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 16, 2001

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 GAMS, Maple, Mathematica, SAS, and RATS all located on the `frank` server. These programs differ by the function they are intended to perform. For example, GAMS is used mostly for numeric computations while Maple and Mathematica are used primarily for symbolic computations. SAS and RATS 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 GAMS will be written in a Unix text editor using the GAMS syntax. The program is then submitted to GAMS. 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 SAS, RATS, 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
- How to create, run, and interpret the results of a SAS program
- More SAS Examples
  1. Data Manipulation
  2. The Least Squares Problem Using SAS
- How to create, run, and interpret the results of a RATS program

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

- More RATS Examples
  1. Preliminaries
  2. The Model
  3. Implementation
- How to create, run, and interpret the results of a Maple program
- 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. For example, students who have e-mail accounts use the `frank` server. 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 an account. To get an account: visit or call the Office of Information Technology (OIT) help desk:

Office of Information Technology
Help Desk
(615) 898-5345
`help@frank.edu`

You will be given a username and password, which will be needed for logging into your account.

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 called Telnet. Usually, this program is located in the communication program group. The name for the icon in the program group is sometimes `TELNET`, `QvtTerm`, `BW VT220`, or `Bw220w`. Additionally, many computers with Microsoft Windows come with a Telnet program. Thus, one may be able to connect to the `frank` server at home. Some places to log into the `frank` are:

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

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

```
frank.mtsu.edu
```

If the program does not prompt you then you must find the open remote option in the telnet menu. Once you have opened a link to the `frank` you should see on your monitor and enter:

```
Unauthorized use is prohibited by the Computer Crimes Act of 1993.

frank [HP-UX]

login: enter your username here
password: enter your password here
```

If you have logged on correctly you should see on your monitor the `frank` prompt:

```
frank $
```

4 Basic Unix Commands

Press the <`return`> key after each of the following commands in Table 1 on page 3. Remember to use the lower case letters unless otherwise indicated.

`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

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
```

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
- Pico is similar to Pine (E-mail)

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 ˆ`x`. 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 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
*;
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;
```

[1] I take the data from Table 5.1 located in Judge, Hill, Griffiths, Lutkepohl, and Lee [7, pg 173].

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.

7 More SAS Examples

7.1 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;
```

7.2 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,

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

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 (1) 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 [7, 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:
```

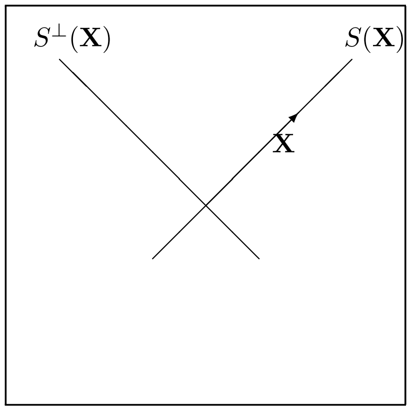


Figure 1: The Spaces $S(\mathbf{X})$ and $S^{\perp}(\mathbf{X})$

```
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 (1) 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 (1). From these equations we obtain the solutions

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^{\top}\mathbf{X}\right)^{-1}\mathbf{X}^{\top}\mathbf{y}. \tag{2}$$

Using the estimator for $\boldsymbol{\beta}$, equation (2), we can write the estimator for the residuals as

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

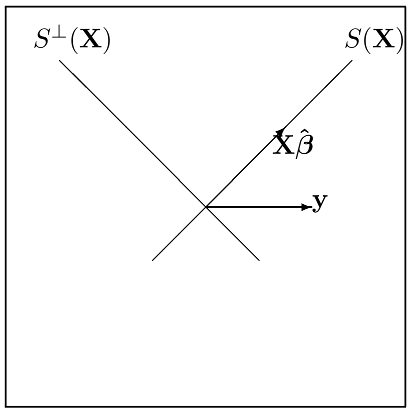


Figure 2: The Projection of $\mathbf{y}$ onto $S(\mathbf{X})$

$$= \quad (\mathbf{I} - \mathbf{P}_X)\,\mathbf{y} = \mathbf{M}_X\mathbf{y}$$

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

$$\mathbf{y}^\top\mathbf{y} = \hat{\mathbf{y}}^\top\hat{\mathbf{y}} + \hat{\mathbf{e}}^\top\hat{\mathbf{e}}, \tag{3}$$

since $2\hat{\mathbf{y}}^\top\hat{\mathbf{e}} = \mathbf{0}$. Equation (3) 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 $2\hat{\mathbf{y}}^\top\hat{\mathbf{e}} = \mathbf{0}$.

8 To Create, Run, and Interpret the Results of a RATS Program

RATS is a macroeconometric time series package. Macroeconomists are interested in time series since the behavior of an economy can only be truly studied by making observations over many periods. Most macroeconomic time series have trends. These trends are removed since they offer little information. Removal of trends is called filtering. A commonly used trend is the Hodrick-Prescott (HP) filter. The HP filter solves for a trend τ_t such that it minimizes the following loss function:

$$\sum_{t=1}^{T}(x_t - \tau_t)^2 + 1600\sum_{t=2}^{T-1}[(\tau_{t+1} - \tau_t) - (\tau_t - \tau_{t-1})]^2$$

Once an estimate of the trend component is found, denoted $\hat{\tau}_t$, the deviation $x_t - \hat{\tau}_t$ is then computed. It is this variable, the cyclical component of the variable x_t or the deviation from the trend, that is of interest to the macroeconomist since it represents the movements of the variable over the "business cycle."

Then, statistics such as standard deviations and cross-correlations are computed on the cyclical variables. The standard deviation, computed as $\sqrt{\sum_{t=1}^{T}(x_t - \bar{x})^2/(T-1)}$, gives the relative volatility of any variable x when compared to an aggregate variable such as GNP. The cross-correlation coefficient of a variable y_t with a variable x_{t-k} is computed as

$$\rho(k) = \frac{\sum_{t=1}^{T}(y_t - \bar{y})(x_{t-k} - \bar{x})}{\sqrt{\sum_{t=1}^{T}(y_t - \bar{y})^2 \sum_{t=1}^{T}(x_t - \bar{x})^2}}.$$

When y is real GNP and $k = 0$ a number close to one indicates x is *procyclical.* A number close to one but of opposite sign indicate x is *countercyclical.* A number close to zero indicates x is *uncorrelated* with the cycle of GNP. Additionally, when $|\rho(k < 0)| > |\rho(0)|$ we say x *lags* the cycle. When $|\rho(k > 0)| > |\rho(0)|$ we say x *leads* the cycle.

In this first example we will create a RATS input file that will read in the data series created in `ex3.sas`. Then, simple time series procedures are performed such as HP filtering, computing cross-correlations, and plotting the data. At the prompt, type and enter:

```
frank $ pico ex5.prg
```

Once in the pico editor, enter and save the following lines of RATS code:

```
envo noecho
open output ex5.out
calendar 1945 3 4
allocate 0 2001:4

source(noecho) /opt/rats/examples/hpfilter.src

*
* open data
*
open data /users/faculty8/sfowler/public_html/data/ex3.dat
data(unit=data, org=obs, missing=-999) 1945:3 2000:4 $
gnpq gcq gcnq gcsq gcdq gpiq gvq ggeq gexq gimq lhours lhem hpwh

*
* to set beginning dates
*
compute begdate = (1959:1)
compute enddate = (1991:2)

