Bus 41202: Analysis of Financial Time Series
Spring 2016, Ruey S. Tsay

Pre-class reading assignment: The R Program and Some Packages

In this note, we briefly introduce the R program to be used extensively in the course. Specific packages and their commands for performing statistical analyses discussed in the lectures will be given when needed. Our goal is to make the empirical analysis as easy as possible so that students can reproduce the results shown in the lecture notes and textbook.
R is a free software available from http://www.r-project.org. It runs on many operating systems, including Linux, MacOS X, and Windows. One can click *CRAN* on the above web page to select a nearby *CRAN Mirror* to download and install the software and selected packages. The simplest way to install the program is to follow the online instructions and to use the default options. Because R is an open-source software, it contains thousands of packages developed by researchers around the world for various statistical analyses. For financial time series analysis, the Rmetrics of Dr. Diethelm Wuertz and his associates has many useful packages, including fBasics and fGarch. We use many functions of these packages in the lectures. We also use some other packages that are powerful and easy to use in R, e.g., the `evir` package for extreme value analysis in R and the `rugarch` package for additional volatility models.
The R commands are case sensitive and must be followed exactly.

0.1 Installation of R packages

Using default options in R installation creates an icon on the desktop of a computer. One can start the R program simply by double clicking the R icon. All Booth computers should have R installed. For Windows, a `RGui` window will appear with command menu and the R `Console`. To install packages, one can click on the command `Packages` to select `Install packages`. A pop-up window appears asking users to select a R mirror (similar to R installation mentioned before). With a selected mirror, another pop-up window appears that contains all available packages. One can click on the desired packages for installation. *You only need to install a package once* on each machine.
With packages installed, one can load them into R by clicking on the command `Packages` followed by clicking `Load packages`. A pop-up window appears that contains all installed packages for users to choose. An alternative approach to load a package is to use the command `library` or `require`. See the demonstration below.

0.2 The quantmod Package

To begin with, we consider a useful R package for downloading financial data directly from some open sources, including Yahoo Finance, Google Finance, and the Federal Reserve

Economic Data (FRED) of Federal Reserve Bank of St. Louis. The package is `quantmod` by Jeffry A. Ryan. It is highly recommended that one installs it. Another useful package for financial data is `quandl`. The basic version of quandl is also free.

Once installed, the **quantmod** package allows users, with Internet connection, to use tick symbols to access daily stock data from Yahoo and Google Finance and to use series name to access thousands of economic and financial time series from FRED. The command is `getSymbols`. The package also has some nice functions, e.g., obtaining time series plots of closing price and trading volume. The command is `chartSeries`. The default option of these two commands is sufficient for basic analysis of financial time series. One can use subcommands to further enhance the capabilities of the package such as specifying the time span of interest in `getSymbols`. Interested readers may consult the document associated with the package for description of the commands available. Here we provide a simple demonstration. Figure 1 shows the time plots of daily closing price and trading volume of Apple stock from January 3, 2008 to January 28, 2015. The plot also shows the price and volume of the last observation. The subcommand `theme=''white''` of `chartSeries` is used to set the background of the time plot. The default is black. Figure 2 shows the time plot of monthly U.S. unemployment rates from January 1948 to November 2011. Figure 3 shows the time plot of daily interest rate of 10-year treasures notes from January 3, 2007 to December 2, 2011. These are the interest rates from the Chicago Board Options Exchange (CBOE) obtained from Yahoo Finance. Since there is no volume, the subcommand `TA=NULL` is used to omit the time plot of volume in `chartSeries`. The commands `head` and `tail` show, respectively, the first and the last six rows of the data.

