Introduction to R

Hendro Sugandi

Contents

1 Packages

```r
# see loaded packages
(.packages())
```

```
## [1] "stats"     "graphics"  "grDevices" "utils"     "datasets"  "methods"
## [7] "base"
```

```r
# # see (the first 10) function within stats package
# ls("package:stats")[1:10]

# # ?, help() operator, access to the documentation pages for R functions,
# # data sets, and other objects
# ?acf

# Download and install packages
# install.packages('tidyverse')

# load and attach add-on packages
library(tidyverse)

# ?group_by
# ?library
# or use function: help()
```

2 Data type

```
# vector
vec_num = c(1,2,3,4) #numeric
class(vec_num)
```

```
## [1] "numeric"
```

```
vec_num = 1:3 #integer
class(vec_num)
```

```
## [1] "integer"
```

```
vec_char = c("a","b","c") # character
class(vec_char)
```

```
## [1] "character"
```

```
# matrix
matrix_a = matrix(1:9, nrow = 3, ncol = 3, byrow = TRUE)
class(matrix_a)
```

```
## [1] "matrix" "array"
```

```
# data frame
df_a = data.frame(id1 = c(111,222,333),
                  id2 = c("a", "b", "c"),
                  age = c(20,30,40))
View(df_a)
class(df_a)
```

```
## [1] "data.frame"
```

```
# select or remove
df_a[2:3,] # select only the second and third rows
```

```
##   id1 id2 age
## 2 222   b  30
## 3 333   c  40
```

```
df_a[-c(1),] # exclude the first row
```

```
##   id1 id2 age
## 2 222   b  30
## 3 333   c  40
```

```
df_a[, 1] # select the first column
```

```
## [1] 111 222 333
```

```
# list
list_a = list(vec_num = vec_num, vec_char, matrix_a, df_a)
View(list_a)
class(list_a)
```

```
## [1] "list"
```

```
list_a$vec_num
```

```
## [1] 1 2 3
```

```
list_a[[1]] # member reference
```

```
## [1] 1 2 3
```

```
list_a[1] # list slice
```

```
## $vec_num
## [1] 1 2 3
```

```
class(list_a[[1]] )
```

```
## [1] "integer"
```

```
class(list_a[1])
```

```
## [1] "list"
```

```
# or use function: str(), to check data structure
str(list_a)
```

```
## List of 4
##  $ vec_num: int [1:3] 1 2 3
##  $        : chr [1:3] "a" "b" "c"
##  $        : int [1:3, 1:3] 1 4 7 2 5 8 3 6 9
##  $        :'data.frame': 3 obs. of  3 variables:
##   ..$ id1: num [1:3] 111 222 333
##   ..$ id2: chr [1:3] "a" "b" "c"
##   ..$ age: num [1:3] 20 30 40
```

3 Code Snippets

Used for quickly inserting common snippets of code, for example:

- fun (function)
- mat (matrix)
- if, el, and ei (conditional expressions)
- for (for loop)

```
# code snippets example: typing "for", then press tab
# for (variable in vector) {
#
# }
```

4 Load and save

```
# check current working directory
# getwd()

# set working directory using the following function
# setwd(dir) # e.g. "C:/Users/.../"

# read csv
FF5 <- read.csv(file = "F-F_Research_Data_5_Factors_2x3.csv")
# FF5 <- read.csv(file = "C:/Users/.../F-F_Research_Data_5_Factors_2x3.csv")
# OR
# FF5 <- read.csv(file = "C:\\Users\\...\\F-F_Research_Data_5_Factors_2x3.csv")
# save data to RData format
save(FF5, file = "FF5.RData")

# save data to csv format
write.csv(FF5, file = "FF5.csv")

# remove the data
rm(FF5)

# load RData
load("FF5.RData")
```

