverse, Parameter Estimates, and SSE

                 INTERCEP            CARS                CO

INTERCEP      114.59571238     -0.179422409      371.62089155
CARS          -0.179422409     0.0004013599      5.4662084078
CO            371.62089155     5.4662084078      8498.6932364
```



```
Dependent Variable: CO

                         Analysis of Variance

                            Sum of          Mean
Source           DF        Squares        Square      F Value        Prob>F

Model             1    74445.48837   74445.48837       96.356        0.0001
Error            11     8498.69324     772.60848
C Total          12    82944.18161

      Root MSE      27.79584     R-square       0.8975
      Dep Mean    2815.21394     Adj R-sq       0.8882
      C.V.           0.98734

                          Parameter Estimates

                     Parameter       Standard     T for H0:
Variable  DF          Estimate          Error    Parameter=0    Prob > |T|

INTERCEP   1        371.620892   297.55271584          1.249        0.2376
CARS       1          5.466208     0.55686090          9.816        0.0001
```

Note that the computer output has the same "form" as the output from an ordinary regression analysis. In particular, the output contains the matrix $C^{(w)}$ (the first two rows and columns printed under the label X'X Inverse), an ANOVA table, and the point estimates of β_0 and β_1 along with their standard errors.

Problems

S8.2.1 Work Problems 8.2.1 through 8.2.4 using the commands discussed in this section.

S8.2.2 Work Exercise 8.4.1 using SAS to do the computations.

8.3 Straight Line Regression–Theil's Method

In this section we explain a macro named **theil** supplied by us on the data disk, that can be used to perform the calculations necessary to obtain point estimates and confidence intervals for the linear combination

$$\theta = a_0\beta_0 + a_1\beta_1$$

in the straight line regression model

$$\mu_Y(x) = \beta_0 + \beta_1 x$$

using Theil's method for straight line regression. We suppose that the assumptions in Box 8.3.1 are satisfied. The SAS commands for this macro are stored in the files **theil.mac**, **theil1.sas**, and **theil2.sas** on the data disk. We illustrate the macro by using the data of Example 8.3.1 which are given in Table 8.3.3 and are also stored in the files **profsal.dat** and **profsal.ssd** on the data disk. In this example the Y data are annual salaries, labeled `salary` in the data set, and the X data are number of years of experience, labeled `yrsexp` in the data set. We compute a point estimate for $\mu_Y(10)$ and a confidence interval for it with confidence coefficient as close to 90% as possible.

Invoke SAS, and on the Command line of the `PROGRAM EDITOR` window type

```
include 'b:\macro\theil.mac'
```

and press `Enter`. This will bring the following statements to the `PROGRAM EDITOR` window.

```
--------------------------------------------------------------------------
00001 Title 'Straight line regression using the method of Theil';
00002 options nodate center ls=75 ps=60;
00003
00004 libname my 'b:\';
00005
00006 ****** On line 00010 enter the name of the SAS data file
00007 ****** that contains the data set you want to use;
00008 data rawdata(keep = yvar xvar);set
00009
00010                          my.filename
00011
00012 ;
00013 ****** On line 00018 enter the name of the response variable as it
00014 ****** appears in the data file;
00015 ****** On line 00020 enter the name of the predictor variable as it
00016 ****** appears in the data file;
00017 rename
00018                          response variable
00019 = yvar
00020                          predictor variable
00021 = xvar
00022
00023 ;%include 'b:\macro\theil1.sas';
00024
00025 proc iml;
00026
00027 ****** On line 00031 enter a(0) and a(1), the coefficients of
00028 ****** beta(0) and beta(1) (in this order), that you want to use;
00029 a={
00030
00031                          1      100
00032 };
00033 ****** On line 00035 enter the confidence coefficient;
00034 cc=
00035                          0.95
00036 ;
00037
00038 %include 'b:\macro\theil2.sas';
--------------------------------------------------------------------------
```


Enter the following information on the indicated lines, replacing the quantities already there if necessary.

```
00010          my.profsal
00018          salary
00020          yrsexp
00031          1    10
00035          0.90
```

After entering these quantities, press the `F10` key to execute the macro commands. The following results will appear in the `OUTPUT` window.

```
-------------------------------------------------------------------

          Straight line regression using the method of Theil

 The point estimate of theta is     46.444444


 For a two-sided confidence interval for theta with confidence
 coefficient equal to  0.875 (this is the value that is closest
 to the desired value of  0.900)

 the lower confidence bound is         44 and
 the upper confidence bound is  48.333333
-------------------------------------------------------------------
```

Recall that exact confidence intervals are available by Theil's method only for a special set of confidence coefficients, so the macro will automatically choose an allowable confidence coefficient that is closest to the desired one. In this example, the desired confidence coefficient is 0.90 and the confidence coefficient that is closest to 0.90 for which an exact confidence interval is available, is 0.875. We thus get,

$$C[44 \leq \mu_Y(10) \leq 48.333333] = 0.875$$

The point estimate of $\mu_Y(10)$ is $\hat{\mu}_Y(10) = 46.444444$.

Problems

S8.3.1 In Example 8.3.1, use the macro discussed in this section and find a point estimate and a confidence interval for β_0, with confidence coefficient as close to 90% as possible (assumptions in Box 8.3.1 are presumed valid).

S8.3.2 In Example 8.3.1, use the macro discussed in this section and find a point estimate and a confidence interval for β_1 with confidence coefficient as close to 90% as possible (assumptions in Box 8.3.1 are presumed valid).

S8.3.3 Work Problem 8.3.1 by using the SAS macro explained in this section.

Chapter 9

Nonlinear Regression

9.1 Overview.

No computing instructions are needed in this section.

9.2 Some Commonly Used Families of Nonlinear Regression Functions

No computing instructions are needed in this section.

9.3 Statistical Assumptions and Inferences for Nonlinear Regression

In this section we describe how to use the program NLIN (short for Non LINear) available in SAS for solving nonlinear regression problems. NLIN is a general purpose nonlinear regression program that, in principle, is capable of fitting any nonlinear regression model. Before invoking this program you must first create a SAS dataset consisting of the data you want to use. The dataset may be a temporary dataset created during a data step of the current SAS session, or it may be a permanent dataset stored in a file.

As part of the instructions for fitting a nonlinear regression model you must provide the following information.

(1) The name of the dataset to be used.

(2) The functional form of the regression function (i.e., $\beta_0 + e^{\beta_1 x}$, *etc.*).

(3) Initial guesses for the model parameters.

(4) Whether or not you want any diagnostic statistics saved in a file, and if you do, then the name of the file where the diagnostic statistics are to be saved.

(5) Maximum number of iterations to be performed.

(6) Criteria for deciding whether or not the algorithm has converged, and

(7) The numerical method to be used for fitting the model.

Many other options are available for controlling the model fitting process and you should refer to the SAS/STAT guide for further details.

We illustrate the use of `proc nlin` with the data from Example 9.3.1 which are given in Table 9.3.1 and are also stored in the file **light.ssd**. You should print and examine these data before proceeding with the analysis. The SAS commands for fitting the model

$$\mu_Y(x) = \beta_1 + \beta_2 e^{-\beta_3 x}$$

and for plotting the results to visually examine the adequacy of the fit are as follows.

COMMANDS FOR USING THE NLIN PROCEDURE

```
options center linesize=75 pagesize=60;
libname my 'b:\';

proc nlin data=my.light method=dud maxiter=20;
model reading = beta1+beta2*exp(-beta3*concentr);
parms beta1=0.0  beta2=2.0  beta3=0.5;
output out=diagnstc p=fits r=residual student=stdresid;

proc plot data=diagnstc;
plot reading*concentr='o' fits*concentr='*'/overlay
     hpos=50  vpos=25;
run;
```

The first set of (two) statements specify some options and the libname. The first statement declares certain options specifying how the output is to be printed. According to this statement, the output will be centered on the page, the width of each line will be 75 characters, and the maximum page size will be 60 lines of text. The second statement is the usual `libname` statement giving the nickname `my` to the directory `b:\` .

The second set of (four) statements relate to the program NLIN. The first statement in this group asks SAS to use the program NLIN to analyze the data in the SAS dataset **light.ssd**. It also specifies that the numerical method to be used is the method called **dud** (which is short for **d**oesn't **u**se **d**erivatives) and that the maximum number of iterations to perform is 20. Statement two in this group specifies the model as

$$\mu_Y(x) = \beta_1 + \beta_2 e^{-\beta_3 x}$$

where the actual name of the predictor variable, viz., `concentr`, is used instead of the symbol x. The initial guesses for the unknown parameters in the nonlinear regression model are specified in the third statement of this second group of statements. The parameters are β_1, β_2, and β_3, and the initial guesses for their values are 0.0, 2.0, and 0.5, respectively. Statement four in this group requests SAS to create a dataset named `diagnstc` and specifies that this dataset is to contain the predicted values (`fits`), the residuals, and the standardized residuals, in addition to all the original variables in the file **light.ssd**. The column containing the predicted values is named `fits`, the column containing the residuals is named `residual`, and the column containing the standardized residuals is named `stdresid`. The syntax here is the same as what you are already familiar with as part of `proc reg` from Chapters 3 and 4.

The last group of four statements are needed to obtain plots of reading (Y) against concentr (X) and fits against concentr. The statement on line 00009 invokes the plot procedure and declares that the data to be used are in the dataset named diagnstc. The statement on line 00010 requests SAS to produce two plots, the first plot being that of reading against concentr (using the symbol o) and the second being that of fits against concentr (using the symbol *). The options following the 'forward slash' symbol / specify that the second plot is to be overlayed on the first plot, and also that the horizontal and vertical dimensions for the plot are 50 characters and 25 characters respectively (these are part of the proc plot statement since no semicolon ends the preceding line. The last statement is the usual run statement.

After entering these commands in the PROGRAM EDITOR window and checking them carefully, press the F10 key to execute the commands. The results from these commands appear in the OUTPUT window and are given below.

```
 Non-Linear Least Squares DUD Initialization    Dependent Variable READING
     DUD          BETA1          BETA2          BETA3 Sum of Squares
      -4              0       2.000000       0.500000       1.688465
      -3       0.100000       2.000000       0.500000       1.506433
      -2              0       2.200000       0.500000       1.145670
      -1              0       2.000000       0.550000       1.725670

                 Non-Linear Least Squares Iterative Phase
                 Dependent Variable READING   Method: DUD
    Iter          BETA1          BETA2          BETA3 Sum of Squares
       0              0       2.200000       0.500000       1.145670
       1       0.089871       2.662030       0.749880       0.473436
       2       0.011669       2.742061       0.692222       0.465652
       3       0.031511       2.721756       0.679821       0.460730
       4       0.027979       2.724874       0.682355       0.460430
       5       0.027898       2.723823       0.681937       0.460429
       6       0.028990       2.722912       0.682828       0.460427
       7       0.028761       2.723274       0.682774       0.460427
       8       0.028763       2.723274       0.682773       0.460427
NOTE: Convergence criterion met.

   Non-Linear Least Squares Summary Statistics     Dependent Variable READING

      Source                DF Sum of Squares       Mean Square

      Regression             3   20.542872863       6.847624288
      Residual               9    0.460427137       0.051158571
      Uncorrected Total     12   21.003300000

      (Corrected Total)     11   10.605891667
```



```
   Parameter     Estimate    Asymptotic              Asymptotic 95 %
                             Std. Error            Confidence Interval
                                                  Lower          Upper
      BETA1   0.028763192 0.17163881268 -0.3595140815  0.4170404648
      BETA2   2.723273503 0.21054950823  2.2469733725  3.1995736341
      BETA3   0.682773200 0.14160078051  0.3624472546  1.0030991454

                       Asymptotic Correlation Matrix

   Corr                BETA1              BETA2              BETA3
   ---------------------------------------------------------------
      BETA1                1       -0.677677022       0.8462620108
      BETA2     -0.677677022                  1       -0.396324924
      BETA3     0.8462620108       -0.396324924                  1

        Plot of READING*CONCENTR.   Symbol used is 'o'.
        Plot of FITS*CONCENTR.  Symbol used is '*'.

          |
  READING |
          |
          |
        3 +
          | o
          | o
          |
          |
          |
        2 +
          |
          |
          |          o
          |          *
          |          o
        1 +                   o
          |
          |                   *          o
          |                   o
          |                              *                    o
          |                              o          o         *
        0 +                                         o         o
          |
          --+---------+---------+---------+---------+---------+---
            0         1         2         3         4         5
                                  CONCENTR
NOTE: 8 obs hidden.
```

The numerical procedure performs some preliminary exploration of the fit of the model at and around the initial guesses for the parameters. The results of this step are reported under the heading `Non-Linear Least Squares DUD Initialization`. After this step, the algorithm performs several iterations in an attempt to find values for the β parameters that might yield a better fit to the data. After each interation, the program prints out the updated values for the parameters and the value of `Sum of Squares`, the sum of squared errors (SSE). When the value of `Sum of Squares` fails to decrease appreciably, and the changes in the parameter values are negligible, the process is terminated and the final results are printed. In the above output, SAS has decided that the algorithm has converged after 8 iterations. At this point summary statistics are printed. These include an ANOVA table, final parameter estimates, their approximate standard errors (labeled `Asymptotic Std. Error`), and one-at-a-time two-sided 95% confidence intervals for each β parameter. Finally, the program prints out the `Asymptotic Correlation Matrix` for the parameter estimates which is useful for more advanced calculations than those discussed in the textbook.

The plots of `reading` against `concentr` and `fits` against `concentr` indicate that the fit of the model to the data is quite good.

You should note that several iteration methods are available in the program NLIN, but all of them, with the exception of the method `dud`, require a knowledge of calculus. For this reason we do not discuss them here, but if you know calculus then you can refer to the SAS/STAT guide for information on relative advantages and disadvantages of the different methods and on how to use the other methods.

Finally, you should note that the computer output in Exhibit 9.3.1 in the textbook was obtained by using the program NLIN. However, only selected portions of the output are given in that Exhibit.

Note that, in this example, convergence was obtained after 8 iterations. This is partly because the choice of the initial values for this problem turned out to be good. This will not always be the case and most problems require many more iterations. If it appears that convergence has not been attained at the end of the specified number of iterations, then you can rerun the program using the parameter values from the final iteration as your initial guess for the new run.

There is no guarantee that the result given by the program is in fact correct even when convergence appears to have been attained. This is true of most nonlinear regression programs. The reader must check the results carefully. It is often useful to try different starting values and see if the final results are the same. If problems are

encountered we recommend that you consult a statistician.

Problems

S9.3.1 Consider the experiment discussed in Problem 9.3.3 where we provided a SAS output containing the results of a nonlinear regression analysis for that problem. See Exhibit 9.3.3. What are the SAS commands required to obtain this output? Use the SAS procedure NLIN and compare your results with those given in Exhibit 9.3.3. For starting values for β_1, β_2, and β_3, you may use the values 0.8, −0.67, and 0.16, respectively. You may use other starting values but convergence can not be guaranteed if starting values are chosen arbitrarily.

S9.3.2 Solve Problem 9.3.2 using the SAS procedure NLIN. Use $\beta_1 = 2$, $\beta_2 = -1$, and $\beta_3 = 14$ as starting values.

S9.3.3 Work Example 9.4.1 using the SAS procedure NLIN. Use $\beta_1 = -3.0$ and $\beta_2 = 150.0$ as starting values.

9.4 Linearizable models

All SAS commands needed in this section have already been discussed.

Problems

S9.4.1 Use the procedure NLIN in SAS to fit the nonlinear model in Example 9.4.1. Use the estimates obtained from the linearization approach as the initial guesses for β_1 and β_2.

S9.4.2 Use the SAS procedure NLIN to fit the nonlinear model in Problem 9.4.1. Use the estimates obtained from the linearization approach as the initial guesses for β_1, β_2, and β_3.

Answers and Solutions

S1.1.1 The SAS commands are

```
data prob111;
input y x z;
cards;
1.5 600 34.5
1.9 590 43.9
1.2 710 30.3
2.1 560 31.7
1.6 610 42.1
1.7 700 39.0
;
run;
```

S1.1.2 The SAS commands are

```
proc print data=prob111;
run;
```

This assumes that you are in the same SAS session in which you created the temporary dataset prob111. Otherwise this temporary dataset will not be available for you to print. Remember to press the F10 key to execute the command statements.

The SAS response which appears in the OUTPUT window is

OBS	Y	X	Z
1	1.5	600	34.5
2	1.9	590	43.9
3	1.2	710	30.3
4	2.1	560	31.7
5	1.6	610	42.1
6	1.7	700	39.0

S1.1.3 The SAS commands which you type in the PROGRAM EDITOR window are

```
proc means data=prob111;
run;
```

The following results appear in the OUTPUT window.

N Obs	Variable	N	Minimum	Maximum	Mean	Std Dev
6	Y	6	1.2000000	2.1000000	1.6666667	0.3141125
	X	6	560.0000000	710.0000000	628.3333333	61.7791766
	Z	6	30.3000000	43.9000000	36.9166667	5.6001488

So $\hat{\mu}_X = 628.3333333$, $\hat{\mu}_Y = 1.6666667$, and $\hat{\mu}_Z = 36.9166667$.

S1.1.4 The required SAS commands are

```
libname my 'b:\';
proc contents data=my.table164;
run;
```

The SAS response which appears in the OUTPUT window is

```
-----------------------------------------------------------------------

                        CONTENTS PROCEDURE

 Data Set Name:  MY.TABLE164            Type:
 Observations:   30                     Record Len: 12
 Variables:      1
 Label:

          -----Alphabetic List of Variables and Attributes-----

 #  Variable  Type  Len  Pos  Label
 1  Y         Num     8    4

-----------------------------------------------------------------------
```

Thus there is one variable (named Y) and 30 observations.

S1.1.5 The appropriate SAS commands are

```
libname my 'b:\';
proc means data=my.table164 mean std;
run;
```

The following result appears in the OUTPUT window.

```
-----------------------------------------------------------------------

                  Analysis Variable : Y


                  N Obs          Mean        Std Dev
                  ---------------------------------
                     30     6.9890000      3.5437827
                  ---------------------------------

-----------------------------------------------------------------------
```

Thus the mean is 6.989 and the standard deviation is 3.5437827.

S1.1.6 The required SAS commands are


```
libname my 'b:\';
proc contents data=my.agebp;
run;
```

The results which appear in the OUTPUT window are

```
-----------------------------------------------------------------------

                        CONTENTS PROCEDURE

 Data Set Name:  MY.AGEBP               Type:
 Observations:   24                     Record Len: 20
 Variables:      2
 Label:

          -----Alphabetic List of Variables and Attributes-----

 #  Variable  Type  Len  Pos  Label
 2  AGE       Num     8   12
 1  BP        Num     8    4

-----------------------------------------------------------------------
```

There are two variables, named bp and age, respectively, and 24 observations in this dataset.

S1.1.7 The appropriate SAS commands are

```
libname my 'b:\';
proc means data=my.agebp max;
run;
```

The results which appear in the OUTPUT window are

```
 ----------------------------
 N Obs  Variable      Maximum
 ----------------------------
    24  BP        177.0000000
        AGE        67.0000000
 ----------------------------
```

The maximum value of BP is 177 and the maximum value of AGE is 67.

S1.1.8 The required SAS commands are

```
libname my 'b:\';
proc means data=my.agebp mean std;
run;
```

The results which appear in the OUTPUT window are

```
---------------------------------------------------------------------

              N Obs  Variable          Mean         Std Dev
              -----------------------------------------
                 24  BP         139.1250000      20.3391088
                     AGE         44.9583333      12.5264214
              -----------------------------------------

---------------------------------------------------------------------
```

Thus, the mean and the standard deviation for BP are 139.125 and 20.3391088, respectively. The mean and the standard deviation for AGE are 44.9583333 and 12.5264214, respectively.

S1.1.9 Use the following SAS commands.

```
libname my 'b:\';
proc print data=my.agebp;
run;
```

The results which appear in the OUTPUT window are

```
---------------------------------------------------------------------

                          OBS     BP     AGE

                            1    116      34
                            2    112      26
                            3    151      51
```



```
                            4    161      58
                            5    122      34
                            6    129      40
                            7    119      31
                            8    158      57
                            9    144      46
                           10    150      53
                           11    111      29
                           12    148      50
                           13    135      40
                           14    126      34
                           15    172      67
                           16    100      23
                           17    139      47
                           18    135      42
                           19    163      61
                           20    128      38
                           21    159      57
                           22    177      66
                           23    135      42
                           24    149      53

---------------------------------------------------------------------
```

S1.1.10 The required SAS commands are

```
proc contents data=my.chol;
run;
```

The results which appear in the OUTPUT window are

```
---------------------------------------------------------------------

                         CONTENTS PROCEDURE

 Data Set Name:  MY.CHOL                      Type:
 Observations:   20                           Record Len: 20
 Variables:      2
 Label:
```

```
-----Alphabetic List of Variables and Attributes-----

 # Variable  Type  Len  Pos  Label
 2 DAILYFAT  Num     8   12
 1 TOTLCHOL  Num     8    4
-----------------------------------------------------------------------
```

Thus the file **chol.ssd** contains two variables named DAILYFAT and TOTLCHOL, respectively, and twenty observations on each variable.

S1.1.11 Use the following SAS commands.