R Demonstration with quantmod package
Output edited. > denotes R prompt and explanation starts with %.

```
> library(quantmod)   % You may use the command "require(quantmod)" too.
> getSymbols("AAPL",from="2008-01-03",to="2015-01-28")  % Use default: from Yahoo
[1] "AAPL"
> dim(AAPL) % See the size of the loaded data set
[1] 1780    6
> head(AAPL) % The first 6 row of data
           AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume AAPL.Adjusted
2008-01-03    195.41    197.39   192.69     194.93   210516600         26.37
2008-01-04    191.45    193.00   178.89     180.05   363958000         24.36
2008-01-07    181.25    183.60   170.23     177.64   518048300         24.03
2008-01-08    180.14    182.46   170.80     171.25   380954000         23.17
2008-01-09    171.30    179.50   168.30     179.40   453470500         24.27
2008-01-10    177.58    181.00   175.41     178.02   370743800         24.08
> tail(AAPL) % The last 6 row of data
           AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume AAPL.Adjusted
2015-01-21    108.95    111.06   108.27     109.55    48575900        109.55
2015-01-22    110.26    112.47   109.72     112.40    53796400        112.40
2015-01-23    112.30    113.75   111.53     112.98    46464800        112.98
2015-01-26    113.74    114.36   112.80     113.10    55615000        113.10
```

```
2015-01-27    112.42    112.48   109.03     109.14    95568700        109.14
2015-01-28    117.63    118.12   115.31     115.31   145448000        115.31
> chartSeries(AAPL,theme="white")  % Obtain time plot of
                                        closing price and trading volume

> getSymbols("UNRATE",src="FRED")  % Load monthly unemployment rate
                                        from FRED (Federal Reserve Bank)
[1] "UNRATE"
> dim(UNRATE)
[1] 804   1
> chartSeries(UNRATE,theme="white")
> head(UNRATE)
           UNRATE
1948-01-01    3.4
1948-02-01    3.8
1948-03-01    4.0
1948-04-01    3.9
1948-05-01    3.5
1948-06-01    3.6
> tail(UNRATE)
           UNRATE
2014-07-01    6.2
2014-08-01    6.1
2014-09-01    5.9
2014-10-01    5.7
2014-11-01    5.8
2014-12-01    5.6
> getSymbols("INTC",src="google") %% Use Google finance
[1] "INTC"
> dim(INTC)  %% Default time span: Jan. 03, 2007 to last day available.
[1] 2032    5
> head(INTC)
           INTC.Open INTC.High INTC.Low INTC.Close INTC.Volume
2007-01-03     20.45     20.88    20.14      20.35    69803965
2007-01-04     20.63     21.33    20.56      21.17    89514297
2007-01-05     21.09     21.15    20.76      21.10    64596375
2007-01-08     21.25     21.34    20.95      21.01    52844607
2007-01-09     21.18     21.21    20.86      21.03    54414980
2007-01-10     21.09     21.62    21.03      21.52    76369883
> tail(INTC)
           INTC.Open INTC.High INTC.Low INTC.Close INTC.Volume
2015-01-22     36.56     37.00    36.14      36.91    31809899
2015-01-23     36.96     37.03    36.38      36.44    27427055
2015-01-26     36.18     36.30    35.57      35.80    30906818
2015-01-27     34.40     34.72    33.55      34.18    58456203
2015-01-28     34.47     34.70    33.72      33.78    34929097
```

