RMSC4002 Tutorial 1

September 14, 2017

1. Introduction to R

1.1 How to get R?

R can be freely downloaded from https://www.r-project.org.

RStudio is highly recommended because it is more user-friendly than R.

1.2 Getting help in R

You can use the commands "help(function)" or "?function".

1.3 Writing comments in R

You can add "#" before the statement. Comments can help others understand the code.

1.4 Setting working directory in R

It is necessary to set the working directory in R if you want to input some datasets.

There are two ways:

(1) By Menus:

Click File −> Change dir −> choose your location (in R)

Click Session −> Set working directory −> choose directory (in RStudio)

(2) By Commands:

Type setwd("F:/Tuto notes") (assuming your location is F:\Tuto notes)

2.

```
> d<-read.csv("fin-ratio.csv")            # read in dataset
> names(d)                                # display the var in d
[1] "EY"    "CFTP"  "ln_MV" "DY"    "BTME"  "DTE"   "HSI"
> x<-d[,1:6]                              # extract the first 6  columns in d
> m<-apply(x,2,mean)                      # save sample mean vector to m
> m
        EY       CFTP      ln_MV         DY       BTME        DTE
-0.6502403 -0.2338956  6.2668068  2.4961735  1.9082626  0.7097322

> S<-var(x)                               # save sample covariance matrix
> S
              EY      CFTP      ln_MV         DY       BTME        DTE
EY    18.4979068 2.9089644  1.1601886  1.9203766  1.4781279  0.3379530
CFTP   2.9089644 3.6930613  0.7662995  1.2371466  1.8228390  0.3287908
ln_MV  1.1601886 0.7662995  2.7439362  0.9720714 -0.7734227 -0.0741322
DY     1.9203766 1.2371466  0.9720714 13.8715626 -0.2575337  0.1581528
BTME   1.4781279 1.8228390 -0.7734227 -0.2575337 68.3081966  1.9617652
DTE    0.3379530 0.3287908 -0.0741322  0.1581528  1.9617652 12.9929072

> cor(x)                                  # sample correlation matrix
              EY       CFTP       ln_MV          DY        BTME         DTE
EY    1.00000000 0.35195234  0.16284719  0.11988433  0.04158285  0.02179926
CFTP  0.35195234 1.00000000  0.24072338  0.17284835  0.11476743  0.04746497
ln_MV 0.16284719 0.24072338  1.00000000  0.15756091 -0.05649285 -0.01241557
DY    0.11988433 0.17284835  0.15756091  1.00000000 -0.00836633  0.01178043
BTME  0.04158285 0.11476743 -0.05649285 -0.00836633  1.00000000  0.06585025
DTE   0.02179926 0.04746497 -0.01241557  0.01178043  0.06585025  1.00000000
```

3.

```
> options(digits=4)                          # control display to 4 decimals
> det(solve(S))                              # det of inverse of S
[1] 5.706e-07
> 1/det(S)                                   # 1/det(S)
[1] 5.706e-07
> eig<-eigen(S)                              # save eigenvalues and vector of S
> names(eig)                                 # display items in eig
[1] "values"  "vectors"
> eval<-eig$values                           # save eigenvalues
> eval                                       # display eigenvalues
[1] 68.487 19.918 13.205 12.899  3.341  2.257

> H<-eig$vectors                             # save matrix of eigenvector
> H                                          # display H
          [,1]     [,2]      [,3]      [,4]     [,5]     [,6]
[1,]  0.031107  0.91185  0.364402  0.016136  0.18218 -0.03637
[2,]  0.029477  0.19142 -0.018447  0.005915 -0.81898  0.53980
[3,] -0.010938  0.09104 -0.043829 -0.020125 -0.53213 -0.84030
[4,] -0.003038  0.34621 -0.911976 -0.188393  0.11223  0.01856
[5,]  0.998380 -0.03383 -0.007556 -0.036516  0.01255 -0.02334
[6,]  0.035663  0.05094 -0.182190  0.981058  0.01304 -0.01720

> round(t(H)%*%H,3)                  # H' H=I, H is orthogonal, HH' =I as well
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    0    0    0    0    0
[2,]    0    1    0    0    0    0
[3,]    0    0    1    0    0    0
[4,]    0    0    0    1    0    0
[5,]    0    0    0    0    1    0
[6,]    0    0    0    0    0    1
> h1<-H[,1]                                  # extract first column of H to h1
> eval[1]*h1                                 # compute lambda1*h1
[1]  2.1304  2.0188 -0.7491 -0.2080 68.3765  2.4425
```

```
> t(S%*%h1)                                  # compute (S*h1)'
        EY  CFTP   ln_MV      DY  BTME   DTE
[1,] 2.130 2.019 -0.7491 -0.2080 68.38 2.442
> round(t(H)%*%S%*%H,3)                      # compute H' SH (should = D)
      [,1]  [,2]  [,3] [,4]  [,5]  [,6]
[1,] 68.49  0.00  0.00  0.0 0.000 0.000
[2,]  0.00 19.92  0.00  0.0 0.000 0.000
[3,]  0.00  0.00 13.21  0.0 0.000 0.000
[4,]  0.00  0.00  0.00 12.9 0.000 0.000
[5,]  0.00  0.00  0.00  0.0 3.341 0.000
[6,]  0.00  0.00  0.00  0.0 0.000 2.257
> D<-diag(eval)                              # form diagonal matrix D
> H%*%D%*%t(H)                               # compute HDH' (should = S)
        [,1]   [,2]     [,3]    [,4]    [,5]     [,6]
[1,] 18.4979 2.9090  1.16019  1.9204  1.4781  0.33795
[2,]  2.9090 3.6931  0.76630  1.2371  1.8228  0.32879
[3,]  1.1602 0.7663  2.74394  0.9721 -0.7734 -0.07413
[4,]  1.9204 1.2371  0.97207 13.8716 -0.2575  0.15815
[5,]  1.4781 1.8228 -0.77342 -0.2575 68.3082  1.96177
[6,]  0.3380 0.3288 -0.07413  0.1582  1.9618 12.99291
> rS<-H%*%sqrt(D)%*%t(H)                     # H*sqrt(D)*H'
> rS
        [,1]   [,2]     [,3]     [,4]     [,5]     [,6]
[1,] 4.26492 0.4603  0.17720  0.22591  0.11267  0.03736
[2,] 0.46026 1.8359  0.19270  0.19921  0.17666  0.05180
[3,] 0.17720 0.1927  1.62490  0.16722 -0.08301 -0.01540
[4,] 0.22591 0.1992  0.16722  3.70834 -0.02570  0.02000
[5,] 0.11267 0.1767 -0.08301 -0.02570  8.26013  0.16422
[6,] 0.03736 0.0518 -0.01540  0.02000  0.16422  3.60017
> rS%*%rS                                      # rS*rS
```