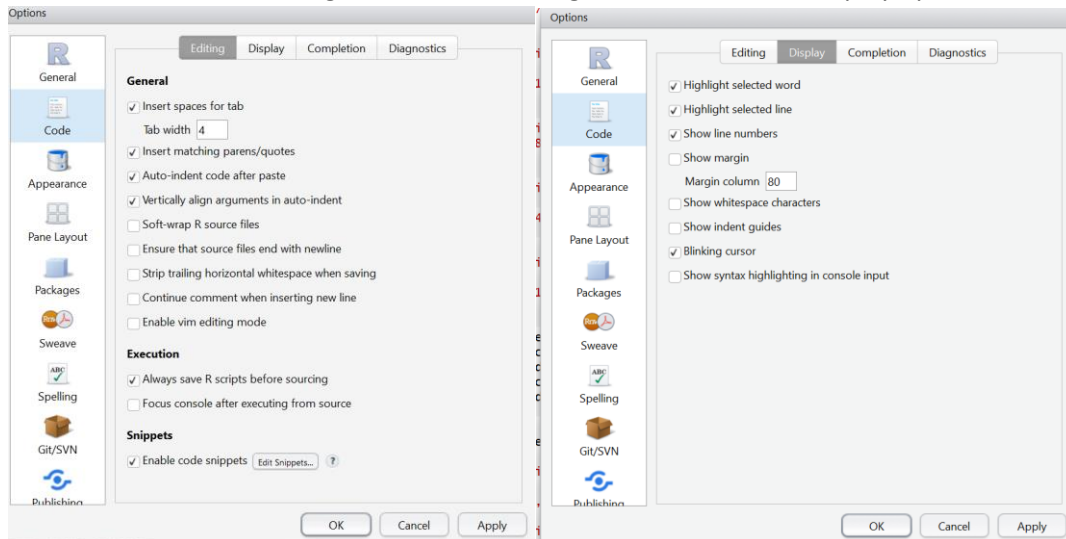# Assignment 1:

- by Sandeep Joshi

Question 1 – Installing IDE and packages:

1. Installed R
2. Installed RStudio and changed the editor settings for tab and other display options.

3. Installed package fBasisc.

```
> install.packages("fBasics")
Installing package into 'C:/Users/joshi/Documents/R/win-library/3.2'
(as 'lib' is unspecified)
also installing the dependencies 'timeDate', 'timeSeries', 'gss', 'stabledis
t'

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.2/timeDate_3012.1
00.zip'
Content type 'application/zip' length 790561 bytes (772 KB)
downloaded 772 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.2/timeSeries_3012
.99.zip'
Content type 'application/zip' length 1586242 bytes (1.5 MB)
downloaded 1.5 MB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.2/gss_2.1-5.zip'
Content type 'application/zip' length 868673 bytes (848 KB)
downloaded 848 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.2/stabledist_0.7-
0.zip'
Content type 'application/zip' length 41309 bytes (40 KB)
downloaded 40 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.2/fBasics_3011.87
.zip'
Content type 'application/zip' length 1552658 bytes (1.5 MB)
downloaded 1.5 MB

package 'timeDate' successfully unpacked and MD5 sums checked
package 'timeSeries' successfully unpacked and MD5 sums checked
package 'gss' successfully unpacked and MD5 sums checked
package 'stabledist' successfully unpacked and MD5 sums checked
package 'fBasics' successfully unpacked and MD5 sums checked
```

4. Load package fBasics

```
> library(fBasics)
Loading required package: timeDate
Loading required package: timeSeries

Attaching package: 'timeSeries'

The following object is masked from 'package:zoo':

    time<-



Rmetrics Package fBasics
Analysing Markets and calculating Basic Statistics
Copyright (C) 2005-2014 Rmetrics Association Zurich
Educational Software for Financial Engineering and Computational Science
Rmetrics is free software and comes with ABSOLUTELY NO WARRANTY.
https://www.rmetrics.org --- Mail to: info@rmetrics.org

Attaching package: 'fBasics'

The following object is masked from 'package:TTR':

    volatility
```

5. Install package quantmod

```
> install.packages("quantmod")
Installing package into 'C:/Users/joshi/Documents/R/win-library/3.2'
(as 'lib' is unspecified)
also installing the dependencies 'xts', 'zoo', 'TTR'

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.2/xts_0.9-7.zip'
Content type 'application/zip' length 662446 bytes (646 KB)
downloaded 646 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.2/zoo_1.7-12.zip'
Content type 'application/zip' length 897113 bytes (876 KB)
downloaded 876 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.2/TTR_0.23-0.zip'
Content type 'application/zip' length 280554 bytes (273 KB)
downloaded 273 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.2/quantmod_0.4-5.
zip'
Content type 'application/zip' length 473485 bytes (462 KB)
downloaded 462 KB

package 'xts' successfully unpacked and MD5 sums checked
package 'zoo' successfully unpacked and MD5 sums checked
package 'TTR' successfully unpacked and MD5 sums checked
package 'quantmod' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\joshi\AppData\Local\Temp\Rtmpy4ohJV\downloaded_packages
```