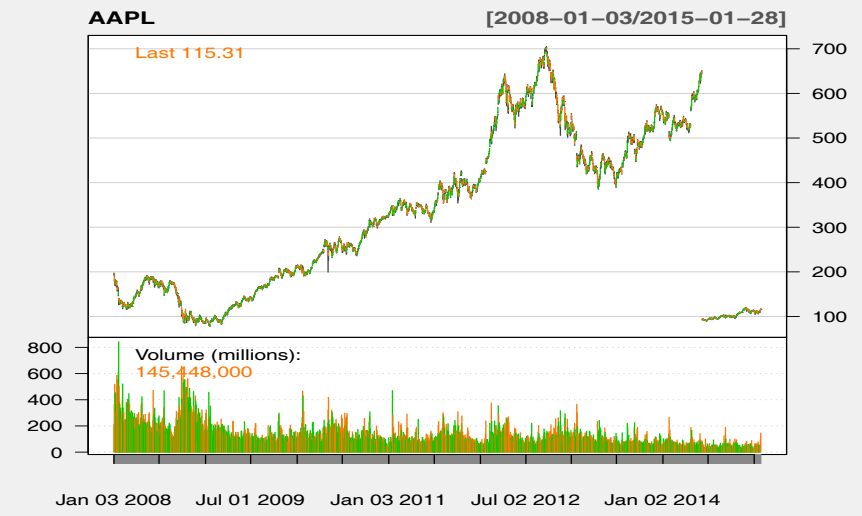

Figure 1: Time plots of daily closing price and trading volume of Apple stock from January 3, 2008 to January 28, 2015.

```
2015-01-29     33.84     34.28     33.46      34.21    28890909

> getSymbols("^TNX")
[1] "TNX"
> head(TNX)
           TNX.Open TNX.High TNX.Low TNX.Close TNX.Volume TNX.Adjusted
2007-01-03     4.66     4.69    4.64      4.66          0          4.66
2007-01-04     4.66     4.66    4.60      4.62          0          4.62
2007-01-05     4.59     4.70    4.58      4.65          0          4.65
2007-01-08     4.67     4.68    4.65      4.66          0          4.66
2007-01-09     4.66     4.67    4.64      4.66          0          4.66
2007-01-10     4.67     4.70    4.66      4.68          0          4.68
> tail(TNX)
           TNX.Open TNX.High TNX.Low TNX.Close TNX.Volume TNX.Adjusted
2015-01-21     1.80     1.86    1.77      1.85          0          1.85
2015-01-22     1.94     1.95    1.81      1.90          0          1.90
2015-01-23     1.82     1.85    1.80      1.82          0          1.82
2015-01-26     1.80     1.84    1.80      1.83          0          1.83
2015-01-27     1.79     1.83    1.75      1.83          0          1.83
2015-01-28     1.82     1.83    1.72      1.72          0          1.72
> chartSeries(TNX,theme="white",TA=NULL) % Obtain time plot without trading volume
```

Remark: The **Quantmod** package updates financial data daily. The default option of `getSymbols` is to download data to the most recent one available.

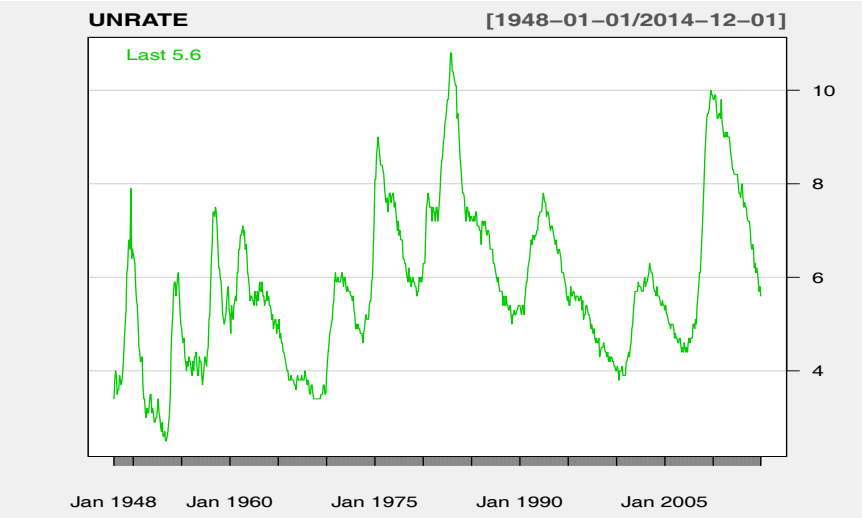

Figure 2: Time plot of U.S. monthly unemployment rates from January 1948 to December 2014.

