

# Software Tools, R - MidTerm Project

**Due date :** 18 Nov 2020, 23:59

---

## Objectives

- Manage Working Directory, R-Files and R-Projects
- Download, Copy, Paste and Find the Data
- Read and Write Data
- Identify and Indexing Data
- Use Logical Operators
- Create a Function
- Plot the Data

**Hints :** You can use these functions and the main web page of our course - [LINK](#)

```
getwd()
setwd()
list.files()
file.path()

read.csv()
read.delim()
read.table()

class()
str()
length()
sim()
attributes()
attr()
typeof()

mean()
sum()
colMeans()
rowMeans()

na.omit()

plot()
View()
```

---

# Instructions

## PART-1 Download, Get or Find and Read Data

### R-STUDIO AND DATA

1. Go to the webpage of the course
2. Open Data “Istanbul\_Goztepe\_Mean\_Temperature\_1839-2013\_Monthly” (.dat) by click the LINK
3. Copy and Paste it in your “Downloads” directory in a text file with “.dat” extension
4. Open your R-Studio, Create an R-Project, Create an R-Script (*you will write and save your all codes from now in this script*), and try to read the data file with each of these ways:

### WAY 1 - GO TO FILE

5. Check your Project Name and your Working directory
6. Go to “Downloads” directory in R-Studio, using console
7. List files and Read Data with three different read functions ( *read.csv()*, *read.delim()*, *read.table()* )
8. Choose the best for you ( *or change the options if it is necessary, regarding header or separators* )
9. Assign your data as “temp\_1”

### WAY 2 - CALL THE FILE

10. Go Back to your Working directory
11. Define your data file path with file.path() ( *remember, you downloaded the data in your Downloads folder* )
12. Assign the path a new variable as “path\_my\_file”
13. Use your best read...() function to read the file with “path\_my\_file”
14. Assign your data as “temp\_2”

### WAY 3 - IMPORT THE FILE

15. Use “Import Dataset”
16. Chose “From Text (base)” option
17. Try to assign your data as “temp\_3” at this time

### WAY 4 - GET THE FILE (WITH URL)

18. Copy the LINK of data
19. Use your best read() function
20. Read the file with this function and LINK
21. Assign your data as “temp\_4”

### LAST STEP

22. Choose your favorite data ( *temp\_1, temp\_2, temp\_3 or temp\_4* ) and assign as just “**temp**”

## PART-2 Play with the Data

### MEET WITH THE DATA

1. Look at class and structure
  2. Learn attributes and dimensions
  3. Rename attributes ( *ex: Months for each column and years for each row* ) ( *attributes()* and *month.name* or *month.abb* )
- *By the way, your data have to include just temperature values when you print it. Years and months have to be just attributes of data.*

### CLEAR NA AND CHOOSE COLUMN

4. Print “temp”. Do you see irrelevant numbers ? If so, change them as **NA**
- *Now your “temp” data must seem like this*

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1839	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	6.9
1840	4.3	3.8	4.3	7.7	16.6	19.0	24.5	22.7	20.3	15.7	12.6	3.7
1841	6.6	4.2	4.9	10.7	15.5	21.3	NA	NA	NA	NA	NA	NA
1842	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1843	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1844	4.5	9.1	8.6	8.8	14.2	21.3	24.7	24.6	21.1	17.2	14.3	5.2
1845	6.3	5.6	10.3	14.2	18.8	21.1	NA	NA	NA	NA	NA	NA
1846	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1847	5.8	8.7	8.5	15.7	19.6	23.4	25.2	25.5	23.4	15.8	11.2	5.9
1848	3.0	5.5	7.7	13.6	16.2	25.0	24.4	23.8	19.0	18.9	12.0	5.4
1849	3.4	2.2	6.0	12.2	16.5	21.8	22.9	23.4	20.9	16.9	12.6	7.0
1850	3.4	3.6	5.0	9.7	15.0	21.7	24.3	25.7	19.3	18.2	10.2	5.7

5. Delete rows which include NA ( *which()* or *na.omit()* )
- *Now your “temp” data must seem like this*

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1840	4.3	3.8	4.3	7.7	16.6	19.0	24.5	22.7	20.3	15.7	12.6	3.7
1844	4.5	9.1	8.6	8.8	14.2	21.3	24.7	24.6	21.1	17.2	14.3	5.2
1847	5.8	8.7	8.5	15.7	19.6	23.4	25.2	25.5	23.4	15.8	11.2	5.9
1848	3.0	5.5	7.7	13.6	16.2	25.0	24.4	23.8	19.0	18.9	12.0	5.4
1849	3.4	2.2	6.0	12.2	16.5	21.8	22.9	23.4	20.9	16.9	12.6	7.0
1850	3.4	3.6	5.0	9.7	15.0	21.7	24.3	25.7	19.3	18.2	10.2	5.7
1851	4.7	3.3	7.5	12.4	19.2	19.3	22.9	24.3	20.1	17.7	15.7	5.4
1852	6.2	5.4	5.2	8.6	16.4	21.4	22.4	22.6	20.0	17.6	12.8	9.1
1853	8.6	9.0	10.1	11.1	17.1	21.6	24.3	23.6	20.9	18.4	11.1	8.5
1855	4.3	6.6	9.9	10.5	19.9	20.9	23.5	22.2	17.3	15.7	13.5	9.8
1857	8.8	4.5	6.4	10.8	15.3	19.3	21.4	22.1	16.9	16.0	9.6	4.3
1858	1.6	1.9	7.8	11.0	17.5	20.1	25.5	24.5	19.7	18.5	12.3	7.7
1859	3.1	4.8	7.8	13.6	18.1	20.3	22.8	24.6	21.0	17.0	12.4	9.2