6. Load package quantmod

```
> library(quantmod)
Loading required package: xts
Loading required package: zoo

Attaching package: 'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric

Loading required package: TTR
Version 0.4-0 included new data defaults. See ?getSymbols.
> ?getSymbols
```
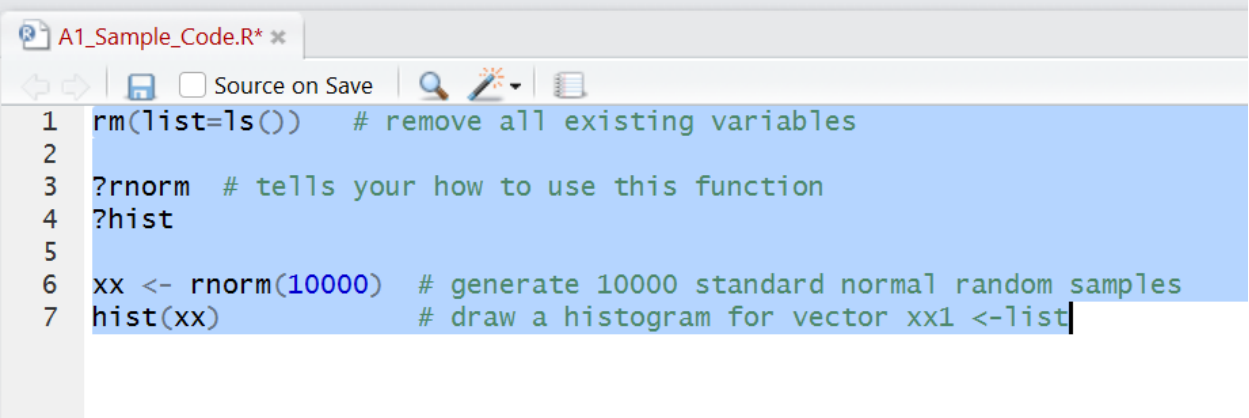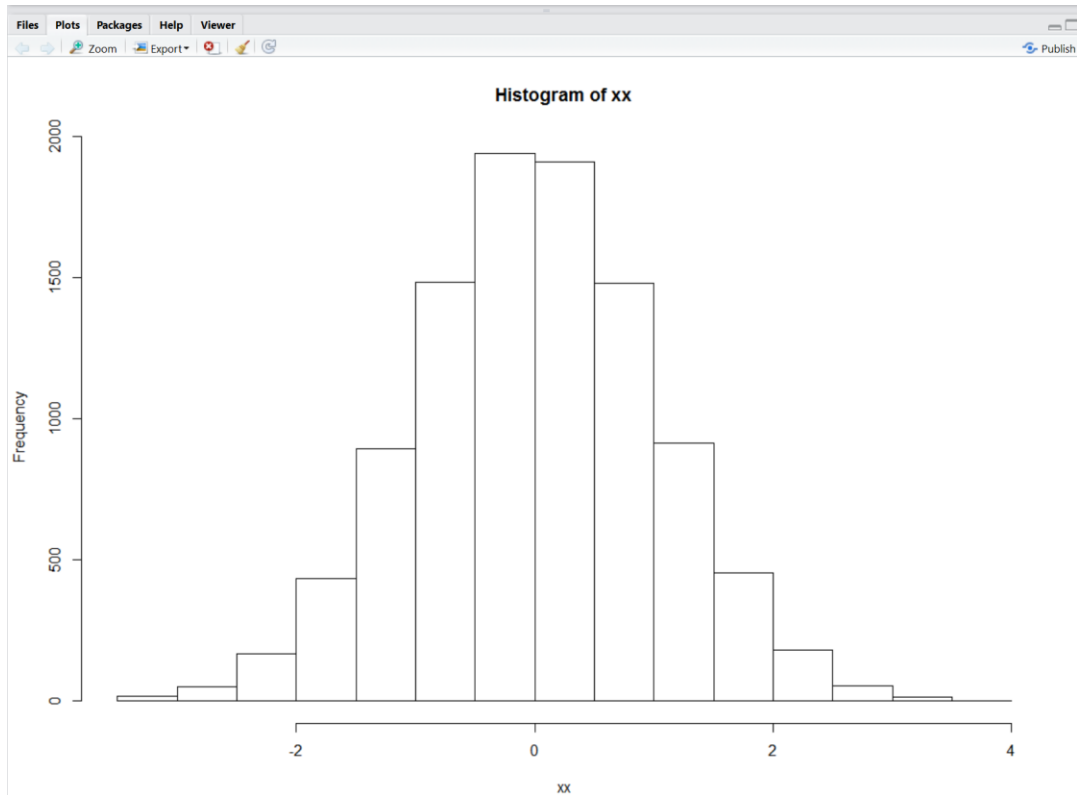
7. Copied code and ran it.

```
A1_Sample_Code.R* ×

    Source on Save    🔍  ✏️ ▾  📋

1  rm(list=ls())    # remove all existing variables
2
3  ?rnorm  # tells your how to use this function
4  ?hist
5
6  xx <- rnorm(10000)   # generate 10000 standard normal random samples
7  hist(xx)             # draw a histogram for vector xx1 <-list
```

