QBUS6830
Financial Times Series and Forecasting
Semester 1, 2019

Lab 1: A Matlab Introductory Tutorial

Matlab is a very popular and powerful tool for statistical, econometric and mathematical modelling, simulation and analysis. It is used by many businesses and is perhaps the 2nd most popular statistical software tool, behind SAS, in the business world.

-- Matlab is a command based language. You tell it what to do using its programming language or code. There are almost no drop-down windows or automatic buttons.
-- You will learn (or at least have access to) its command coding in this unit.

Commands are typed after the symbol:

```
>>
```

which automatically appears when Matlab is started and re-appears after each command is entered. Use the 'Start → programs' button in Windows to start Matlab.

-- We will learn some basic Matlab commands below.

-- Matlab does have some nice pull-down options for data acquisition and input.

Let's import some real data into Matlab. This is easy!

1. First save the dataset for lab 1 'CBA_Daily_Jan1999_Feb2018_yahoo.csv' into a suitable folder (e.g. your flash-drive or PC hard-drive).

2. You can EITHER:

   (i) Open the data in Excel and then copy and paste the required data which are the last four columns (just the numbers!) into Matlab. To do this type:

   ```
   CBAdata = [ <paste the data here> ];
   ```

   Now you have a matrix variable called CBAdata.

   OR

   (ii) In Matlab go to the Menu: 'File -> Import data'. Then choose the folder where you saved the data and select the data file. A Data Wizard should appear. There are a number of ways Matlab can import your data. Under the dropdown box for 'Output Type' you can choose Column vectors, Numeric Matrix, Cell Array, and Table. Select "Numeric Matrix", and then in the wizard adjust the 'box/frame' so that it only includes the rows and columns you want to import. Here we shall just use the default which is to select all columns and all rows. Above the data Matlab will provide a suggested name for your new matrix such as "CBADailyJan1999Feb2018yahoo". You can change the variable name to 'CBAdata' by selecting the name and typing over the Matlab chosen name in the Wizard before you import, OR after you import via the commands.

```
CBAdata = CBADailyJan1999Feb2018yahoo;
clear CBADailyJan1999Feb2018yahoo
```

3. Type `whos` into the Matlab command screen and you will see any variables that Matlab has in the current workspace:

```
Your variables are:

CBAdata
```

Note: Variable names must start with a letter (not a number or any other symbol like %).

You should be able to see that CBAdata is a matrix of size 4835×7, i.e., a matrix with 4835 rows and 7 columns

Now type

```
CBAp = CBAdata(:,6);
```

in Matlab. This extracts the Adjusted Closing prices for CBA in column 6 of this matrix into a vector (a 1-column matrix) called 'CBAp'. Column 5 contains the daily closing prices for CBA stock, on the ASX.

Note that I put a semi-colon ';' after the command, to stop it printing out the whole series to the screen. The ';' is always optional, as sometimes you want to see the result of your command appear on the screen. Include the ';' if you don't want to see the output.

We can do some simple functions on this data series of prices. It is stored as a vector in Matlab. Just type

```
CBAp
```

and it will print the whole series to the screen. Type

```
plot(CBAp)
```

and a time plot will appear on the screen. We can do lots of operations on this series as shown below. Some of these require having certain Toolboxes. In lab 2 we have all the toolboxes: Statistics, Optimisation, Econometrics, Finance, Financial time series, are the ones we will use in this unit (you might check the others out too for interest). You will be able to use ALL the functions below:

Command	Function
`mean(prices)`	Calculates the mean of the series
`var(prices)`	Calculates the variance

`qqplot(prices)`	Plots a QQ plot for normality
`autocorr(prices)`	ACF plot of series
`parcorr(prices)`	PACF of series
`price2ret(prices)`	Forms log returns from a price series
`length(prices)`	Sample size of series
`log(prices)`	Takes natural logarithms of whole series
`log10(prices)`	Takes logarithm to base 10 of series
`diff(prices)`	Takes first difference of series
`100*prices`	Multiplies each series element by 100
`prices.^2`	Squares each series element
`prctile(prices,p)`	Returns the pth percentile of series (0<p<100)

Note that the command ‘`prices.^2`’ used a full-stop ‘.’ This tells Matlab to square *each* ***element*** of the vector ‘prices’ individually. If the full-stop wasn’t there Matlab would do a matrix square product and, under matrix rules, form a 4835*4835 matrix (`length(prices) = 4835`) which would probably take ALL of the available memory in, and hence crash, Matlab. So, it is very important NOT to forget the full-stop.

Help in Matlab is really (usually) quite fantastic (and I don’t work for Matlab). You should consult this whenever you get into trouble or lost in Matlab. *Click on the ‘Getting Started’ button in the left hand bottom corner of the Matlab Command window for help*. This includes general help or help with specific toolboxes (like ‘Econometrics’).

