Economic Computing Resources at Middle Tennessee State University: a User's Guide

Stuart J. Fowler*
Economics and Finance Department
Middle Tennessee State University
Murfreesboro, TN 37132

August 29, 2011

1 Introduction

There are many economic computational resources available to students and faculty of Middle Tennessee State University. Examples of these resources are programming languages such as Matlab, Maple, Mathematica, SAS, and STATA all located on the `frank` server. These programs differ by the function they are intended to perform. For example, Matlab is used mostly for numeric computations while Maple and Mathematica are used primarily for symbolic computations. SAS and STATA are econometric packages used in statistical analysis of economic data. However, though these programs differ by the functions they were designed for, the programs are operated in a very similar manner. This is due to the fact that all the programs located on the `frank` server run on a *Unix* operating system.

Unix has the feature that all computer language programs use the same *text editor* to write programs for submittal to the language programs. For instance, a program written for Matlab will be written in a Unix text editor using the Matlab syntax. The program is then submitted to Matlab. Finally, the output of the program is viewed or printed by the same Unix text editor. It is evident, to run language programs on Unix, knowledge of Unix and the Unix text editors is essential. Thus, the purpose of these pages is twofold. First, the reader is introduced to the Unix operating system and the using of text editors. Second, examples of how to use the Matlab, SAS, and Maple programs for economic uses are given. The format of these pages follow:

- How to obtain a `frank` account
- Where and how to log into your `frank` account
- Basic Unix commands
- How to create and edit files in Unix
- FTP'ing
- How to create, run, and interpret the results of a Matlab program
- More Matlab Examples
- How to create, run, and interpret the results of a SAS program
- More SAS Examples

*For questions or comments E-mail: `Stuart.Fowler@mtsu.edu` or Phone: (615) 898-2383

1. Data Manipulation
2. The Least Squares Problem Using SAS

- How to log off of your `frank` account

Please note that Unix is *case sensitive*. That is to say, capitalize where capitalized and don't if not.

2 How to Obtain a `frank` Account

The `frank` is a general purpose server available to the MTSU community. Besides e-mail and web-pages, statistical and mathematical programs are also located on `frank`. Most servers, including `frank`, run on a Unix operating system. Thus, we will need to know some Unix commands to start. First, however, you will need to apply for an account. To get an account: visit or call the Information Technology Division (ITD) help desk:

Office of Information Technology
Help Desk
(615) 898-5345
`http://www.mtsu.edu/itd`

You will apply online at: `http://www.mtsu.edu/passwords.shtml`. You will need your MTSU ID numer.

3 Where and How to Log Into Your `frank` Account

MTSU has many computer centers and labs on campus. Most allow the user to *telnet* to the `frank`. This connection to `frank` is obtained through the TCP/IP network. The software package used to communicate over the internet is often just called `putty` (downloaded at: `http://www.putty.org/`). Usually, this program is located in the communication program group. The name for the icon in the program group is sometimes `putty` or `frank`. Some places to log into the `frank` are:

- Business Computer Lab (BCL).
- Economic and Finance Department Computing Lab (EFDCL).
- At home using Putty (HOME).

`Putty` is run by opening or double clicking on the program icon. `Putty` should prompt you for a remote host address. The remote address for the frank that should be entered is:

```
frank.mtsu.edu
```

Once you have opened a link to the `frank` you should see on your monitor and enter:

```
login as: sfowler
Using keyboard-interactive authentication.
Password:
Last   successful login for sfowler: Fri Aug 26 15:22:54 CST6CDT 2011 on tty
```

```
Last unsuccessful login for sfowler: Fri Jul 29 12:13:34 CST6CDT 2011
Last login: Fri Aug 26 15:22:54 2011 from 161.45.248.165:
(c)Copyright 1983-2000 Hewlett-Packard Co.,  All Rights Reserved.
(c)Copyright 1979, 1980, 1983, 1985-1993 The Regents of the Univ. of California
(c)Copyright 1980, 1984, 1986 Novell, Inc.
(c)Copyright 1986-1992 Sun Microsystems, Inc.
(c)Copyright 1985, 1986, 1988 Massachusetts Institute of Technology
(c)Copyright 1989-1993  The Open Software Foundation, Inc.
(c)Copyright 1986 Digital Equipment Corp.
(c)Copyright 1990 Motorola, Inc.
(c)Copyright 1990, 1991, 1992 Cornell University
(c)Copyright 1989-1991 The University of Maryland
(c)Copyright 1988 Carnegie Mellon University
(c)Copyright 1991-2000 Mentat Inc.
(c)Copyright 1996 Morning Star Technologies, Inc.
(c)Copyright 1996 Progressive Systems, Inc.
(c)Copyright 1991-2000 Isogon Corporation, All Rights Reserved.


                           RESTRICTED RIGHTS LEGEND
Use, duplication, or disclosure by the U.S. Government is subject to
restrictions as set forth in sub-paragraph (c)(1)(ii) of the Rights in
Technical Data and Computer Software clause in DFARS 252.227-7013.

                           Hewlett-Packard Company
                           3000 Hanover Street
                           Palo Alto, CA 94304 U.S.A.

Rights for non-DOD U.S. Government Departments and Agencies are as set
forth in FAR 52.227-19(c)(1,2).
frank $
```