```
proc print data=my.chol;
run;
```

The following result appears in the OUTPUT window.

OBS	TOTLCHOL	DAILYFAT
1	130	21
2	163	29
3	169	43
4	136	52
5	187	56
6	193	64
7	170	77
8	115	81
9	196	84
10	237	93
11	214	98
12	239	101
13	258	107
14	283	109
15	242	113
16	289	120
17	298	127
18	271	134
19	297	148
20	316	157

S1.1.12 The required SAS statements are


```
proc means data=my.chol mean std;
run;
```

The following response appears in the OUTPUT window.

N Obs	Variable	Mean	Std Dev
20	TOTLCHOL	220.1500000	61.6008160
	DAILYFAT	90.7000000	38.1646020

S1.1.13 The required SAS statements are

```
proc means data=my.chol min max;
run;
```

The SAS response is

N Obs	Variable	Minimum	Maximum
20	TOTLCHOL	115.0000000	316.0000000
	DAILYFAT	21.0000000	157.0000000

S1.6.1 The SAS commands are

```
libname my 'b:\';
data tab164;
set my.table164;
proc contents data=tab164;
proc print data=tab164;
run;
```

S1.6.2 The SAS commands for obtaining the mean, the standard deviation, and the standard error of the mean are as follows.

```
libname my 'b:\';
proc means data=my.table164 mean std stderr;
run;
```

As usual, you type these commands in the PROGRAM EDITOR window and press the F10 key. The following result appears in the OUTPUT window.

```
------------------------------------------------------------------------
              Analysis Variable : Y

              N Obs          Mean         Std Dev      Std Error
              ---------------------------------------------------
                 30     6.9890000       3.5437827      0.6470032
              ---------------------------------------------------
------------------------------------------------------------------------
```

In particular, we get $\hat{\mu}_Y = 6.989$.

S1.6.3 Using the results from the preceding output you can obtain a 80% two-sided confidence interval for μ_Y, and from this you can obtain a 90% upper confidence bound. Refer to the appropriate formula in Table 1.6.2. You will need the tabled t-value from Table T-2 in Appendix T. The degrees of freedom are $n - 1 = 29$ and hence from the table we get $t_{0.90,29} = 1.311$. The 80% two-sided confidence interval for μ_Y is given by

$$C[6.1408 \leq \mu_Y \leq 7.8372] = 0.80$$

Hence the desired one-sided confidence statement is

$$C[\mu_Y \leq 7.8372] = 0.90$$

S1.6.4 Using the procedure described in Box 1.6.1 we get

$$t_C = (6.989 - 4.5)/0.6470032 = 3.847$$

with degrees of freedom equal to 29.

S1.6.5 From TableT-2 in Appendix T we find that the value of $1 - \alpha/2$ for which $t_{1-\alpha/2:29} = |t_C| = 3.847$ is between 0.9995 and 1.0 (this is so because $t_{0.9995:29} = 3.659$, which is less than 3.847, and $t_{1.0:29}$ is infinity, which is greater than 3.847). Hence the value of α for which $t_{1-\alpha/2:29} = |t_C| = 3.847$ is between 0 and 0.001. Thus the P-value is a number between 0 and 0.001.

The SAS statements to compute and print the exact P-value for a test with a two-sided alternative, and the corresponding SAS response are as follows.

```
data temp;
pvalue=2*(1-probt(3.847,29));
proc print data=temp;
run;
```

```
------------------------------------------------------------------------
                         OBS      PVALUE

                          1      .00060513
------------------------------------------------------------------------
```

Hence the P-value is 0.0006 (rounded to four decimals).

S1.6.6 Using the procedure in Box 1.6.1 we get $t_C = (6.989 - 5.0)/0.647 = 3.074$. The SAS statements to compute and print the P-value for the required one-sided test (as in part (b) of Table 1.6.3), and the corresponding SAS output are given below.

```
data temp;
pvalue=1-probt(3.074,29);
proc print data=temp;
run;
```

```
------------------------------------------------------------------------
                       OBS     PVALUE

                        1     .0022841
------------------------------------------------------------------------
```

Thus the P-value is 0.0023 (rounded to four decimals).

Note: If you want to compute the P-value for a one-sided test of NH: $\mu_Y \geq 5.0$ versus AH: $\mu_Y < 5.0$ (as in part (c) of Table 1.6.3) use the following SAS statements.

```
data temp;
pvalue=probt(3.074,29);
proc print data=temp;
run;
```

S1.8.1 (a) The SAS commands for reading y and X into the computer are as follows. As usual, you type the commands in the PROGRAM EDITOR window and press the F10 key to execute the statements.

```
proc iml;
reset nolog;
X={
12 28 21,
14 31 46,
20 21 31,
11 19 21,
16 13 34,
39 26 30,
25 37 15
};
y={9, 13, 28, 6, 32, 16, 24};
print X y;
```

The results which appear in the OUTPUT window are

```
---------------------------------------------------------------------------

                    X                                       Y
                   12          28          21               9
                   14          31          46              13
                   20          21          31              28
                   11          19          21               6
                   16          13          34              32
                   39          26          30              16
                   25          37          15              24

---------------------------------------------------------------------------
```


We first give the instructions for computing the matrices needed in (b)–(h). All of the computed matrices will be printed at the end. These commands should be issued during the same IML session during which the matrices X and y were created. Otherwise, SAS will not remember what these matrices are.

(b) The command to compute X^T, and place the result in a matrix named XTRAN, is

XTRAN = X‘;

The command to compute $X^T\mathbf{X}$, and place the result in a matrix named XTRANX, is

XTRANX = X‘ * X;

(c) The command to compute $\mathbf{X}^T y$, and place the result in a vector named XTRANy, is

XTRANy = X‘ * y;

(d) The command to compute $(X^TX)^{-1}$, and place the result in a matrix named C, is

C = inv(X‘ * X);

(e) The command to compute $(X^TX)^{-1}X^Ty$, and place the result in a vector named BETA, is

BETA = inv(X‘ * X) * X‘ * y;

(f) The command to compute y^Ty, and place the result in a scalar (i.e., 1 by 1 matrix) named SUMSQY, is

SUMSQY = y‘ * y;

(g) The command to compute $y^T[I - X(X^TX)^{-1}X^T]y$, and place the result in a scalar named SSE, is

SSE = y‘ * (I(7) − X * inv(X‘ * X) * X‘) * y;

The statement I(7) tells SAS to create the 7×7 identity matrix **I**.

(h) In SAS/IML, the expression j(r,c,k) represents a r by c matrix whose elements are all equal to k. So, to create a 7 by 7 matrix J whose elements are all equal to one, the SAS command is

```
J=j(7,7,1);
```

Thus the command to create E is

E = I(7) − (1/7) * J;

(i) The command is K=E*E;

(j) The command to compute $y^T[(1/7)J]y$, and place the result in a scalar named ybarsq, is

$$\texttt{ybarsq} = (1/7) * \texttt{y}' * \texttt{J} * \texttt{y};$$

(k) To compute $\bar{y}$, and place the result in a scalar named ybar, the command is

$$\texttt{ybar} = (1/7) * \texttt{j}(1,7,1) * \texttt{y};$$

(l) The command to obtain the sum of all elements in a matrix named A, and place the result in a scalar named sumA, is

$$\texttt{sumA} = \texttt{sum(A)}$$

This command can be used for vectors as well, since vectors are special cases of matrices. So SSY may be computed using the command

$$\texttt{SSY} = (\texttt{y} - (1/7) * \texttt{sum(y)})' * (\texttt{y} - (1/7) * \texttt{sum(y)});$$

(m) The command to compute EJ, and place the result in a matrix named G, is

$$\texttt{G} = \texttt{E} * \texttt{J};$$

Note: In all the preceding commands, it makes no difference whether you use upper or lower case letters or a mixture.

The matrices which were computed above may be printed using the following statement.

```
print XTRANX XTRANy C BETA SUMSQY E SSE K ybarsq ybar SSY G;
```

The results which appear in the OUTPUT window are

```
   XTRANX                         XTRANY         C
     3263      3546      3836       2652 0.0017193 -0.000975 -0.000301
     3546      4761      4841       3077 -0.000975 0.0015474 -0.000601
     3836      4841      6240       3709 -0.000301 -0.000601 0.0008115
```



```
                    BETA     SUMSQY
               0.4448968       2926
               -0.053762
               0.3626025

          E

0.8571429 -0.142857 -0.142857 -0.142857 -0.142857 -0.142857 -0.142857
-0.142857 0.8571429 -0.142857 -0.142857 -0.142857 -0.142857 -0.142857
-0.142857 -0.142857 0.8571429 -0.142857 -0.142857 -0.142857 -0.142857
-0.142857 -0.142857 -0.142857 0.8571429 -0.142857 -0.142857 -0.142857
-0.142857 -0.142857 -0.142857 -0.142857 0.8571429 -0.142857 -0.142857
-0.142857 -0.142857 -0.142857 -0.142857 -0.142857 0.8571429 -0.142857
-0.142857 -0.142857 -0.142857 -0.142857 -0.142857 -0.142857 0.8571429

                              SSE
                        566.66789

          K
0.8571429 -0.142857 -0.142857 -0.142857 -0.142857 -0.142857 -0.142857
-0.142857 0.8571429 -0.142857 -0.142857 -0.142857 -0.142857 -0.142857
-0.142857 -0.142857 0.8571429 -0.142857 -0.142857 -0.142857 -0.142857
-0.142857 -0.142857 -0.142857 0.8571429 -0.142857 -0.142857 -0.142857
-0.142857 -0.142857 -0.142857 -0.142857 0.8571429 -0.142857 -0.142857
-0.142857 -0.142857 -0.142857 -0.142857 -0.142857 0.8571429 -0.142857
-0.142857 -0.142857 -0.142857 -0.142857 -0.142857 -0.142857 0.8571429

                 YBARSQ7      YBAR       SSY
               2340.5714 18.285714 585.42857

          G
 2.22E-16  2.22E-16  2.22E-16  2.22E-16  2.22E-16  2.22E-16  2.22E-16
 2.22E-16  2.22E-16  2.22E-16  2.22E-16  2.22E-16  2.22E-16  2.22E-16
 1.11E-16  1.11E-16  1.11E-16  1.11E-16  1.11E-16  1.11E-16  1.11E-16
 1.11E-16  1.11E-16  1.11E-16  1.11E-16  1.11E-16  1.11E-16  1.11E-16
 1.11E-16  1.11E-16  1.11E-16  1.11E-16  1.11E-16  1.11E-16  1.11E-16
1.665E-16 1.665E-16 1.665E-16 1.665E-16 1.665E-16 1.665E-16 1.665E-16
 2.22E-16  2.22E-16  2.22E-16  2.22E-16  2.22E-16  2.22E-16  2.22E-16
```

You can perform some of the calculations by hand to convince yourself that the results are correct.

Note: The elements of the matrix G are all supposed to be *exactly* equal to zero in the absence of rounding errors. However, rounding errors are not uncommon when doing numerical calculations using a computer and so the results may not agree *exactly* with their theoretical values. You should observe that, while the elements of G are not all exactly zero, they indeed are all zero to at least 15 decimal places.

S1.8.2 The SAS commands are

```
proc iml;
reset nolog;
libname keep 'c:\';
reset storage='keep.Xy';
store X y;
```

These commands should be issued during the same IML session in which the matrices X and y were created.

S1.8.3 To exit SAS, go to any *Command* line, type `bye` , and press `Enter` . Now invoke SAS and type the following in the PROGRAM EDITOR window.

```
proc iml;
reset nolog;
libname keep 'c:\';
reset storage='keep.Xy';
load X y;
```

S1.9.1 The SAS commands are

```
libname my 'b:\';
proc contents data=my.bivgauss;
proc contents data=my.bivngaus;
run;
```

S1.9.2 Use the following SAS commands.

```
option linesize=75 pagesize=35;
libname my 'b:\';
```



```
proc chart data=my.bivgauss;
vbar x2;
run;
```

The SAS response is as follows.

```
---------------------------------------------------------------------------

                          FREQUENCY OF X2

FREQUENCY

    |                             **  **
150 +                         **  **  **
    |                         **  **  **
    |                         **  **  **
    |                         **  **  **
120 +                     **  **  **  **
    |                     **  **  **  **
    |                 **  **  **  **  **  **
    |                 **  **  **  **  **  **
 90 +                 **  **  **  **  **  **
    |                 **  **  **  **  **  **
    |                 **  **  **  **  **  **
    |                 **  **  **  **  **  **  **
 60 +                 **  **  **  **  **  **  **
    |                 **  **  **  **  **  **  **
    |             **  **  **  **  **  **  **  **
    |             **  **  **  **  **  **  **  **  **
 30 +         **  **  **  **  **  **  **  **  **  **
    |         **  **  **  **  **  **  **  **  **  **  **
    |         **  **  **  **  **  **  **  **  **  **  **
    |     **  **  **  **  **  **  **  **  **  **  **  **  **
    ------------------------------------------------------------
      -24 -20 -16 -12  -8  -4   0   4   8  12  16  20  24  28  32

                            X2 MIDPOINT

---------------------------------------------------------------------------
```

S1.9.3 The SAS commands are

```
libname my 'b:\';
proc chart data=my.bivgauss;
hbar x2;
run;
```

S1.9.4 The required SAS commands are

```
libname my 'b:\';
options linesize=75 pagesize=35;
proc chart data=my.bivngaus;
vbar x1;
hbar x1;
run;
```

SAS responds as follows.

```
                         FREQUENCY OF X1

    FREQUENCY

      |                        ** ** **
      |                        ** ** **
      |                        ** ** **
  120 +                     ** ** ** **
      |                     ** ** ** **
      |                     ** ** ** ** **
      |                     ** ** ** ** **
   90 +                  ** ** ** ** ** **
      |                  ** ** ** ** ** ** **
      |                  ** ** ** ** ** ** **
      |                  ** ** ** ** ** ** **
   60 +               ** ** ** ** ** ** ** **
      |               ** ** ** ** ** ** ** **
      |               ** ** ** ** ** ** ** ** **
      |               ** ** ** ** ** ** ** ** **
   30 +               ** ** ** ** ** ** ** ** **
      |            ** ** ** ** ** ** ** ** ** ** **
      |            ** ** ** ** ** ** ** ** ** ** **
      |      ** ** ** ** ** ** ** ** ** ** ** ** ** ** **
      ----------------------------------------------------------
         -
         1  -  -  -  -  -  -                       1  1  1  1
         0  9  7  6  4  3  1  0  1  3  4  6  7  9  0  2  3  5
         .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
         5  0  5  0  5  0  5  0  5  0  5  0  5  0  5  0  5  0

                            X1 MIDPOINT
```



```
                                FREQUENCY OF X1

   X1                                            CUM              CUM
MIDPOINT                                  FREQ   FREQ  PERCENT  PERCENT
         |
  -10.5  |*                                  3      3     0.30     0.30
   -9.0  |                                   2      5     0.20     0.50
   -7.5  |*                                  4      9     0.40     0.90
   -6.0  |**                                 9     18     0.90     1.80
   -4.5  |*****                             26     44     2.60     4.40
   -3.0  |***********                       57    101     5.70    10.10
   -1.5  |******************                89    190     8.90    19.00
    0.0  |************************         118    308    11.80    30.80
    1.5  |***************************** 144    452    14.40    45.20
    3.0  |****************************     142    594    14.20    59.40
    4.5  |****************************     139    733    13.90    73.30
    6.0  |*********************            104    837    10.40    83.70
    7.5  |****************                  81    918     8.10    91.80
    9.0  |********                          42    960     4.20    96.00
   10.5  |*****                             23    983     2.30    98.30
   12.0  |**                                11    994     1.10    99.40
   13.5  |*                                  5    999     0.50    99.90
   15.0  |                                   1   1000     0.10   100.00
         --------+-------+-------+-----
                40      80     120

                    FREQUENCY
```

S1.9.5 Use the following SAS commands.

```
libname my 'b:\';
proc plot data=my.bivngaus;
plot x1*x2='+'/hpos=50 vpos=15;
run;
```

SAS responds as follows.

```
-------------------------------------------------------------------------

              Plot of X1*X2.  Symbol used is '+'.

       X1 |
       20 +
          |
          |         +        +
          |              ++++++
       10 +       +   ++++++++++
          |      ++++++++++++++
          |    + ++++++++++++++
          |     + +++ ++++++++++++++++++++ ++    +
        0 +                 ++++++++++++++++ +
          |                  ++++++++++++  +
          |                  ++++++++++    +
          |                   +  ++ + + + +
      -10 +                  ++
          |
          --+--------+--------+--------+--------+--------+---
           -40      -20       0       20       40       60

                              X2

NOTE: 874 obs hidden.
-------------------------------------------------------------------------
```

Note that we have used hpos=50 and vpos=15 in the above command. You should try other values for hpos and vpos to obtain a plot with the scale you like.

S1.9.6 The SAS commands are

```
libname my 'b:\';
proc plot data=my.bivgauss;
plot x2*x1='o'/hpos=50 vpos=15;
run;
```

The results which appear in the OUTPUT window are


```
-------------------------------------------------------------------------

              Plot of X2*X1.  Symbol used is 'o'.

   50 +
      |
      |                                       o
   X2 |            o    oo   oo          oo
      |        ooo oo   ooooooooo  o   o
      | o   oooooooooooooooooooooooo
      | o ooooooooooooooooooooooooo
    0 +     ooooooooooooooooooooooooooooo
      |  o     oooooooooooooooooooooo  o  o
      |   o oo     ooo ooooooooooo    oo
      |            oo  o   oooo   o
      |                  o
      |
      |
  -50 +
      --+--------------+--------------+--------------+--------
       -10             0             10             20

                               X1

NOTE: 844 obs hidden.
-------------------------------------------------------------------------
```

Again, you may try other values for hpos and vpos to obtain a plot having a scale you like.

S1.9.7 The SAS commands are

```
libname my 'b:\';
proc plot data=my.bivgauss;
plot x1*x2='o'/hpos=60 vpos=20;
run;
```

SAS responds as follows.

```
--------------------------------------------------------------------------

                 Plot of X1*X2.  Symbol used is 'o'.

  X1 |
  20 +
     |
     |
     |                          o
     |                       o
     |                        ooo oo     o
  10 +                  o o    o  ooo ooooo  o o  oo
     |                      o oooooo ooooo oo  o     o
     |               oo o ooooooooooooooooooooooo  o o o  o
     |              o   o oooooooooooooooooooooooo oo
     |            o  oo oooooooooooooooooooooooooooo
     |           o    ooooo ooooooooooooooooooooo ooooo
   0 +             o   o oooooooooooooooooooooo ooo oo
     |                ooooooooooooooooooooooooooooooo  o
     |              ooo  ooooo oooooooooo o ooo  o
     |                        ooooooo ooooo o o o    o
     |                   o o       o ooo o o oo o o
     |                   o              oooo     o
 -10 +                        o        o
     --+-------------+-------------+-------------+-------------+--
      -40           -20            0            20            40

                                    X2

NOTE: 720 obs hidden.
--------------------------------------------------------------------------
```

Note that we have used hpos=60 and vpos=20 here. You may experiment with other values for hpos and vpos .

S2.3.1 The appropriate SAS commands are

```
data auto;
infile 'b:\car.dat';
input id y x1 x2;
run;
```


S2.3.2 If the temporary data set auto has been created in this SAS session, then use the following SAS commands. Observe that we use the option vardef=n in the proc means command since we are working with population data.

```
proc means data=auto mean std vardef=n;
var y x1;
run;
```

If the temporary dataset auto has not been created in this SAS session, you must create it with the commands in Problem S2.3.1.

The results of the preceding command are as follows.

N Obs	Variable	Mean	Std Dev
1242	Y	526.1417069	105.9232892
	X1	19647.75	5835.83

Thus, from this output you can read the mean and standard deviation of the variables Y and X_1.

S2.3.3 You can obtain the answers by using the following SAS statements.

```
data auto;
infile 'b:\car.dat';
input id y x1 x2;
proc means data=auto min max;
var x1 x2;
run;
```

You can omit the first three SAS statements above if you have already created the temporary dataset auto during the current SAS session. The results which appear in the OUTPUT window are

```
---------------------------------------------
N Obs  Variable          Minimum       Maximum
---------------------------------------------
 1242  X1                7200.00      38300.00
       X2                1600.00      18500.00
---------------------------------------------
```

S2.3.4 Use the following commands.