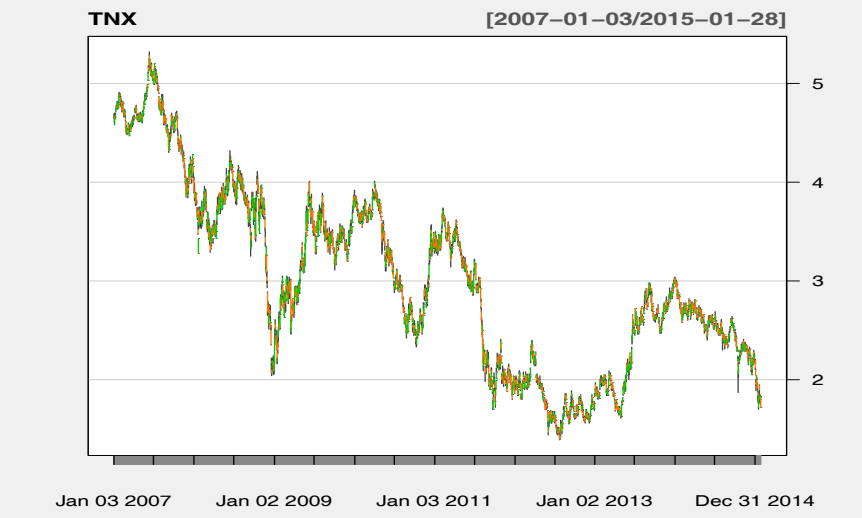

Figure 3: Time plot of Chicago Board Options Exchange interest rates of 10-year treasury note from January 3, 2007 to January 28, 2015.

0.3 Some Basic R commands

After starting R, the first thing to do is to set the working directory. By working directory, we mean the computer directory where data sets reside and output will be stored. This can be done in two ways. The first method is to click on the command `File`. A pop-up window appears that allows one to select the desired directory. The second method is to type in the desired directory in the R Console using the command `setwd`, which stands for set working directory. See the demonstration below.
R is an object oriented program. It handles many types of object. For the purposes of the course, we do not need to study details of an object in R. Explanations will be given when needed. It suffices now to say that R allows one to assign values to variables and refer to them by names. The assignment operator is $\leftarrow$, but $=$ can also be used. For instance, x $\leftarrow$ 10 assigns the value 10 to the variable "x". Here R treats "x" as a sequence of real numbers with the first element being 10. There are several ways to load data into the R working space, depending on the data format. For simple text data, the command is `read.table`. For `*.csv` files, the command is `read.csv`. The data file is specified in either a single or double quotes; see the R demonstration. R treats the data as an object and refer to them by the assigned name. For both loading commands, R stores the data in a matrix framework. As such, one can use the command `dim` (i.e., dimension) to see the size of the data. Finally, the basic operations in R are similar to those we commonly use and the command to exit R is `q()`.

R Demonstration

```
> setwd("C:/Users/rst/teaching/bs41202/sp2016") % Set my working directory
> x <- 10  % Assign value, here "x" is a variable.
> x  % See the value of x.
[1] 10 % Here [1] signifies the first element.
> 1+2 % Basic operation: addition
[1] 3
> 10/2 % Basic operation: division
[1] 5
% Use * and ^ for multiplication and power, respectively.
% Use log for the natural logarithm.
> da=read.table('d-ibm-0110.txt',header=T) % Load text data with names.
> head(da) % See the first 6 rows
      date    return
1 20010102 -0.002206
2 20010103  0.115696
 ....
6 20010109 -0.010688
> dim(da)  % Dimension of the data object "da".
[1] 2515    2
> da <- read.csv("d-vix0411.csv",header=T) % Load csv data with names.
> head(da)  % See the first 6 rows
      Date VIX.Open VIX.High VIX.Low VIX.Close
```