4 Basic Unix Commands

Press the <**return**> key after each of the following commands in Table 1 on page 4. Remember to use the lower case letters unless otherwise indicated.
As an example, to list the files in your directory, use the `ls` command. At the `frank` prompt type:

```
frank $ ls
```

and press the <**return**> key. Your listing might resemble the following

```
bin
mail
```

`pwd`	Gives the identity and the pathname of your present directory
`cd` *dirname*	Changes your present working directory to *dirname*.
`cd`	Changes present working directory to your home directory.
`mkdir` *dirname*	Creates a directory called *dirname*. Your present working directory will be the parent of this new directory, unless you specify another pathname.
`rmdir` *dirname*	Deletes a directory called *dirname*, if the directory is completely empty. If you are not in *dirname's* parent directory, you must use *dirname's* full pathname.
`cp` *filename1 filename2*	Makes a copy of a file called *filename1* and names a copy *filename2*.
`rm` *filename*	Deletes a file called *filename*.
`pico` *filename*	Uses the Pico editor to open a file called *filename*. If a file of that name does not already exist, it starts a new one.
`exit`	Logs off the Unix system

Table 1: Basic Unix Commands

When you first receive your account, your home directory will have files, like these, so that your account will operate properly.

5 Creating and Editing Files in Unix

A text editor lets you enter, edit, re-arrange text in files. Unix computers support several text editors including *pico*, *vi*, and others. If you are a beginning Unix user, you might prefer pico to vi for each of the following reasons:

- Pico is easier to learn
- Pico contains on-line help

To create a new file `ex1.txt` using `pico`, at the `frank` prompt enter:

```
frank $ pico ex1.txt
```

This starts the text editor and opens a blank file called `ex1.txt`. Editing commands and cursor movements are listed at the bottom of the screen and are given to `pico` by typing special control key sequences. A caret, ˆ, is used to denote the control key, sometimes marked <`CTRL`> on your keyboard. Thus the <`CTRL-x`> key combination is written as `ˆx`. This means to hold down the <`CTRL`> key while you press the x key.

Once in this file type

```
Robert Lucas won the Nobel Peace Prize
```

To save this file and exit `pico`:

1. Press <`CTRL-x`>.
2. In response to the question: Save before leaving (y/n)?, type `y`. (Note: If no changes have been made to the current opened file then question, "Save before leaving (y/n)?," will not appear.)
3. In response to the prompt: Filename to write, press the <`return`> or enter a new filename.

To re-open and edit the file with `pico`, at the `frank` prompt enter:

```
frank $ pico ex1.txt
```

6 FTP'ing

FTP is a program that can transfer files from frank to your home computer. A good freeware version is WinSCP that can be found at: `http://en.wikipedia.org/wiki/WinSCP`.

7 To Create, Run, and Interpret the Results of a Matlab Program

One of the distinguishing features of a Unix computer is that the programs on the computer can communicate with another. Specifically, programs such as Matlab can use files created in the pico editor as input programs. Thus, one would write the Matlab input program in pico, submit to Matllab this input program, and then view the output files from Matlab with the pico editor.

This example is to create an input Matlab file called `ex1.m`. To start, create a new file called `ex1.m` using `pico`. At the `frank` prompt enter:

```
%
% ex1.m is an example matlab program
%

% ----- save output to file ----- %

clear all
diary ex1.mout
diary off
delete ex1.mout
diary ex1.mout

% ----- display date and time of computation ----- %

format short g
date
time0 = clock;

% ----- enter matrices ----- %
```

```
a  = [5;
      4;
      2]

b  = [1;
      1;
      2]

AA = [2 1 3;
      1 3 2;
      3 2 5];

% ----- create special matrices in matlabl ----- %

iota = ones(10,1);
ident = eye(10);

iota, ident

% ----- use of the addtion, multiplication, and transpose operators ----- %

temp = a'+b'
temp1 = (a+b)'

temp = a'*AA
temp1 = AA*AA
temp2 = AA^2

% ----- enter data into matrices ----- %

y = [15.3
     19.91
     20.94
     19.66
     21.32
     18.33
     19.59
     21.3
     20.93
     21.64
     21.9
     20.5
     22.83
     23.49
     24.2
     23.05
     24.01
     25.83
     25.15
     25.06]

x = [1 17.3
     1 21.91
     1 23.14
     1 21.86
     1 23.72
     1 20.73
     1 22.19
     1 23.9
     1 23.73
     1 24.44
     1 24.9
     1 23.5
     1 26.03
```

```
     1 26.69
     1 27.6
     1 26.45
     1 27.61
     1 29.43
     1 28.95
     1 28.86 ]


% ----- turn off diary ----- %

comptime = etime(clock, time0)
diary off
```