*
* compute logs
*
```

```
set gnpq   = log(gnpq)
set gcq    = log(gcq)
set gpiq   = log(gpiq)
set ggeq   = log(ggeq)
set gexq   = log(gexq)
set gimq   = log(gimq)
set lhours = log(lhours)
set lhem   = log(lhem)
set hpwh   = log(hpwh)

*
* compute the hp filter growth components
*
@hpfilter gnpq   begdate enddate ggnpq
@hpfilter gcq    begdate enddate ggcq
@hpfilter gpiq   begdate enddate ggpiq
@hpfilter ggeq   begdate enddate gggeq
@hpfilter gexq   begdate enddate ggexq
@hpfilter gimq   begdate enddate ggimq
@hpfilter lhours begdate enddate glhours
@hpfilter lhem   begdate enddate glhem
@hpfilter hpwh   begdate enddate ghpwh

*
* compute cyclical components
*
set cgnpq   = gnpq   - ggnpq
set cgcq    = gcq    - ggcq
set cgpiq   = gpiq   - ggpiq
set cggeq   = ggeq   - gggeq
set cgexq   = gexq   - ggexq
set cgimq   = gimq   - ggimq
set clhours = lhours - glhours
set clhem   = lhem   - glhem
set chpwh   = hpwh   - ghpwh

*
* compute simple statistics
*
statistics cgnpq   begdate enddate
statistics cgcq    begdate enddate
statistics cgpiq   begdate enddate
statistics cggeq   begdate enddate
statistics cgexq   begdate enddate
statistics cgimq   begdate enddate
statistics clhours begdate enddate
statistics clhem   begdate enddate
statistics chpwh   begdate enddate

*
* compute cross-correlations
*
cross cgnpq cgnpq   begdate enddate
cross cgnpq cgcq    begdate enddate
cross cgnpq cgpiq   begdate enddate
cross cgnpq cggeq   begdate enddate
cross cgnpq cgexq   begdate enddate
cross cgnpq cgimq   begdate enddate
cross cgnpq clhours begdate enddate
cross cgnpq clhem   begdate enddate
cross cgnpq chpwh   begdate enddate

*
* plot data
*
```

```
set res = $
   t>=1953:2.and.t<=1954:2.or. $
   t>=1957:3.and.t<=1958:2.or. $
   t>=1960:2.and.t<=1961:1.or. $
   t>=1969:4.and.t<=1970:4.or. $
   t>=1973:4.and.t<=1975:1.or. $
   t>=1980:1.and.t<=1980:3.or. $
   t>=1981:3.and.t<=1982:4.or. $
   t>=1990:3.and.t<=1991:1

open plot ex5a.gsp
spgraph(vfields=2,hfields=1)
graph(shading=res,header='Real GNP and Growth Component') 2
# gnpq  begdate enddate
# ggnpq begdate enddate
graph(shading=res,header='Cyclical GNP') 1
# cgnpq  begdate enddate
spgraph(done)

open plot ex5b.gsp
spgraph(vfields=2,hfields=1)
graph(shading=res,header='Total Hours and Growth Component') 2
# lhours  begdate enddate
# glhours begdate enddate
graph(shading=res,header='Cyclical Hours') 1
# clhours  begdate enddate
spgraph(done)

open plot ex5c.gsp
spgraph(vfields=2,hfields=1)
graph(shading=res,header='Cyclical Employment',min=-.03,max=.03) 1
# clhem  begdate enddate
graph(shading=res,header='Cyclical Hours per Worker',min=-.03,max=.03) 1
# chpwh  begdate enddate
spgraph(done)

* /opt/rats/bin/rgf2pst ex5a.gsp ex5a.ps
* /opt/rats/bin/rgf2pst ex5b.gsp ex5b.ps
* /opt/rats/bin/rgf2pst ex5c.gsp ex5c.ps

end
```

To run the program enter:

```
frank $ rats ex5.prg
```

RATS creates an output file `ex5.out`. In this file the results of the program are given:

```
Statistics on Series CGNPQ
Quarterly Data From 1959:01 To 1991:02
Observations   130
Sample Mean    -0.0000000000           Variance              0.000313
Standard Error  0.0177008573           SE of Sample Mean     0.001552
t-Statistic       -0.00000             Signif Level (Mean=0) 1.00000000
Skewness          -0.38321             Signif Level (Sk=0)   0.07787154
Kurtosis           0.05747             Signif Level (Ku=0)   0.89642105
```

```
Statistics on Series CGCQ
Quarterly Data From 1959:01 To 1991:02
Observations   130
Sample Mean    -0.0000000000            Variance                 0.000199
Standard Error  0.0141058669            SE of Sample Mean        0.001237
t-Statistic       -0.00000              Signif Level (Mean=0) 1.00000000
Skewness          -0.22273              Signif Level (Sk=0)   0.30546421
Kurtosis           0.05554              Signif Level (Ku=0)   0.89989387
```

```
Statistics on Series CGPIQ
Quarterly Data From 1959:01 To 1991:02
Observations   130
Sample Mean    -0.0000000000            Variance                 0.005935
Standard Error  0.0770370119            SE of Sample Mean        0.006757
t-Statistic       -0.00000              Signif Level (Mean=0) 1.00000000
Skewness          -0.78460              Signif Level (Sk=0)   0.00030627
Kurtosis           0.90513              Signif Level (Ku=0)   0.04034237
```

```
Statistics on Series CGGEQ
Quarterly Data From 1959:01 To 1991:02
Observations   130
Sample Mean    -0.0000000000            Variance                 0.000300
Standard Error  0.0173341155            SE of Sample Mean        0.001520
t-Statistic       -0.00000              Signif Level (Mean=0) 1.00000000
Skewness           0.09026              Signif Level (Sk=0)   0.67794135
Kurtosis           0.39441              Signif Level (Ku=0)   0.37165801
```

```
Statistics on Series CGEXQ
Quarterly Data From 1959:01 To 1991:02
Observations   130
Sample Mean    -0.0000000000            Variance                 0.002139
Standard Error  0.0462447059            SE of Sample Mean        0.004056
t-Statistic       -0.00000              Signif Level (Mean=0) 1.00000000
Skewness          -0.29403              Signif Level (Sk=0)   0.17611577
Kurtosis           0.13185              Signif Level (Ku=0)   0.76519661
```

```
Statistics on Series CGIMQ
Quarterly Data From 1959:01 To 1991:02
Observations   130
Sample Mean    -0.0000000000            Variance                 0.002882
Standard Error  0.0536816699            SE of Sample Mean        0.004708
t-Statistic       -0.00000              Signif Level (Mean=0) 1.00000000
Skewness          -1.04011              Signif Level (Sk=0)   0.00000171
Kurtosis           1.76215              Signif Level (Ku=0)   0.00006567
```

```
Statistics on Series CLHOURS
Quarterly Data From 1959:01 To 1991:02
Observations   130
Sample Mean    -0.0000000000            Variance                 0.000203
Standard Error  0.0142615865            SE of Sample Mean        0.001251
t-Statistic       -0.00000              Signif Level (Mean=0) 1.00000000
Skewness          -0.37957              Signif Level (Sk=0)   0.08073938
```