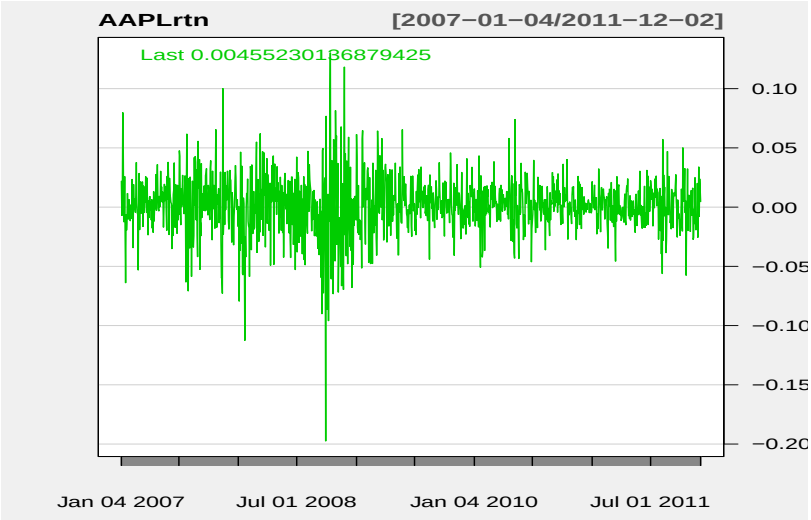

Figure 4: Time plot of daily log returns of Apple stock from January 4, 2008 to January 28, 2015.

```
1 1/2/2004    17.96    18.68   17.54     18.22
2 1/5/2004    18.45    18.49   17.44     17.49
  ....
6 1/9/2004    16.15    16.88   15.57     16.75
```

1 Examples of Financial Data

In this section, we examine some of the return series in finance. Figure **??** shows the time plot of daily log returns of Apple stock from January 4, 2007 to January 28, 2015. As defined before, daily log returns are simply the change series of log prices. In R, a change series can easily be obtained by taking the *difference* of the log prices. Specifically, $r_t = \ln(P_1) - \ln(P_{t-1})$, where P_t is the stock price at time t. Note that in the demonstration, I used *adjusted* daily price to compute log returns because adjusted price takes into consideration the stock splits, if any, during the sample period. From the plot, we see that (a) there exist some large outlying observations and (b) the returns were volatile in certain periods, but stable in others. The latter characteristic is referred to as volatility clustering in asset returns. The former, on the other hand, are indicative that the returns have heavy tails.
Figure **??** shows the time plot of daily changes in yield-to-maturity (YTM) of the 10-year Treasury notes also from January 4, 2008 to January 28, 2015. The changes in YTM exhibit similar characteristics as those of daily returns of Apple stock. Figure 6 provides the time plot of daily log returns of the Dollar-Euro exchange rate. Again, the log returns of exchange rates have the same features as those of the daily log returns of stock. The daily Dollar-Euro exchange rate is given in Figure 7. The exchange rates are downloaded from the database FRED.

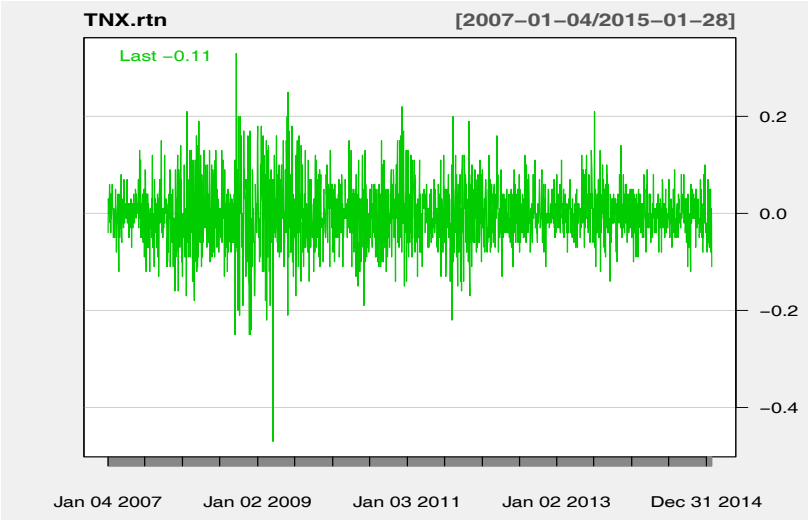

Figure 5: Time plot of daily changes in the yield to maturity for the U.S. 10-year Treasury notes from January 4, 2007 to January 28, 2015.