Histogram of xx

8. Ran code '?basicStats and learnt the overview.

BasicStatistics {fBasics}   R Documentation

Basic Time Series Statistics

Description

Computes basic financial time series statististics.

List of Functions:

`basicStats` Computes an overview of basic statistical values.

Usage

```
basicStats(x, ci = 0.95)
```

Arguments

ci confidence interval, a numeric value, by default 0.95, i.e. 95 percent.

x   an object of class `"timeSeries"` or any other object which can be transformed by the function `as.timeSeries` into an object of class `timeSeries`. The latter case, other then `timeSeries` objects, is more or less untested.

9. Ran code '??basicStats' which searches for the term in an embedded web browser and eventually yields same result as before.

Question 2 – Basic Data Types:

Three ways to create vector:

```
12
13   # Three ways to create vector
14   way1 <- 1:3   # way 1
15   print(way1)
16   v1 <- seq(-10,10, by = 0.1)
17   print(v1)
18   v2 <- rep(c('0','1','2'),3)
19   print(v2)
```

```
> print(way1)
[1] 1 2 3
> v1 <- seq(-10,10, by = 0.1)
> print(v1)
  [1] -10.0  -9.9  -9.8  -9.7  -9.6  -9.5  -9.4  -9.3  -9.2  -9.1
 [11]  -9.0  -8.9  -8.8  -8.7  -8.6  -8.5  -8.4  -8.3  -8.2  -8.1
 [21]  -8.0  -7.9  -7.8  -7.7  -7.6  -7.5  -7.4  -7.3  -7.2  -7.1
 [31]  -7.0  -6.9  -6.8  -6.7  -6.6  -6.5  -6.4  -6.3  -6.2  -6.1
 [41]  -6.0  -5.9  -5.8  -5.7  -5.6  -5.5  -5.4  -5.3  -5.2  -5.1
 [51]  -5.0  -4.9  -4.8  -4.7  -4.6  -4.5  -4.4  -4.3  -4.2  -4.1
 [61]  -4.0  -3.9  -3.8  -3.7  -3.6  -3.5  -3.4  -3.3  -3.2  -3.1
 [71]  -3.0  -2.9  -2.8  -2.7  -2.6  -2.5  -2.4  -2.3  -2.2  -2.1
 [81]  -2.0  -1.9  -1.8  -1.7  -1.6  -1.5  -1.4  -1.3  -1.2  -1.1
 [91]  -1.0  -0.9  -0.8  -0.7  -0.6  -0.5  -0.4  -0.3  -0.2  -0.1
[101]   0.0   0.1   0.2   0.3   0.4   0.5   0.6   0.7   0.8   0.9
[111]   1.0   1.1   1.2   1.3   1.4   1.5   1.6   1.7   1.8   1.9
[121]   2.0   2.1   2.2   2.3   2.4   2.5   2.6   2.7   2.8   2.9
[131]   3.0   3.1   3.2   3.3   3.4   3.5   3.6   3.7   3.8   3.9
[141]   4.0   4.1   4.2   4.3   4.4   4.5   4.6   4.7   4.8   4.9
[151]   5.0   5.1   5.2   5.3   5.4   5.5   5.6   5.7   5.8   5.9
[161]   6.0   6.1   6.2   6.3   6.4   6.5   6.6   6.7   6.8   6.9
[171]   7.0   7.1   7.2   7.3   7.4   7.5   7.6   7.7   7.8   7.9
[181]   8.0   8.1   8.2   8.3   8.4   8.5   8.6   8.7   8.8   8.9
[191]   9.0   9.1   9.2   9.3   9.4   9.5   9.6   9.7   9.8   9.9
[201]  10.0
> v2 <- rep(c('0','1','2'),3)
> print(v2)
[1] "0" "1" "2" "0" "1" "2" "0" "1" "2"
```