5 Vector and matrix

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{bmatrix}$$

```
# create vector and matrix
vec1 <- matrix(c(10, 20), nrow = 2, ncol = 1)
vec2 <- c(10,20)
mat1 <- matrix(c(10, 20, 30, 40, 50, 60), nrow = 2) # default: byrow = FALSE
mat2 <- matrix(c(10, 20, 30, 40, 50, 60), nrow = 2, byrow = TRUE)

# check the data dimension
dim(vec1)
```

```
## [1] 2 1
```

```
dim(vec2)
```

```
## NULL
```

```
length(vec2)
```

```
## [1] 2
```

```
dim(mat2)
```

```
## [1] 2 3
```

```
# transpose
t(vec1)
```

```
##      [,1] [,2]
## [1,]   10   20
```

```
t(vec2)
```

```
##      [,1] [,2]
## [1,]   10   20
```

```
class(vec2); class(t(vec2))
```

```
## [1] "numeric"
```

```
## [1] "matrix" "array"
```

```
t(mat2)
```

```
##      [,1] [,2]
## [1,]   10   40
## [2,]   20   50
## [3,]   30   60
```

```
# add
vec1+vec2
```

```
##      [,1]
## [1,]   20
## [2,]   40
```

```
mat1+mat2
```

```
##      [,1] [,2] [,3]
## [1,]   20   50   80
## [2,]   60   90  120
```

```
# multiplication with constant
2*vec1
```

```
##      [,1]
## [1,]   20
## [2,]   40
```

```
2*mat2
```

```
##      [,1] [,2] [,3]
## [1,]   20   40   60
## [2,]   80  100  120
```

```
# matrix multiplication
vec2 %*% t(vec2)
```

```
##      [,1] [,2]
## [1,]  100  200
## [2,]  200  400
```

```
vec1 %*% t(vec1)
```

```
##      [,1] [,2]
## [1,]  100  200
## [2,]  200  400
```

```
t(vec1) %*% mat1
```

```
##      [,1] [,2] [,3]
## [1,]  500 1100 1700
```

```
mat1 %*% t(mat2)
```

```
##      [,1] [,2]
## [1,] 2200 4900
## [2,] 2800 6400
```

```
t(mat1) %*% mat2
```

```
##      [,1] [,2] [,3]
## [1,]  900 1200 1500
## [2,] 1900 2600 3300
## [3,] 2900 4000 5100
```

```
# element wise multiplication
mat1 * mat1
```

```
##      [,1] [,2] [,3]
## [1,]  100  900 2500
## [2,]  400 1600 3600
```

$$x^tYx = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

```
# FF5 returns are in percentage
FF5 <- read.csv(file = "F-F_Research_Data_5_Factors_2x3.csv")
x_vec <- matrix(c(0.2, 0.8), nrow = 2)
sigma_mat <- cov(FF5 %>% select(SMB, HML)) # using dplyr
sigma_mat2 <- cov(FF5[, c(3,4)] )

# variance of portfolio
t(x_vec) %*% sigma_mat %*% x_vec
```

```
##         [,1]
## [1,] 5.71319
```

```
var(FF5$SMB*0.2+FF5$HML*0.8)
```

```
## [1] 5.71319
```

Inverse

```
solve(sigma_mat)
```

```
##             SMB         HML
## SMB 0.108879024 0.002633249
## HML 0.002633249 0.118340274
```

```
round(solve(sigma_mat) %*% sigma_mat, 2) # identity matrix
```

```
##     SMB HML
## SMB   1   0
## HML   0   1
```