```
Kurtosis           0.25680              Signif Level (Ku=0)   0.56078824



Statistics on Series CLHEM
Quarterly Data From 1959:01 To 1991:02
Observations   130
Sample Mean    -0.0000000000            Variance                0.000105
Standard Error  0.0102634258            SE of Sample Mean       0.000900
t-Statistic       -0.00000              Signif Level (Mean=0) 1.00000000
Skewness          -0.48349              Signif Level (Sk=0)   0.02611346
Kurtosis           0.38249              Signif Level (Ku=0)   0.38627676



Statistics on Series CHPWH
Quarterly Data From 1959:01 To 1991:02
Observations   130
Sample Mean    0.00000000000            Variance           2.921298e-05
Standard Error 0.00540490353            SE of Sample Mean       0.000474
t-Statistic        0.00000              Signif Level (Mean=0) 1.00000000
Skewness          -0.17775              Signif Level (Sk=0)   0.41345064
Kurtosis           0.42569              Signif Level (Ku=0)   0.33492660


Cross Correlations of Series CGNPQ and CGNPQ
Quarterly Data From 1959:01 To 1991:02
   -32:  0.0550898  0.1184379  0.1383661  0.1677698  0.1763284  0.1641785
   -26:  0.1493195  0.1439446  0.1374692  0.1180071  0.1085986  0.0857859
   -20:  0.0590593 -0.0047808 -0.0594137 -0.1274535 -0.1958610 -0.2868021
   -14: -0.3447004 -0.3774779 -0.3906004 -0.3726468 -0.3612697 -0.3537516
    -8: -0.3340565 -0.2742525 -0.1615015 -0.0086449  0.1942469  0.4129138
    -2:  0.6439169  0.8511612  1.0000000  0.8511612  0.6439169  0.4129138
     4:  0.1942469 -0.0086449 -0.1615015 -0.2742525 -0.3340565 -0.3537516
    10: -0.3612697 -0.3726468 -0.3906004 -0.3774779 -0.3447004 -0.2868021
    16: -0.1958610 -0.1274535 -0.0594137 -0.0047808  0.0590593  0.0857859
    22:  0.1085986  0.1180071  0.1374692  0.1439446  0.1493195  0.1641785
    28:  0.1763284  0.1677698  0.1383661  0.1184379  0.0550898
Cross Correlations of Series CGNPQ and CGCQ
Quarterly Data From 1959:01 To 1991:02
   -32: -0.0473870  0.0100244  0.0633968  0.1256339  0.1620080  0.1963086
   -26:  0.1930661  0.1653767  0.1545130  0.1357836  0.1172317  0.0973566
   -20:  0.0768394  0.0319883  0.0021832 -0.0504608 -0.0944249 -0.1924404
   -14: -0.2733973 -0.3345584 -0.3847736 -0.4070141 -0.4091831 -0.4177911
    -8: -0.4361406 -0.4007590 -0.3040879 -0.1650669  0.0118996  0.2343378
    -2:  0.4695124  0.6928992  0.8720218  0.8623088  0.7519813  0.5935086
     4:  0.4371594  0.2454196  0.0845873 -0.0509904 -0.1676314 -0.2367771
    10: -0.2773005 -0.3027345 -0.3282854 -0.3441047 -0.3703528 -0.3447225
    16: -0.3066334 -0.2744753 -0.2218508 -0.1678048 -0.1094518 -0.0559124
    22:  0.0028348  0.0359094  0.0670351  0.1041541  0.1456729  0.1627789
    28:  0.1955052  0.2116901  0.1930350  0.1998454  0.1745313
Cross Correlations of Series CGNPQ and CGPIQ
Quarterly Data From 1959:01 To 1991:02
   -32:  0.0313072  0.0842721  0.0758053  0.0719462  0.0783600  0.0736255
   -26:  0.1001107  0.1547706  0.1998696  0.2330999  0.2708038  0.2742052
   -20:  0.2517694  0.1708358  0.0867879 -0.0034660 -0.1062338 -0.1932603
   -14: -0.2641307 -0.3185175 -0.3765922 -0.4043597 -0.4337238 -0.4811861
    -8: -0.4910923 -0.4449402 -0.3342659 -0.1856921  0.0434900  0.2808031
    -2:  0.5282719  0.7515083  0.9116070  0.7705203  0.5855879  0.3929814
     4:  0.2090874  0.0623999 -0.0379475 -0.1180298 -0.1532978 -0.1890765
    10: -0.2222097 -0.2651467 -0.3159152 -0.3478595 -0.3270736 -0.2885634
    16: -0.2137704 -0.1611710 -0.1041313 -0.0719436 -0.0226675 -0.0022306
    22:  0.0106746  0.0292526  0.0906230  0.1454643  0.1776824  0.2289954
    28:  0.2623558  0.2595288  0.2472462  0.2209297  0.1420155
Cross Correlations of Series CGNPQ and CGGEQ
```