When you finish typing, save and exit `ex1.m` using the three steps that precede this example.

To run Matlab, at the `frank` prompt enter:

```
frank $ matlab
```

You will see:

```
Warning: Unable to open display , MATLAB is starting without a display.
  You will not be able to display graphics on the screen.
Warning:
  MATLAB is starting without a display, using internal event queue.
  You will not be able to display graphics on the screen.


                                          < M A T L A B >
                              Copyright 1984-2005 The MathWorks, Inc.
                              Version 7.1.0.183 (R14) Service Pack 3
                                          August 02, 2005


  To get started, type one of these: helpwin, helpdesk, or demo.
  For product information, visit www.mathworks.com.

>>
```

At the Matlab prompt enter:

```
>> ex1.m
```

Exit Matlab and list the files in your directory (might resemble the following):

```
bin        ex1.m    ex1.mout   mail
```

It is important to note that `ex1.mout` is the output file from your matlab session. To view `ex1.mout` at `frank` enter:

```
frank $ pico ex1.mout
```

You should see on your monitor:

```
ans =

29-Aug-2011


a =

     5
     4
     2


b =

     1
     1
     2


iota =

     1
     1
     1
     1
     1
     1
     1
     1
     1
     1


ident =

     1     0     0     0     0     0     0     0     0     0
     0     1     0     0     0     0     0     0     0     0
     0     0     1     0     0     0     0     0     0     0
     0     0     0     1     0     0     0     0     0     0
     0     0     0     0     1     0     0     0     0     0
     0     0     0     0     0     1     0     0     0     0
```

```
     0     0     0     0     0     0     1     0     0     0
     0     0     0     0     0     0     0     1     0     0
     0     0     0     0     0     0     0     0     1     0
     0     0     0     0     0     0     0     0     0     1


temp =

     6     5     4


temp1 =

     6     5     4


temp =

    20    21    33


temp1 =

    14    11    23
    11    14    19
    23    19    38


temp2 =

    14    11    23
    11    14    19
    23    19    38


y =

         15.3
        19.91
        20.94
        19.66
        21.32
        18.33
        19.59
         21.3
        20.93
        21.64
         21.9
         20.5
        22.83
        23.49
         24.2
        23.05
        24.01
        25.83
        25.15
        25.06


x =

            1         17.3
            1        21.91
            1        23.14
            1        21.86
```

```
         1        23.72
         1        20.73
         1        22.19
         1         23.9
         1        23.73
         1        24.44
         1         24.9
         1         23.5
         1        26.03
         1        26.69
         1         27.6
         1        26.45
         1        27.61
         1        29.43
         1        28.95
         1        28.86


comptime =

     0.007962
```

When you finish looking at `ex1.mout` save and exit it using the three steps that precede this example.

8 More Matlab Examples

8.1 OLS

The OLS problem is to find some k-vector $\boldsymbol{\beta}$ such that the distance between $\mathbf{y}$ and $\mathbf{X}\boldsymbol{\beta}$ is minimized;

$$\min_{\boldsymbol{\beta}} \sum_{t=1}^{n} (y_t - \mathbf{X}_t \boldsymbol{\beta})^2 = \min_{\boldsymbol{\beta}} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

The first order condition gives

$$\mathbf{X}^\top \left(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\right) = 0,$$

From these equations we obtain the solutions

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{y}. \tag{1}$$

In Matlab, the command

```
>> inv(x'*x)*x'*y

ans =

       1.4178
      0.82481
```

returns the numerical version of the OLS estimator $\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{y}$. Using the estimator for $\boldsymbol{\beta}$, equation (1), we can write the estimator for the residuals as

$$\begin{aligned}\hat{\mathbf{e}} &= \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{y} - \mathbf{X}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{y} \\ &= \left(\mathbf{I} - \mathbf{P}_X\right)\mathbf{y} = \mathbf{M}_X\mathbf{y}\end{aligned}$$

where $\mathbf{P}_X = \mathbf{X}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top$ and $\mathbf{M}_X = (\mathbf{I} - \mathbf{P}_X)$. Then, the estimated SSR is

$$\mathbf{y}^\top\mathbf{M}_X^\top\mathbf{M}_X\mathbf{y} = \mathbf{y}^\top\mathbf{M}_X\mathbf{y}.$$