Explicit Coercion:

```
21   # Explicit coercion
22   # Convert v2 to numeric
23   v2num <- as.numeric(v2)
24   print(v2num)
25   # Convert v2 to logical
26   v2NA <- as.logical(v2)
27   print(v2NA)
28   # Convert v2num to logical
29   v2logical <- as.logical(v2num)
30   print(v2logical)
```

```
> # Explicit coercion
> # Convert v2 to numeric
> v2num <- as.numeric(v2)
> print(v2num)
[1] 0 1 2 0 1 2 0 1 2
> # Convert v2 to logical
> v2NA <- as.logical(v2)
> print(v2NA)
[1] NA NA NA NA NA NA NA NA NA
> # Convert v2num to logical
> v2logical <- as.logical(v2num)
> print(v2logical)
[1] FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE
```

Conclusion: logical conversion didn't work for string vector but it worked for numeric vector with all non-zeros as True and zeros as False.

Matrix and List:

Create matrix m2

```
32   # Matrix and List
33   # 3x3 matrix
34   m2 <- matrix(v2num, nrow = 3, byrow = T)
```

```
> print(m2)
     [,1] [,2] [,3]
[1,]    4    7    2
[2,]    0    1    2
[3,]    0    1    7
```

Inverse m2

```
35   # Calculating inverse of m2
36   solve(m2)
37   # Doesn't work as [m2] yields 0, so changing few enteries
38   m2[1, 1] = 4
39   m2[3, 3] = 7
40   solve(m2)
```

```
> solve(m2)
Error in solve.default(m2) :
  Lapack routine dgesv: system is exactly singular: U[1,1] = 0
> m2[1, 1] = 4
> m2[1, 2] = 7
> solve(m2)
Error in solve.default(m2) :
  Lapack routine dgesv: system is exactly singular: U[3,3] = 0
> m2[1, 1] = 4
> m2[3, 3] = 7
> solve(m2)
     [,1]  [,2] [,3]
[1,] 0.25 -2.35  0.6
[2,] 0.00  1.40 -0.4
[3,] 0.00 -0.20  0.2
```

List and Subsets:

```
43   # My First List
44   myFirstList <- list(char = v2, integer = v2num, NAs = v2NA,
45                       bool = v2logical, mat = m2)
46   # Subseting 4th element by indexing
47   myFirstList[4]
48   # Subseting 4th element by element name
49   myFirstList["bool"]
```

```
> # Subseting 4th element by indexing
> myFirstList[4]
$bool
[1] FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE

> # Subseting 4th element by element name
> myFirstList["bool"]
$bool
[1] FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE
```

Question 3 - Loops:

Fibonacci series

```
52
53   # Fibonacci
54   # Wrong one as nth term <> (n-1)th term + (n-2)th term
55   a <- 0
56   b <- 1
57   MAX = 1000
58   while (TRUE)
59 - {
60       print (a)
61       if (a > MAX)
62           break
63       a <- b
64       b <- a + b
65   }
66
67   # Right one as given in the code but goes beyond MAX
68   a <- 0
69   b <- 1
70   MAX = 10000
71   while(TRUE)
72 - {
73       print (a)
74       if(a > MAX)
75           break
76       tmp <- a
77       a <- b
78       b <- tmp + b
79   }
80
81   # learning repeat
82   ?Repeat
83   # Fibonacci using repeat. Doesn't go be beyond MAX specified
84   MAX <- 10000
85   a <- 0
86   b <- 1|
87   repeat
88 - {
89       if(a > MAX)
90           break
91       print (a)
92       tmp <- a
93       a <- b
94       b <- tmp + b
95   }
96
```

```
+ }
[1] 0
[1] 1
[1] 2
[1] 4
[1] 8
[1] 16
[1] 32
[1] 64
[1] 128
[1] 256
[1] 512
[1] 1024
>
```

```
+ }
[1] 0
[1] 1
[1] 1
[1] 2
[1] 3
[1] 5
[1] 8
[1] 13
[1] 21
[1] 34
[1] 55
[1] 89
[1] 144
[1] 233
[1] 377
[1] 610
[1] 987
[1] 1597
[1] 2584
[1] 4181
[1] 6765
[1] 10946
```

```
[1] 0
[1] 1
[1] 1
[1] 2
[1] 3
[1] 5
[1] 8
[1] 13
[1] 21
[1] 34
[1] 55
[1] 89
[1] 144
[1] 233
[1] 377
[1] 610
[1] 987
[1] 1597
[1] 2584
[1] 4181
[1] 6765
```