```
proc print data=auto;
var y;
run;
```

We have assumed that the temporary dataset auto has been created during the current SAS session.

S2.3.5 The appropriate SAS statements are

```
proc chart data=auto;
hbar y;
run;
```

We have assumed that the temporary dataset auto has been created during the current SAS session. If not, use the commands in Problem S2.3.1 to create it first.

The results which appear in the OUTPUT window are


```
------------------------------------------------------------------------------

                              FREQUENCY OF Y

   Y                                                   CUM              CUM
MIDPOINT                                       FREQ   FREQ  PERCENT  PERCENT
         |
  360    |*****                                  38     38     3.06     3.06
  400    |****************                      128    166    10.31    13.37
  440    |****************************** 233    399    18.76    32.13
  480    |****************************      214    613    17.23    49.36
  520    |********************              155    768    12.48    61.84
  560    |******************                139    907    11.19    73.03
  600    |************                       91    998     7.33    80.35
  640    |************                       88   1086     7.09    87.44
  680    |********                           61   1147     4.91    92.35
  720    |*****                              39   1186     3.14    95.49
  760    |****                               29   1215     2.33    97.83
  800    |**                                 14   1229     1.13    98.95
  840    |*                                   9   1238     0.72    99.68
  880    |                                    3   1241     0.24    99.92
  920    |                                    1   1242     0.08   100.00
          --------+-------+-------+-------
                 60     120     180

                     FREQUENCY

------------------------------------------------------------------------------
```

S2.3.6 The SAS commands are

```
data auto;
infile 'b:\car.dat';
input id y x1 x2;
options pagesize=35 linesize=75;
proc chart data=auto;
vbar x1;
vbar x2;
run;
```

The first three statements are used to create the temporary dataset auto. These statements may be omitted if this dataset has already been created during the current SAS session.

The results which appear in the OUTPUT window are

```
--------------------------------------------------------------------------

                          FREQUENCY OF X1

FREQUENCY

    |                  **
150 +                  **  **  **  **
    |                  **  **  **  **
    |              **  **  **  **  **
    |              **  **  **  **  **
    |              **  **  **  **  **  **
100 +              **  **  **  **  **  **  **
    |          **  **  **  **  **  **  **  **
    |          **  **  **  **  **  **  **  **
    |          **  **  **  **  **  **  **  **
    |          **  **  **  **  **  **  **  **  **
 50 +      **  **  **  **  **  **  **  **  **  **
    |      **  **  **  **  **  **  **  **  **  **  **
    |      **  **  **  **  **  **  **  **  **  **  **
    |      **  **  **  **  **  **  **  **  **  **  **  **
    |  **  **  **  **  **  **  **  **  **  **  **  **  **  **  **
    ----------------------------------------------------------------
              1   1   1   1   1   2   2   2   2   2   3   3   3   3   3
          8   0   2   4   6   8   0   2   4   6   8   0   2   4   6   8
          0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
          0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
          0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0

                            X1 MIDPOINT
```



```
                          FREQUENCY OF X2

FREQUENCY

    |                              **  **
    |                              **  **  **
    |                              **  **  **
150 +                          **  **  **  **
    |                          **  **  **  **  **
    |                          **  **  **  **  **
    |                          **  **  **  **  **
    |                          **  **  **  **  **
100 +                      **  **  **  **  **  **  **
    |                      **  **  **  **  **  **  **
    |                      **  **  **  **  **  **  **
    |                      **  **  **  **  **  **  **
    |                  **  **  **  **  **  **  **  **
 50 +                  **  **  **  **  **  **  **  **  **
    |                  **  **  **  **  **  **  **  **  **
    |              **  **  **  **  **  **  **  **  **  **  **
    |          **  **  **  **  **  **  **  **  **  **  **  **
    |      **  **  **  **  **  **  **  **  **  **  **  **  **  **
    ------------------------------------------------------------
                                      1   1   1   1   1   1   1   1
          1   3   4   5   6   7   9   0   1   2   3   5   6   7   8
          8   0   2   4   6   8   0   2   4   6   8   0   2   4   6
          0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
          0   0   0   0   0   0   0   0   0   0   0   0   0   0   0

                            X2 MIDPOINT
--------------------------------------------------------------------------
```

S2.3.7 The following SAS statements may be used to obtain the answers to this problem and also Problem S2.3.8.

```
proc iml;
reset nolog;
use auto;
read all var {y} into y;
```

```
n=nrow(y);
meany=sum(y)/n;
ssy=(y-meany)`*(y-meany);
sumsqy=y`*y;
print ssy sumsqy;
```

The SAS response is

```
--------------------------------------------------------------------------------

                              SSY     SUMSQY
                         13934921 357751690

--------------------------------------------------------------------------------
```

From this we get $SSY = 13,934,921$.

S2.3.8 See the commands and the output for Problem S2.3.7. We get $\sum_{i=1}^{1242} Y_i^2 = 357,751,690$.

S2.3.9 The appropriate SAS commands are

```
proc means data=auto std vardef=n;
var x2;
run;
```

We have assumed that the temporary dataset auto has been created during the current SAS session. If not, use the commands in Problem S2.3.1 to create it.

The results which appear in the OUTPUT window are

```
--------------------------------------------------------------------------------

                     Analysis Variable : X2

                     N Obs        Std Dev
                     -------------------
                      1242        3083.15
                     -------------------

--------------------------------------------------------------------------------
```


From the preceding output we get $\sigma_{X_2} = 3083.15$.

S2.3.10 The SAS commands are

```
data auto;
infile 'b:\car.dat';
input id y x1 x2;

data newdata;
set auto;
u=x1+3*x2;
keep u;
proc means data=newdata mean std vardef=n;
run;
```

The first three statements may be omitted if you have already created the temporary dataset auto during the current SAS session. The results which appear in the OUTPUT window are

```
--------------------------------------------------------------------------------

                    Analysis Variable : U

                    N Obs          Mean        Std Dev
                    ----------------------------------
                     1242      52991.22       11074.83
                    ----------------------------------

--------------------------------------------------------------------------------
```

From this we get $\mu_U = 52,991.22$ and $\sigma_U = 11,074.83$.

S2.3.11 The appropriate SAS commands are

```
libname my 'b:\';
proc plot data=my.car;
plot price*mtcost/vpos=15 hpos=50;
run;
```

The `libname` command is not needed if you have already issued this command during the current SAS session. The results which appear in the `OUTPUT` window are

```
---------------------------------------------------------------------------

       Plot of PRICE*MTCOST.  Legend: A = 1 obs, B = 2 obs, etc.

       40000 +
             |          A A     AA  B  A
             |        A AAAA A  A  A  B  A    A
       PRICE |       BC ABCDC BCBB  BBBBB        A A
             |      AABDCFGFGCFDCBABCDBDAB B   AA
             |       BAGLJOGHFLHFIAGBEDDCACAAA AB A
             |      ADAJLPQKMKIOKGHHGCEDDADDDC AAB  A
       20000 +       CBIIOYISKGHIFGHCAHFCDCBAD C  A    A
             |      AGEGPSNNWOKILLGIJFDFBCDEDAA BA
             |       DKJISNMKLJMIMIGGGDDBADAAA BA  A A
             |       EFFCHFKMHHCDDCBGCBBAB BAB
             |       ABB CFBCBBEBA  AAB    A A    A
             |
             |
           0 +
             -+----------+----------+----------+----------+-
             200        400        600        800       1000

                                 MTCOST

---------------------------------------------------------------------------
```

S2.3.12 The appropriate SAS commands are

```
libname my 'b:\';
proc plot data=my.car;
plot mtcost*miles/vpos=15 hpos=50;
run;
```

The `libname` command is not needed if you have already issued this command during the current SAS session. The results which appear in the `OUTPUT` window are


```
---------------------------------------------------------------------------

       Plot of MTCOST*MILES.  Legend: A = 1 obs, B = 2 obs, etc.

       1000 +
            |                                         AA
            |                                       BFAB
     MTCOST |                                     BEHIFD
            |                                   ARVOJDA
            |                                HNTZZZMD
            |                            DHWZZZZVMA
        500 +                      AAGNOUZZZZZZWH
            |              AGFADHGEQNTZZZZZZZWMGA
            |           A   BCEBDEHDGBJEFC
            |
            |
            |
            |
          0 +
            -+---------+----------+----------+----------+-
             0        5000      10000      15000      20000

                                 MILES

NOTE: 183 obs hidden.

---------------------------------------------------------------------------
```

S2.3.13 Use the following SAS commands.

```
libname my 'b:\';
data newdata;
set my.car;
if carno=792;
proc print data=newdata;
run;
```

The results which appear in the `OUTPUT` window are

```
---------------------------------------------------------------------------

                 OBS    CARNO    MTCOST    PRICE    MILES

                   1     792       528     13800    11800
---------------------------------------------------------------------------
```

From this we get the first-year maintenance cost of car 792 to be $528.00. You could also get this value from Table D-1 in Appendix D.

An alternative, perhaps more convenient, way to solve this problem is by using the following SAS statements.

```
proc print data=my.car;
where carno=792;
run;
```

SAS responds as follows.

```
------------------------------------------------------------------------

                OBS    CARNO    MTCOST    PRICE    MILES

                792     792       528     13800    11800

------------------------------------------------------------------------
```

The `where` statement is used to instruct SAS to carry out the commands using only the subset of observations for which `carno = 792`; in this case, the subset consists of only one observation. You should consult the SAS reference manuals for learning about more advanced uses of the `where` statement.

S2.3.14 You may use the following SAS statements to answer parts (a)–(d).

```
libname my 'b:\';
data newdata;
set my.car;
if price=12500;
proc print data=newdata;
proc means data=newdata mean std vardef=n;
var mtcost price miles;
run;
```

The SAS response is as follows.


```
------------------------------------------------------------------------

             OBS    CARNO    MTCOST    PRICE    MILES

               1      292      484     12500    10800
               2      415      397     12500     7700
               3     1125      438     12500     8500
               4     1127      432     12500    10300

           N Obs  Variable          Mean         Std Dev
           -------------------------------------------------
               4  MTCOST     437.7500000      30.9546039
                  PRICE         12500.00               0
                  MILES          9325.00         1269.60
           -------------------------------------------------

------------------------------------------------------------------------
```

From the preceding output we see that there are four cars that sold for $12,500, and from the column labeled CARNO we see that the car numbers are 292, 415, 1125, and 1127 (you can check this result by looking up Table D-1 in Appendix D). The column labeled MTCOST lists the first-year maintenance costs associated with these cars. From the preceding output you can also obtain the mean of MTCOST as $437.75 and the standard deviation of MTCOST as $30.95.

An alternative set of SAS commands, using the `where` statement, that will also yield the above results is as follows.

```
proc print data=my.car;
where price=12500;
proc means data=my.car mean std vardef=n;
where price=12500;
var mtcost price miles;
run;
```

S2.3.15 The required SAS commands are

```
libname my 'b:\';
data newdata;
set my.car;
if price=9600;
```

```
run;
proc print data=newdata;
proc means data=newdata mean std vardef=n;
var mtcost price miles;
run;
```

The results which appear in the OUTPUT window are

```
--------------------------------------------------------------------------
              OBS    CARNO    MTCOST    PRICE    MILES

               1       141      450      9600      9600
               2       900      621      9600     13200
               3       932      773      9600     17400
               4      1045      490      9600     11000
               5      1206      650      9600     15300

             N Obs  Variable         Mean          Std Dev
             -------------------------------------------
                 5  MTCOST      596.8000000    116.1195935
                    PRICE           9600.00              0
                    MILES          13300.00        2821.35
             -------------------------------------------
--------------------------------------------------------------------------
```

From this output you can obtain the answers. An alternative set of commands, using the `where` statement, that will yield the above results is

```
proc print data=my.car;
where price=9600;
proc means data=my.car mean std vardef=n;
where price=9600;
var mtcost price miles;
run;
```

S3.4.1 The appropriate SAS commands are as follows.

```
libname my 'b:\';
```



```
proc print data=my.table323;
run;
```

The SAS response is given below.

```
--------------------------------------------------------------------------
                    OBS    SCORE    HOURS

                      1      44        0
                      2      86       10
                      3      87       10
                      4      58        3
                      5      85       10
                      6      55        1
                      7      63        4
                      8      48        0
                      9      57        3
                     10      54        2
                     11      82       10
                     12      90       12
                     13      56        3
                     14      67        5
                     15      81        8
                     16      57        4
                     17      47        1
                     18      47        1
                     19      44        0
                     20      48        0
                     21      54        3
                     22      45        0
                     23      51        1
                     24      91       12
                     25      58        3
                     26     100       12
--------------------------------------------------------------------------
```

S3.4.2 The SAS commands are

```
libname my 'b:\';
proc plot data=my.table323;
plot score*hours='*';
run;
```

S3.4.3 The SAS commands which you enter in the PROGRAM EDITOR window are

```
libname my 'b:\';
proc reg data=my.table323;
model score=hours;
run;
```

The results which appear in the OUTPUT window are

```
---------------------------------------------------------------------------

Model: MODEL1
Dependent Variable: SCORE

                         Analysis of Variance

                         Sum of          Mean
      Source      DF    Squares        Square      F Value     Prob>F

      Model        1   7519.38674   7519.38674     947.335     0.0001
      Error       24    190.49787      7.93741
      C Total     25   7709.88462

          Root MSE       2.81734     R-square       0.9753
          Dep Mean      63.65385     Adj R-sq       0.9743
          C.V.           4.42603


                         Parameter Estimates

                     Parameter       Standard    T for H0:
    Variable  DF      Estimate          Error   Parameter=0     Prob > |T|

    INTERCEP   1     45.509647     0.80795970        56.327         0.0001
    HOURS      1      3.997874     0.12989050        30.779         0.0001
---------------------------------------------------------------------------
```

From the preceding output we get $\hat{\beta}_0 = 45.510$, $\hat{\beta}_1 = 3.998$, $\hat{\mu}_Y(x) = 45.510 + 3.998x$, and $\hat{\sigma} = 2.817 =$ Root MSE.

S3.4.4 Use the following SAS commands.

```
libname my 'b:\';
proc print data=my.table324;
```



```
proc plot data=my.table324;
plot score*hours='*';
proc reg data=my.table324;
model score=hours;
run;
```

The SAS response is given below.

```
---------------------------------------------------------------------------

                      OBS     SCORE     HOURS

                        1       41         1
                        2       59         4
                        3       90        11
                        4       88        11
                        5       52         2
                        6       53         2
                        7       53         1
                        8       63         5
                        9       87        10
                       10       74         8


              Plot of SCORE*HOURS.  Symbol used is '*'.
        (NOTE: 2 obs hidden.)
   100 +
       |
       |                                                     *      *
       |
    75 +                                               *
       |
 SCORE |                              *
       |                         *
    50 +*    *
       |*
       |
       |
    25 +
       -+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-
        1     2     3     4     5     6     7     8     9    10    11

                                    HOURS
```

```
Model: MODEL1
Dependent Variable: SCORE

                    Analysis of Variance

                      Sum of         Mean
Source        DF     Squares       Square      F Value      Prob>F

Model          1  2692.71845   2692.71845      241.279      0.0001
Error          8    89.28155     11.16019
C Total        9  2782.00000

    Root MSE      3.34069     R-square      0.9679
    Dep Mean     66.00000     Adj R-sq      0.9639
    C.V.          5.06165

                    Parameter Estimates

                 Parameter      Standard    T for H0:
Variable  DF      Estimate         Error   Parameter=0    Prob > |T|

INTERCEP   1     43.038835    1.81689460        23.688        0.0001
HOURS      1      4.174757    0.26876433        15.533        0.0001
-------------------------------------------------------------------------
```

From the preceding output we get $\hat{\beta}_0 = 43.039$, $\hat{\beta}_1 = 4.175$, $\hat{\mu}_Y(x) = 43.039 + 4.175x$, and $\hat{\sigma} = 3.34 = $ Root MSE.

S3.4.5 Use the SAS commands given below.

```
libname my 'b:\';
proc print data=my.arsenic;
proc plot data=my.arsenic;
plot measured*true='*';
run;
```

S3.4.6 The appropriate SAS commands are

```
libname my 'b:\';
proc reg data=my.arsenic;
model measured=true;
run;
```


The results which appear in the OUTPUT window are

```
-------------------------------------------------------------------------

Model: MODEL1
Dependent Variable: MEASURED

                    Analysis of Variance

                      Sum of         Mean
Source        DF     Squares       Square      F Value      Prob>F

Model          1   163.89538    163.89538     4663.009      0.0001
Error         30     1.05444      0.03515
C Total       31   164.94982

    Root MSE      0.18748     R-square      0.9936
    Dep Mean      3.56156     Adj R-sq      0.9934
    C.V.          5.26392

                    Parameter Estimates

                 Parameter      Standard    T for H0:
Variable  DF      Estimate         Error   Parameter=0    Prob > |T|

INTERCEP   1      0.104583    0.06050825         1.728        0.0942
TRUE       1      0.987708    0.01446424        68.286        0.0001
-------------------------------------------------------------------------
```

From the preceding output we get $\hat{\beta}_0 = 0.1046$, $\hat{\beta}_1 = 0.9877$, $\hat{\mu}_Y(x) = 0.1046 + 0.9877x$, and $\hat{\sigma} = 0.18748 = $ Root MSE.

S3.5.1 The appropriate SAS commands are given below.

```
libname my 'b:\';
proc contents data=my.car20;
run;
proc print data=my.car20;
run;
```

The results which appear in the OUTPUT window are

```
--------------------------------------------------------------------------------
                              CONTENTS PROCEDURE

 Data Set Name:  MY.CAR20                  Type:
 Observations:   20                        Record Len: 20
 Variables:      2
 Label:

          -----Alphabetic List of Variables and Attributes-----

 #  Variable  Type  Len  Pos  Label
 2  MILES     Num     8   12
 1  MTCOST    Num     8    4

                            OBS    MTCOST    MILES

                              1      456     11200
                              2      828     17300
                              3      500     11100
                              4      489     11000
                              5      387      6700
                              6      553     13700
                              7      531     12400
                              8      650     15300
                              9      475     11300
                             10      474      8200
                             11      533     12300
                             12      396      7700
                             13      618     14300
                             14      474      8800
                             15      639     13600
                             16      457      7100
                             17      460      8700
                             18      433      6500
                             19      621     13100
                             20      460      9900
--------------------------------------------------------------------------------
```

From the preceding output we see that the file **car20.ssd** has twenty observations and two variables. The variable are named `mtcost` and `miles`, respectively.

S3.5.2 Execute the following SAS commands.

```
libname my 'b:\';
proc plot data=my.car20;
plot mtcost*miles='*';
run;
```

S3.5.3 In Problem S3.5.6 you are asked to print the values of y_i, x_i, r_i, $\hat{e}_i$, and $\hat{\mu}_Y(x_i)$, so we will compute them here and store them in the file `diagnstc` using the commands given below. We will print them later when answering Problem S3.5.6.

Execute the following SAS commands.

```
libname my 'b:\';
proc reg data=my.car20;
model mtcost=miles;
output out=diagnstc student=standres r=resd p=fitval;
run;
```

The results which appear in the `OUTPUT` window are

```
--------------------------------------------------------------------------------
Model: MODEL1
Dependent Variable: MTCOST

                        Analysis of Variance

                             Sum of         Mean
     Source          DF     Squares       Square      F Value     Prob>F

     Model            1 171857.06013 171857.06013      79.911     0.0001
     Error           18  38711.13987   2150.61888
     C Total         19 210568.20000

         Root MSE      46.37477     R-square       0.8162
         Dep Mean     521.70000     Adj R-sq       0.8059
         C.V.           8.88916
```

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
INTERCEP	1	177.006587	39.92948082	4.433	0.0003
MILES	1	0.031307	0.00350222	8.939	0.0001

S3.5.4 From the preceding output we get $\hat{\beta}_0 = 177.006587$, $\hat{\beta}_1 = 0.031307$, $\hat{\mu}_Y(x) = 177.006587 + 0.031307x$, and $\hat{\sigma} = 46.37477 =$ `Root MSE`.

S3.5.5 From Problem S3.5.4 we get

$$\hat{\mu}_Y(9400) = 177.006587 + 0.031307(9400) = 471.29239$$

S3.5.6 Recall that, in Problem S3.5.3, we saved the necessary information for this problem in the temporary SAS dataset **diagnstc**. We now print the contents of this dataset. The following commands must be issued in the same SAS session during which the dataset was created.

