Financial Econometrics

Introduction to R

Darya Yuferova

This lab session is designed to give you an overview of the tasks commonly performed in R. Closely related R code can be found in the pre-recorded video lecture.

Task 1: First steps

- Open R-Studio
- Set the working directory to the folder you want to work in

```
# This is just the one that I have picked on my computer
setwd("C:/Users/s13163/Dropbox/FIE401/data/data_labs")
```

- Install the “plm” package and load it for the current session

```
# install.packages("plm") # uncomment if you need to install the package
require(plm) # or library(plm)
```

- Declare two variables with a name of your choosing that consist of 100 randomly generated numbers each. (Hint: *rnorm(...)* generates normally distributed random variables) `rnorm(...)` generates normally distributed random variables)

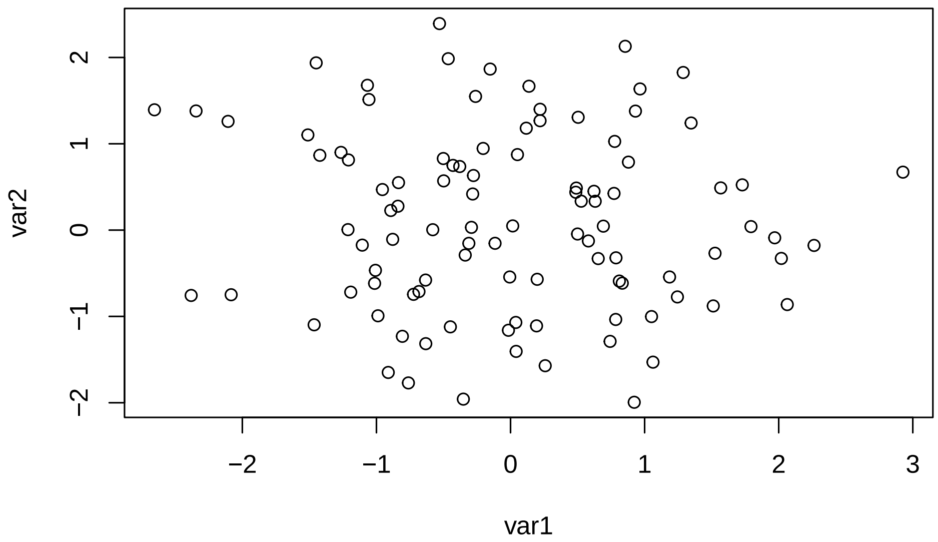
```
var1 <- rnorm(100)
var2 <- rnorm(100)
```

- Check if both variables you declared are in working memory (Hint: *ls()* might help)

```
ls()
```

- Make a scatterplot with one variable on the x-axis and the other one on the y-axis (Hint: *plot(x = , y =)* makes a plot)

```
plot(x = var1, y = var2)
```



```
# This plot is based on random numbers generated,
# hence your plot will look different
```

- Delete both variables and check if they are still in working memory

```
rm(var1, var2)
ls()
```

Task 2: Data objects and structures

- Make a vector holding the numbers from 1:100

```
x <- 1:100
```

- Construct a dataframe, in which column one is called "digits" and contains the previous vector, column two is called "year" which holds the number 1843 for each entry, and the last column is called "text" which contains some characters of your choosing.

```
first_data <- data.frame(digits = x,
                         year = 1843,
                         text = sample(letters, 100, replace = T))
# the third column consists of randomly sampled letters from a-z
```

Task 3: Import, export, and save data

- Read the *factor_returns.csv* file into R (use *read.csv(...)*). The file contains the daily returns of the three Fama-French factors. Note that you first have to download the file and save in your working directory (Check out wikipedia if you want to know more about the three Fama-French factors)

```
factor_returns <- read.csv("factor_returns.csv")
```

- Investigate the data (Hint: use *summary()*)

```
summary(factor_returns)
```

- Save the data to your working directory (Hint: use *save()*)

```
save(factor_returns, file = "factor_returns.RData")
```

- Write the data as a ".txt" file to your working directory. Use "\t" (tab) as a separating parameter (Hint: use *write.table(..., sep = "\t")*. (Hint: either search for an example online or type *?write.table* for explanations.) Open the resulting file in any text reader to see if it is tab delimited.

```
write.table(factor_returns, file = "factor_returns_tab.txt", sep = "\t")
```

- Check if the two files created in the previous tasks are in your working directory (Hint: use *list.files()*)

```
list.files()
```

Task 4: Manipulation

- Extract smb (column 3) for all days after end of 1999. Note, it is not strictly necessary to transform the dates into dates class. It is actually easier, to use the dates as numerics for this assignment and use logical operators, such as > and <.

```
smb_2000<-factor_returns[factor_returns$date >= 20000000, "smb"]
```

- Apply following tasks to the series extracted: compute mean, standard deviation, and summary, and display a histogram

```
mean(smb_2000)
sd(smb_2000)
summary(smb_2000)
hist(smb_2000)
```

Task 5: Loops [optional]

- Compute and print mean and standard deviation of smb separately for each individual year from 2000 to 2005. Approach: (1) Write a loop that iterates over the sequence from 2000 to 2005 (Hint: Use *2000:2005*), (2) For each iteration, compute mean and standard deviation. (Hint: use dates as numerics and logical operators such as > and < to subset the data), and (3) print the results to the console (Hint: use *print()*)

```
for(y in 2000:2005)
  {
    # each iteration y will have a different value of y
    smb<-factor_returns[factor_returns$date>=y*10000 & factor_returns$date<(y+1)*10000, "smb"]
    Mean<-mean(smb)
    StDev<-sd(smb)
    # output to console
    print(y)
    print(Mean)
    print(StDev)
  }
```

Task 6: Tidyverse [optional]

- Take the factor returns, (1) construct the market return (mkt.rf + rf), (2) limit to pre 2000, (3) order smb in decreasing order, and (4) limit the data.frame to hml. Only use tidyverse syntax. Use package "dplyr".

```r
require(dplyr)
factor_returns2<-factor_returns %>%
  mutate(mkt = mkt.rf + rf) %>%
  filter(date < 20000000) %>%
  arrange(-smb) %>% # arrange orders in increasing frequency. If you use (-),
  # then the order is reversed, or arrange(desc(smb))
  select(hml)
```