```
Quarterly Data From 1959:01 To 1991:02
   -32:  0.1219785  0.1244362  0.0998125  0.1255193  0.1366481  0.1042447
   -26:  0.0656905  0.0141304 -0.0461838 -0.1197992 -0.1864918 -0.2573965
   -20: -0.3013785 -0.3235654 -0.3220817 -0.3060553 -0.2760225 -0.2509676
   -14: -0.2030229 -0.1464758 -0.0483725  0.0620879  0.1425734  0.2118623
    -8:  0.2923578  0.3359127  0.3238893  0.3231027  0.2868685  0.2199584
    -2:  0.1990169  0.2025753  0.1821066  0.1071173  0.0474796 -0.0044400
     4: -0.0530554 -0.0967536 -0.1354997 -0.1847463 -0.1997587 -0.1571417
    10: -0.1333307 -0.1324665 -0.1404821 -0.1208370 -0.0938521 -0.0559034
    16:  0.0024425  0.0422412  0.0734835  0.1187839  0.1588636  0.1707145
    22:  0.1607299  0.1361698  0.1069774  0.0540145 -0.0083840 -0.0681242
    28: -0.1794840 -0.2840632 -0.3717532 -0.4372448 -0.4711428
Cross Correlations of Series CGNPQ and CGEXQ
Quarterly Data From 1959:01 To 1991:02
   -32:  0.2568064  0.2615529  0.2434342  0.1921374  0.1379700  0.0570082
   -26: -0.0082097 -0.0635539 -0.1178536 -0.1779053 -0.2055457 -0.1867614
   -20: -0.2220245 -0.1973782 -0.1861083 -0.1546981 -0.1605371 -0.1753719
   -14: -0.1665615 -0.1523113 -0.1117091 -0.0213709  0.0596678  0.1253949
    -8:  0.1840296  0.2622339  0.3256237  0.4043616  0.4582851  0.4535594
    -2:  0.4450038  0.4211622  0.2834083  0.0699322 -0.1480413 -0.3065282
     4: -0.3983396 -0.4355316 -0.4688476 -0.4104038 -0.4232784 -0.3337187
    10: -0.2256798 -0.1227700 -0.0867380 -0.0112481  0.0708844  0.1100277
    16:  0.1506669  0.1789594  0.2054383  0.2118843  0.2383949  0.2215270
    22:  0.1795203  0.1500343  0.0451157 -0.0401544 -0.0958320 -0.0726889
    28: -0.0840995 -0.0395813  0.0182452  0.0439345  0.0515510
Cross Correlations of Series CGNPQ and CGIMQ
Quarterly Data From 1959:01 To 1991:02
   -32: -0.0019206  0.0276140  0.0250230  0.0206553  0.0603631  0.0834536
   -26:  0.1360642  0.1811308  0.2178710  0.2363708  0.2466525  0.2537376
   -20:  0.1901472  0.1280489  0.0588478  0.0147705 -0.0648778 -0.1158531
   -14: -0.1727255 -0.2473367 -0.3318313 -0.3651847 -0.3920537 -0.4738994
    -8: -0.5106721 -0.4533761 -0.3500027 -0.1892881  0.0340397  0.2324834
    -2:  0.4547853  0.6751638  0.7419428  0.6695106  0.5140212  0.3889236
     4:  0.2791274  0.1991277  0.1215301  0.0667907 -0.0624824 -0.1261773
    10: -0.1657818 -0.2110001 -0.2932376 -0.3651403 -0.3633402 -0.3348857
    16: -0.2838925 -0.2422721 -0.1826963 -0.1499064 -0.0946005 -0.0436060
    22: -0.0269784  0.0043189  0.0467997  0.1210467  0.1550961  0.2269228
    28:  0.2152345  0.1984398  0.1981687  0.1613399  0.0936323
Cross Correlations of Series CGNPQ and CLHOURS
Quarterly Data From 1959:01 To 1991:02
   -32:  0.1233538  0.1753943  0.1648826  0.1361559  0.1108131  0.1113555
   -26:  0.1104855  0.1300200  0.1436724  0.1519877  0.1390284  0.1112349
   -20:  0.0373510 -0.0731581 -0.1613700 -0.2245935 -0.2959220 -0.3480426
   -14: -0.3839280 -0.4178044 -0.4303634 -0.3945304 -0.3580261 -0.3243272
    -8: -0.2666396 -0.1293796  0.0377968  0.2249718  0.4073519  0.5913419
    -2:  0.7295511  0.8378594  0.8514769  0.6962361  0.4879791  0.2987715
     4:  0.1044542 -0.0543673 -0.1830286 -0.2721885 -0.3726713 -0.4142581
    10: -0.4183005 -0.4027624 -0.4175299 -0.3729974 -0.3049646 -0.2312531
    16: -0.1711102 -0.1092559 -0.0474865  0.0280665  0.0822218  0.1256746
    22:  0.1452718  0.1430915  0.1305329  0.1413055  0.1398462  0.1378428
    28:  0.1119674  0.1126205  0.0978925  0.0879996  0.0250450
Cross Correlations of Series CGNPQ and CLHEM
Quarterly Data From 1959:01 To 1991:02
   -32:  0.1558976  0.1878420  0.1661014  0.1212425  0.0827147  0.0778821
   -26:  0.0838252  0.1032428  0.1192102  0.1182743  0.1040521  0.0807706
   -20:  0.0235239 -0.0801979 -0.1684729 -0.2247525 -0.2898359 -0.3450836
   -14: -0.3833831 -0.4085453 -0.4132807 -0.3733157 -0.3267999 -0.2819813
    -8: -0.2033196 -0.0542962  0.1199497  0.3084228  0.4921401  0.6618730
    -2:  0.7879612  0.8643073  0.8243694  0.6329684  0.4022274  0.1892430
     4:  0.0040739 -0.1406372 -0.2458612 -0.3320974 -0.3992054 -0.4342383
    10: -0.4505768 -0.4319665 -0.4242491 -0.3783815 -0.3087876 -0.2239402
    16: -0.1552189 -0.0834653 -0.0201668  0.0451348  0.0973730  0.1446839
    22:  0.1710389  0.1672336  0.1575647  0.1594976  0.1581236  0.1566760
    28:  0.1414738  0.1199747  0.1020639  0.0853372  0.0239615
Cross Correlations of Series CGNPQ and CHPWH
```

```
Quarterly Data From 1959:01 To 1991:02
   -32:  0.0294505  0.1061070  0.1196540  0.1290375  0.1353279  0.1459358
   -26:  0.1323546  0.1470270  0.1527302  0.1764481  0.1692602  0.1401326
   -20:  0.0538859 -0.0407491 -0.1058822 -0.1658363 -0.2304589 -0.2630757
   -14: -0.2850372 -0.3266441 -0.3507905 -0.3321300 -0.3241378 -0.3203251
    -8: -0.3174798 -0.2382822 -0.1280415  0.0079521  0.1403247  0.3035002
    -2:  0.4287542  0.5695662  0.6813364  0.6351652  0.5238067  0.4289946
     4:  0.2678810  0.1236018 -0.0160779 -0.0875839 -0.2252896 -0.2684979
    10: -0.2481391 -0.2424788 -0.2960995 -0.2656929 -0.2183315 -0.1849509
    16: -0.1567514 -0.1297937 -0.0870048 -0.0116498  0.0320511  0.0568681
    22:  0.0585324  0.0600051  0.0452279  0.0699827  0.0687411  0.0662037
    28:  0.0267956  0.0693436  0.0644927  0.0701515  0.0205840