6 Example 1: summary statistics, function, and plot

- Load and check data
- Calculate summary statistics
- Create function
- Create plot

```
# read data
FF5 <- read.csv(file = "F-F_Research_Data_5_Factors_2x3.csv")
# check first n row
head(FF5)
```

```
##     date Mkt.RF   SMB   HML   RMW   CMA   RF
## 1 196307  -0.39 -0.45 -0.94  0.66 -1.15 0.27
## 2 196308   5.07 -0.82  1.82  0.40 -0.40 0.25
## 3 196309  -1.57 -0.48  0.17 -0.76  0.24 0.27
## 4 196310   2.53 -1.30 -0.04  2.75 -2.24 0.29
## 5 196311  -0.85 -0.85  1.70 -0.45  2.22 0.27
## 6 196312   1.83 -1.90 -0.06  0.07 -0.30 0.29
```

```r
head(FF5, n = 10)
```

```
##      date Mkt.RF   SMB   HML   RMW   CMA   RF
## 1  196307  -0.39 -0.45 -0.94  0.66 -1.15 0.27
## 2  196308   5.07 -0.82  1.82  0.40 -0.40 0.25
## 3  196309  -1.57 -0.48  0.17 -0.76  0.24 0.27
## 4  196310   2.53 -1.30 -0.04  2.75 -2.24 0.29
## 5  196311  -0.85 -0.85  1.70 -0.45  2.22 0.27
## 6  196312   1.83 -1.90 -0.06  0.07 -0.30 0.29
## 7  196401   2.24  0.08  1.53  0.22  1.50 0.30
## 8  196402   1.54  0.31  2.86  0.06  0.85 0.26
## 9  196403   1.41  1.40  3.37 -2.01  2.93 0.31
## 10 196404   0.10 -1.50 -0.66 -1.35 -1.08 0.29
```

```r
tail(FF5)
```

```
##       date Mkt.RF   SMB   HML   RMW   CMA RF
## 691 202101  -0.03  6.88  2.85 -3.33  4.68  0
## 692 202102   2.78  4.51  7.08  0.09 -1.97  0
## 693 202103   3.08 -0.97  7.40  6.43  3.44  0
## 694 202104   4.93 -3.06 -0.74  2.26 -2.71  0
## 695 202105   0.29  1.27  7.04  2.37  3.20  0
## 696 202106   2.79 -0.22 -7.70 -1.97 -1.03  0
```

```r
colnames(FF5)
```

```
## [1] "date"   "Mkt.RF" "SMB"    "HML"    "RMW"    "CMA"    "RF"
```

```r
dim(FF5); nrow(FF5); ncol(FF5)
```

```
## [1] 696   7
```

```
## [1] 696
```

```
## [1] 7
```

```r
# check data structure
str(FF5)
```

```
## 'data.frame':    696 obs. of  7 variables:
##  $ date  : int  196307 196308 196309 196310 196311 196312 196401 196402 196403 196404 ...
##  $ Mkt.RF: num  -0.39 5.07 -1.57 2.53 -0.85 1.83 2.24 1.54 1.41 0.1 ...
##  $ SMB   : num  -0.45 -0.82 -0.48 -1.3 -0.85 -1.9 0.08 0.31 1.4 -1.5 ...
##  $ HML   : num  -0.94 1.82 0.17 -0.04 1.7 -0.06 1.53 2.86 3.37 -0.66 ...
##  $ RMW   : num  0.66 0.4 -0.76 2.75 -0.45 0.07 0.22 0.06 -2.01 -1.35 ...
##  $ CMA   : num  -1.15 -0.4 0.24 -2.24 2.22 -0.3 1.5 0.85 2.93 -1.08 ...
##  $ RF    : num  0.27 0.25 0.27 0.29 0.27 0.29 0.3 0.26 0.31 0.29 ...
```

```r
# summary statistics
summary(FF5)
```

```