As a basis for deriving ESS let us write the least squares prediction equation as

$$\mathbf{y} = \mathbf{X}\hat{\boldsymbol{\beta}} + \hat{\mathbf{e}} = \hat{\mathbf{y}} + \hat{\mathbf{e}}.$$

Squaring both sides

$$\mathbf{y}^\top\mathbf{y} = \hat{\mathbf{y}}^\top\hat{\mathbf{y}} + \hat{\mathbf{e}}^\top\hat{\mathbf{e}}, \tag{2}$$

since $2\hat{\mathbf{y}}^\top\hat{\mathbf{e}} = \mathbf{0}$. Equation (2) is the decomposition of $TSS = ESS + SSR$. Thus we see

$$\begin{aligned}ESS &= \hat{\mathbf{y}}^\top\hat{\mathbf{y}} \\ &= \mathbf{y}^\top\mathbf{P}_X^\top\mathbf{P}_X\mathbf{y} = \mathbf{y}^\top\mathbf{P}_X\mathbf{y}\end{aligned}$$

Another interesting estimator is the so-called coefficient of determination given by:

$$R^2 = \frac{ESS}{TSS}$$

Here are some possible exercises.

1. Show algebraically that $\mathbf{M}_X$ and $\mathbf{P}_X$ are idempotent. That is

$$\mathbf{M}_X\mathbf{M}_X = \mathbf{M}_X \quad \text{and} \quad \mathbf{P}_X\mathbf{P}_X = \mathbf{P}_X$$

2. By altering the codes above, show that numerically $\mathbf{M}_X$ and $\mathbf{P}_X$ are idempotent.
3. Show algebraically that $\mathbf{M}_X$ and $\mathbf{P}_X$ annihilate each other. That is $\mathbf{M}_X\mathbf{P}_X = \mathbf{0}$.
4. By altering the codes above, show that numerically $\mathbf{M}_X$ and $\mathbf{P}_X$ annihilate each other.
5. Show algebraically that $2\hat{\mathbf{y}}^\top\hat{\mathbf{e}} = \mathbf{0}$.

9 To Create, Run, and Interpret the Results of a SAS Program

One of the distinguishing features of a Unix computer is that the programs on the computer can communicate with another. Specifically, programs such as SAS can use files created in the pico editor as input programs. Thus, one would write the SAS input program in pico, submit to SAS this input program, and then view the output files from SAS with the pico editor.

This example is to create an input SAS file called `ex2.sas` to enter[1] both quarterly real domestic consumption and income. Also, the data will be printed and plotted. To start, create a new file called `ex2.sas` using pico. At the `frank` prompt enter:

```
*
* ex2.sas is a sample sas program
*;
```

[1] I take the data from Table 5.1 located in Judge, Hill, Griffiths, Lutkepohl, and Lee [1, pg 173].

```
options pagesize=30 linesize=72 nodate;
*
* enter data
*;
data ex2;
input date con in;
cards;
1980 15.30 17.30
1981 19.91 21.91
1982 20.94 23.14
1983 19.66 21.86
1984 21.32 23.72
1985 18.33 20.73
1986 19.59 22.19
1987 21.30 23.90
1988 20.93 23.73
1989 21.64 24.44
1990 21.90 24.90
1991 20.50 23.50
1992 22.83 26.03
1993 23.49 26.69
1994 24.20 27.60
1995 23.05 26.45
1996 24.01 27.61
1997 25.83 29.43
1998 25.15 28.95
1999 25.06 28.86
run;

*
*to print and plot
*;
proc print data=ex2;
proc plot data=ex2;
plot con*date in*date;
run;
```

When you finish typing, save and exit `ex2.sas` using the three steps that precede this example.