Normal Completion
```

From the output and figures we can notice a few stylized facts:

1. Real GNP (output) grows at a steady constant rate.
2. The consumption component of GNP is smoother than investment as measured by the standard deviation.
3. Employment fluctuates more than hours per worker indicating that the majority of changes in total hours is caused by employment volatility and not hours per worker.
4. The investment component of GNP is highly procyclical.
5. The consumption, export, and import components of GNP are procyclical. Government expenditures are uncorrelated with GNP.
6. Employment lags the cycle.

Figure 3:

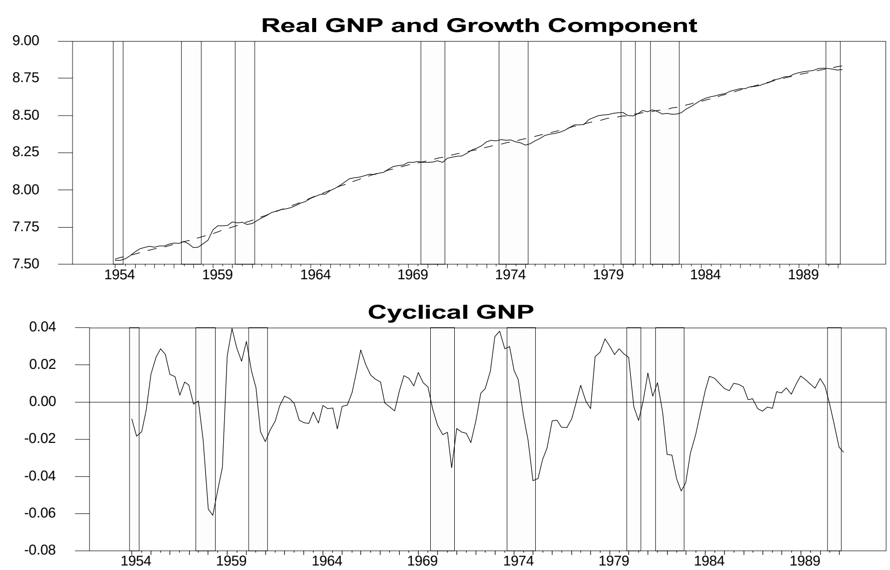


Figure 4:

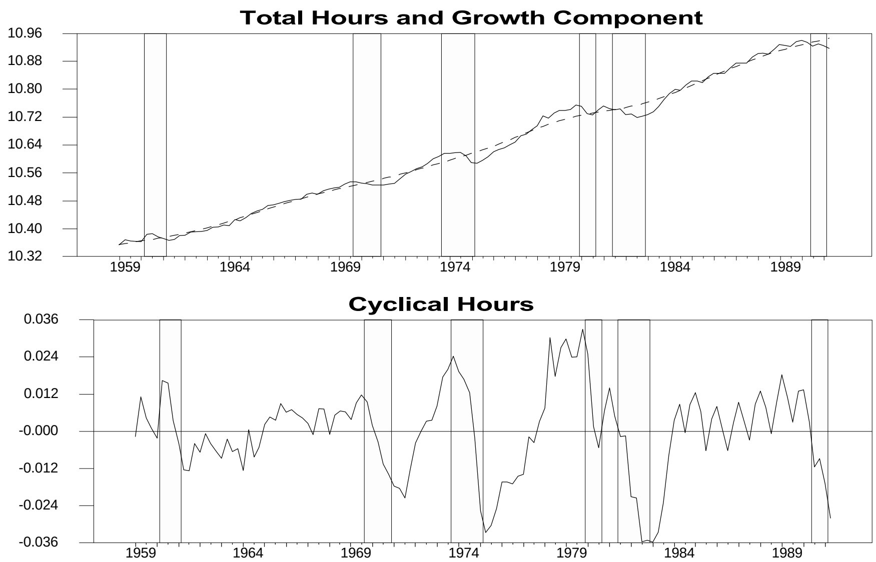
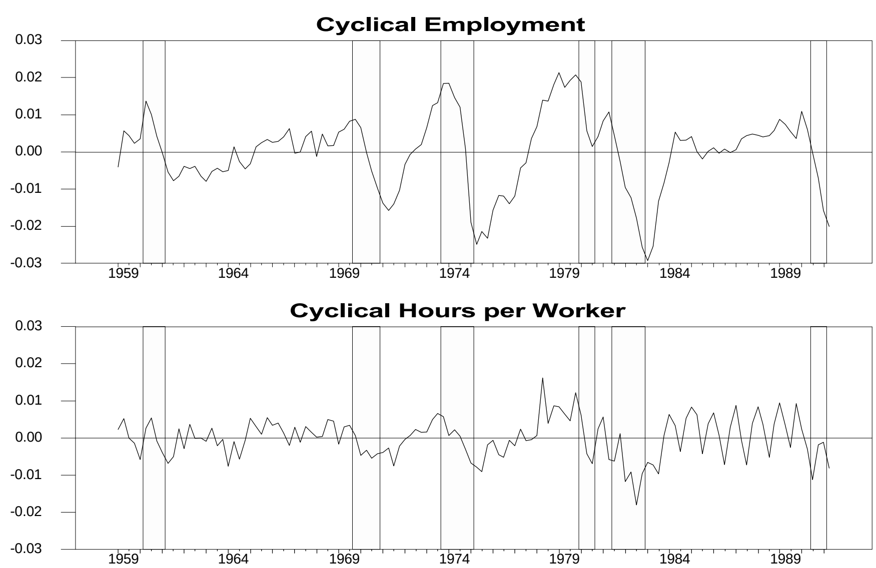


Figure 5:



9 More RATS Examples

In this section we do more than download and plot data. Specifically, we attempt to replicate the results of Hansen and Singleton [5] by using the *generalized method of moments (GMM)* procedure in RATS. This section is organized into three subsections. First, subsection 9.1 presents necessary preliminaries on GMM for the understanding of how the RATS GMM procedure works. Second, subsection 9.2 presents the formalized model to be estimated from Hansen and Singleton [5]. Finally, subsection 9.3 shows how to implement the GMM estimation in RATS.

9.1 Preliminaries

GMM estimators, see Hansen [4], are produced by RATS from population orthogonality conditions that are expressed as

$$\begin{aligned} G(\beta_0) &= E\left[h_t(\beta_0) \otimes z_t\right] \\ &= 0, \end{aligned}$$

where $h_t(\beta_0) = (h_{1,t}(\beta_0), \ldots, h_{n,t}(\beta_0))'$ is a n-dimensional vector of residuals at time t, $z_t = (z_{1,t}, \ldots, z_{r,t})'$ is a r-dimensional vector of instruments at time t, $\beta_0 = (\beta_{0,1}, \ldots, \beta_{0,l})'$ is a l-dimensional vector of coefficients, and $E(\cdot)$ is the unconditional expectations operator.

The GMM estimator of β_0, denoted β_T, is derived by using a sample counterpart of the population orthogonality condition. Define the sample moment:

$$G_T(\beta) = \frac{1}{T}\sum_t h_t(\beta) \otimes z_t$$

and suppose that a law of large numbers can be applied to $G_T(\beta)$ so that it converges to its population mean for all β with probability one (or almost surely):

$$\lim_{T\to\infty} \frac{1}{T}\sum_t h_t(\beta) \otimes z_t = E\left[h_t(\beta) \otimes z_t\right] \quad \text{a.s.}$$

The GMM estimator is chosen to set the sample counterpart of the population orthogonality conditions as close to zero as possible by minimizing the quadratic form:

$$J_T(\beta) = G_T(\beta)' W_T G_T(\beta) \tag{4}$$

with respect to β. In equation (4), W_T is a positive definite weighting matrix which converges in probability to a positive definite matrix W_0:

$$\lim_{T\to\infty} W_T = W_0.$$

Under some regularity conditions, the GMM estimator β_T is a consistent estimator of β_0, see Hansen [4]. The nonlinear GMM estimators are discussed by Gallant [3].

Under the regularity conditions, $\sqrt{T}(\beta_T - \beta_0)$ is asymptotically normally distributed with mean zero and covariance matrix:

$$\Lambda = (D_0' W_0 D_0)^{-1} D_0' W_0 S_g W_0 D_0 (D_0' W_0 D_0)^{-1\prime} \tag{5}$$

where $D_0 = \partial G(\beta_0)/\partial\beta_0$ and $S_g = cov(u, z)$. When the number of orthogonality conditions is greater than or equal to the number of coefficients, $n \cdot r \geq l$, it can be shown that the asymptotically efficient GMM estimator or one which has the smallest covariance among estimators with weighting matrices W_0 is obtained by setting $W_0 = S_g^{-1}$. In this case, equation (5) reduces to

$$\Lambda = (D_0' W_0 D_0)^{-1}.$$

With serially correlated residuals of order p, a consistent estimator of S_g is provided by:

$$S_T = \sum_{j=-p}^{p} \left\{ \frac{p+1-|j|}{p+1} \right\}^{\Theta} \left\{ \frac{1}{T} \sum_t \left[h_t(\hat{\beta}_T) \otimes z_t \right] \left[h_{t-j}(\hat{\beta}_T) \otimes z_{t-j} \right]' \right\},$$

where $\hat{\beta}_T$ is the first pass estimator from equation (4) with $W_T = I$. Newey and West [8] prove several results regarding S_T when $\Theta = 1$ and recommend setting $p = T^{\frac{1}{4}}$.

When the number of orthogonality conditions is greater than the number of parameters, $n \cdot r > l$, there are $n \cdot r - l$ linearly independent remaining orthogonality conditions that are not set to zero during the estimation but should be close to zero if the restrictions of the model are true. This fact provides a way of developing a test of the model under the null hypothesis that all $n \cdot r$ orthogonality conditions are equal to zero. Hansen [4] has shown that the test statistic

$$T \cdot J_T(\beta_T) \equiv T \cdot G_T(\beta_T)' W_T G_T(\beta_T)$$

converges in distribution to a chi-square random variable with $n \cdot r - l$ degrees of freedom.

In summary, RATS obtains GMM estimates by first finding $\hat{\beta}_T$ that minimizes equation (4) given $h_t(\beta)$, z_t, and $W_T = I$. Second, S_T is obtained given Θ, p, and $\hat{\beta}_T$. Next, β_T is found by minimizing equation (4) with $W_T = S_T^{-1}$. Finally, over-identifying restrictions are tested using the $\chi^2_{(n \cdot r - l)}$ test statistic $T \cdot J_T(\beta_T)$.