##       date            Mkt.RF              SMB                HML          
##  Min.   :196307   Min.   :-23.2400   Min.   :-14.890   Min.   :-13.9600  
##  1st Qu.:197779   1st Qu.: -1.9450   1st Qu.: -1.492   1st Qu.: -1.3900  
##  Median :199207   Median :  0.9250   Median :  0.090   Median :  0.2400  
##  Mean   :199207   Mean   :  0.5829   Mean   :  0.240   Mean   :  0.2717  
##  3rd Qu.:200634   3rd Qu.:  3.4325   3rd Qu.:  2.040   3rd Qu.:  1.7125  
##  Max.   :202106   Max.   : 16.1000   Max.   : 18.080   Max.   : 12.5800
##       RMW               CMA                RF        
##  Min.   :-18.480   Min.   :-6.8600   Min.   :0.0000  
##  1st Qu.: -0.820   1st Qu.:-1.0300   1st Qu.:0.1500  
##  Median :  0.220   Median : 0.1100   Median :0.3800  
##  Mean   :  0.255   Mean   : 0.2619   Mean   :0.3693  
##  3rd Qu.:  1.252   3rd Qu.: 1.5000   3rd Qu.:0.5100  
##  Max.   : 13.380   Max.   : 9.5600   Max.   :1.3500  
```

```r
# change returns from % to actual
FF5 <- FF5[,c(2:7)]/100

# ways of calculating mean of each column
colMeans(FF5)
```

```
##      Mkt.RF         SMB         HML         RMW         CMA          RF 
## 0.005828879 0.002399713 0.002717385 0.002549713 0.002619253 0.003693103
```

```r
sapply(FF5, mean)
```

```
##      Mkt.RF         SMB         HML         RMW         CMA          RF 
## 0.005828879 0.002399713 0.002717385 0.002549713 0.002619253 0.003693103
```

```r
apply(FF5, MARGIN = 2, FUN = mean)
```

```
##      Mkt.RF         SMB         HML         RMW         CMA          RF 
## 0.005828879 0.002399713 0.002717385 0.002549713 0.002619253 0.003693103
```

```r
c(mean(FF5$Mkt.RF), mean(FF5$SMB), mean(FF5$HML),
  mean(FF5$RMW) , mean(FF5$CMA), mean(FF5$RF))
```

```
## [1] 0.005828879 0.002399713 0.002717385 0.002549713 0.002619253 0.003693103
```

```r
c(mean(FF5[,1]), mean(FF5[,2]), mean(FF5[,3]),
  mean(FF5[,4]), mean(FF5[,5]), mean(FF5[,6]))
```

```
## [1] 0.005828879 0.002399713 0.002717385 0.002549713 0.002619253 0.003693103
```

```r
# using dplyr
FF5 %>%
  summarise_all(.funs = ~ mean(.))
```

```
##        Mkt.RF         SMB         HML         RMW         CMA          RF
## 1 0.005828879 0.002399713 0.002717385 0.002549713 0.002619253 0.003693103
```

```r
# covariance and correlation
round(cov(FF5), 6)
```

```
##            Mkt.RF       SMB       HML       RMW       CMA      RF
## Mkt.RF  0.001984  0.000388 -0.000275 -0.000194 -0.000332 -1.1e-05
## SMB     0.000388  0.000919 -0.000020 -0.000227 -0.000057 -4.0e-06
## HML    -0.000275 -0.000020  0.000845  0.000056  0.000391  7.0e-06
## RMW    -0.000194 -0.000227  0.000056  0.000475 -0.000009  0.0e+00
## CMA    -0.000332 -0.000057  0.000391 -0.000009  0.000400  4.0e-06
## RF     -0.000011 -0.000004  0.000007  0.000000  0.000004  7.0e-06
```

```r
round(cor(FF5), 2)
```

```
##        Mkt.RF   SMB   HML   RMW   CMA    RF
## Mkt.RF   1.00  0.29 -0.21 -0.20 -0.37 -0.10
## SMB      0.29  1.00 -0.02 -0.34 -0.09 -0.04
## HML     -0.21 -0.02  1.00  0.09  0.67  0.09
## RMW     -0.20 -0.34  0.09  1.00 -0.02  0.01
## CMA     -0.37 -0.09  0.67 -0.02  1.00  0.08
## RF      -0.10 -0.04  0.09  0.01  0.08  1.00
```

```r
cor(FF5$Mkt.RF, FF5$SMB)
```

```
## [1] 0.2869758
```

