
LECTURE 13 Nov

Table of Contents

Part I	1
Plot	1
Output statements: fprintf	2
Save workspace or variables, clear and load	3
Part II	4
Extract data	4
Create tables	4
Extracting portions of a Table	5
Applying functions to tables	5
Exporting tables	5
Compare logical operator with an array	5

Part I

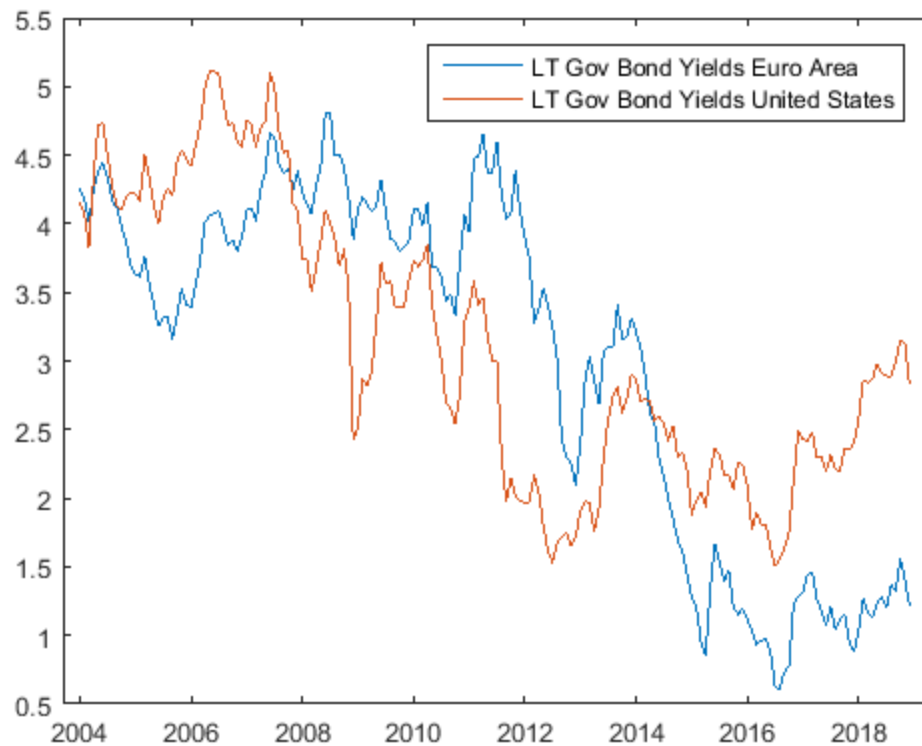
```
% Download data from fred
url = 'https://fred.stlouisfed.org/';
c = fred(url);

% the start and end dates (specified as datetime variables).
startdate = datenum('01-2004','mm-yyyy'); %startdate =
    datenum('Q1-2004','QQ-yyyy') or datenum('04-2004','mm-yyyy');
enddate = datenum('12-2018','mm-yyyy');

%Long Term Gov Bond Yields Euro Area
d = fetch(c,'IRLTLT01EZM156N',startdate,enddate);
%Long Term Gov Bond Yields United States
d2 = fetch(c,'IRLTLT01USM156N',startdate,enddate);
```

Plot

```
% time space
dt = linspace(startdate,enddate,180);
% convert
dtDates = datetime(dt,'ConvertFrom','datenum');
% plot the data
figure('color','w')
plot(dtDates,d.Data(:,end),dtDates,d2.Data(:,end)); hold on % you can
    also extract the data with d.Data(:,2)
legend('LT Gov Bond Yields Euro Area','LT Gov Bond Yields United
    States')
hold off
```