9.2 The Model

The model studied in Hansen and Singleton [5] is of a representative household that ranks stochastic consumption streams, $\{C_t\}$, according to the utility function

$$E\left[\sum_{t=0}^{\infty} \beta^t u(C_t)\right].$$

The household faces the following sequence of budget constraints:

$$P_t C_t + Q_t^e Z_{t+1} + Q_t B_{t+3} \leq (P_t Y_t + Q_t^e) Z_t + B_t, \quad t = 0, 1, \ldots,$$

where P_t denotes price of consumption, Z_t is the household's beginning-of-period share holdings, Q_t^e denotes the price of a share after dividends have been paid, Y_t is the dividend, and Q_t and B_t are the price and quantity of a bond that pays a sure unit of output at time $t+3$, respectively. The market clearing conditions for this economy for $t = 0, 1, \ldots$, are:

$$\begin{aligned} C_t &= Y_t, \\ Z_{t+1} &= 1, \\ B_{t+3} &= 0. \end{aligned}$$

Additionally, the variable Y_t is assumed to display equilibrium growth. To meet the stationarity requirement of GMM estimation, we specify the nonstationary variable in terms of a scaled strictly stationary variable with a strictly positive mean. That is, define:

$$Y_t = e^{\alpha + \mu t} y_t. \tag{6}$$

Also, we assume output is modelled as an exogenous Gaussian $AR(1)$ process that takes the form

$$\log(y_{t+1}) = (1-\phi)\log(\overline{y}) + \phi \log(y_t) + \varepsilon_{t+1}$$

with $\varepsilon_t \sim$ i.i.d. $N(0, \sigma^2)$.

The estimation strategy focuses on several *efficiency conditions* that must be satisfied given markets clear and agents optimize. Suppose, for example, that the household wants to decrease current consumption by one unit and buy one unit of the equity. The cost would be $Q_t^e u'(C_t)/P_t$ where $u'(C) = \partial u(C)/\partial C$. The benefit is that the household next period receives the dividend and price of the stock. The discounted value of this to the household is $\beta u'(C_{t+1})/P_{t+1}\left[Q_{t+1}^e + P_{t+1}Y_{t+1}\right]$. If the household's undisturbed plan were, indeed, optimal, as we suppose, then the costs and benefits of the above plan must exactly match. Thus,

$$\frac{Q_t^e u'(C_t)}{P_t} - E_t\left\{\frac{\beta u'(C_{t+1})}{P_{t+1}}\left[Q_{t+1}^e + P_{t+1}Y_{t+1}\right]\right\} = 0 \tag{7}$$

where $E_t\{\cdot\}$ is the conditional expectations operator. Similarly for bonds, the efficiency condition is

$$\frac{Q_t u'(C_t)}{P_t} - E_t\left\{\frac{\beta^3 u'(C_{t+3})}{P_{t+3}}\right\} = 0. \tag{8}$$

Using the market clearing conditions, equations (6), (7), (8), iterated expectations, and letting $u(C_t) = C_t^{1-\rho}/(1-\rho)$ we can rewrite equations (6), (7) and (8) as:

$$\begin{aligned} h_t(\beta,\rho,\mu,\alpha,\bar{c},\phi,\sigma_c^2) &= E\left\{\begin{array}{l} \widehat{x}_{t+1} - \log(\bar{c}) \\ \{(\widehat{x}_{t+1} - \log(\bar{c})) - \phi(\widehat{x}_t - \\ \quad \log(\bar{c}))\}(\widehat{x}_t - \log(\bar{c})) \\ (\widehat{x}_{t+1} - \log(\bar{c}))^2 - \sigma_c^2 \\ 1 - \frac{\beta e^{-\rho\mu}\left(e^{\log(C_{t+1})-\alpha-\mu(t+1)}\right)^{-\rho}}{\left(e^{\log(C_t)-\alpha-\mu t}\right)^{-\rho}} R_{t+1}^e \\ 1 - \frac{\beta^3 e^{-3\rho\mu}\left(e^{\log(C_{t+3})-\alpha-\mu(t+3)}\right)^{-\rho}}{\left(e^{\log(C_t)-\alpha-\mu t}\right)^{-\rho}} R_t \end{array}\right\} \\ &= 0 \end{aligned} \tag{9}$$

where R_{t+1}^e is the real, relative to the consumption good, gross return on equity, R_t is the real gross return on bonds, and $\widehat{x}_t \equiv \log(C_t) - \alpha - \mu(t)$. Letting z_t denote a vector of instruments, with each element of equation (9) multiplied by z_t and using iterated expectations we get:

$$G(\beta,\rho,\mu,\alpha,\bar{c},\phi,\sigma_c^2) = E\left[h_t(\beta,\rho,\mu,\alpha,\bar{c},\phi,\sigma_c^2) \otimes z_t\right] \tag{10}$$

which are the orthogonality conditions needed for GMM.

9.3 Implementation

For the estimation of the parameters in the orthogonality conditions in equation (10) we use monthly data that runs from January 1959 to September 1995, which gives $T = 441$ observations. The time series C_t, P_t, and R_t are taken from Citibase. We identify the real consumption series with the consumption of nondurables (GMCNQ) plus services (GMCSQ) divided by total population (POP). The price series is identified as the consumer price index (PUNEW). The nominal return on the riskless security is identified as return on 3-month T-bill (FYGN3) and made real by dividing by the inflation rate from the price series. The time series R_t^e is taken from the *Center for Research in Security Prices (CRSP)* database. The series is identified as the value-weighted equity return for the NYSE (VWRETD).

The data are located in my public directory `/users/faculty8/sfowler/public_html/data` in two files, `ex6a.dat` and `ex6b.dat`, which contain, respectively, the Citibase and CRSP data. To implement the GMM estimation, we now write and run a RATS program, `ex6.prg` which will read these data files and estimate our model. At the prompt, type and enter:

```
frank $ pico ex6.prg
```

Once in the pico editor, enter and save the following lines of RATS code:

```
envo noecho
open output ex6.out
calendar 1925 12 12
allocate 0 1995:12

*
* to set the begining and ending dates
*
compute begdate = (1959:1)
compute enddate = (1995:9)

*
* the names for the data are
*
* gmcnq  = real nondurable consumption (bbls)
* gmcsq  = real service consumption (bbls)
* pop    = total population (ths)
* punew  = cpi
* fygn3  = 3-month T-bill (yr rate)
* vwretd = value-weighted equity return for the NYSE
*
open data /users/faculty8/sfowler/public_html/data/ex6a.dat
data(unit=data,org=obs,missing=-999) 1947:1 1995:9 $
gmcnq gmcsq pop punew fygn3

open data /users/faculty8/sfowler/public_html/data/ex6b.dat
data(unit=data,org=obs,missing=-999) 1925:12 1995:12 $
vwretd

*
* to treat data
*
* 1. to make population into billions
* 2. to set cpi deflator
* 3. to calculate total per capita consumption
* 4. to calculate 3-month t-bill real return from 1yr discount rate
* 5. to calculate gross real equity-weigted return on NYSE
* 6. to calculate log of consumption
* 7. to set trend
*
compute ad = 90/360
set n      = pop/1000000000
set def    = punew/100
set c      = (gmcnq + gmcsq)/n(t)
set rtbill = (1/(1-ad*(fygn3(t)/100)))/(def(t+3)/def(t))
set requit = (1+vwretd(t))/(def(t)/def(t-1))
set lnc    = log(c)
set trend  = t

*
* define parameters
*
```

```
* beta  = rate of time preference
* rho   = utlity function parameter
* mu    = growth rate of consumption
* alpha = intercept
* xi    = E[ln(c)]
*
nonlin beta rho mu alpha xi phi sigma

*
* define efficiency conditions
*
frml h1 = (lnc(t+1)-alpha-mu*trend(t+1)-xi)

frml h2 = ((lnc(t+1)-alpha-mu*trend(t+1)-xi)-$
phi*(lnc(t)-alpha-mu*trend(t)-xi))*$
(lnc(t)-alpha-mu*trend(t)-xi)

frml h3 = ((lnc(t+1)-alpha-mu*trend(t+1)-xi)-$
phi*(lnc(t)-alpha-mu*trend(t)-xi))**2 - sigma

frml h4 = $
beta*exp(-rho*mu)*(exp(lnc(t+1)-alpha-mu*trend(t+1))/exp(lnc(t)-$
alpha-mu*trend(t)))**(-rho)*requit(t+1)-1

frml h5 = $
(beta**3)*exp(-3*rho*mu)*(exp(lnc(t+3)-alpha-mu*trend(t+3))/exp(lnc(t)-$
alpha-mu*trend(t)))**(-rho)*rtbill(t)-1

*
* set intial conditions
*
compute beta  = 0.9
compute rho   = 1.0
compute mu    = 0.0025
compute alpha = 14.0
compute xi    = -1.0
compute phi   = 0.96
compute sigma = 0.0002

*
* do gmm with p=7, Theta=1 and
* over-identified system
*
dofor nlag = 0 1 2
   instruments constant rtbill{3 to (nlag+3)} requit{0 to nlag}
   nlsystem(instruments,zudep,lags=5,damp=1,iterations=200) begdate enddate $
   h1 h2 h3 h4 h5
   cdf chisqr %uzwzu/%nobs 5*(3+2*nlag)-7
end dofor

end
```

After completing and saving ex6.prg, run the program by entering

```
frank $ rats ex6.prg
```

Completing of the program is indicated by the return of the titan prompt. Listing of the files in your directory should resemble:

```
bin        ex2.log    ex2.sas   ex6.out
ex1.txt    ex2.lst    ex3.sas   ex6.prg  mail
```

The file ex6.out is the RATS output file where all the estimation results are placed. To view the estimation results, open ex6.out by typing

```
frank $ pico ex6.out
```

For more information on the RATS programing language see Doan [2].

10 To Create, Run, and Interpret the Results of a Maple Program

In this chapter, we study how to compute *Gauss quadrature rules* with the help of Maple while closely following von Matt [9]. We consider the efficiency equations of the previous section in the integral forms:

$$\begin{aligned} \overline{Q}^e(c) - \beta e^{-\rho\mu} \int_C (c'/c)^{-\rho} \, [\overline{Q}^e(c') + c'] f(c'|c) \; dc' &= 0 \\ \overline{Q}(c) - \beta e^{-\rho\mu} \int_C (c'/c)^{-\rho} f(c'|c) \; dc' &= 0 \end{aligned}$$

where $\overline{Q}^e(c)$ and $\overline{Q}^e(c)$ are, respectively, the real price of equity and bonds, c' and c are next and current period consumption, and $f(c'|c)$ is the conditional density for c' given c. Additionally, we assume consumption is modelled as an exogenous Gaussian $AR(1)$ process that takes the form

$$c_{t+1} = (1-\phi)\mu_c + \phi c_t + \varepsilon_{t+1}$$

with $\varepsilon_t \sim$ i.i.d. $N(0, \sigma^2)$.

The purpose of Gaussian quadrature is to approximate the integrals in the above equations by a finite sum. With the integrals $\int_C (c'/c)^{-\rho} \, [\overline{Q}^e(c') + c'] f(c'|c) \; dc'$ and $\int_C (c'/c)^{-\rho} f(c'|c) \; dc'$ written respectively as $\int_C g_1(c', c) w(c') \, dc'$ and $\int_C g_2(c', c) w(c') \, dc'$, where $w(c')$ is some positive weighting function, a Gaussian quadrature is

$$\int_C g_j(c', c) w(c') \, dc' \approx \sum_{i=1}^{n} g_j(\widehat{c}_i, c) w_i, \quad \text{for } j \in \{1, 2\}$$

where the $\widehat{c}_i$'s are called the abscissas and the w_i's, weights. The weights are determined such that all polynomials to as high degree as possible are integrated exactly. In economics, quadrature are used, for example, in the solving of the above integral equations for the optimal prices $\overline{Q}^e(c)$ and $\overline{Q}(c)$.

Orthogonal polynomials play a key role for Gaussian quadrature. Orthogonal polynomials corresponding to the interval $[a, b]$ and the weighting function $w(x)$ are calculated because the zeroes of the polynomials, which are unique, are the abscissas for Gaussian quadrature. As a start, we first make some necessary definitions.

Definition 10.1 *The inner product of two functions f and g over a weight function $w(x)$ is denoted as*

$$\langle f|g\rangle \equiv \int_{[a,b]} f(x)g(x)w(x)\,dx. \tag{11}$$

Definition 10.2 *The polynomials $p_0(x), p_1(x), \ldots$ are called orthogonal polynomials with respect to the inner product (11), if the following conditions are met:*

1. *$p_k(x)$ is of degree k.*
2. *The polynomial $p_k(x)$ satisfies the orthogonality condition*

   $$\langle p_k|p\rangle = 0$$

   for all polynomials $p(x)$ of degree less than k.

Given $w(x)$, the following theorems ensures the existence of orthogonal polynomials and their recursive representation with respect to the inner product (11). The reader is directed to von Matt [9] for proofs.

Theorem 10.1 *For any admissible product there exists a sequence $\{p_k(x)\}_{k=0}^{\infty}$ of orthogonal polynomials.*

Theorem 10.2 *The orthogonal polynomials $\{p_k(x)\}_{k=0}^{\infty}$ satisfy the three-term relationship*

$$xp_{k-1} = \beta_{k-1}p_{k-2} + \alpha_k p_{k-1} + \beta_k p_k, \quad k = 1, 2, \ldots \tag{12}$$

with

$$\begin{aligned}\alpha_k &= \langle xp_{k-1}|p_{k-1}\rangle \\ \beta_k &= \langle xp_{k-1}|p_k\rangle\end{aligned}$$

where $\beta_0 = 0, p_{-1}(x) = 0,$and $p_0(x) = \pm\sqrt{1/\langle 1|1\rangle}$.

Evidently, given $\beta_0 = 0, p_{-1}(x) = 0,$and $p_0(x) = \pm\sqrt{1/\langle 1|1\rangle}$ the sequences α_k and β_k can be recursively calculated. This recursive scheme is known as Lanczos algorithm.

Computation of the sequences α_k and β_k are important because they are directly used in computation of the Gaussian abscissas and weights.

Theorem 10.3 *A polynomial $p_n(x)$ of degree n, which can be represented by the tree-term recurrence relationship (12), has as zeroes the unique eigenvalues of the tridiagonal matrix*

$$T_n = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_{n-1} \\ & & \beta_{n-1} & \alpha_n \end{bmatrix}$$

Furthermore, let $T_n = QXQ^T$ be the eigenvalue decomposition of T_n. Each $\langle 1|1\rangle \cdot Q_{1,i}^2$ for $i \in \{1, \ldots, n\}$ represents the weight corresponding to the i'th abscissa or i'th eigenvalue.

When the weight function, $w(x)$, is a probability density function (p.d.f.), as in our case, computation of the α's and β's is straight forward. For all p.d.f.'s $\langle 1|1\rangle = 1$, giving $\alpha_1 = \langle x|1\rangle = E[x]$. Furthermore, every subsequent α and β will be functions of the mean and variance. As an example, we assume $c \in [a,b]$ and has a normalized Gaussian p.d.f. of the form

$$f(c'|c) = \frac{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(c'-(1-\phi)\mu_c-\phi c)^2}{2\sigma^2}\right)}{\frac{1}{\sqrt{2\pi\sigma^2}} \int_{[a,b]} \exp\left(\frac{-(c'-(1-\phi)\mu_c-\phi c)^2}{2\sigma^2}\right) dc'}.$$

When computing the abscissas, $\widehat{c}_i$'s, it is easier to choose a weighting function not dependent on c. We can multiply and divide the integral equations by the numerator of the conditional p.d.f. of c' given c is at it's mean. This gives a weighting funtion of

$$f(c'|c = \mu_c) = w(c') = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(c'-\mu_c)^2}{2\sigma^2}\right), \qquad \text{for } c' \in [a,b].$$

Implementation in Maple is done by first creating an input file called `ex4.map`. At the `frank` prompt enter