Make average function

```r
f_average <- function(x) {
  N <- length(x)
  average <- sum(x)/N
  return(average)
}

sapply(FF5, f_average)
```

```
##      Mkt.RF         SMB         HML         RMW         CMA          RF
## 0.005828879 0.002399713 0.002717385 0.002549713 0.002619253 0.003693103
```

```r
f_average(x = FF5$Mkt.RF)
```

```
## [1] 0.005828879
```

Using dplyr to calculate yearly average of monthly returns

```r
FF5 <- read.csv(file = "F-F_Research_Data_5_Factors_2x3.csv")

library(lubridate) # for ymd function
```

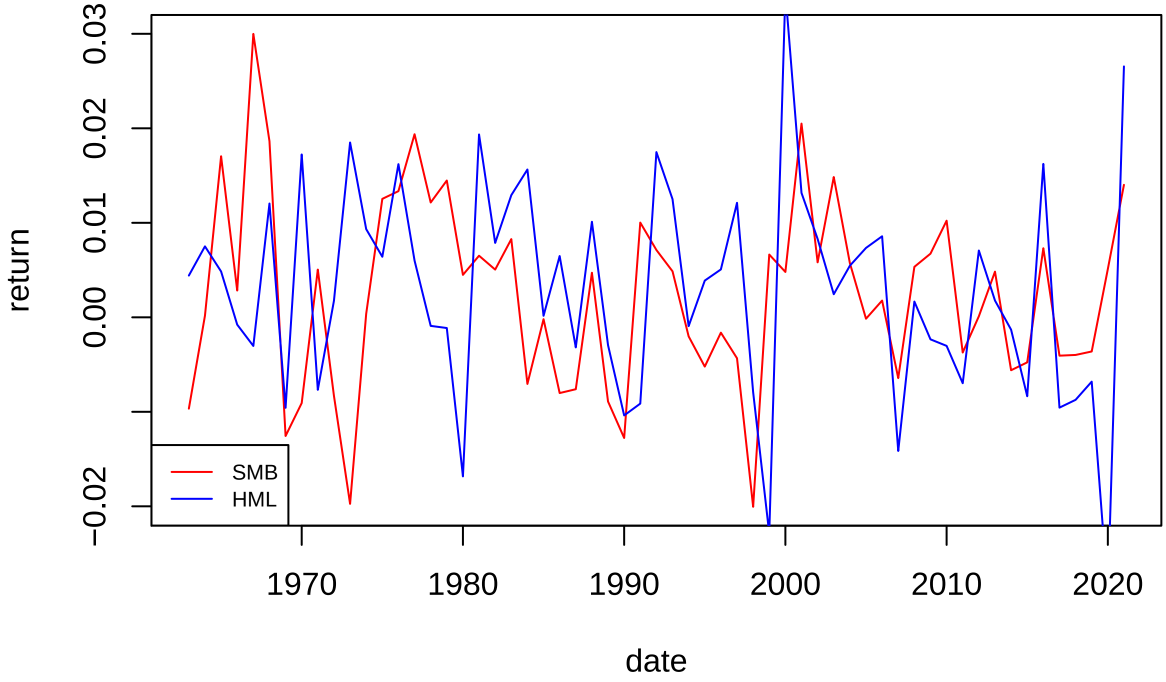
```
## Warning: package 'lubridate' was built under R version 4.0.5
```

```
##
## Attaching package: 'lubridate'
```