```
proc print data=diagnstc;
run;
```

The output is

OBS	MTCOST	MILES	FITVAL	RESD	STANDRES
1	456	11200	527.648	-71.648	-1.58529
2	828	17300	718.623	109.377	2.77121
3	500	11100	524.518	-24.518	-0.54243
4	489	11000	521.387	-32.387	-0.71652
5	387	6700	386.766	0.234	0.00550
6	553	13700	605.917	-52.917	-1.19700
7	531	12400	565.217	-34.217	-0.76144
8	650	15300	656.008	-6.008	-0.14094
9	475	11300	530.779	-55.779	-1.23435
10	474	8200	433.726	40.274	0.91290
11	533	12300	562.086	-29.086	-0.64674
12	396	7700	418.073	-22.073	-0.50523
13	618	14300	624.701	-6.701	-0.15332
14	474	8800	452.511	21.489	0.48254
15	639	13600	602.786	36.214	0.81782
16	457	7100	399.288	57.712	1.33975
17	460	8700	449.380	10.620	0.23881
18	433	6500	380.504	52.4959	1.23954
19	621	13100	587.132	33.8677	0.75930
20	460	9900	486.949	-26.9489	-0.59842

S3.5.7 The appropriate SAS commands are

```
proc plot data=diagnstc;
plot standres*miles='*';
run;
```

The results which appear in the OUTPUT window are

```
                 Plot of STANDRES*MILES.  Symbol used is '*'.
      (NOTE: 1 obs hidden.)
S  4 +
t    |
u    |                                                                 *
d    |
e  2 +
n    |        *
t    |     *        *                          *  *
i    |                 *
z  0 +      *          *                              *     *
e    |           *           *      *      *
d    |                               *     *      *
     |                               *
R -2 +
e    ---+----------+----------+----------+----------+----------+----------+
s     6000       8000       10000      12000      14000      16000      18000
i
                                       MILES
```

S3.5.8 The following SAS commands can be used for computing the standardized residuals and their nscores, and store both in a temporary dataset named data2. The commands also ask SAS to print the contents of data2.

```
libname my 'b:\';
proc reg data=my.car20;
model mtcost=miles;
output out=data1 student=stdresid;

proc rank normal=blom data=data1 out=data2;
var stdresid;
ranks nscores;

proc print data=data2;
var stdresid nscores;
run;
```

SAS responds as follows.

```
---------------------------------------------------------------------------
Model: MODEL1
Dependent Variable: MTCOST

                          Analysis of Variance

                            Sum of          Mean
      Source        DF     Squares        Square     F Value    Prob>F

      Model          1 171857.06013 171857.06013      79.911    0.0001
      Error         18  38711.13987   2150.61888
      C Total       19 210568.20000

          Root MSE     46.37477     R-square      0.8162
          Dep Mean    521.70000     Adj R-sq      0.8059
          C.V.          8.88916

                          Parameter Estimates

                     Parameter      Standard    T for H0:
     Variable  DF     Estimate         Error   Parameter=0    Prob > |T|

     INTERCEP   1   177.006587   39.92948082         4.433        0.0003
     MILES      1     0.031307    0.00350222         8.939        0.0001
```



```
                              OBS    STDRESID     NSCORES

                                1    -1.58529    -1.86824
                                2     2.77121     1.86824
                                3    -0.54243    -0.31457
                                4    -0.71652    -0.74414
                                5     0.00550     0.18676
                                6    -1.19700    -1.12814
                                7    -0.76144    -0.91914
                                8    -0.14094     0.06193
                                9    -1.23435    -1.40341
                               10     0.91290     0.91914
                               11    -0.64674    -0.58946
                               12    -0.50523    -0.18676
                               13    -0.15332    -0.06193
                               14     0.48254     0.44777
                               15     0.81782     0.74414
                               16     1.33975     1.40341
                               17     0.23881     0.31457
                               18     1.23954     1.12814
                               19     0.75930     0.58946
                               20    -0.59842    -0.44777
---------------------------------------------------------------------------
```

S3.5.9 Recall that, in Problem S3.5.8 we saved the standardized residuals and the nscores in the temporary dataset named data2. We assume that you are in the same SAS session as the one where the dataset data2 was created. Then the appropriate SAS commands are

```
proc plot data=data2;
plot stdresid*nscores='*';
run;
```

The SAS response is

```
--------------------------------------------------------------------------------

              Plot of STDRESID*NSCORES.  Symbol used is '*'.

     S  4 +
     t    |
     u    |                                                          *
     d    |
     e  2 +
     n    |                                                     *
     t    |                                          * *  *  *
     i    |                                        *
     z  0 +                                  * * **
     e    |                          * * * **
     d    |              *   *  *
          |    *
     R -2 +
     e    ---+-------------+-------------+-------------+-------------+--
     s      -2            -1             0             1             2
     i
                          RANK FOR VARIABLE STDRESID
--------------------------------------------------------------------------------
```

S3.6.1-S3.6.4 The results of the following commands can be used to answer Problems S3.6.1–S3.6.4.

```
libname my 'b:\';
proc reg data=my.arsenic;
model measured=true;
output out=diagnstc p=fits r=residual student=stdresid;

proc plot data=diagnstc;
plot measured*true='*';
plot stdresid*measured='*';
plot stdresid*true='*';
plot stdresid*fits='*';
run;
```

The results of the preceding commands are given below.


```
--------------------------------------------------------------------------------
Model: MODEL1
Dependent Variable: MEASURED

                          Analysis of Variance

                              Sum of          Mean
      Source          DF     Squares        Square      F Value        Prob>F

      Model            1   163.89538     163.89538     4663.009        0.0001
      Error           30     1.05444       0.03515
      C Total         31   164.94982

          Root MSE       0.18748     R-square       0.9936
          Dep Mean       3.56156     Adj R-sq       0.9934
          C.V.           5.26392

                          Parameter Estimates

                      Parameter      Standard     T for H0:
     Variable  DF      Estimate         Error    Parameter=0    Prob > |T|

     INTERCEP   1      0.104583    0.06050825          1.728        0.0942
     TRUE       1      0.987708    0.01446424         68.286        0.0001


              Plot of MEASURED*TRUE.  Symbol used is '*'.
          (NOTE: 20 obs hidden.)
      7.5 +                                                              *
          |                                                              *
          |                                                       *
          |
      5.0 +                                               *
          |                                               *
 MEASURED |                                       *
          |                                       *
      2.5 +                               *
          |                       *
          |              *
          |              *
      0.0 +    *
          ---+---------+---------+---------+---------+---------+---------+---------+--
             0         1         2         3         4         5         6         7

                                          TRUE
```

```
          Plot of STDRESID*MEASURED.  Symbol used is '*'.
      (NOTE: 4 obs hidden.)
S  2 +
t    |                          *            *             *
u    |    *      *             *      *            *
d    |   *             *      *                    *      *
e  0 +   *      *                                 *      *
n    |  *             *      *             *      *
t    |                             *      *
i    |        *
z -2 +              *                                  *
e    |                           *
d    |
     |
R -4 +
e   ----+-------------+-------------+-------------+-------------+--
s       0             2             4             6             8
i
                                   MEASURED
```

```
                Plot of STDRESID*TRUE.  Symbol used is '*'.
      (NOTE: 4 obs hidden.)
S  2 +
t    |                              *                   *                   *
u    |*         *                   *         *                   *
d    |*                   *         *                             *         *
e  0 +*         *                                                 *         *
n    |*                   *         *                   *         *
t    |                                        *         *
i    |          *
z -2 +                    *                                                 *
e    |                                        *
d    |
     |
R -4 +
e    -+---------+---------+---------+---------+---------+---------+---------+-
s     0         1         2         3         4         5         6         7
i
                                         TRUE
```



```
                 Plot of STDRESID*FITS.  Symbol used is '*'.
      (NOTE: 4 obs hidden.)
S  2 +
t    |                              *                   *                   *
u    |*         *                   *         *                   *
d    |*                   *         *                             *         *
e  0 +*         *                                                 *         *
n    |*                   *         *                   *         *
t    |                                        *         *
i    |          *
z -2 +                    *                                                 *
e    |                                        *
d    |
     |
R -4 +
e    -+---------+---------+---------+---------+---------+---------+---------+-
s  0.1046    1.0923    2.0800    3.0677    4.0554    5.0431    6.0308   7.0185
i
                            Predicted Value of MEASURED
```

From the preceding output we get $\hat{\beta}_0 = 0.104583$, $SE(\hat{\beta}_0) = 0.06050825$, $\hat{\beta}_1 = 0.987708$, $SE(\hat{\beta}_0) = 0.01446424$, $\hat{\sigma} = 0.18748 =$ Root MSE.

S3.6.5 We use the macro **citheta** to obtain a 90% confidence interval for β_0. On the Command line of the PROGRAM EDITOR window type

```
include 'b:\macro\citheta.mac'
```

and press the Enter key. This will bring the following statements to the screen.

```
00001 Title 'Confidence interval for theta';
00002 libname my 'b:\';proc iml; reset nolog;
00003
00004 ****** On line 00007 enter the name of the SAS data file
00005 ****** you want to use;
00006 use
00007                                my.filename
00008 ;
00009 ****** On line 00013 enter the name of the response variable
00010 ****** exactly as it appears in the data file;
00011
00012 read all var{
```

```
00013                         response variable
00014 } into yvar;
00015
00016 ****** On line 00020 enter the name of the predictor variable
00017 ****** exactly as it appears in the data file;
00018
00019 read all var{
00020                         predictor variable
00021 } into xvar;
00022
00023 ****** On line 00025 enter the desired confidence coefficient;
00024 cc=
00025                         0.95
00026 ;
00027 ****** On line 00029 enter the vector a;
00028 a={
00029                         0   1
00030
00031 };%include 'b:\macro\citheta.sas';
----------------------------------------------------------------------------
```

On lines 00007, 00013, 00020, 00025, and 00029, replace the quantities there with `my.arsenic`, `measured` , `true` , `0.90`, and `1 0` , respectively. Press the `F10` key to execute the macro. The following result appears in the `OUTPUT` window.

```
----------------------------------------------------------------------------
                    Confidence interval for theta


        The point estimate of theta is   0.1046

        For a two-sided  90%   confidence interval for theta

        the lower confidence bound is   0.0019   and

        the upper confidence bound is   0.2073
----------------------------------------------------------------------------
```

Hence $C[0.0019 \leq \beta_0 \leq 0.2073] = 0.90$.

S3.6.6 As you did in Problem S3.6.5, use the macro **citheta**. Enter the following information on the indicated lines. On lines 00007, 00013, 00020, 00025, and 00029

replace the quantities there with `my.arsenic`, `measured`, `true`, `0.90`, and `0 1` , respectively. Press the `F10` key to execute the macro commands. SAS responds as follows.

```
----------------------------------------------------------------------------

                    Confidence interval for theta


        The point estimate of theta is   0.9877

        For a two-sided  90%   confidence interval for theta

        the lower confidence bound is   0.9632       and

        the upper confidence bound is   1.0123
----------------------------------------------------------------------------
```

This gives you the required 90% confidence interval for β_1.

S3.6.7 As in Problem S3.6.5, use the macro **citheta** and enter the following on the indicated lines. On lines 00007, 00013, 00020, 00025, and 00029, replace the quantities there with `my.arsenic`, `measured`, `true`, `0.95`, and `1 3` respectively. Press the `F10` key to execute the macro commands. SAS responds as follows.

```
----------------------------------------------------------------------------

                    Confidence interval for theta


        The point estimate of theta is   3.0677

        For a two-sided  95%   confidence interval for theta

        the lower confidence bound is   2.9984   and

        the upper confidence bound is   3.1370

----------------------------------------------------------------------------
```

From this we get $\hat{\mu}_Y(3) = 3.0677$

S3.6.8 From the output for the previous problem we get $C[2.9984 \leq \mu_Y(3) \leq 3.1370] = 0.95$.

S3.6.9 We use the macro **predy** to obtain a point estimate and a 95% confidence interval for $Y(3)$. On the Command line of the PROGRAM EDITOR window type

```
include 'b:\macro\predy.mac'
```

and press the Enter key. This will bring the following SAS statements to the PROGRAM EDITOR window.

```
-------------------------------------------------------------------------

00001 Title 'Predicted values and prediction intervals';
00002 libname my 'b:\';proc iml; reset nolog;
00003
00004 ****** On line 00007 enter the name of the SAS data file
00005 ****** you want to use;
00006 use
00007                             my.filename
00008 ;
00009 ****** On line 00013 enter the name of the response variable
00010 ****** exactly as it appears in the data file;
00011
00012 read all var{
00013                             response variable
00014 } into yvar;
00015
00016 ****** On line 00020 enter the name of the predictor variable
00017 ****** exactly as it appears in the data file;
00018
00019 read all var{
00020                             predictor variable
00021 } into xvar;
00022
00023 ****** On line 00025 enter the desired confidence coefficient;
00024 cc=
00025                             0.95
00026 ;
00027 ****** On line 00029 enter the value of x;
00028 x=
```



```
00029                                 100
00030
00031 ;%include 'b:\macro\predy.sas';

-------------------------------------------------------------------------
```

Enter the following information on the indicated lines. On lines 00007, 00013, 00020, 00025, and 00029, replace the quanities there with my.arsenic, measured, true, 0.95, and 3 respectively. Press the F10 key and the macro commands will be executed. The SAS response is as follows.

```
-------------------------------------------------------------------------

                    Prediction Interval for Y(x)

      The point estimate of Y(x) for x =   3.00 is   3.0677

      For a two-sided 95.0% prediction interval for Y(x)

      the lower confidence bound is    2.6786   and

      the upper confidence bound is    3.4568

-------------------------------------------------------------------------
```

So we get $\hat{Y}(3) = 3.0677$.

S3.6.10 From the output for the previous problem we get

$$C[2.6786 \leq Y(3) \leq 3.4568] = 0.95$$

S3.7.1 To solve this problem we use the macro **test**. On the Command line of the PROGRAM EDITOR window type

```
include 'b:\macro\test.mac'
```

and the following statements will appear in the PROGRAM EDITOR window.

```
00001 Title 'Test for theta';
00002 libname my 'b:\';proc iml; reset nolog;
00003
00004 ****** On line 00007 enter the name of the SAS data file
00005 ****** you want to use;
00006 use
00007                                 my.filename
00008 ;
00009 ****** On line 00013 enter the name of the response variable
00010 ****** exactly as it appears in the data file;
00011
00012 read all var{
00013                                 response variable
00014 } into yvar;
00015
00016 ****** On line 00020 enter the name of the predictor variable
00017 ****** exactly as it appears in the data file;
00018
00019 read all var{
00020                                 predictor variable
00021 } into xvar;
00022
00023 ****** On line 00025 enter the value of q;
00024 q=
00025                                 0
00026 ;
00027 ****** On line 00029 enter the vector a;
00028 a={
00029                                 0   1
00030
00031 };%include 'b:\macro\test.sas';
```

On line 00007 replace `my.filename` with `my.crystal`, on line 00013 replace `response variable` with `weight` , on line 00020 replace `predictor variable` with `time` , on line 00025 replace 0 with 50, and on line 00029 replace `0 1` with `6 264` . After these values are entered and checked, press the `F10` key to execute the commands. SAS responds as follows.


```
                             Test for theta

For NH: theta    =     50.0000 vs AH: theta not =   50.0000, P value = 0.000

For NH: theta < or =   50.0000 vs AH: theta     >   50.0000, P value = 0.000

For NH: theta > or =   50.0000 vs AH: theta     <   50.0000, P value = 1.000
```

The P-value for this test is on the second line of the preceding output and is 0.000 (within machine accuracy) so certainly NH would be rejected.

S3.7.2

(a) To solve this problem we use the macro **test**. As in Problem S3.7.1, bring the statements in the file **test.mac** to the `PROGRAM EDITOR` window and enter the following data on the indicated lines. On line 00007 replace `my.filename` with `my.shelflif`. On line 00013 replace `response variable` with `days` . On line 00020 replace `predictor variable` with `temp` . On lines 00025 and 00029, the entries are 0 and `0 1` , respectively. After these values are entered and checked press the `F10` key to execute the macro commands. The results are as follows.

```
                             Test for theta

For NH: theta    =      0.0000 vs AH: theta not =    0.0000, P value = 0.000

For NH: theta < or =    0.0000 vs AH: theta     >    0.0000, P value = 1.000

For NH: theta > or =    0.0000 vs AH: theta     <    0.0000, P value = 0.000
```

The result we are interested in is on the first line, so the P-value is 0.000 (within machine accurcy) so it is certainly less than 0.001.

(b) For this problem use the macro test again. On line 00007 replace `my.filename` with `my.shelflif`, on line 00013 replace `response variable` with `days` , on line 00020 replace `predictor variable` with `temp` , on line 00025 replace 0 with 650, and on line 00029 replace `0 1` with `1 13` . After these values are entered and checked, press the `F10` key to execute the macro commands. The results are as follows.

```
--------------------------------------------------------------------------
                          Test for theta

For NH: theta   =     650.0000 vs AH: theta not =  650.0000, P value = 0.000

For NH: theta < or =  650.0000 vs AH: theta     >  650.0000, P value = 0.000

For NH: theta > or =  650.0000 vs AH: theta     <  650.0000, P value = 1.000
--------------------------------------------------------------------------
```

The result we are interested in is on the second line in the preceding output, so the P-value is 0.000 (within machine precision).

S3.8.1 Use the following SAS commands.

```
libname my 'b:\';
proc reg data=my.shelflif;
model days=temp;
run;
```

The results are as follows.

```
--------------------------------------------------------------------------
Model: MODEL1
Dependent Variable: DAYS
                           Analysis of Variance

                            Sum of         Mean
      Source          DF   Squares        Square       F Value     Prob>F

      Model            1  91645.47420  91645.47420     315.375     0.0001
      Error           16   4649.47024    290.59189
      C Total         17  96294.94444

          Root MSE      17.04676      R-square        0.9517
          Dep Mean     630.05556      Adj R-sq        0.9487
          C.V.           2.70560

                           Parameter Estimates

                     Parameter      Standard     T for H0:
    Variable  DF      Estimate         Error    Parameter=0     Prob > |T|

    INTERCEP   1    925.752666   17.12865578         54.047         0.0001
    TEMP       1    -13.753354    0.77445262        -17.759         0.0001
--------------------------------------------------------------------------
```


S3.8.2 The relevant SAS commands are

```
libname my 'b:\';
proc reg data=my.agebp;
model bp=age;
run;
```

The results are as follows.

```
--------------------------------------------------------------------------

Model: MODEL1
Dependent Variable: BP

                           Analysis of Variance

                            Sum of         Mean
      Source          DF   Squares        Square       F Value     Prob>F

      Model            1  9337.72938   9337.72938     1161.307     0.0001
      Error           22   176.89562      8.04071
      C Total         23  9514.62500

          Root MSE       2.83561      R-square        0.9814
          Dep Mean     139.12500      Adj R-sq        0.9806
          C.V.           2.03818

                           Parameter Estimates

                     Parameter      Standard     T for H0:
    Variable  DF      Estimate         Error    Parameter=0     Prob > |T|

    INTERCEP   1     66.808082    2.19962517         30.372         0.0001
    AGE        1      1.608532    0.04720155         34.078         0.0001

--------------------------------------------------------------------------
```

From the preceding output you can obtain the required ANOVA table.

S3.8.3 Use the following SAS commands.

```
libname my 'b:\';
proc reg data=my.grades26;
```

```
model score=hours;
run;
```

S3.11.1 Use the following SAS commands.

```
libname my 'b:\';
proc reg data=my.gravity;
model ftpersec=sec /noint;
run;
```

The results are as follows.

```
--------------------------------------------------------------------------
Model: MODEL1
NOTE: No intercept in model. R-square is redefined.
Dependent Variable: FTPERSEC

                          Analysis of Variance

                          Sum of         Mean
      Source        DF     Squares        Square      F Value     Prob>F

      Model          1 580888.02857 580888.02857   877600.619     0.0001
      Error          6      3.97143      0.66190
      U Total        7 580892.00000

          Root MSE       0.81358     R-square       1.0000
          Dep Mean     257.42857     Adj R-sq       1.0000
          C.V.           0.31604

                          Parameter Estimates

                    Parameter      Standard    T for H0:
    Variable  DF     Estimate         Error   Parameter=0   Prob > |T|

    SEC        1    32.207143    0.03437983       936.803       0.0001
--------------------------------------------------------------------------
```

From this we get $\hat{\beta}_1 = 32.207143$ and $\hat{\sigma} = 0.81358 =$ Root MSE.


```
libname my 'b:\';
proc contents data=my.table444;
proc print data=my.table444;
run;
```

SAS responds as follows.