To run SAS using `ex2.sas` as an input file, at the `frank` prompt enter:

```
frank $ sas ex2.sas
```

Completion of the program is indicated by the appearance of the `frank` prompt. It should take a few seconds to run the program so please be patient. To view the SAS output files, at the titan prompt enter:

```
frank $ ls
```

Your listing might resemble the following

```
bin        ex2.log    ex2.sas
ex1.txt    ex2.lst    mail
```

It is important to note that `ex2.lst` is the output file containing the data and graph, and `ex2.log` is the log file where SAS puts any error statement. To view `ex2.log`, at `frank` enter:

```
frank $ pico ex2.log
```

You should see on your monitor:

```
1                                                          The SAS System                          07:43
Thursday, August 16, 2001

NOTE: Copyright (c) 1999-2000 by SAS Institute Inc., Cary, NC, USA.
NOTE: SAS (r) Proprietary Software Release 8.1 (TS1M0)
      Licensed to MIDDLE TENNESSEE STATE UNIVERSITY, Site 0028968011.
NOTE: This session is executing on the HP-UX B.11.00 platform.

NOTE: Running on HP Model 9000/800 Serial Number 75635.



This message is contained in the SAS news file, and is presented upon
initialization.  Edit the files "news" in the "misc/base" directory to
display site-specific news and information in the program log.
The command line option "-nonews" will prevent this display.



NOTE: SAS initialization used:
      real time           1.94 seconds
      cpu time            0.15 seconds

1          *
2          * ex2.sas is a sample sas program
3          *;
4          options pagesize=30 linesize=72 nodate;
5          *
6          * enter data
7          *;
8          data ex2;
2                            The SAS System

9          input date con in;
10         cards;

NOTE: The data set WORK.EX2 has 20 observations and 3 variables.
NOTE: DATA statement used:
      real time           0.52 seconds
      cpu time            0.04 seconds


31         run;
32
33         *
34         *to print and plot
35         *;
36         proc print data=ex2;
```

```
NOTE: There were 20 observations read from the data set WORK.EX2.
NOTE: The PROCEDURE PRINT printed page 1.
NOTE: PROCEDURE PRINT used:
      real time           0.68 seconds
      cpu time            0.04 seconds


37         proc plot data=ex2;
38         plot con*date in*date;
39         run;

NOTE: There were 20 observations read from the data set WORK.EX2.
3                            The SAS System

NOTE: The PROCEDURE PLOT printed pages 2-3.
NOTE: PROCEDURE PLOT used:
      real time           0.06 seconds
      cpu time            0.01 seconds


NOTE: SAS Institute Inc., SAS Campus Drive, Cary, NC USA 27513-2414
NOTE: The SAS System used:
      real time           3.44 seconds
      cpu time            0.27 seconds
```

When you finish looking at `ex2.log` save and exit it using the three steps that precede this example. To view `ex2.lst` follow the above procedures. Note if there is no `ex2.lst` then some error might have occurred in your program. Please go back and review `ex2.sas` for any mistakes such as missing ";" or misspelling.