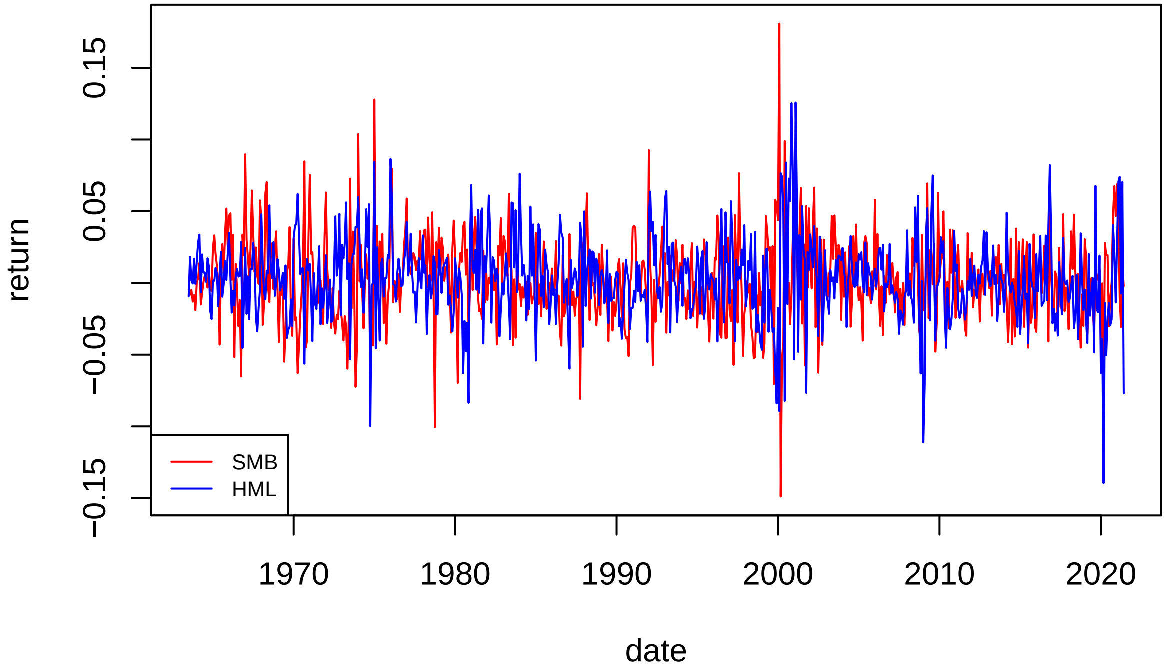
```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
FF5 <- FF5 %>%
  # change returns from % to actual
  mutate_at(.vars = vars(Mkt.RF:RF),
            .funs = list( ~ . /100) ) %>%
  # make new variables of date_format and date_year
  mutate(date_format = ymd(paste0(date, "01")),
         date_year = year(date_format))

# FF5_sum: FF5 summary
FF5_sum <- FF5 %>%
  # calculate average monthly returns for each year
  group_by(date_year) %>%
  summarise_at(.vars = vars(Mkt.RF:RF),
               .funs = list( ~ mean(.))) %>%
  ungroup()

# plot SMB and HML, average monthly returns for each year
```

```r
plot(x = FF5_sum$date_year, y = FF5_sum$SMB, type = "l",
     col = "red", xlab = "date", ylab = "return")
lines(x = FF5_sum$date_year, y = FF5_sum$HML, col = "blue")
legend("bottomleft", legend=c("SMB", "HML"),
       col=c("red", "blue"), lty = 1, cex = 0.7)
```



```r
# plot SMB and HML, monthly returns
plot(x = FF5$date_format, y = FF5$SMB, type = "l",
     col = "red", xlab = "date", ylab = "return")
lines(x = FF5$date_format, y = FF5$HML, col = "blue")
legend("bottomleft", legend=c("SMB", "HML"),
       col=c("red", "blue"), lty = 1, cex = 0.7)
```



7 Example 2: regression and for loop

- Simulate data
- Run regressions
- Find the true relationship between Y and X

```
set.seed(1) # for replication

N_obs = 1000

# generate random numbers for X
df_regression <- data.frame(
  #Random number UNIForm
  X1 = runif(n = N_obs, min = -5, max = 10) ,
  #Random number NORMal
  X2 = rnorm(n = N_obs, mean = 3, sd = 5),
  X3 = rexp(n = N_obs, rate = 1),
  X4 = rbinom(n = N_obs, size = 10, prob = 0.3),
  X5 = rpois(n = N_obs, lambda = 3),
  noise = rnorm(n = N_obs, mean = 0, sd = 1)
)

# define Y
df_regression$Y <- 3*df_regression$X1 + df_regression$noise
```

```r
# prepare data frame to store output of loop
df_out <- data.frame(coef = rep(NA, 5),
                     ts = rep(NA, 5),
                     R2 = rep(NA, 5))

# run univariate regressions
for (i in 1:5) {
  # # run regression at each loop, using X_i
  # reg_temp <- lm(df_regression$Y ~ df_regression[, i])

  # alternatively
  f <- as.formula(paste0("Y ~ X",i))
  reg_temp <- lm(f, data = df_regression)

  # store the results
  df_out[i,] <- c(reg_temp$coefficients[2],
                  summary(reg_temp)$coefficients[2,3],
                  summary(reg_temp)$r.squared)

}

round(df_out,2)
```

```
##   coef     ts   R2
## 1 3.00 408.09 0.99
## 2 0.02   0.27 0.00
## 3 0.24   0.64 0.00
## 4 0.14   0.48 0.00
## 5 0.21   0.88 0.00
```

8 Further readings

- R for Data Science, free book
- RMarkdown, for writing reports and interactive documents
- Stackoverflow, question and answer website for programming