```
--------------------------------------------------------------------------
                          CONTENTS PROCEDURE

 Data Set Name:  MY.TABLE444             Type:
 Observations:   16                      Record Len: 36
 Variables:      4
 Label:

          -----Alphabetic List of Variables and Attributes-----

 #  Variable  Type  Len  Pos  Label
 1  POPITEM   Num     8    4
 4  PRESSURE  Num     8   28
 2  STRENGTH  Num     8   12
 3  TEMP      Num     8   20

              OBS    POPITEM    STRENGTH    TEMP    PRESSURE

                1      1150       36.6       260       10
                2      1186       20.7       230       18
                3       200       36.5       290       18
                4      1305       16.4       200       16
                5       783       23.2       200       10
                6      1066       26.6       230       14
                7      1023       22.5       210       16
                8       448       17.0       200       20
                9       945       32.7       290       18
               10       508       34.4       260       10
               11       704       32.4       260       12
               12      1135       24.8       240       18
               13       107       26.8       220       12
               14       742       37.7       280       12
               15       749       26.7       260       20
               16      1585       24.6       250       20
--------------------------------------------------------------------------
```

S4.4.2 We explain the SAS/IML commands required to solve parts (a)–(d). You must issue these commands within the same SAS/IML session because the commands for each part use the results from earlier parts.

(a) The required SAS commands are

```
libname my 'b:\';
proc iml;
reset nolog;
use my.table444;
read all var{strength} into y;
read all var{temp pressure} into q;
n=nrow(q);
ones=j(n,1,1);
x=ones||q;
print y x;
```

In the above command we create the vector y and the matrix q from the data; then we create the matrix X and finally we print y and X. The results which appear in the OUTPUT window are

```
---------------------------------------------------------------------------
              Y          X
           36.6          1        260        10
           20.7          1        230        18
           36.5          1        290        18
           16.4          1        200        16
           23.2          1        200        10
           26.6          1        230        14
           22.5          1        210        16
             17          1        200        20
           32.7          1        290        18
           34.4          1        260        10
           32.4          1        260        12
           24.8          1        240        18
           26.8          1        220        12
           37.7          1        280        12
           26.7          1        260        20
           24.6          1        250        20
---------------------------------------------------------------------------
```

Next we compute X^TX, $C = (X^TX)^{-1}$, and X^Ty with the following commands.


```
xtranx=x'*x;
c=inv(x'*x);
xtrany=x'*y;
print xtranx c xtrany;
```

The results which appear in the OUTPUT window are

```
---------------------------------------------------------------------------
   XTRANX                            C                                XTRANY
       16      3880       244 5.1300776 -0.016582 -0.068616   439.6
     3880    955400     59200 -0.016582  0.000069 -9.626E-6  109372
      244     59200      3936 -0.068616 -9.626E-6 0.0046525  6530.2
---------------------------------------------------------------------------
```

(b)–(d) The SAS/IML commands to compute $\hat{\beta}$, $\hat{e}$, and $\hat{\sigma}$ are given below (we are assuming that the matrix X and the vector y have been created during the same SAS/IML session).

```
betahat=inv(x'*x)*x'*y;
ehat=y-x*betahat;
n=nrow(x);
p=ncol(x);
df=n-p;
sigmahat=sqrt(ehat'*ehat/df);
print betahat ehat sigmahat;
```

SAS responds as follows.

```
---------------------------------------------------------------------------
          BETAHAT        EHAT   SIGMAHAT
        -6.522361   1.3702079   1.487388
        0.1926927   -2.070657
        -0.834794   2.1677822
                    -2.259465
                    -0.468231
                    0.4901656
                    1.9136078
                    1.6797119
                    -1.632218
                    -0.829792
                    1.160203
```

```
                    0.1024161
                    0.9475037
                     0.285943
                    -0.181849
                    -0.354922
------------------------------------------------------------------------
```

(e) Use the following SAS commands.

```
libname 'b:\';
proc reg data=my.table444;
model strength=temp pressure;
run;
```

The output from the above commands is given below.

```
------------------------------------------------------------------------

Model: MODEL1
Dependent Variable: STRENGTH

                         Analysis of Variance

                           Sum of         Mean
      Source          DF   Squares        Square      F Value    Prob>F

      Model            2   678.56980    339.28490     153.361    0.0001
      Error           13    28.76020      2.21232
      C Total         15   707.33000

          Root MSE       1.48739     R-square       0.9593
          Dep Mean      27.47500     Adj R-sq       0.9531
          C.V.           5.41361

                         Parameter Estimates

                    Parameter      Standard     T for H0:
    Variable  DF     Estimate         Error   Parameter=0    Prob > |T|

    INTERCEP   1    -6.522361    3.36888546        -1.936        0.0749
    TEMP       1     0.192693    0.01235387        15.598        0.0001
    PRESSURE   1    -0.834794    0.10145367        -8.228        0.0001
------------------------------------------------------------------------
```


From the preceding output we see that the values for $\hat{\beta}$ and $\hat{\sigma}$ are the same as the values obtained using matrices.

S4.4.3 The relevant SAS/IML commands are given below.

```
libname my 'b:\';
proc iml;
reset nolog;
use my.table444;
read all var{strength} into y;
read all var{temp} into q1;
n=nrow(q1);
ones=j(n,1,1);
x1=ones||q1;

betahat1=inv(x1`*x1)*x1`*y;
ehat1=y-x1*betahat1;
sse1=ehat1`*ehat1;
n=nrow(x1);
p=ncol(x1);
df=n-p;
mse1=sse1/df;
sigmaht1=sqrt(mse1);
print betahat1 ehat1 sigmaht1;
```

The results which appear in the OUTPUT window are

```
------------------------------------------------------------------------

              BETAHAT1        EHAT1     SIGMAHT1
             -18.83414    5.7831034    3.5711791
             0.1909655    -4.387931
                          -0.045862
                          -2.958966
                          3.8410345
                           1.512069
                          1.2313793
                          -2.358966
                          -3.845862
                          3.5831034
                          1.5831034
```

```
-2.197586
3.6217241
3.0637931
-4.116897
-4.307241
```

From these we get $\hat{\beta}_0^{(A)} = -18.83414$, $\hat{\beta}_1^{(A)} = 0.1909655$, and $\hat{\sigma}_{Y|X_1} = 3.5711791$.

S4.4.4 The appropriate SAS commands are as follows.

```
libname my 'b:\';
proc reg data=my.table444;
model strength=temp;
run;
```

The results are as follows.

Model: MODEL1
Dependent Variable: STRENGTH

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	1	528.78352	528.78352	41.462	0.0001
Error	14	178.54648	12.75332		
C Total	15	707.33000			

Root MSE	3.57118	R-square	0.7476
Dep Mean	27.47500	Adj R-sq	0.7295
C.V.	12.99792		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
INTERCEP	1	-18.834138	7.24703332	-2.599	0.0210
TEMP	1	0.190966	0.02965703	6.439	0.0001

The values for $\hat{\beta}_0^{(A)}$, $\hat{\beta}_1^{(A)}$, and $\hat{\sigma}_{Y|X_1}$ from the above output agree (within rounding error accuracy) with the values that we obtained for them in Problem S4.4.3.

S4.5.1 SAS commands to answer (a) through (e) are as follows.

```
libname my 'b:\';
proc reg data=my.table444;
model strength=temp pressure;
output out=diagnstc p=fits r=residual student=stdresid;

proc rank normal=blom data=diagnstc out=newdata;
var stdresid;
ranks nscores;

proc print data=newdata;
run;
```

The first four statements ask SAS to perform a regression of strength on temp and pressure, compute the fitted values, the residuals, and the standardized residuals, and store these along with the original data in a temporary SAS dataset named diagnstc. The next three statements instruct SAS to compute the nscores for the standardized residuals and store them along with the rest of the variables in another temporary dataset named newdata. The final two statements ask SAS to print the information in the dataset newdata. The results are shown below.

Model: MODEL1
Dependent Variable: STRENGTH

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	2	678.56980	339.28490	153.361	0.0001
Error	13	28.76020	2.21232		
C Total	15	707.33000			

Root MSE	1.48739	R-square	0.9593
Dep Mean	27.47500	Adj R-sq	0.9531
C.V.	5.41361		

```
                    Parameter Estimates

                  Parameter      Standard    T for H0:
Variable  DF       Estimate         Error   Parameter=0    Prob > |T|

INTERCEP   1      -6.522361    3.36888546        -1.936        0.0749
TEMP       1       0.192693    0.01235387        15.598        0.0001
PRESSURE   1      -0.834794    0.10145367        -8.228        0.0001
```

```
OBS  POPITEM  STRENGTH  TEMP  PRESSURE    FITS    RESIDUAL  STDRESID   NSCORES

  1    1150     36.6     260     10     35.2298   1.37021   1.03884   0.76184
  2    1186     20.7     230     18     22.7707  -2.07066  -1.47494  -1.28155
  3     200     36.5     290     18     34.3322   2.16778   1.68383   1.76883
  4    1305     16.4     200     16     18.6595  -2.25947  -1.68822  -1.76883
  5     783     23.2     200     10     23.6682  -0.46823  -0.37926  -0.39573
  6    1066     26.6     230     14     26.1098   0.49017   0.34362   0.39573
  7    1023     22.5     210     16     20.5864   1.91361   1.38608   1.28155
  8     448     17.0     200     20     15.3203   1.67971   1.34590   0.98815
  9     945     32.7     290     18     34.3322  -1.63222  -1.26783  -0.98815
 10     508     34.4     260     10     35.2298  -0.82979  -0.62912  -0.56918
 11     704     32.4     260     12     33.5602  -1.16020  -0.83814  -0.76184
 12    1135     24.8     240     18     24.6976   0.10242   0.07251   0.07720
 13     107     26.8     220     12     25.8525   0.94750   0.68899   0.56918
 14     742     37.7     280     12     37.4141   0.28594   0.21643   0.23349
 15     749     26.7     260     20     26.8818  -0.18185  -0.13559  -0.07720
 16    1585     24.6     250     20     24.9549  -0.35492  -0.26203  -0.23349
--------------------------------------------------------------------------------
```

(f) Use the following SAS commands to obtain the required plots.

```
proc plot data=newdata;
plot stdresid*fits='*';
plot stdresid*strength='*';
plot stdresid*nscores='*';
plot stdresid*temp='*';
plot stdresid*pressure='*';
run;
```


S4.5.2 The following SAS commands will perform the necessary computations and generate the required plots to answer parts (a) through (f).

```
libname my 'b:\';
proc reg data=my.gpa;
model gpa=satmath satverb hsmath hsengl;
output out=diagnstc p=fits r=residual student=stdresid;

proc rank normal=blom data=diagnstc out=newdata;
var stdresid;
ranks nscores;

proc print data=newdata;

proc plot data=newdata;
plot stdresid*fits='*';
plot stdresid*gpa='*';
plot stdresid*nscores='*';
plot stdresid*satmath='*';
plot stdresid*satverb='*';
plot stdresid*hsmath='*';
plot stdresid*hsengl='*';
run;
```

S4.5.3 Use the following SAS commands to generate several linear combinations of the variables Y, X_1, X_2, X_3, and X_4, and obtain rankit plots for them.

```
libname my 'b:\';
data lincomb;
set my.gpa;
u1=200*gpa+satmath-satverb-200*hsmath;
u2=200*gpa+satmath+satverb+200*hsmath+200*hsengl;
u3=200*gpa-satmath-satverb-200*hsmath+200*hsengl;
keep u1 u2 u3;

proc rank normal=blom data=lincomb out=newdata;
var u1 u2 u3;
ranks nscoreu1 nscoreu2 nscoreu3;

options linesize=75 pagesize=20;
```

```
proc plot data=newdata;
plot u1*nscoreu1;
plot u2*nscoreu2;
plot u3*nscoreu3;
run;
```

SAS response to the above commands is given below.

```
--------------------------------------------------------------------------------

       Plot of U1*NSCOREU1.  Legend: A = 1 obs, B = 2 obs, etc.

    U1 |
   500 +
       |                                                   A     A
       |                                               A
       |
       |                                       A A  A
     0 +                              A A AA A
       |                         A AA
       |              A  A  A A
       |          A
       |    A
  -500 +
       ---+-------------+-------------+--------------+-------------+--
         -2            -1             0              1             2

                            RANK FOR VARIABLE U1

       Plot of U2*NSCOREU2.  Legend: A = 1 obs, B = 2 obs, etc.

    U2 |
  4000 +
       |                                                         A
       |                                       A A  A  A   A
  3000 +                                  AA A
       |                            A A A
       |                    A A A A
  2000 +          A   A  A
       |    A
       |
  1000 +
       |
       ---+-------------+-------------+--------------+-------------+--
         -2            -1             0              1             2

                            RANK FOR VARIABLE U2
```



```
       Plot of U3*NSCOREU3.  Legend: A = 1 obs, B = 2 obs, etc.

    U3 |
     0 +                                                         A
       |                                               A   A
       |                                         A  A
  -250 +                                     A A
       |
       |
  -500 +                                A  B
       |                    A A A AA A
       |          A   A  A
  -750 +    A
       |
       ---+-------------+-------------+--------------+-------------+--
         -2            -1             0              1             2

                            RANK FOR VARIABLE U3


             Plot of U3*NSCOREU3.  Symbol used is '*'.
        (NOTE: 1 obs hidden.)
   500 +
       |
       |
       |
     0 +                                                         *
       |                                         *  *  *   *
    U3 |                                     * *
       |
  -500 +                                *  *
       |          *   *  *  * * * ** *
       |    *
       |
 -1000 +
       ---+-------------+-------------+--------------+-------------+--
         -2            -1             0              1             2

                            RANK FOR VARIABLE U3
--------------------------------------------------------------------------------
```

S4.6.1 The SAS commands for obtaining the results in Exhibit 4.6.2 in the textbook

```
libname my 'b:\';
proc reg data=my.electric;
model bill=income persons area/i;
run;
```

Note the use of the /i option in the model statement, which asks SAS to print the matrix $(X^T X)^{-1}$ as part of the output. The results which appear in the OUTPUT window are

```
---------------------------------------------------------------------
Model: MODEL1
              X'X Inverse, Parameter Estimates, and SSE

                   INTERCEP           INCOME           PERSONS

    INTERCEP     2.153683547     -0.001377673      -0.25570104
    INCOME      -0.001377673     1.0099689E-6      0.000175464
    PERSONS      -0.25570104      0.000175464      0.046002015
    AREA        0.0021517892     -1.655901E-6      -0.00029998
    BILL        -358.4415686      0.075136905     55.087632718

              X'X Inverse, Parameter Estimates, and SSE

                              AREA             BILL

         INTERCEP     0.0021517892     -358.4415686
         INCOME       -1.655901E-6      0.075136905
         PERSONS       -0.00029998     55.087632718
         AREA         2.7878446E-6     0.2811036938
         BILL         0.2811036938     550163.42009

Dependent Variable: BILL

                      Analysis of Variance

                         Sum of         Mean
    Source       DF     Squares       Square      F Value       Prob>F

    Model         3 3151504.8152 1050501.6051      57.283       0.0001
    Error        30 550163.42009  18338.78067
    C Total      33 3701668.2353

       Root MSE     135.42075     R-square       0.8514
       Dep Mean     619.41176     Adj R-sq       0.8365
       C.V.          21.86280
```



```
                           Parameter Estimates

                     Parameter       Standard     T for H0:
    Variable  DF      Estimate          Error   Parameter=0    Prob > |T|

    INTERCEP   1   -358.441569   198.73583019        -1.804        0.0813
    INCOME     1      0.075137     0.13609408         0.552        0.5850
    PERSONS    1     55.087633    29.04515215         1.897        0.0675
    AREA       1      0.281104     0.22610987         1.243        0.2234

---------------------------------------------------------------------
```

Compare this output with Exhibit 4.6.2 in the textbook.

To work Problem 4.6.6 in the textbook use the macro **cilinear**. On the Command line of the PROGRAM EDITOR window type

```
include 'b:\macro\cilinear.mac'
```

and press Enter , and the following SAS statements will appear on the screen.

```
---------------------------------------------------------------------

00001 Title 'Confidence interval for theta';
00002 libname my 'b:\';proc iml; reset nolog;
00003
00004 ****** On line  00007 enter the name of the SAS data file
00005 ****** that contains the data you want to use;
00006 use
00007                   my.filename
00008 ;
00009
00010 ****** On line 00013 enter the name of the response variable
00011 ****** exactly as it appears in the data file;
00012 read all var {
00013                   response variable
00014 } into yvar;
00015
00016 ****** Use lines 00022 to 00024 to enter the names of the predictor
00017 ****** variables exactly as they appear in the data file. You can
00018 ****** enter as many variable names on a line as will fit.
00019 ****** Leave at least one space between variable names.
00020 ****** Do not use any punctuation marks;
00021 read all var {
00022                   predictor1  predictor2  predictor3
```

```
00023                   predictor4  ... etc.
00024
00025 } into xvar;
00026
00027 ****** On line  00029 enter the confidence coefficient;
00028 cc=
00029                   0.95
00030 ;
00031 ****** On line 00038 enter the vector a. The first element of the
00032 ****** vector  a  must correspond to the intercept (which is
00033 ****** assumed to be present in the model). The order of the
00034 ****** remaining coefficients in the vector  a  must correspond
00035 ****** to the order in which you entered the names of the predictor
00036 ****** variables on lines 00022--00024;
00037 a={
00038                  0  0  0  1  0
00039
00040 };%include 'b:\macro\cilinear.sas';
```

For this problem we want a point estimate and a 95% upper confidence bound for β_2. We compute a 90% two-sided confidence interval for β_2 and use the upper bound. Thus, on line 00007 replace `my.filename` with `my.electric`. On line 00013 replace the words `response variable` with `bill`. On lines 00022–00024, replace the words that appear there with the names of the predictor variables for this problem. One way to enter these names is given below.

```
00022                    income  persons  area
00023
00024
```

On line 00029 replace `0.95` with `0.90`, and on line 00038 replace the values there with `0 0 1 0`. Press the `F10` key to execute the macro commands. The following results appear in the `OUTPUT` window.


```
------------------------------------------------------------------------------
                     Confidence interval for theta


       The point estimate of theta is       55.0876

       The standard error of this estimate is      29.0452


       For a two-sided  90% confidence interval for theta

       the lower confidence bound is      5.7904   and

       the upper confidence bound is    104.3848

------------------------------------------------------------------------------
```

From this we get $\hat{\beta}_2 = 55.0876$ and the confidence statement is

$$C[\beta_2 \leq \$104.3848] = 0.95$$

To work Problem 4.6.7 in the textbook we need a point estimate and a 90% two-sided confidence interval for $1000\beta_1$. We use the macro **cilinear**. On line 00007 replace `my.filename` with `my.electric`, on line 00013 replace the words `response variable` with `bill`, on lines 00022–00024 replace the words there with the names `income persons area` with no punctuation marks anywhere, on line 00029 replace `0.95` with `0.90`, and on line 00038 replace the values there with `0 1000 0 0`. Press `F10` to execute the macro statements. The following result appears in the `OUTPUT` window.

```
------------------------------------------------------------------------------
                     Confidence interval for theta


       The point estimate of theta is       75.1369

       The standard error of this estimate is     136.0941


       For a two-sided  90% confidence interval for theta

       the lower confidence bound is   -155.8503   and

       the upper confidence bound is    306.1241

------------------------------------------------------------------------------
```

Thus we get $1000\hat{\beta}_1 = \$75.14$. The confidence statement is

$$C[1000\beta_1 \leq \$306.12] = 0.95.$$

For Problem 4.6.8 in the textbook we need a 90% two-sided confidence interval for $500\beta_3$. Again use the macro **cilinear**. Lines 00007, 00013, 00022–00024, 00029, and 00038 should contain the following information.

```
00007                    my.electric
00013                    bill
00022                    income persons area
00023
00024
00029                    0.90
00038                    0  0  0  500
```

On executing the macro commands the following result is obtained.

```
-----------------------------------------------------------------------------
                      Confidence interval for theta

       The point estimate of theta is     140.5518

       The standard error of this estimate is     113.0549

       For a two-sided  90% confidence interval for theta

       the lower confidence bound is    -51.3319   and

       the upper confidence bound is    332.4356
-----------------------------------------------------------------------------
```

From this we get $500\hat{\beta}_3 = \$140.55$ and the confidence statement is

$$C[500\beta_3 \leq \$332.44] = 0.95$$

S4.7.1 To work this problem we use the macro **testmult**. Bring the SAS statements contained in the file **testmult.mac** to the PROGRAM EDITOR window and enter the following information on the indicated lines. On line 00007 replace my.filename with

my.gpa. On line 00013 replace the words response variable with gpa. One way to enter the names of the predictor variables on lines 00022–00024 is shown below.

```
00022                    satmath  satverb
00023                    hsmath   hsengl
00024
```

For part (a), replace 0 on line 00029 with 0.003, and on line 00038 replace the values there with 0 1 0 0 0.

For part (b), replace 0 on line 00029 with 0.001, and on line 00038 replace the values there with 0 0 1 0 0.

For part (c), replace 0 on line 00029 with 2.5, and on line 00038 replace the values there with 1 500 613 3.10 2.90.

S4.7.2 The hypothesis of interest is $NH : \beta_1 = 0$ against $AH : \beta_1 \neq 0$. Use the macro testmult with $a_0 = 0$, $a_1 = 1$, $a_2 = 0$, $a_3 = 0$, and $q = 0$. The filename on line 00008 should be my.electric. The response variable name on line 00013 is bill. The predictor variable names for lines 00022–00024 are

income   persons   area

Since $q = 0$, leave the value 0 on line 00029 unchanged, and on line 00038 replace the values there with 0 1 0 0. Press F10 to execute the macro commands. The results which appear in the OUTPUT window are given below.