Output statements: fprintf

```
% Output statements display strings and the results of expressions,  
% fprintf contains the text to be printed as well as formatting  
% information  
% for the expressions to be printed
```

```
LTGB = d.Data(:,end);
```

```
fprintf('mean %d \n',mean(LTGB))  
fprintf('mode %d \n',mode(LTGB))  
fprintf('median %d \n',median(LTGB))  
fprintf('std %d \n',std(LTGB))  
fprintf('min %d \n',min(LTGB))  
fprintf('max %d \n',max(LTGB))
```

```
% the %d specifies where the value is to be printed
```

```
% There are others  
% %d integers (decimal integer)  
% %f floats  
% %c single characters  
% %s strings
```

```
% \n let the output moves down to the next line.

% 5%d would indicate a field width of 5 for printing an integer
fprintf('mean %5d \n',mean(LTGB))
fprintf('mode %5d \n',mode(LTGB))
fprintf('median %5d \n',median(LTGB))
% 10%d would indicate a field width of 10 for printing an integer
fprintf('std %10d \n',std(LTGB))
fprintf('min %10d \n',min(LTGB))
fprintf('max %10d \n',max(LTGB))

% 6.2%f means a field width of 6 (including the decimal point and
% the decimal places) with two decimal places.
fprintf('mean %6.2f \n',mean(LTGB))
fprintf('mode %6.2f \n',mode(LTGB))
fprintf('median %6.2f \n',median(LTGB))

% 6.4%f means a field width of 6 (including the decimal point and
% the decimal places) with four decimal places.

fprintf('std %6.4f \n',mean(LTGB))
fprintf('min %6.4f \n',mode(LTGB))
fprintf('max %6.4f \n',median(LTGB))

mean 3.015288e+00
mode 4.089700e+00
median 3.422700e+00
std 1.284655e+00
min 6.133000e-01
max 4.814500e+00
mean 3.015288e+00
mode 4.089700e+00
median 3.422700e+00
std 1.284655e+00
min 6.133000e-01
max 4.814500e+00
mean 3.02
mode 4.09
median 3.42
std 3.0153
min 4.0897
max 3.4227
```

Save workspace or variables, clear and load

```
% To save all workspace variables in a file, the command is:
save filename

%To save just one variable to a file, the format is:
save filename2 LTGB
```

```
% Clear workspace and command window
clear
clc
```

```
% Load a file
load filename
```

Part II

Download data from fred

```
% 12-Month London Interbank Offered Rate (LIBOR), based on U.S.
Dollar
% USD12MD156N

% 12-Month London Interbank Offered Rate (LIBOR), based on British
Pound
% GBP12MD156N

% 12-Month London Interbank Offered Rate (LIBOR), based on Euro
% EUR12MD156N

% 12-Month London Interbank Offered Rate (LIBOR), based on Japanese
Yen
% JPY12MD156N

% 12-Month London Interbank Offered Rate (LIBOR), based on Swiss
Franc
% CHF12MD156N

startdate = datenum('01-2000','dd-yyyy');
enddate = datenum('12-2018','dd-yyyy');

d3 = fetch(c,'USD12MD156N',startdate,enddate);
d4 = fetch(c,'GBP12MD156N',startdate,enddate);
d5 = fetch(c,'EUR12MD156N',startdate,enddate);
d6 = fetch(c,'JPY12MD156N',startdate,enddate);
d7 = fetch(c,'CHF12MD156N',startdate,enddate);
```

Extract data

```
datedd = d3.Data(:,1);
usdd = d3.Data(:,end);
gbpd = d4.Data(:,end);
eurd = d5.Data(:,end);
jpyd = d6.Data(:,end);
chfd = d7.Data(:,end);
```

Create tables

```
Tablibor = table(datedd,usdd,gbpd,eurd,jpyd,chfd);
```

Extracting portions of a Table

```
% Create a table named bafin that contains the date and US and GBP
base
% LIBOR the first 20 rows.

bafin = Tablibor(1:20,{'datedd','usdd','gbpd'});

% Create a table named banov that contains the Dates and chfd
variables for
% rows seven to the end of Tablibor.
banov = Tablibor(7:end,{'datedd','chfd'});
```

Applying functions to tables

```
maxfin = @max

avgStocks = varfun(maxfin,bafin)
```

```
maxfin =
```

```
    @max
```

```
avgStocks =
```

<u>max_datedd</u>	<u>max_usdd</u>	<u>max_gbpd</u>
7.3051e+05	6.6513	6.8877

Exporting tables

```
writetable(bafin,'exdata.csv')
```

Compare logical operator with an array

the operation is applied to each element of the array and the result is returned in the array having the same size as the input array.

```
I = usdd>7;
```

```
% I contains elements that are either true or false
% (true if usdd is greater than 7)
```

Published with MATLAB® R2015b