Let’s change the prices to log returns and store them in a variable called ‘ret’. We simply type:

```
ret = diff(log(CBAp));
```

Here the Log() command takes the natural logarithm of each element in the price series and the diff() command then takes the difference between sequential elements of the logged series which provides $\log(P_t)-\log(P_{t-1}) = \log(P_t/P_{t-1}) = \log(R_t)$

You could also use the function:

```
ret = price2ret(CBAp)
```

To plot these on the SAME plot as the prices type

```
hold on; plot(ret,'r')
```

The first command (‘hold on’) says to use the SAME plot as before and overlay the new data. The ‘r’ says to plot in red colour. We can’t really see these returns (they are too small compared to the prices) on this plot. Let’s change to percentage log returns.

```
pret = 100*ret;

plot(pret, 'g')
```

Now we see the green (‘g’) percentage returns. This is a nice plot showing risk and return properties over time for CBA in the last decade. To make a new plot just type

```
figure; plot(pret)
```

Or, to keep only one figure, but write over or lose the existing plot, type

```
hold off; plot(pret)
```

You can see that blue is the default colour (‘b’). Type

```
help plot
```

to see the various options, colours, symbols and line types we can use in Matlab plots.

The code for loops in Matlab is

```
for k = 1:10
    x(k) = k^2;
end
```

which will return, after typing

```
x
```

the answer:

```
x =

     1     4     9    16    25    36    49    64    81   100
```

as expected. If we want x to be a column vector instead of a row we can type

```
>>x=x'
```

which returns

```
x =

     1
     4
     9
    16
    25
    36
    49
    64
    81
   100
```

There are many ways to do the same thing in Matlab (as with all of MATHS in general!). For example:

```
X =(1:10).^2
```

does the same thing as the loop above. The command

```
1:10
```

creates a row vector of the integers between 1 and 10.
NB. Loops are SLOW in matlab. It is faster to do x=(1:10).^2 than run the loop above. The difference in time gets much larger as the size of the series gets bigger.

To create numbers in between set limits the command is

```
x = 1:1:10
```

which simply returns the integers 1,2, …, 10 again. The extra ':1' here says the gap between each number should be 1. If we change the gap to 0.5, the command and output is:

```
x = 1:0.5:10

x =

  Columns 1 through 15

    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000
5.0000    5.5000    6.0000    6.5000    7.0000    7.5000    8.0000

  Columns 16 through 19

    8.5000    9.0000    9.5000   10.0000
```

which jumps between 1 and 10 in steps of size 0.5.

Statistical functions

With access to the Statistics Toolbox we can do most interesting statistical functions. Some that we will use are listed below:

Command	Function
`normcdf`	Calculates the cdf function for a normal distribution
`norminv`	Calculates the inverse cdf fuction for a normal distribution
`Tcdf`	Calculates the cdf fuction for a Student-t(v) distribution
`Tinv`	Calculates the inverse cdf fuction for a Student-t(v) distribution
`Rand`	Simulate from a Unif(0,1)
`Randn`	Simulate from a N(0,1)
`normrnd`	Simulate from a Normal distribution with mean m and standard deviation s
`Trnd`	Simulate from a Student-t(v) distribution with mean 0, variance v/(v-2)

Type 'help' and then the command name for specific instructions on each command.
For example:

```
help tcdf
```

Calculations

Matlab can also be a calculator. e.g. type

```
3*2+6-4
```

and Matlab returns

```
ans =

      8
```

Finding previous commands

Sometimes we wish to repeat previous commands. To execute the previous command, simply press the up arrow ↑ key. The previous command will display, in this case it was

```
3*2+6-4
```

just press return to execute it again. We could also change the expression by editing this command. Again press the up arrow and then edit the command using the right and left arrows and delete/backspace keys.

To find previous commands we can also type the first letter of that command and again use the up arrow. For example, to find the loop above we type

```
f
```

and then use up arrow. Try this!

We will learn many other Matlab commands and functions throughout the unit. This exercise was meant as a simple introduction to the environment of Matlab and to give you a reference for some common Matlab functions. **You might like to add your own commands to this document as needed throughout semester.**

Exercises:

(i) Plot the CBA prices over time. Discuss the properties of CBA prices over the past decade or so. Are they predictable? Stable?

平均值，中位数，标准差，偏度和峰度

(ii) Convert the price series to percentage log returns. Find the sample mean, median, standard deviation, skewness and kurtosis for the CBA percentage log return data. Discuss the general properties of this data.

(iii) Plot a histogram of the percentage log returns. Use 'hist" and type 'help hist' for assistance. Discuss the distribution of percentage log returns for CBA.

(iv) Find the following percentiles of this data: 0.5%, 1%, 10%, 25%, 50%, 75%, 90%, 99%, 99.5%. Further describe the distribution of log returns

(v) What return might an investor expect to suffer on only 0.1% of days when investing in CBA stock?

(vi) Plot the CBA log returns over time. Discuss this plot and the time properties of CBA returns. What aspects seem to change with time? Which seem to remain fairly constant or static over time?