```
-----------------------------------------------------------------------------
                              Test for theta

For NH: theta      =   0.000 vs AH: theta not =   0.000, P value = 0.5850

For NH: theta < or =   0.000 vs AH: theta     >   0.000, P value = 0.2925

For NH: theta > or =   0.000 vs AH: theta     <   0.000, P value = 0.7075
-----------------------------------------------------------------------------
```

The appropriate P-value for this problem is given on the first line of the output. Thus the P-value is 0.585, so NH is not rejected.

S4.8.1 You can obtain an ANOVA table using the `proc reg` command. The data are in the file **electric.ssd**.

S4.8.2 You can obtain an ANOVA table using the `proc reg` command. The data are in the file **grocery.ssd**.

S4.8.3 You can obtain an ANOVA table using the `proc reg` command. The data are in the file **age18.ssd**.

S4.9.1 We need an 80% two-sided confidence interval for σ_A and for σ_B. Bring the SAS statements in the file **ratiosgm.mac** to the PROGRAM EDITOR window. They are reproduced below.

```
--------------------------------------------------------------------------------
00001 Title 'Confidence intervals for sigma(A), sigma(B), sigma(B)/sigma(A)';
00002 proc iml;
00003
00004 ****** On line 00007 enter the confidence
00005 ****** coefficient for sigma(A);
00006 ca=
00007                     0.95
00008 ;
00009 ****** On line 00012 enter the confidence
00010 ****** coefficient for sigma(B);
00011 cb=
00012                     0.95
00013 ;
00014 ****** On line 00016 enter the estimate of sigma(A);
00015 sa=
00016                     10.00
00017 ;
00018 ****** On line 00020 enter the degrees of freedom for sigma(A);
00019 dfa=
00020                     15
00021 ;
00022 ****** On line 00024 enter the estimate of sigma(B);
00023 sb=
00024                     30.00
00025 ;
00026 ****** On line 00028 enter the degrees of freedom for sigma(B);
00027 dfb=
00028                     25
00029
00030 ;%include 'b:\macro\ratiosgm.sas';
--------------------------------------------------------------------------------
```


To use the macro for this problem, enter the following information on the indicated lines to replace the quantities there. On line 00007, and also on line 00012, enter 0.80. On line 00016 enter the value of $\hat{\sigma}_A$, which is 1.00366. On line 00020 enter the degrees of freedom for $\hat{\sigma}_A$, which equals 12. These can be obtained from the output of the `proc reg` command regressing Y on X_1, X_2, X_3, X_4, X_5, X_6, X_7. On lines 00024 and 00028, enter the value of $\hat{\sigma}_B$, which is 0.93045, and the corresponding degrees of freedom, which equals 16, respectively. These can be obtained from the output of the `proc reg` command regressing Y on X_1, X_2, X_3. These estimates and degrees of freedom can also be obtained from Exhibit 4.9.2 in the textbook. Press F10 to execute the macro statements. The following results appear in the OUTPUT window.

```
--------------------------------------------------------------------------------
      Confidence intervals for sigma(A), sigma(B), sigma(B)/sigma(A)


       For a two-sided  80.0% confidence interval for sigma(A)

       the lower confidence bound is    0.8073 and

       the upper confidence bound is    1.3848


       For a two-sided  80.0% confidence interval for sigma(B)

       the lower confidence bound is    0.7671 and

       the upper confidence bound is    1.2196


       For a two-sided confidence interval for sigma(B)/sigma(A)
       with confidence coefficient greater than or equal to  60%

       the lower confidence bound is    0.5539 and

       the upper confidence bound is    1.5108
--------------------------------------------------------------------------------
```

From the preceding output we get

$$C[0.8073 < \sigma_A < 1.3848] = 0.80$$

$$C[0.7671 \leq \sigma_B \leq 1.2196] = 0.80$$

S4.9.2 Since we want the confidence coefficient to be greater than or equal 95%, this means $1 - \alpha_A - \alpha_B = 0.95$. If we want $\alpha_A = \alpha_B$, then we must use $\alpha_A = \alpha_B = 0.025$, and hence $1 - \alpha_A = 0.975 = 1 - \alpha_B$. Use the macro **ratiosgm** with the same entries as in Problem S4.9.1, except that, on lines 00007 and 00012 enter 0.975. The result of executing the macro is given below.

```
---------------------------------------------------------------------

      Confidence intervals for sigma(A), sigma(B), sigma(B)/sigma(A)


       For a two-sided  97.5% confidence interval for sigma(A)

       the lower confidence bound is    0.6881 and

       the upper confidence bound is    1.7948


       For a two-sided  97.5% confidence interval for sigma(B)

       the lower confidence bound is    0.6658 and

       the upper confidence bound is    1.5126


       For a two-sided confidence interval for sigma(B)/sigma(A)
       with confidence coefficient greater than or equal to  95%

       the lower confidence bound is    0.3709 and

       the upper confidence bound is    2.1983
---------------------------------------------------------------------
```

From this we get $C[0.3709 \leq \sigma_B/\sigma_A \leq 2.1983] \geq 0.95$.

S4.11.1 Bring the following SAS statements contained in the file **lackfit.mac** to the PROGRAM EDITOR window.


```
---------------------------------------------------------------------------
00001 Title 'Lack-of-fit Analyses';
00002 libname my 'b:\';data rawdata(keep= yvar xvar);
00003
00004 ****** On line 00007 enter the name of the SAS data file
00005 ****** that contains the data you want to use;
00006 set
00007                               my.filename
00008 ;
00009 ****** On line 00012 enter the name of the response variable, and
00010 ****** on line 00014 enter the name of the predictor variable;
00011 rename
00012                          response variable
00013 = yvar
00014                          predictor variable
00015
00016 = xvar;proc iml;
00017
00018 ****** On line 00020 enter the confidence coefficient;
00019 cc=
00020                               0.95
00021
00022 ;%include 'b:\macro\lackfit.sas';
---------------------------------------------------------------------------
```

On line 00007 replace `my.filename` with `my.car17` , on line 00012 replace the words `response variable` with `y` , and on line 00014 replace the words `predictor variable` with `x1` . The quantity 0.95 on line 00020 is the one we want to use, so there is no need to change this. Press F10 to execute the macro statements. The following results appear in the OUTPUT window.

```
---------------------------------------------------------------------------

                          Lack-of-fit Analyses

          The estimate of beta(0) is  202.2887
          The estimate of beta(1) is    0.0149


          The estimate of sigma (pure error) is   15.0180
```

```
The estimate of the theta(1) is   21.9485
The estimate of the theta(2) is   23.1065
The estimate of the theta(3) is   -6.7354
The estimate of the theta(4) is   -9.5773
The estimate of the theta(5) is  -20.1649
The estimate of the theta(6) is  -27.8488
The estimate of the theta(7) is   -8.3660
The estimate of the theta(8) is   -1.0498
The estimate of the theta(9) is   28.6873


The standard error of the estimate of theta(1) is    9.5429
The standard error of the estimate of theta(2) is   12.4956
The standard error of the estimate of theta(3) is    9.9540
The standard error of the estimate of theta(4) is   13.5376
The standard error of the estimate of theta(5) is    8.6157
The standard error of the estimate of theta(6) is    8.6284
The standard error of the estimate of theta(7) is   10.0113
The standard error of the estimate of theta(8) is    9.6460
The standard error of the estimate of theta(9) is    9.9528


The confidence interval for theta(1) is  -13.9195  to   57.8164
The confidence interval for theta(2) is  -23.8594  to   70.0725
The confidence interval for theta(3) is  -44.1482  to   30.6774
The confidence interval for theta(4) is  -60.4595  to   41.3049
The confidence interval for theta(5) is  -52.5479  to   12.2180
The confidence interval for theta(6) is  -60.2793  to    4.5817
The confidence interval for theta(7) is  -45.9943  to   29.2624
The confidence interval for theta(8) is  -37.3052  to   35.2055
The confidence interval for theta(9) is   -8.7211  to   66.0956

The sum of squares for lackfit is 5647.6178 with df=   7

The sum of squares for pure error is 1804.3333 with df=   8

The computed F value for the lack-of-fit test is    3.5772

The P-value for the lack-of-fit test is  0.047
------------------------------------------------------------------------------
```

These results are the same as in the textbook (within rounding errors). From this output you can obtain the required answers.

S5.2.1 First we create a temporary dataset named `modified` using the `COMMAND TO CHANGE A VALUE IN A DATA SET`, given as part of Problem S5.2.1. This dataset is used (during the same SAS session in which it is created) to compute the diagnostic statistics in Exhibit 5.2.2 and store them in the file `diagnstc`. The required SAS statements are as follows.

```
proc reg data=modified;
model premium=age price;
output out=diagnstc p=fits r=residual student=stdresid rstudent=tresid;
proc print data=diagnstc;
run;
```

The result is in Exhibit 5.2.2.

S5.4.1 The appropriate SAS commands for this problem are given below.

```
libname my 'b:\';
proc reg data=my.gpa;
model  gpa=satmath satverb hsmath hsengl;
output out=diagnstc  p=fits  r=residual  student=stdresid
       rstudent=tresid  cookd=cooksd  dffits=dffits  h=hatvals;
proc print data=diagnstc;
var fits residual stdresid tresid cooksd dffits hatvals;
run;
```

The output from this command is

```
------------------------------------------------------------------------------

Model: MODEL1
Dependent Variable: GPA

                         Analysis of Variance

                             Sum of         Mean
   Source          DF       Squares       Square       F Value       Prob>F

   Model            4       6.26432      1.56608        21.721       0.0001
   Error           15       1.08150      0.07210
   C Total         19       7.34582

       Root MSE       0.26851     R-square        0.8528
       Dep Mean       2.59300     Adj R-sq        0.8135
       C.V.          10.35535
```

```
                         Parameter Estimates

                 Parameter      Standard    T for H0:
Variable  DF      Estimate         Error   Parameter=0     Prob > |T|

INTERCEP   1      0.161550    0.43753205         0.369         0.7171
SATMATH    1      0.002010    0.00058444         3.439         0.0036
SATVERB    1      0.001252    0.00055152         2.270         0.0383
HSMATH     1      0.189440    0.09186804         2.062         0.0570
HSENGL     1      0.087564    0.17649628         0.496         0.6270

OBS     FITS   RESIDUAL   STDRESID     TRESID     COOKSD     DFFITS   HATVALS

  1  1.78211    0.18789    0.79656    0.78636   0.03755    0.42775   0.22833
  2  3.18328   -0.44328   -1.90390   -2.11217   0.23926   -1.21341   0.24814
  3  2.39453   -0.20453   -0.85952   -0.85161   0.04038   -0.44520   0.21463
  4  2.40309    0.19691    0.87079    0.86336   0.06218    0.55284   0.29079
  5  3.09807   -0.11807   -0.46553   -0.45303   0.00524   -0.15744   0.10776
  6  1.53397    0.11603    0.51367    0.50068   0.02180    0.32178   0.29231
  7  1.84287    0.04713    0.19980    0.19328   0.00236    0.10506   0.22808
  8  2.37485    0.00515    0.02265    0.02188   0.00004    0.01378   0.28392
  9  2.32710    0.33290    1.43352    1.49079   0.13847    0.86533   0.25201
 10  1.96000   -0.00000   -0.00002   -0.00002   0.00000   -0.00001   0.37231
 11  3.24100   -0.10100   -0.41271   -0.40100   0.00694   -0.18101   0.16928
 12  2.38476   -0.42476   -1.68509   -1.80806   0.07651   -0.66365   0.11873
 13  2.31968   -0.11968   -0.57169   -0.55842   0.04218   -0.44856   0.39218
 14  3.36100    0.53900    2.25104    2.67246   0.26102    1.35628   0.20481
 15  2.18478   -0.16478   -0.69187   -0.67934   0.02595   -0.35367   0.21324
 16  3.33018    0.27982    1.21546    1.23673   0.10649    0.74246   0.26493
 17  3.04136    0.02864    0.15488    0.14975   0.00532    0.15768   0.52579
 18  2.78446   -0.15446   -0.62949   -0.61634   0.015652  -0.27390   0.16493
 19  3.07261    0.03739    0.16446    0.15903   0.002136   0.09993   0.28306
 20  3.24028   -0.04028   -0.16221   -0.15684   0.000891  -0.06453   0.14476
-------------------------------------------------------------------------------
```

From this output we get

(a) $h_{4,4} = 0.29079$ (b) $DFFITS_2 = -1.21341$ (c) $c_9 = 0.13847$
(d) $r_6 = 0.51367$ (e) $\hat{e}_2 = -0.44328$ (f) $T_7 = 0.19328$

S6.2.1 To obtain the answers for this problem we use the macro **pred**.

(a) Invoke SAS and on the Command line of the PROGRAM EDITOR window type


```
include 'b:\macro\pred.mac'
```

to bring the SAS statements in the file **pred.mac** to the screen. Then enter the following information on the indicated lines, replacing the quantities already there if necessary.

```
00007          my.usedcars
00013          mtcost
00022          miles  age  odometer
00023
00024
00040          1  10.0  20.0  30.2
00041
00042
00043
00044
00045
00046
00050          0.90
```

Note that, since we are only interested in a point estimate, it does not matter what value (between 0 and 1) we use for the confidence coefficient. However, we have used the value 0.90 because this is needed to answer Problem S6.2.2. On pressing the F10 key the following result will appear in the OUTPUT window.

```
-----------------------------------------------------------------------

        Predicted value and prediction interval for YA


     The estimate of YA is          YAhat =     149.4864
     The value of SE(YAhat) is                   55.4201

         A  90% prediction interval for YA is
               56.0506 to     242.9223

-----------------------------------------------------------------------
```

Thus the required answer is $\hat{Y}_A = \$149.49$.

(b) Use the macro **pred** and input the same quantities as in part (a) except on line 00040 , where you must enter

```
1  8.5  15.0  15.0
```

The result is as follows.

```
----------------------------------------------------------------------
          Predicted value and prediction interval for YA


     The estimate of YA is          YAhat =        47.8063
     The value of SE(YAhat) is                     56.3996

         A  90% prediction interval for YA is
              -47.2809 to      142.8934
----------------------------------------------------------------------
```

Thus the answer is $\hat{Y}_A = \$47.81$.

(c) Use the same macro and input the same information as in part (a) except on line 00040 , where you must enter

```
1  6.5  24.0  28.0
```

The result in the OUTPUT window is

```
----------------------------------------------------------------------
          Predicted value and prediction interval for YA


     The estimate of YA is          YAhat =        56.0990
     The value of SE(YAhat) is                     55.3364

         A  90% prediction interval for YA is
              -37.1957 to      149.3936
----------------------------------------------------------------------
```

Thus the answer is $\hat{Y}_A = \$56.10$.

S6.2.2 The answer is obtained from the output from part (a) of Problem S6.2.1, and is

$C[\$56.05 \le Y_A \le \$242.92] = 0.90$

S6.2.3 Use the macro **pred** and enter the following information on the specified lines.

```
00007                  my.usedcars
00013                  mtcost
00022                  miles  age  odometer
00023
00024
00040                  1  10.0  20.0  30.2,
00041                  1   8.5  15.0  15.0,
00042                  1   6.5  24.0  28.0
00043
00044
00045
00046
00050                  0.90
```

Execute the macro commands and the following results appear in the OUTPUT window.

```
----------------------------------------------------------------------
          Predicted value and prediction interval for YA


     The estimate of YA is          YAhat =        84.4639
     The value of SE(YAhat) is                     37.5508

         A  90% prediction interval for YA is
               21.1550 to      147.7728
----------------------------------------------------------------------
```

From this we get $C[\$21.16 \le Y_A \le \$147.77] = 0.90$. Since $h = 3$ multiply the bounds by three to obtain

$$C[\$63.47 \le Y_S \le \$443.32] = 0.90$$

S6.3.1 Use the macro **toleranc** to solve this problem. Invoke SAS and on the Command line of the PROGRAM EDITOR window type

```
include 'b:\macro\toleranc.mac'
```

to bring the SAS statements in the file **toleranc.mac** to the screen. Enter the following information on the indicated lines.

```
00007                  my.table631
00013                  y
00022                  x
00023
00024
00029                  0.20
00033                  0.95
00043                  1   3.0
```

Execute the macro commands. The results are as follows.

```
------------------------------------------------------------------------

        Estimates and Confidence Intervals for Tolerance Points


        The estimate of lambda, the number such that  20% of the
        subpopulation Y values are below it, is      -1.1134

        A 95% confidence interval for lambda is
                 -1.9165 to      -0.5688
------------------------------------------------------------------------
```

Thus we have $\hat{\lambda}_{0.20}(3.0) = -1.1134$ and the confidence statement is

$$C[-1.9165 \le \lambda_{0.20}(3.0) \le -0.5688] = 0.95$$

S6.3.2
(a) Bring the SAS statements in the file **toleranc.mac** to the PROGRAM EDITOR window as usual. Enter the following information on the indicated lines, replacing the quantities present there if necessary.

```
00007                  my.bpweight
00013                  bp
00022                  weight
00023
00024
00029                  0.99
00033                  0.95
00043                  1   210
```


Press the F10 key to execute the macro commands. SAS responds as follows.

```
------------------------------------------------------------------------

        Estimates and Confidence Intervals for Tolerance Points


        The estimate of lambda, the number such that  99% of the
        subpopulation Y values are below it, is     149.4056

        A 95% confidence interval for lambda is
                 145.2061 to     156.9342

------------------------------------------------------------------------
```

Thus we have $\hat{\lambda}_{0.99}(210) = 149.4$, and $C[145.2 \le \lambda_{0.99}(210) \le 156.9] = 0.95$.

(b) Use the macro **toleranc** and input the following information on the indicated lines.

```
00007                  my.bpweight
00013                  bp
00022                  weight
00023
00024
00029                  0.95
00033                  0.80
00043                  1   240
```

Press the F10 key to execute the macro commands. SAS responds as follows.

```
------------------------------------------------------------------------

        Estimates and Confidence Intervals for Tolerance Points


        The estimate of lambda, the number such that  95% of the
        subpopulation Y values are below it, is     157.9589

        A 80% confidence interval for lambda is
                 154.9297 to     162.1810

------------------------------------------------------------------------
```

Thus we get $C[154.9 \leq \lambda_{0.95}(240) \leq 162.2] = 0.80$. Based on this result one might conclude that a blood pressure value of 210 units *is indeed* in the upper 5% of the subpopulation of blood pressures of individuals who weigh 240 pounds.

(c) Use the macro **toleranc** and input the following information on the indicated lines.

```
00007          my.bpweight
00013          bp
00022          weight
00023
00024
00029          0.99
00033          0.95
00043          1   160
```

Press the F10 key to execute the macro commands. SAS responds as follows.

```
------------------------------------------------------------------------
     Estimates and Confidence Intervals for Tolerance Points


      The estimate of lambda, the number such that  99% of the
      subpopulation Y values are below it, is     128.4599

      A 95% confidence interval for lambda is
               124.1385 to     136.1227
------------------------------------------------------------------------
```

Thus

$$C[124.1 \leq \lambda_{0.99}(160) \leq 136.1] = 0.95$$

S6.4.1

(a) For Problems 6.4.1 and 6.4.2 in the textbook we need a point estimate and a 99% confidence interval for x_0, the value of the dial setting if the *average* temperature of the reaction chamber is to be 400°F. This is a regulation problem since we are interested in *average* temperature. Hence we use the macro **regul**. Invoke SAS and bring the statements in the file **regul.mac** to the PROGRAM EDITOR window. Input the following


```
00007          my.chamber
00012          chambtmp
00018          dialset
00023          400
00027          0.99
```

Check the entries and if they are correct, press F10 and the following result appears in the OUTPUT window.

```
------------------------------------------------------------------------

                            Regulation


        The point estimate of x0 is      66.5200

        A finite width  99% confidence interval for x0 exists.

        The lower bound is      65.0709

        The upper bound is      68.0258
------------------------------------------------------------------------
```

So $\hat{x}_0 = 66.52$ and the confidence statement for x_0 is

$$C[65.07 \leq x_0 \leq 68.03] = 0.99$$

S6.4.2. Since the temperature of an individual is desired, the macro to use is **calib**. Bring the SAS statements in the file **calib.mac** to the PROGRAM EDITOR window and input the following information as indicated.

```
00007          my.thermom
00012          reading
00018          knowntmp
00023          100
00027          0.90
```

```
-----------------------------------------------------------------------
                              Calibration

         The point estimate of x0 is    100.1123

         A finite width  90% confidence interval for x0 exists.

         The lower bound is      99.6294

         The upper bound is     100.5885
-----------------------------------------------------------------------
```

So the point estimate of x_0, the temperature of the patient, is 100.1, and the confidence statement is

$$C[99.6 \leq x_0 \leq 100.6] = 0.90$$

S6.4.3 For Problems 6.4.5 and 6.4.6 in the textbook we use the data in the file **crystal.ssd** and find the number of hours, x_0, that crystals need to grow so they will weigh an average of 5 grams. You should recognize this as a regulation problem. Bring the SAS statements in the file **regul.mac** to the `PROGRAM EDITOR` window and enter the following information on the indicated lines.