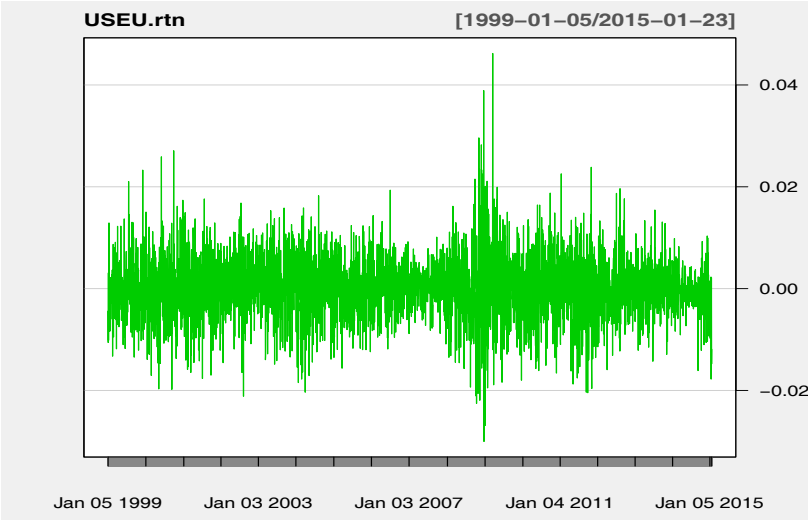

Figure 6: Time plot of daily log returns of the Dollar-Euro exchange rates from January 5, 1999 to January 23, 2015. The rate is dollars per Euro.

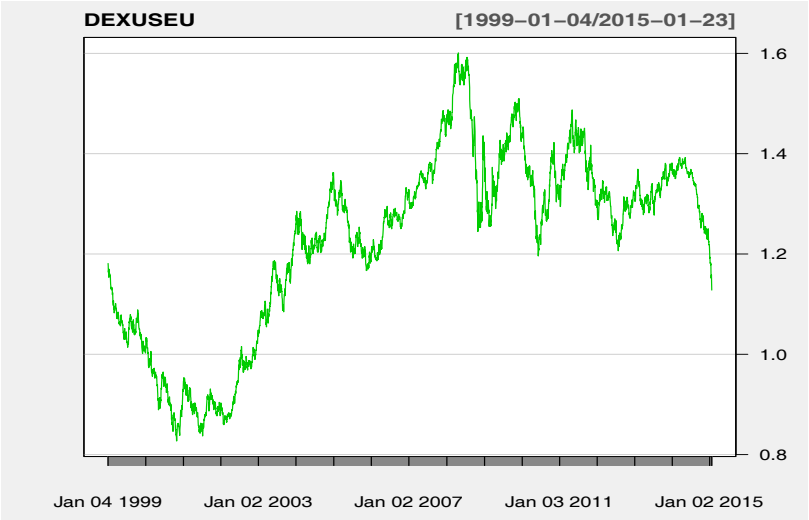

Figure 7: Time plot of daily Dollar-Euro exchange rates from January 4, 1999 to January 23, 2015. The rate is dollars per Euro.

R Demonstration

```
> require(quantmod)
> getSymbols("AAPL",from="2008-01-03",to="2015-01-28") %Specify period
[1] "AAPL"
> AAPL.rtn=diff(log(AAPL$AAPL.Adjusted)) % Compute log returns
> chartSeries(AAPL.rtn,theme="white")
> getSymbols("^TNX",from="2007-01-03",to="2015-01-28")
[1] "TNX"

> TNX.rtn=diff(TNX$TNX.Adjusted) % Compute changes
> chartSeries(TNX.rtn,theme="white")
> getSymbols("DEXUSEU",src="FRED") %Obtain exchange rates from FRED
[1] "DEXUSEU"
> head(DEXUSEU)
           DEXUSEU
1999-01-04  1.1812
1999-01-05  1.1760
 ....
1999-01-11  1.1534
> tail(DEXUSEU)
           DEXUSEU
2011-12-09  1.3368
 ....
2011-12-16  1.3025
> USEU.rtn=diff(log(DEXUSEU$DEXUSEU))
> chartSeries(DEXUSEU,theme="white")
```

```
> chartSeries(USEU.rtn,theme="white")
```