```
frank $ pico ex7.map
```

and enter the following lines of Maple code:

```
#
# ex7.map is a Maple program for finding
# Gaussian quadrature rules
#
restart:
settime := time():
kernelopts( printbytes = false ):
with(linalg):

#
# to write to output file
#
writeto( `ex7.mapout` );
print( `output for Maple program ex7.mapout` );

#
# define parameters
#
sigma := 0.032496:
c     := 0.005725:
phi   := 0.99425:
dss   := c/(1-phi):

#
# use Lanczos algorithm to
# estimate alphas and betas of
```

```
# othogonal polynomials
#
a := 0.85:
b := 1.15:
n := 6:

omega := t -> ( exp( -((t-dss)^2)/(2*sigma^2) )/
 sqrt(2*Pi*sigma^2) ):
iprod := (f,g,x) -> int(f*g*omega(x), x=a..b):
mu0 := iprod(1,1,x):

  alpha := array (1..n):
  if n > 1 then beta := array (1..n-1) else beta := 'beta' fi:

  p[0] := collect (1 / sqrt (iprod (1, 1, x)), x, simplify):
  alpha[1] := normal (iprod (x*p[0], p[0], x)):
  q := collect ((x - alpha[1]) * p[0], x, simplify):
  for k from 2 to n do
    beta[k-1] := normal (sqrt (iprod (q, q, x))):
    p[k-1] := collect (q / beta[k-1], x, simplify):
    alpha[k] := normal (iprod (x*p[k-1], p[k-1], x)):
    q := collect ((x - alpha[k]) * p[k-1] - beta[k-1]*p[k-2],
                  x, simplify):
  od:

#
# to compute weights and
# abcissas using the Eigenvalue
# decomposition
#
  T := array (1..n, 1..n, symmetric, [[0$n]$n]):
  for i from 1 to n do
    T[i,i] := alpha[i]:
  od:
  for i from 1 to n-1 do
    T[i,i+1] := beta[i]:
    T[i+1,i] := beta[i]:
  od:

  # compute the eigenvalue decomposition T = QXQ'
  X := evalf( Eigenvals (T, Q)):

  # abscissas
  x := array (1..n, [seq (X[i], i=1..n)]);

  # weights
  w := array (1..n, [seq (evalf(mu0) * Q[1,i]^2, i=1..n)]);

#
# finish
#
cpu_time := (time() - settime)*seconds;
```

```
writeto(terminal ):
cpu_time := (time() - settime)*seconds;
```

Once `ex7.map` is saved and closed, type and enter

```
frank $ maple
```

You should see on your screen

```
    |\^/|     Maple V Release 4 (Middle Tennessee State University)
._|\|   |/|_. Copyright (c) 1981-1996 by Waterloo Maple Inc. All
 \  MAPLE  /  rights reserved. Maple and Maple V are registered
 <____ ____>  trademarks of Waterloo Maple Inc.
      |       Type ? for help.
>
```

Then, enter the following statement

```
>read `ex7.map`;
```

Once maple has finished processing, enter

```
>quit;
```

To open the output file `ex7.mapout`, at the `frank` prompt enter

```
frank $ pico ex7.mapout
```

You should see on your screen

```
                output for Maple program ex7.mapout

 x := [.9010058781, .9488747877, .9923073990, 1.035708914,
       1.083578566, 1.236867705]


w := [.009334945827, .1945848840, .5231244566, .2562004398,
    .01675502747, .0000002469550046]

                cpu_time := 1.809 seconds
```

which are the abscissas and weights corresponding to our Gaussian quadrature rule. For more information on the Maple programing language see Heck [6]. For more on Gaussian quadrature rules see Davis and Rabinowitz [1].

11 How to Log off Your `frank` Account

To log off your `frank` account type

```
frank $ exit
```

References

[1] Davis, P.J. and P. Rabinowitz, *Methods of Numerical Integration*, Academic Press, Orlando, 1984.

[2] Doan, T.A., *RATS User's Manual Version 4*. Evanston, IL: Estima, 1995.

[3] Gallant, A., "Three-Stage Least-Squares Estimation for a System of Simultaneous, Nonlinear, Implicit Equations." *Journal of Econometrics*, Vol 5, (1977), pp. 71-88.

[4] Hansen, Lars P., "Large Sample Properties of Generalized Method of Moments Estimators." *Econometrica*, Vol. 50, (1982), pp. 1029-1054.

[5] Hansen, L.P. and K.J. Singleton, "Generalized Instrumental Variables Estimation of Non-Linear Rational Expectations Models." *Econometrica*, Vol. 50, (1982), pp. 1269-1286.

[6] Heck, A., *Introduction to Maple*. Budapest: Springer-Verlag, 1996.

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

[8] Newey, W. and K. West, "A Simple Positive-Definite Heteroskedasticity and Autocorrelation Consistent Covariance Matrix." *Econometrica*, Vol. 55, (1987), pp. 703-708.

[9] von Mat, Urs, "Gauss Quadrature." in W. Grander and J. Hrebicek (eds.), *Solving Problems in Scientific Computing Using Maple and Matlab*. Heidelburg: Springer-Verlag, 1995, pp. 237-265.