```
00007            my.crystal
00012            weight
00018            time
00023            5
00027            0.90
```

Press the `F10` key to execute the macro statements. The results are as follows.

```
-----------------------------------------------------------------------
                              Regulation

         The point estimate of x0 is       9.9291

         A finite width  90% confidence interval for x0 exists.

         The lower bound is       8.6503

         The upper bound is      11.0479
-----------------------------------------------------------------------
```


S6.5.1 To solve this problem use the macro **compare**. Bring the SAS statements in the file **compare.mac** to the `PROGRAM EDITOR` window and input the following information on the indicated lines.

```
00007                   my.eggshell
00016                   y1 x1
00017                   y2 x2
00018                   y3 x3
00019
00020
00021
00026                   0.90
00036                   1   0  -1   0   0   0,
00037                   1   0   0   0  -1   0,
00038                   0   0   1   0  -1   0
00039
00040
00041
00042
00043
00044
00045
00046
00047
```

Lines that are shown as blank lines above should be left blank. If there is some information already present there then it should be erased. Check the entries carefully and if they are correct, press `F10` to execute the macro statements. The results are

```
-----------------------------------------------------------------------
                   Comparison of Regression Lines

      The point estimates and simultaneous confidence intervals
      for the thetas with confidence coefficient greater than
      or equal 90% are given below

          THETA     ESTIMATE       LOWER        UPPER
            1       -0.5012      -3.4688       2.4665
            2        0.9436      -1.8135       3.7008
            3        1.4448      -1.5545       4.4441
-----------------------------------------------------------------------
```

Thus we have are least 90% confident that the following are simultaneously correct.

$$-3.4688 \leq \alpha_1 - \alpha_2 \leq 2.4665$$

$$-1.8135 \leq \alpha_1 - \alpha_3 \leq 3.7008$$

$$-1.5545 \leq \alpha_2 - \alpha_3 \leq 4.4441$$

S6.6.1 To solve this problem use the macro **inter**. On the `Command` line of the `PROGRAM EDITOR` window type

```
include 'b:\macro\inter.mac'
```

to bring the macro statements in the file **inter.mac** to the screen. Enter the following information on the indicated lines, replacing the quantities already there if necessary.

```
00008            my.eggshell
00021            y1
00023            x1
00025            y3
00027            x3
00034            0.90
```

Execute the macro statements by pressing the `F10` key. The results are as follows.

```
---------------------------------------------------------------------------
      Intersection of two straight line regression functions

         The point estimate of x0 is   -0.3387


     A finite width  90% confidence interval for x0 exists
     and it is given by

     the interval from      -1.2476 to       0.4485
---------------------------------------------------------------------------
```

The point estimate of x_0 is -0.3387 which indicates that the two regression lines do not intersect in the interval $2 \leq X \leq 20$. We have 90% confidence that the point of intersection is in the interval $[-1.2476, 0.4485]$. So, for all practical purposes, it appears

that for units of the food supplement in the interval [2, 20] the average hardness of eggshells for breed 1 is higher than for breed 3.

S6.6.2 Repeat the procedure used to solve Problem S6.6.1, but use the following information on the indicated lines.

```
00008            my.eggshell
00021            y1
00023            x1
00025            y2
00027            x2
00034            0.90
```

The result is

```
---------------------------------------------------------------------------
      Intersection of two straight line regression functions

         The point estimate of x0 is    0.2564


     A finite width  90% confidence interval for x0 exists
     and it is given by

     the interval from       1.0276 to       1.2335
---------------------------------------------------------------------------
```

S6.7.1 To solve this problem use the macro **quadr**. Bring the macro statements in the file **quadr.mac** to the `PROGRAM EDITOR` window and input the following information on the indicated lines.

```
00007            my.concrete
00016            strength
00018            sand
00026            0.90
```

Upon executing the macro statements the following results appear in the `OUTPUT` window.

```
---------------------------------------------------------------------------

          Maximum or minimum of a quadratic regression model

             The point estimate of x0 is   31.8240
```

```
A finite width  90% confidence interval for x0 exists and is given by

the interval from   29.9008 to   33.9782

--------------------------------------------------------------------------
```

Thus $\hat{x}_0 = 31.82$ and $C[29.90 \le x_0 \le 33.98] = 0.90$.

S6.7.2 The command for plotting **strength** against **sand** for the data in the file **concrete.ssd** is given below.

```
00001 libname my 'b:\';
00002 proc plot data=my.concrete;
00003 plot strength*sand='*';
00004 run;
```

Execute these statements and the following result appears in the OUTPUT window.

```
--------------------------------------------------------------------------

             Plot of STRENGTH*SAND.  Symbol used is '*'.

       8 +                             *    *
         |
         |                        *
         |                   *
       6 +              *                         *
         |         *
STRENGTH |                                                  *
         |
       4 +                                                          *
         |    *
         |
         |
       2 + *
         -+---------+---------+---------+---------+---------+--------+-
          0        10        20        30        40        50       60

                                     SAND

--------------------------------------------------------------------------
```

From this plot it appears that the maximum is close to 32 units of sand.

S6.8.1 (a) Use the macro **spline** to compute the needed quantities. Invoke SAS and bring the macro statements in the file **spline.mac** to the PROGRAM EDITOR window. Then enter the following information on the indicated lines.

```
00007                my.sales
00014                sales
00016                advbudgt
00023                50
00028                0.90
00036                1  0  0  0
```

After the entries have been made and checked press the F10 key and the following will appear in the OUTPUT window.

```
--------------------------------------------------------------------------
                         Spline regression

    The point estimates of alpha1, beta1, alpha2, and beta2,
    respectively, are

       201.4454,      5.0218,     404.2462,      0.9658

    The point estimate of sigma is      11.0488

    The point estimate of theta is     201.4454

    A  90% confidence interval for theta is given by
    the interval from     181.0934   to     221.7973
--------------------------------------------------------------------------
```

From this you can obtain the point estimate and a 90% confidence statement for α_1.

(b) The 90% confidence interval for α_1 can be obtained from the output in part (a).

(c) To plot the estimated spline regression function, you plot the line

$$\hat{\mu}_Y(x) = 201.4454 + 5.0218x \quad \text{for } 0 \le x \le 50$$

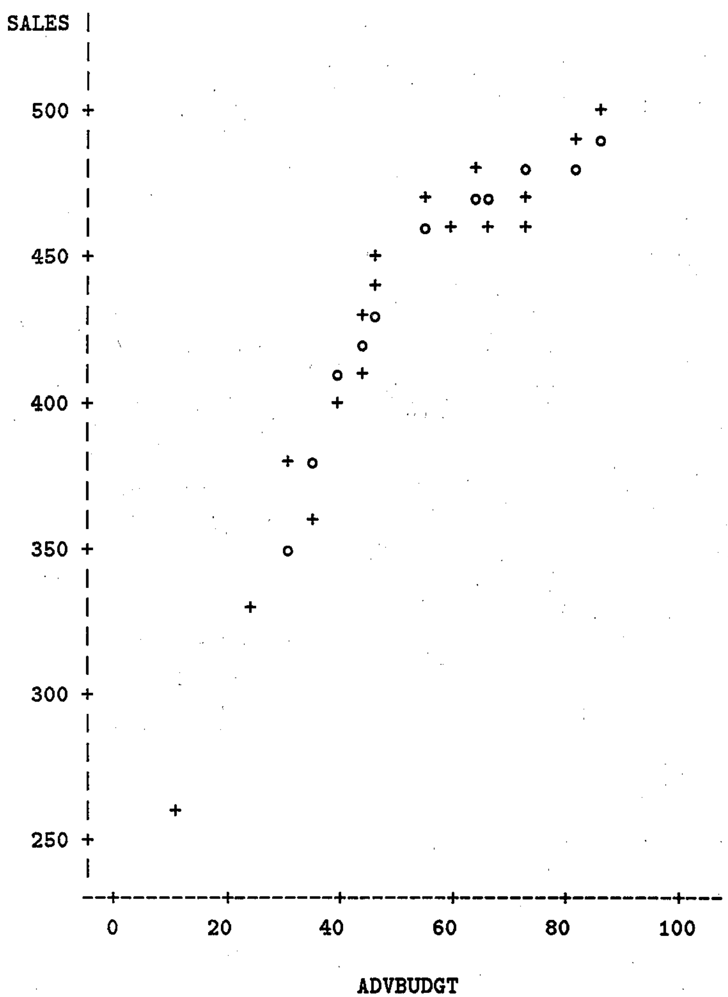
and plot the line

$$\hat{\mu}_Y(x) = 404.2462 + 0.9658x \quad \text{for } x \le 50$$

The appropriate SAS statements and the resulting plot are given below. Note that we

```
data temp;
set my.sales;
if advbudgt <= 50 then fits=201.4454+5.0218*advbudgt;
if advbudgt >50 then fits=404.2462+0.9658*advbudgt;
proc plot data=temp;
plot sales*advbudgt='+' fits*advbudgt='o'/overlay hpos=50 vpos=25;
run;
```

```
Plot of SALES*ADVBUDGT.  Symbol used is '+'.
Plot of FITS*ADVBUDGT.   Symbol used is 'o'.
```



NOTE: 12 obs hidden.

(d) Since $q < 75$, $\mu_Y(75) = \alpha_2 + 75\beta_2$. So use the macro **spline** and input the following information on the indicated lines.

00007	my.sales
00014	sales
00016	advbudgt
00023	50
00028	0.90
00036	0 0 1 75

After the entries have been made and checked, press the F10 key to execute the macro statements. The results are as follows.

```
-----------------------------------------------------------------------

                        Spline regression


    The point estimates of alpha1, beta1, alpha2, and beta2,
    respectively, are

        201.4454,      5.0218,    404.2462,      0.9658

    The point estimate of sigma is       11.0488

    The point estimate of theta is      476.6788

    A  90% confidence interval for theta is given by
    the interval from     469.1915    to     484.1661

-----------------------------------------------------------------------
```

Thus we get $\hat{\mu}_Y(75) = 476.6788$ and the confidence statement is

$$C[469.19 \leq \mu_Y(75) \leq 484.17] = 0.90$$

(e) We want a point estimate and a confidence interval for

$$\mu_Y(80) - \mu_Y(60)$$

and since both 60 and 80 are to the right of the knot point $q = 50$, we get

$$\mu_Y(60) = \alpha_2 + 60\beta_2$$

and

$$\mu_Y(80) = \alpha_2 + 80\beta_2$$

Hence

$$\mu_Y(80) - \mu_Y(60) = 20\beta_2$$

Thus, use the macro **spline** and input the following information.

```
00007                   my.sales
00014                   sales
00016                   advbudgt
00023                   50
00028                   0.90
00036                   0  0  0  20
```

The output is

```
------------------------------------------------------------------------

                         Spline regression


      The point estimates of alpha1, beta1, alpha2, and beta2,
      respectively, are

         201.4454,      5.0218,    404.2462,      0.9658

      The point estimate of sigma is      11.0488

      The point estimate of theta is      19.3154

      A  90% confidence interval for theta is given by
      the interval from      10.9712   to      27.6595
------------------------------------------------------------------------
```

Thus $\hat{\mu}_Y(80) - \hat{\mu}_Y(60) = 19.3154$, and the confidence statement is

$$C[10.9712 \leq \mu_Y(80) - \mu_Y(60) \leq 27.6595] = 0.90$$


```
libname my 'b:\';
proc reg data=my.table733;
model y=x1 x2 x3 x4 x5 x6 x7/selection=rsquare adjrsq cp rmse best=5;
run;
```

S7.3.2 In Problem 7.3.2 in the textbook, the SAS commands for obtaining the eight best models for each subset size, using the C_p criterion, are as follows.

```
libname my 'b:\';
proc reg data=my.table733;
model y=x1 x2 x3 x4 x5 x6/selection=rsquare adjrsq cp rmse best=8;
run;
```

When the total number of subset models of a given size is less than eight, then all of the possible subset models of this size are listed in the output.

S7.4.1 The required SAS commands are

```
libname my 'b:\';
proc reg data=my.table742;
model y = x1 x2 x3/selection = stepwise sle = 0.15 sls = 0.15;
run;
```

The SAS response is

```
------------------------------------------------------------------------

           Stepwise Procedure for Dependent Variable Y

Step 1   Variable X1 Entered     R-square = 0.36878068   C(p) = 9.52041064

                  DF        Sum of Squares      Mean Square      F   Prob>F
Regression         1            3.13795477       3.13795477   4.67   0.0626
Error              8            5.37104523       0.67138065
Total              9            8.50900000

                   Parameter       Standard         Type II
Variable            Estimate          Error  Sum of Squares      F   Prob>F

INTERCEP          9.87279394     0.67060796    145.51627393 216.74   0.0001
X1                0.33181292     0.15348091      3.13795477   4.67   0.0626

Bounds on condition number:             1,             1
------------------------------------------------------------------------

Step 2   Variable X2 Entered     R-square = 0.57322206   C(p) = 6.49360912
```

	DF	Sum of Squares	Mean Square	F	Prob>F
Regression	2	4.87754652	2.43877326	4.70	0.0508
Error	7	3.63145348	0.51877907		
Total	9	8.50900000			

Variable	Parameter Estimate	Standard Error	Type II Sum of Squares	F	Prob>F
INTERCEP	7.69169227	1.32897896	17.37760970	33.50	0.0007
X1	0.46716887	0.15383716	4.78418214	9.22	0.0189
X2	0.59912718	0.32717987	1.73959175	3.35	0.1098

Bounds on condition number: 1.30017, 5.20068

Step 3 Variable X3 Entered R-square = 0.75597837 C(p) = 4.00000000

	DF	Sum of Squares	Mean Square	F	Prob>F
Regression	3	6.43261993	2.14420664	6.20	0.0287
Error	6	2.07638007	0.34606335		
Total	9	8.50900000			

Variable	Parameter Estimate	Standard Error	Type II Sum of Squares	F	Prob>F
INTERCEP	1.73606481	3.01189220	0.11497638	0.33	0.5853
X1	0.18549989	0.18287282	0.35607758	1.03	0.3496
X2	1.55405234	0.52377148	3.04651408	8.80	0.0251
X3	1.14382850	0.53958923	1.55507341	4.49	0.0783

Bounds on condition number: 8.376369, 48.37692

Step 4 Variable X1 Removed R-square = 0.71413120 C(p) = 3.02893757

	DF	Sum of Squares	Mean Square	F	Prob>F
Regression	2	6.07654235	3.03827118	8.74	0.0125
Error	7	2.43245765	0.34749395		
Total	9	8.50900000			

Variable	Parameter Estimate	Standard Error	Type II Sum of Squares	F	Prob>F
INTERCEP	0.26255698	2.64388162	0.00342697	0.01	0.9237
X2	1.79658293	0.46697699	5.14340615	14.80	0.0063
X3	1.54152337	0.37149911	5.98317796	17.22	0.0043

Bounds on condition number: 3.954151, 15.8166

All variables left in the model are significant at the 0.1500 level.
No other variable met the 0.1500 significance level for entry into the model.

Summary of Stepwise Procedure for Dependent Variable Y

Step	Variable Entered	Variable Removed	Number In	Partial R**2	Model R**2	C(p)	F	Prob>F
1	X1		1	0.3688	0.3688	9.5204	4.6739	0.0626
2	X2		2	0.2044	0.5732	6.4936	3.3532	0.1098
3	X3		3	0.1828	0.7560	4.0000	4.4936	0.0783
4		X1	2	0.0418	0.7141	3.0289	1.0289	0.3496

Notice that the variables entered or removed at the various steps agree with the results in Example 7.4.5. The final model includes the predictor variables X_2 and X_3, but not X_1.

S7.5.1

(a) The value of k is 5. The value of m is 24. The value of p is 3.

(b) $t_1 = 4$, $t_2 = 6$, $t_3 = 8$, $t_4 = 10$, and $t_5 = 12$.

(c)

$$X = \begin{bmatrix} 1 & 4 & 16 \\ 1 & 6 & 36 \\ 1 & 8 & 64 \\ 1 & 10 & 100 \\ 1 & 12 & 144 \end{bmatrix}$$

(d)

$$y_3 = \begin{bmatrix} 3.5 \\ 11.3 \\ 18.4 \\ 22.5 \\ 25.3 \end{bmatrix} \qquad y_{10} = \begin{bmatrix} 3.6 \\ 11.4 \\ 18.3 \\ 21.6 \\ 23.9 \end{bmatrix}$$

(e) $\mu_{15}(t) = \alpha_{15} + \beta_{15}t + \gamma_{15}t^2$.

(f) We compute $\hat{\beta}$ using the macro **growth**. The commands for the macro are in the files **growth.mac** and **growth.sas** on the data disk. Bring the macro statements in the file **growth.mac** to the `PROGRAM EDITOR` window and input the following information on the indicated lines.

```
00007          my.pumpkin
00012          5
00016          4  6  8  10 12
00022          3
00027          0  0  1
00031          0.90
```

Execute the macro statements by pressing the `F10` key. The results are given below.

```
------------------------------------------------------------------------
                    Growth curve analysis

 The estimated beta coefficients are

              -18.94333
              6.1738988
              -0.245908

 The estimated value of theta is -0.245908
 and its standard error is       0.0068214

 For a two-sided confidence interval for theta with
 confidence coefficient equal to  90%

 the lower confidence bound is  -0.257599  and
 the upper confidence bound is  -0.234217
------------------------------------------------------------------------
```


Thus a 90% confidence interval for γ is given by

$$C[-0.2576 \leq \gamma \leq -0.2342] = 0.90$$

(g) From the computer output in part (f) we get

$$\hat{\mu}_Y(t) = -18.94333 + 6.1738988t - 0.245908t^2$$

(h)

$$\hat{\mu}_Y(8) = -18.94333 + 6.1738988(8) - 0.245908(8)^2 = 14.7097484$$

(i) $\boldsymbol{a} = [0\ \ 1\ \ 0]^T$.

(j) $\boldsymbol{a} = [1\ \ t\ \ t^2]^T$.

(k) $\mu_Y(12) - \mu_Y(4) = (\alpha + 12\beta + 144\gamma) - (\alpha + 4\beta + 16\gamma) = 8\beta + 128\gamma$. So the population parameters that need to be estimated are β and γ.

S7.5.2

(a) $m = 20$, $k = 4$, and $p = 3$.

(b) $\boldsymbol{X} = \begin{bmatrix} 1 & 8.0 & 64.00 \\ 1 & 8.5 & 72.25 \\ 1 & 9.0 & 81.00 \\ 1 & 9.5 & 90.25 \end{bmatrix}$

(c) $\boldsymbol{a} = [1\ \ 0\ \ 0]^T$.

(d) The estimate of the population growth curve is

$$\hat{\mu}_Y(t) = 26.885 + 3.441t - 0.0915t^2$$

This result is slightly different from what is given in Problem 7.6.2 in the textbook because of rounding errors.

(e) Use the macro **growth** to obtain the required confidence intervals. First, bring the statements in the file **growth.mac** to the `PROGRAM EDITOR` window. To obtain a 95% confidence interval for β, input the following information on the indicated lines

```
00007          my.ramus
00012          4
00016          8 8.5 9 9.5
00022          3
00027          0  1  0
00031          0.95
```

Execute the macro commands. The results are as follows.

```
------------------------------------------------------------------------

                      Growth curve analysis

 The estimated beta coefficients are

                     26.885
                      3.441
                     -0.09

 The estimated value of theta is     3.441
 and its standard error is       3.6685724

 For a two-sided confidence interval for theta with
 confidence coefficient equal to  95%

 the lower confidence bound is   -4.23741  and
 the upper confidence bound is   11.11941
------------------------------------------------------------------------
```

Thus, the required confidence statement is

$$C[-4.23741 \leq \beta \leq 11.11941] = 0.95$$

To obtain a 95% confidence interval for γ, enter the quantities given below to replace those on the indicated lines.

```
00007          my.ramus
00012          4
00016          8 8.5 9 9.5
00022          3
00027          0  0  1
00031          0.95
```


Execute the macro commands and obtain the following results.

```
------------------------------------------------------------------------

                      Growth curve analysis

 The estimated beta coefficients are

                     26.885
                      3.441
                     -0.09

 The estimated value of theta is     -0.09
 and its standard error is        0.211374

 For a two-sided confidence interval for theta with
 confidence coefficient equal to  95%

 the lower confidence bound is  -0.532411  and
 the upper confidence bound is  0.3524108
------------------------------------------------------------------------
```

Thus, the required confidence statement is

$$C[-0.532411 \leq \gamma \leq 0.3524108] = 0.95$$

(f) $\mu_Y(t) = \alpha + \beta t + \gamma t^2$.

(g) $\mu_Y(8.5) = \alpha + 8.5\beta + 72.25\gamma$.

(h) According to the confidence statement in part (e), we are 95% confident that γ is somewhere in the interval $[-0.532411, 0.3524108]$. There is not enough information to decide whether or not γ is less than 0.002 in magnitude.

S8.2.1 The SAS commands and output for Problem 8.2.1 in the textbook are given below.