10 More SAS Examples

10.1 Data Manipulation

As you will see later, SAS is a very powerful statistical program. However, researchers often use SAS for data manipulation. The next example does just this. First, two data sets are read in, `macrodriqtr.txt` and `macrodrimth.txt`. The first data set has quarterly macro series (GNP, consumption, etc.) while the second contains monthly macro series (employment, hourly earnings, etc.). Second, the monthly series are converted to quarterly using the `expand` procedure. Finally, the series are combined and dumped to a data file `ex3.dat`.

```
*
* ex3.sas is an example file of how
* to read in data, treat data, then
* dump the data to a file
*;
options pagesize=30 linesize=72 nodate;

*
* define the files where data are stored
*;
filename ex3a '/users/faculty8/sfowler/public_html/data/macrodriqtr.txt';
filename ex3b '/users/faculty8/sfowler/public_html/data/macrodrimth.txt';


*
* enter data set 1, note
* the values are quarterly
```

```
*;
data ex3a;
 infile ex3a missover firstobs=6;
 input DATE:date9. GCDQ GCNQ GCQ GCSQ GEXQ GGEQ GIMQ GNPQ GPIQ GVQ;
 format date yyqc6.;
 label gnpq = "real gnp"
       gcq  = "real consumption"
       gcnq = "real consumption of nondurables"
       gcsq = "real consumption of services"
       gcdq = "real consumption of durables"
       gpiq = "gross private domestic investment"
       gvq  = "change in business inventories"
       ggeq = "govt purchases of goods and services"
       gexq = "exports"
       gimq = "imports";
run;

proc sort;
by date;
run;

*
* enter data set 2, note
* the values are monthly
*;
data ex3b;
 infile ex3b missover firstobs=6;
 input DATE:date9. LHEM LHEMR LHOURS LPHRM LPMHU LPNAG;
 format date monyy.;
 label lhem   = "civilian employment (household survey, 9-3)"
       lhemr  = "total employment (household survey, 9-3 discont.)"
       lhours = "manhours employed per week (household survey 9-10)"
       lphrm  = "average weekly manufacturing hours (establishment survey 9-15)"
       lpmhu  = "total nonag hours (establishment survey 9-16)"
       lpnag  = "total nonag employment (establishment survey 9-13)";
 lhours = 4*lhours;
 run;

*
* sort data by date
*;
proc sort data=ex3b;
by date;
run;

*
* expand monthly data to
* quarterly
*;
proc expand data=ex3b out=ex3c from=month to=qtr;
 id date;
 convert lhours / observed = total;
 convert lpmhu  / observed = total;
 convert lhem   / observed = average;
 convert lpnag  / observed = average;
run;

*
* treat data
*;
data ex3c;
set ex3c;
 hpwh = lhours/lhem;
 hpwe = lpmhu/lpnag;
 date = intnx('qtr','31dec1946'd,_n_);
```

```
 label hpwh = "hours per worker (household survey)"
       hpwe = "hours per worker (establishment survey)";
run;

*
* sort qtrly data by date
*;
data ex3a;
 set ex3a;
 date = intnx('qtr','31dec1944'd,_n_+2);
run;

proc sort;
by date;
run;

*
* merge data sets
*;
data ex3;
 merge ex3a ex3c;
 by date;
run;

*
* change missing value code and
* dump data to a file
*;
filename ex3 '/users/faculty8/sfowler/public_html/data/ex3.dat';

data ex3;
 set ex3;
 if gnpq   = . then gnpq   = -999;
 if gcq    = . then gcq    = -999;
 if gcnq   = . then gcnq   = -999;
 if gcsq   = . then gcsq   = -999;
 if gcdq   = . then gcdq   = -999;
 if gpiq   = . then gpiq   = -999;
 if gvq    = . then gvq    = -999;
 if ggeq   = . then ggeq   = -999;
 if gexq   = . then gexq   = -999;
 if gimq   = . then gimq   = -999;
 if lhours = . then lhours = -999;
 if lhem   = . then lhem   = -999;
 if hpwh   = . then hpwh   = -999;
 file ex3;
 put gnpq gcq gcnq gcsq gcdq gpiq gvq ggeq gexq gimq lhours lhem hpwh;
run;
```

10.2 The Least Squares Problem Using SAS

The most commonly used estimation technique in econometrics is ordinary least squares, or OLS. The OLS problem is to find some k-vector $\boldsymbol{\beta}$ such that the distance between $\mathbf{y}$ and $\mathbf{X}\boldsymbol{\beta}$ is minimized. In terms of linear regression, $\mathbf{y}$ is an n-vector called the regressand and $\mathbf{X} \equiv [\mathbf{x}_1, \ldots, \mathbf{x}_k]$ is an $n \times k$ matrix of regressors. Evidently, minimizing the distance between $\mathbf{y}$ and $\mathbf{X}\boldsymbol{\beta}$ is equivalent to minimizing the square of this distance. That is,

$$\min_{\boldsymbol{\beta}} \sum_{t=1}^{n} (y_t - \mathbf{X}_t \boldsymbol{\beta})^2 = \min_{\boldsymbol{\beta}} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}), \tag{3}$$

where y_t and $\mathbf{X}_t\boldsymbol{\beta}$ denote, respectively, the t^{th} element of the vector $\mathbf{y}$ and t^{th} row of the matrix $\mathbf{X}$. The difference between y_t and $\mathbf{X}_t\boldsymbol{\beta}$ is commonly referred to as a residual, thus equation (3) is called the sum of squared residuals, or SSR. A close parallel to the SSR is the explained sum of squares, ESS. The total sum of squares, or TSS, is $SSR + ESS$.

As an example of the least squares problem using SAS, I take the data from Table 5.1 located in Judge, Hill, Griffiths, Lutkepohl, and Lee [1, pg 173]. It is assumed that consumption is the regressand $\mathbf{y}$ and a constant and income are the regressors $\mathbf{X} \equiv [\mathbf{x}_1, \mathbf{x}_2]$, where $x_{1,t} = 1, \forall t$. Using the pico editor, the following lines of code are entered and saved in a file called `ex4.sas`.

```
*
* ex4.sas is a sas program
* for OLS example
*;
options pagesize=30 linesize=72 nodate;

*
* enter data
*;
data ex4;
input con in;
cards;
15.30 17.30
19.91 21.91
20.94 23.14
19.66 21.86
21.32 23.72
18.33 20.73
19.59 22.19
21.30 23.90
20.93 23.73
21.64 24.44
21.90 24.90
20.50 23.50
22.83 26.03
23.49 26.69
24.20 27.60
23.05 26.45
24.01 27.61
25.83 29.43
25.15 28.95
25.06 28.86
run;

*
* run OLS regression
*;
proc reg data=ex4;
model con = in;
run;
```

Only two lines of code are needed to run the regression. These lines are the `proc reg data=ex4` and `model con = in` statements. It is interesting to note that the intercept does not need to be included in the model statement. Like most statistical programs, SAS assumes the intercept to be there.