6. Assign it as new “temp” again
7. Select summer season
8. Assign it as “temp\_summer” ( *three months* )

## USE LOGICAL OPERATORS

9. Compare June and July: Find and print the years when June temperatures were **less than** July. ( *which()* )
10. Calculate mean temperature for each months ( *you will probably need the **na.rm** (NA Remove) option* ) and assign it as "avg\_month"
11. Print minimum and maximum values for "avg\_month" and find which year and which month they were observed
12. Calculate the mean temperature for each years ( *you will probably need the **na.rm** (NA Remove) option* ) "avg\_year"
13. Calculate the mean temperature of all data and assign it as "avg\_temp" and print it
14. Print the years of "avg\_year" **which** are **greater than or equal to** "avg\_temp"

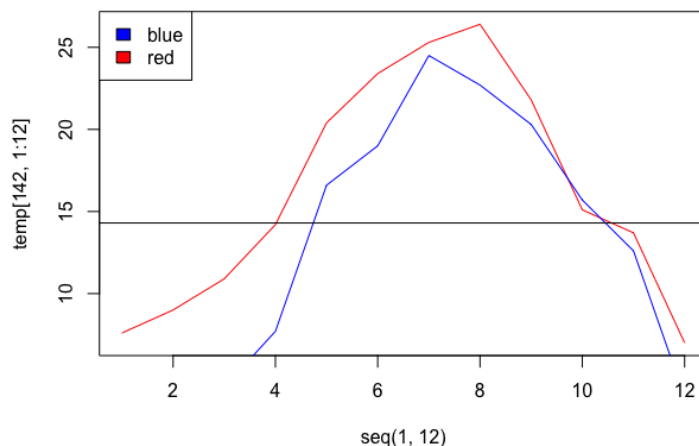
## PLOT

14. Plot temperature for June ( *for each year* ), add title and unit
15. Edit y-axis and x-axis label ( *to see which years are they* )
16. What about July and August ? Plot them on the same figure with June.
17. Is there any strangeness thing, what do you think ? Compare three plots
18. Plot "avg\_month", make the type of line as "line" and colorful. What is reason of the *inverted V letter* shape
19. Plot "avg\_year", make the line dashed line. What do you think
20. Plot just temperature for just 1990 ( *my birth year* )

## THE LAST BUT NOT LEAST

21. **Run these codes.** Can you explain what am I doing with my codes which were given below? What are these x-axis and y-axis in here? ( *which years or months are they?* ) What is the result of this figure, what do you see and what do you think? what do colors mean? Can you edit title and x-axis and y-axis labels?

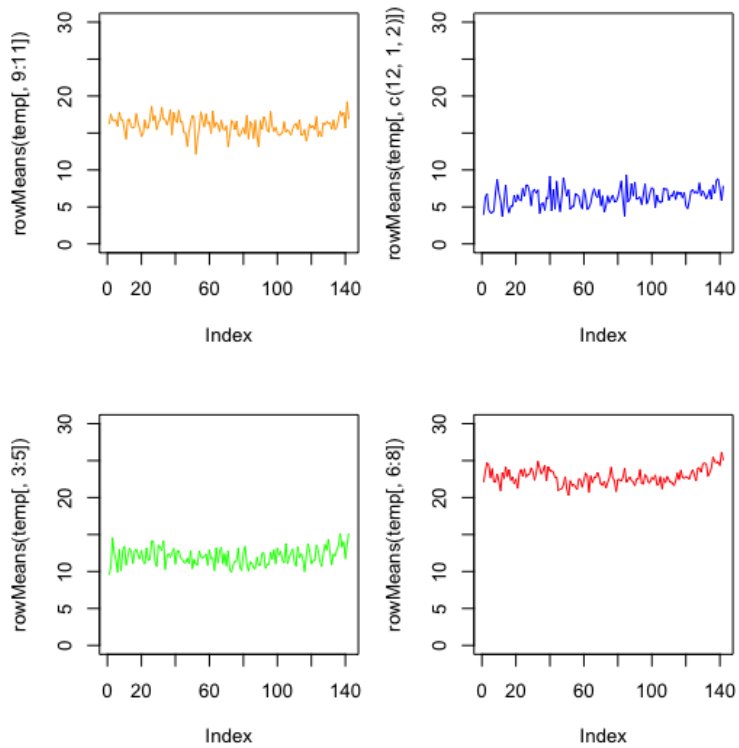
```
plot(seq(1,12), temp[142,1:12] , type = "l", col = "red")
lines(seq(1,12), temp[1,1:12] , type = "l", col = "blue")
legend("topleft",c("blue","red"),fill=c("blue","red"))
abline(h = mean(colMeans(temp, na.rm = T)))
```



```

par(mfrow = c(2, 2))    #or try par(mfrow = c(4, 1))
plot(ylim = c(0,30), rowMeans(temp[,9:11])      , type="l", col="orange")
plot(ylim = c(0,30), rowMeans(temp[,c(12,1,2)]) , type="l", col="blue")
plot(ylim = c(0,30), rowMeans(temp[,3:5])        , type="l", col="green")
plot(ylim = c(0,30), rowMeans(temp[,6:8])        , type="l", col="red")

```




---

*For questions or problems, please use Ninova*

*Emir*

---