```
libname my 'b:\';
```

```
data so2;
set my.so2;
wts=1/(tonperhr**2);
proc print data=so2;
proc reg data=so2;
model mgpermt3=tonperhr/i;
weight wts;
run;
```

Note that the statement `wts = 1/(tonperhr**2)` defines the weights as 1/(tonperhr)^2. The symbol `**2` means "raising to the power 2." The result which appears in the `OUTPUT` window is given below.

OBS	MGPERMT3	TONPERHR	WTS
1	5.21	1.92	0.27127
2	7.36	3.92	0.06508
3	16.26	6.80	0.02163
4	10.10	6.32	0.02504
5	5.80	2.00	0.25000
6	8.06	4.32	0.05358
7	4.76	2.40	0.17361
8	6.93	2.96	0.11413
9	9.36	3.52	0.08071
10	10.90	4.24	0.05562
11	12.48	5.12	0.03815
12	11.70	5.84	0.02932
13	7.44	3.60	0.07716
14	6.99	2.80	0.12755

Model: MODEL1

X'X Inverse, Parameter Estimates, and SSE

	INTERCEP	TONPERHR	MGPERMT3
INTERCEP	5.2968078814	-1.546900208	1.7214483337
TONPERHR	-1.546900208	0.5231912757	1.7761563188
MGPERMT3	1.7214483337	1.7761563188	1.3374565627

Dependent Variable: MGPERMT3

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	1	6.02979	6.02979	54.101	0.0001
Error	12	1.33746	0.11145		
C Total	13	7.36724			

Root MSE	0.33385	R-square	0.8185
Dep Mean	6.97294	Adj R-sq	0.8033
C.V.	4.78777		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
INTERCEP	1	1.721448	0.76834511	2.240	0.0448
TONPERHR	1	1.776156	0.24147905	7.355	0.0001

All the quantities needed to answers Problems 8.2.1–8.2.4 can be obtained from the SAS output above.

S8.2.3 The SAS commands and output for Exercise 8.4.1 are given below.

```
libname my 'b:\';
data soyburgr;
set my.soyburgr;
wts = 1/(filler**4);
proc print data=soyburgr;
proc reg data=soyburgr;
model texture = filler/i;
weight wts;
run;
```

Note that the statement `wts = 1/(filler**4)` defines the weights as 1/(filler)^4; the symbol '**4' stands for "raising to the power 4."

OBS	TEXTURE	FILLER	WTS
1	2.5	0.5	16.0000
2	2.9	1.0	1.0000
3	3.4	1.5	0.1975
4	3.7	2.0	0.0625
5	4.3	2.5	0.0256
6	4.5	3.0	0.0123
7	4.9	3.5	0.0067
8	5.8	4.0	0.0039
9	6.4	4.5	0.0024
10	6.8	5.0	0.0016
11	6.5	5.5	0.0011
12	8.0	6.0	0.0008
13	8.4	6.5	0.0006
14	8.5	7.0	0.0004
15	7.4	7.5	0.0003
16	9.9	8.0	0.0002

Model: MODEL1

X'X Inverse, Parameter Estimates, and SSE

	INTERCEP	FILLER	TEXTURE
INTERCEP	0.3612284207	-0.547297327	2.0697940124
FILLER	-0.547297327	0.9870041652	0.8577620719
TEXTURE	2.0697940124	0.8577620719	0.005028283

Dependent Variable: TEXTURE

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	1	0.74544	0.74544	2075.501	0.0001
Error	14	0.00503	0.00036		
C Total	15	0.75047			

Root MSE	0.01895	R-square	0.9933
Dep Mean	2.54543	Adj R-sq	0.9928
C.V.	0.74454		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
INTERCEP	1	2.069794	0.01139034	181.715	0.0001
FILLER	1	0.857762	0.01882805	45.558	0.0001

All the quantities needed to answer the questions in Exercise 8.4.1 can be obtained from the above output.

S8.3.1 To solve this problem we use the macro **theil**. Bring the SAS statements in the file **theil.mac** to the `PROGRAM EDITOR` window and enter the following information on the indicated lines, replacing the information there if necessary.

00010	my.profsal
00018	salary
00020	yrsexp
00031	1 0
00035	0.90

After entering these quantities, press the `F10` key to execute the macro commands. The following results will appear in the `OUTPUT` window.

```
Straight line regression using the method of Theil

The point estimate of theta is        28

For a two-sided confidence interval for theta with confidence
coefficient equal to  0.875 (this is the value that is closest
to the desired value of  0.900)

the lower confidence bound is         20 and
the upper confidence bound is  31.111111
```

S8.3.2 Use the macro **theil** as in Problem S8.3.1, but replace the quantity on line 00031 with 0 1. All other quantities remain the same. The SAS response is

```
--------------------------------------------------------------------------
 The point estimate of theta is         2


 For a two-sided confidence interval for theta with confidence
 coefficient equal to  0.875 (this is the value that is closest
 to the desired value of  0.900)

 the lower confidence bound is  1.7222222 and
 the upper confidence bound is      2.375
--------------------------------------------------------------------------
```

S8.3.3 For parts (e) and (g) use the macro **theil**. Bring the macro statements in the file **theil.mac** to the PROGRAM EDITOR window and input the following information.

```
00010                          my.so2
00018                          mgpermt3
00020                          tonperhr
00031                          1    0
00035                          0.90
```

Then execute the macro commands to obtain the answers for parts (e) and (g).

For parts (f) and (h), replace the quantitity on line 00031 with 0   1 .

For part (j), use 1   5 on line 00031. Since no confidence interval is required you can leave the quantity 0.90 on line 00035 unchanged.

For part (l), use 0   2.5 on line 00031 since we are interested in the quantity $2.5\beta_1$. Use 0.90 on line 00035.

S9.3.1 The SAS commands for this problem are displayed below.

```
libname my 'b:\';
options center linesize=75 pagesize=60;
proc nlin data=my.absorpt method=dud maxiter=20;
model concentr=1/(beta1+beta2*time+beta3*time**2);
parms beta1=0.8 beta2=-0.67 beta3=0.16;
output out=diagnstc p=fits r=residual student=stdresid;

proc plot data=diagnstc;
```



```
plot concentr*time='o' fits*time='+'/overlay
     hpos=50 vpos=25;
run;
```

Selected portions of the SAS output are given below.

```
--------------------------------------------------------------------------
Non-Linear Least Squares DUD Initialization   Dependent Variable CONCENTR
     DUD          BETA1         BETA2           BETA3 Sum of Squares
      -4       0.800000     -0.670000        0.160000       4.913749
      -3       0.880000     -0.670000        0.160000      18.185283
      -2       0.800000     -0.737000        0.160000    3472.813142
      -1       0.800000     -0.670000        0.176000      16.368102

                 Non-Linear Least Squares Iterative Phase
                 Dependent Variable CONCENTR   Method: DUD
     Iter         BETA1         BETA2           BETA3 Sum of Squares
       0       0.800000     -0.670000        0.160000       4.913749
       1       0.820060     -0.670091        0.161026       1.683922
       2       0.819479     -0.670119        0.160990       1.596250
       3       0.818862     -0.670749        0.161210       1.502415
       4       0.819419     -0.671137        0.161159       1.463014
       5       0.816934     -0.676072        0.162825       1.230167
       6       0.817747     -0.676638        0.162941       1.229876
       7       0.816737     -0.675710        0.162743       1.229869
       8       0.816882     -0.675994        0.162841       1.229741
       9       0.818078     -0.677346        0.163205       1.229594
      10       0.818104     -0.677369        0.163210       1.229594
      11       0.818314     -0.677561        0.163251       1.229593
      12       0.818287     -0.677551        0.163253       1.229592
      13       0.818306     -0.677590        0.163266       1.229591
      14       0.818306     -0.677590        0.163266       1.229591
NOTE: Convergence criterion met.

Non-Linear Least Squares Summary Statistics    Dependent Variable CONCENTR

      Source                DF Sum of Squares     Mean Square

      Regression             3   195.23040870     65.07680290
      Residual               9     1.22959130      0.13662126
      Uncorrected Total     12   196.46000000

      (Corrected Total)     11    81.14000000
```

Parameter	Estimate	Asymptotic Std. Error	Asymptotic 95 % Confidence Interval Lower	Upper
BETA1	0.8183058960	0.06534957701	0.67047362448	0.96613816756
BETA2	-.6775897861	0.06163049008	-.81700882692	-.53817074524
BETA3	0.1632664357	0.01450233506	0.13045959543	0.19607327597

Asymptotic Correlation Matrix

Corr	BETA1	BETA2	BETA3
BETA1	1	-0.984680841	0.9433940967
BETA2	-0.984680841	1	-0.985946029
BETA3	0.9433940967	-0.985946029	1

```
        Plot of CONCENTR*TIME.  Symbol used is 'o'.
        Plot of FITS*TIME.      Symbol used is '+'.

    10.0 +
         |
         |
         |             o
         |
         |
     7.5 +
         |                 +
         |                 o
CONCENTR |
         |         o
         |
     5.0 +
         |
         |
         |                     o
         |     +
         |     o                   o
     2.5 +
         | +                       +
         | o                           o
         |                             +   o       o
         |                                 +   +
         |                                     o   +   +
     0.0 +                                             o
         --+---+---+---+---+---+---+---+---+---+---+---+----
          0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0

                                 TIME
NOTE: 3 obs hidden.
```

S9.3.2 The SAS commands for this problem are

```
libname my 'b:\';
options center linesize=75 pagesize=60;
proc nlin data=my.coil method=dud maxiter=20;
model sensitvy=beta1*(1-exp(-exp(-(beta2+beta3*thicknes))));
parms beta1=0.2 beta2=-1 beta3=14;
output out=diagnstc p=fits r=residual student=stdresid;

proc plot data=diagnstc;
plot sensitvy*thicknes='o' fits*thicknes='+'/overlay
     hpos=50 vpos=25;
run;
```

Selected portions of the SAS output are given below.

NOTE: Convergence criterion met.

Non-Linear Least Squares Summary Statistics   Dependent Variable SENSITVY

Source	DF	Sum of Squares	Mean Square
Regression	3	15.975543499	5.325181166
Residual	13	0.136656501	0.010512039
Uncorrected Total	16	16.112200000	
(Corrected Total)	15	2.569800000	

Parameter	Estimate	Asymptotic Std. Error	Asymptotic 95 % Confidence Interval Lower	Upper
BETA1	1.94810405	0.4724350775	0.9274707307	2.968737376
BETA2	-1.27025028	0.6808704554	-2.7411805853	0.200680035
BETA3	14.36440678	2.7037305519	8.5233555422	20.205458027

Asymptotic Correlation Matrix

Corr	BETA1	BETA2	BETA3
BETA1	1	0.9818657694	-0.911471536
BETA2	0.9818657694	1	-0.968319743
BETA3	-0.911471536	-0.968319743	1

```
       Plot of SENSITVY*THICKNES.  Symbol used is 'o'.
       Plot of FITS*THICKNES.      Symbol used is '+'.

     2.0 +
         |
         |
         |
         |
         | +
     1.5 + o o o
         |     + o
         |       + o
SENSITVY |         +
         |           o
         |           +
     1.0 +             o
         |               +
         |               o +
         |                   +
         |                 o o +
         |                     o o
     0.5 +                         o o o o
         |                             +
         |                               +
         |
         |
         |
     0.0 +
         --+---------+---------+---------+------------------
         0.05      0.10      0.15      0.20

                              THICKNES
```

NOTE: 5 obs hidden.

S9.3.3 The SAS commands for this problem are given below.

```
libname my 'b:\';
proc nlin data=my.contrast method=dud maxiter=20;
model y = 1/(1+exp(-(beta1+beta2*x)));
parms beta1=-3.0 beta2= 150.0;
output out=diagnstc p=fits r=residual student=stdresid;
```



```
proc plot data=diagnstc;
plot y*x='o' fits*x='+'/overlay hpos=50 vpos=25;
run;
```

Selected portions of the SAS output are given below.

NOTE: Convergence criterion met.

Non-Linear Least Squares Summary Statistics   Dependent Variable Y

Source	DF	Sum of Squares	Mean Square
Regression	2	3.6923266027	1.8461633013
Residual	8	0.0018733973	0.0002341747
Uncorrected Total	10	3.6942000000	
(Corrected Total)	9	1.3516400000	

Parameter	Estimate	Asymptotic Std. Error	Asymptotic 95 % Confidence Interval Lower	Upper
BETA1	-4.0261134	0.1290017914	-4.32359496	-3.72863189
BETA2	171.6634747	5.2988898636	159.44409485	183.88285450

Asymptotic Correlation Matrix

Corr	BETA1	BETA2
BETA1	1	-0.963046247
BETA2	-0.963046247	1

```
                Plot of Y*X.     Symbol used is 'o'.
                Plot of FITS*X.  Symbol used is '+'.

     1.00 +                                              o
          |                                         o    +
          |
          |                                    o
          |
          |                               o
     0.75 +                               +
          |
          |
        Y |
          |                          +
          |                          o
     0.50 +
          |
          |
          |                     +
          |                     o
          |
     0.25 +
          |                +
          |                o
          |
          |           o
          |      o
     0.00 + o
          --+---------+---------+---------+---------+---------+--------
          0.00      0.01      0.02      0.03      0.04      0.05

                                          X

NOTE: 5 obs hidden.
```

S9.4.1 The SAS commands for this problem are given below.

```
libname my 'b:\';
proc nlin data=my.contrast method=dud maxiter=30;
model y = 1/(1+exp(-(beta1+beta2*x)));
parms beta1=-3.96 beta2= 174.76;
output out=diagnstc p=fits r=residual student=stdresid;
```



```
proc plot data=diagnstc;
plot y*x='o' fits*x='+'/overlay hpos=50 vpos=25;
run;
```

Selected portions of the output is given below.

```
NOTE: Convergence criterion met.

     Non-Linear Least Squares Summary Statistics     Dependent Variable Y

       Source                 DF Sum of Squares      Mean Square

       Regression              2   3.6923266027     1.8461633013
       Residual                8   0.0018733973     0.0002341747
       Uncorrected Total      10   3.6942000000

       (Corrected Total)       9   1.3516400000


       Parameter    Estimate    Asymptotic                Asymptotic 95 %
                                Std. Error              Confidence Interval
                                                         Lower        Upper
          BETA1   -4.0261321  0.1290043081   -4.32361948  -3.72864480
          BETA2  171.6643751  5.2990063520  159.44472665 183.88402355


                          Asymptotic Correlation Matrix

                  Corr                 BETA1                BETA2
                  -------------------------------------------------
                     BETA1                 1         -0.963047854
                     BETA2      -0.963047854                    1
```

S9.4.2 The SAS commands for this problem are given below.

```
libname my 'b:\';
proc nlin data=my.absorpt method=dud maxiter=30;
model concentr = 1/(beta1+beta2*time+beta3*time**2);
parms beta1= 1.40 beta2= -1.29 beta3=0.28;
output out=diagnstc p=fits r=residual student=stdresid;
```

```
proc plot data=diagnstc;
plot concentr*time='o' fits*time='+'/overlay hpos=50 vpos=25;
run;
```

Selected portions of the SAS output are given below.

```
--------------------------------------------------------------------------------

NOTE: Convergence criterion met.

Non-Linear Least Squares Summary Statistics     Dependent Variable CONCENTR

      Source                 DF Sum of Squares      Mean Square

      Regression              3   195.23040867      65.07680289
      Residual                9     1.22959133       0.13662126
      Uncorrected Total      12   196.46000000

      (Corrected Total)      11    81.14000000


      Parameter     Estimate    Asymptotic               Asymptotic 95 %
                                Std. Error             Confidence Interval
                                                       Lower          Upper
         BETA1  0.8182943392 0.06531707354 0.67053559627 0.96605308221
         BETA2  -.6775773333 0.06159831483 -.81692358802 -.53823107857
         BETA3  0.1632632017 0.01449462904 0.13047379385 0.19605260964

                       Asymptotic Correlation Matrix

      Corr                   BETA1                 BETA2                 BETA3
      -------------------------------------------------------------------------
         BETA1                   1          -0.984670624          0.9433594061
         BETA2        -0.984670624                     1          -0.985937799
         BETA3        0.9433594061          -0.985937799                     1

--------------------------------------------------------------------------------
```


INDEX

Alternate assumptions 160
Analysis of variance 74, 96
Arithmetic 21
ASCII 8
Assumptions 54, 160

Basic statistical concepts 1

Calibration 122
Coefficient of correlation 75
Coefficient of determination 75
Command for changing one value 111
Comparison of several straight line
  regressions 127
Comparison of two regression functions
  (nested case) 99
Comparison of two regression functions
  (non-nested case) 103
Confidence intervals 23
  for βs 61, 91
  for calibration 122
  for comparing regression lines 127
  for lack-of-fit 103
  for linear combination of β_i 65, 91
  for maximum or minimum of a
    quadratic regression 134
  for $\mu_Y(x)$ 65
  for nonlinear regression 172
  for prediction 115
  for ratio of sigmas 99, 101
  for regulation 122
  for σ 62
  for σ_Y 62
  for spline regression 136
  for tolerance points 119
  for two intersecting lines 130
  for $Y(x)$ 68
  for $Y(x_1, \ldots, x_k)$ 115
Cook's distance 111
Correlation 75

Data file
  ASCII 8
  SAS 8, 10
Dataset
  name 4
  SAS 8
  temporary 8
Diagnostic commands 86
  Cook's distance 111
  DFFITS 111
  fits 55, 86, 107
  hat values 55, 86, 111
  influential observations 111
  nscores 57, 86,
  residuals 55, 86, 107
  standardized residuals 55, 86, 107
  studentized deleted residuals 108
Display manager windows 2, 3

Estimates of regression coefficients
  in multiple linear regression 77
  in nonlinear regression 172
  in spline regression 136

in straight line regression 50
in straight line regression through the origin 76
using matrices 78
Estimate of σ 50
Exit SAS 2

Fitted values 55, 86, 107
Function keys 3

Gaussian assumptions 86
Gaussian population 33
Gaussian scores (nscores) 57
Growth curves 153

Hat values 55, 86, 111
Hbar charts 33
Histograms 33

Ill conditioning 112
IML 24
exiting 25
invoking 25
Initial guesses (nonlinear regression) 168
Intersection regression lines 130
Invoking SAS 2

Keyboard 4

Lack-of-fit 103
Leverages 111
Libname 10
Linear combinations 87
Linearizable models 173

Macro
calibration 122
cilinear 91
citheta 65
compare 127
growth 153, 157
intersection 130
lackfit 103
maximum(minimum) of a quadratic regression 134
pred 115
predy 68
ratiosgm 99
regulation 122, 124
sgmaconf 62
spline 136
testmult 95
test 71
Theil 164
tolerance 119
Matrix commands
add 30
creating matrices from SAS data file 79
creating vector of ones 79
entering via keyboard 25
entering from a data file 28
invert 30
load 28
multiply 30
print 26
regression 52, 82
store 27
show storage 28
subtract 30
transpose 30
Maximum of a quadratic regression function 134
Minimum of a quadratic regression function 134
Multicollinearity 112

Name
dataset 4, 5
variable 5
Nonlinear regression 167

number of iterations 168
starting values 168
Nscores 87

Options 169

Plot 36
Prediction interval
for $Y(x)$ 68
for $Y(x_1, \ldots, x_k)$ 114
Print 6
Proc commands
proc contents 11
proc means 15, 23
proc plot 36
proc print 6, 109
proc reg 52, 82
proc univariate 12, 13

Rankit plot 57, 59, 88
Read 4, 5, 8
from file 8
from keyboard 5
Regression 39
diagnostics 107
nonlinear regression 167
dud 169
maximum number of iterations 169
through the origin 76
Regression computations
using proc reg command 52, 82
using matrices 78
Residual analysis 85

Set command 21, 111
Single observation 45
Spline regression 136
Standard deviation 23
Standard errors 23
in straight line regression 53
in multiple linear regression 84
Standardized residuals 55, 86, 107
Storing Matrices 27
Studentized deleted residuals 108
Subpopulation 42

Tests 23, 71
Theil's method 164
Tolerance intervals 119

Unequal standard deviations 160

Variance inflation factor 113
Variables selection
$Adj - R^2$ 141
backward elimination 152
best m models 143
best subsets 141, 143
C_p 141
forward selection 151
R^2 141
s 141
sle 145
sls 145
stepwise 145
Vbar charts 33

Weighted regression 162
Weights 161
Windows 2
LOG 2
OUTPUT 2
PROGRAM EDITOR 2
Working directory 2

Zooming 4