After running the program the `ex4.lst` file shows the results:

```
                              The SAS System

Model: MODEL1
Dependent Variable: CON

                           Analysis of Variance

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

 Model            1     124.19011    124.19011     2240.992       0.0001
 Error           18       0.99751      0.05542
 C Total         19     125.18762

     Root MSE       0.23541     R-square       0.9920
     Dep Mean      21.74700     Adj R-sq       0.9916
     C.V.           1.08249

                           Parameter Estimates

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

  INTERCEP   1       1.417838    0.43265101         3.277        0.0042
  IN         1       0.824813    0.01742349        47.339        0.0001
```

The parameter estimates were found to be $\hat{\boldsymbol{\beta}} = [1.417, 0.824]^\top$ and $SSR = 0.997$.

One can question how SAS solved equation (3) to come up with the parameter estimates. The question can be answered through a geometric interpretation and a little bit of linear algebra. Returning to the beginning of this chapter, the regressand $\mathbf{y}$ and the regressors $\mathbf{x}_1$ and $\mathbf{x}_2$ can be thought of as points in n-dimensional Euclidean space, E^n. Assuming the two regressors are linearly independent, they span a 2-dimensional subspace of E^n which I denote $S(\mathbf{X})$. The orthogonal complement of $S(\mathbf{X})$ in E^n, which is denoted $S^{\perp}(\mathbf{X})$, is the set of all points $\mathbf{w}$ in E^n, such that for all $\mathbf{z}$ in E^n, $\mathbf{w}^\top\mathbf{x} = \mathbf{0}$. Figure 1 illustrates these concepts for the case $n = 2$ and $k = 1$. Notice that $S(\mathbf{X})$ and $S^{\perp}(\mathbf{X})$ form a right angle to each other.

The geometry of OLS is illustrated in Figure 2. The regressand is now shown as the vector $\mathbf{y}$. The vector $\mathbf{X}\hat{\boldsymbol{\beta}}$ in $S(\mathbf{X})$ is to be made as close to $\mathbf{y}$ as possible. It is evident that the line joining $\mathbf{y}$ and $\mathbf{X}\hat{\boldsymbol{\beta}}$ must form a right angle with $S(\mathbf{X})$ at $\mathbf{X}\hat{\boldsymbol{\beta}}$ for it to be the closest point. This line is simply the vector $\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}$. Since $\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}$ must form a right angle with $S(X)$, $\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}$ must be orthogonal to all columns of $\mathbf{X}$. This is written algebraically as

$$\mathbf{X}^\top\left(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\right) = 0,$$

which are the first order conditions for equation (3). From these equations we obtain the solutions

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{y}. \tag{4}$$

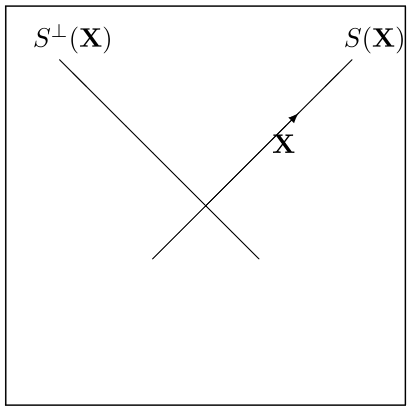

Figure 1: The Spaces $S(\mathbf{X})$ and $S^{\perp}(\mathbf{X})$

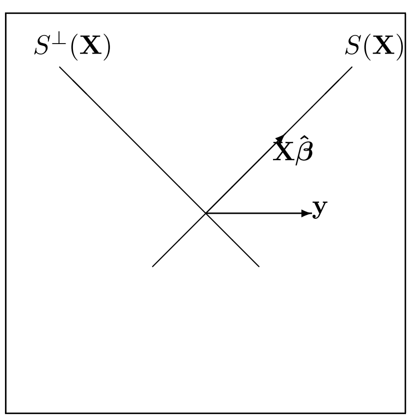

Figure 2: The Projection of $\mathbf{y}$ onto $S(\mathbf{X})$

Using the estimator for $\boldsymbol{\beta}$, equation (4), we can write the estimator for the residuals as

$$\begin{aligned}\hat{\mathbf{e}} &= \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{y} - \mathbf{X}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{y} \\ &= \left(\mathbf{I} - \mathbf{P}_X\right)\mathbf{y} = \mathbf{M}_X\mathbf{y}\end{aligned}$$

where $\mathbf{P}_X = \mathbf{X}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top$ and $\mathbf{M}_X = (\mathbf{I} - \mathbf{P}_X)$. Then, the estimated SSR is

$$\mathbf{y}^\top\mathbf{M}_X^\top\mathbf{M}_X\mathbf{y} = \mathbf{y}^\top\mathbf{M}_X\mathbf{y}.$$

As a basis for deriving ESS let us write the least squares prediction equation as

$$\mathbf{y} = \mathbf{X}\hat{\boldsymbol{\beta}} + \hat{\mathbf{e}} = \hat{\mathbf{y}} + \hat{\mathbf{e}}.$$

Squaring both sides

$$\mathbf{y}^\top\mathbf{y} = \hat{\mathbf{y}}^\top\hat{\mathbf{y}} + \hat{\mathbf{e}}^\top\hat{\mathbf{e}}, \tag{5}$$

since $2\hat{\mathbf{y}}^\top\hat{\mathbf{e}} = \mathbf{0}$. Equation (5) is the decomposition of $TSS = ESS + SSR$. Thus we see

$$\begin{aligned}ESS &= \hat{\mathbf{y}}^\top\hat{\mathbf{y}} \\ &= \mathbf{y}^\top\mathbf{P}_X^\top\mathbf{P}_X\mathbf{y} = \mathbf{y}^\top\mathbf{P}_X\mathbf{y}\end{aligned}$$

Another interesting estimator is the so-called coefficient of determination given by:

$$R^2 = \frac{ESS}{TSS}$$

The formulas for the estimators derived above can be used to test whether our results are the same as in SAS. To do this we use the `proc iml` statement. Enter the following lines of code to the previous file `ex4.sas`.

```
*
* now lets use the short-hand
* linear algebra notation
*;
proc iml;

start appex4a;

use ex4;
read all var{con} into y;
read all var{in} into x2;
t  = nrow(y);
x1 = j(t,1,1);
x  = x1||x2;

* define estimators;
beta = inv(x‘*x)*x‘*y;
px   = x*inv(x‘*x)*x‘;
mx   = i(ncol(px))-px;
ssr  = y‘*mx*y;
ess  = y‘*px*y;
tss  = ssr + ess;
rsq  = ess/tss;
```

```
print / "the estimates are", beta ssr ess, tss rsq;

finish;
run appex4a;
```

The output file ex4.lst shows the following:

```
                              The SAS System

                           the estimates are

                          BETA       SSR       ESS
                     1.4178381 0.9975145 9582.8303
                     0.8248128


                               TSS       RSQ
                          9583.8278 0.9998959
```

Though the estimates for the coefficients and SSR are the same, the estimates for ESS, TSS, and R^2 are different. Why is this? This is because most statistical programs report the ESS centered about the mean of $\mathbf{y}^2$. To do this we add the following lines of code to the program:

```
* now lets do centered sums of squares;
start appex4b;

ybar = x1`*y/t;
cess = ess - t*ybar**2;
ctss = cess + ssr;
crsq = cess/ctss;

print / "the centered estimates are", ybar cess ctss, crsq;

finish;
run appex4b;
```

The output reveals the same as SAS gave us earlier:

```
                              The SAS System

                      the centered estimates are
```

```
    YBAR      CESS      CTSS
  21.747 124.19011 125.18762


               CRSQ
           0.9920318
```

Here are some possible exercises.

1. Show algebraically that $\mathbf{M}_X$ and $\mathbf{P}_X$ are idempotent. That is

$$\mathbf{M}_X\mathbf{M}_X = \mathbf{M}_X \quad \text{and} \quad \mathbf{P}_X\mathbf{P}_X = \mathbf{P}_X$$

2. By altering the codes above, show that numerically $\mathbf{M}_X$ and $\mathbf{P}_X$ are idempotent.
3. Show algebraically that $\mathbf{M}_X$ and $\mathbf{P}_X$ annihilate each other. That is $\mathbf{M}_X\mathbf{P}_X = \mathbf{0}$.
4. By altering the codes above, show that numerically $\mathbf{M}_X$ and $\mathbf{P}_X$ annihilate each other.
5. Show algebraically that $2\hat{\mathbf{y}}^\top\hat{\mathbf{e}} = \mathbf{0}$.

11 How to Log off Your `frank` Account

To log off your `frank` account type

```
frank $ exit
```

References

[1] Judge, G., Hill, R., Griffiths, W., and T. Lee. *Introduction to the Theory and Practice of Econometrics.* New York: John Wiley and Sons, 1988.