

Name-Surname :ÖZGÜR KAN

Email :ozgurkan2020@gmail.com

No :15011702

Signature :

HOME WORK 1 (Return by 09.01.2021)

BLM3590 – Statistical Data Analysis

T1(10)	T2(15)	T3(15)	T4(15)	T5(15)	T6(15)	T7(15)				Total(100)

The attached Excel file (**SdA-HW**) consists of two data types (embolic signals (**type 1**), and Doppler speckle (**type 2**)) recorded from stroke patients and some relevant numerical variables (**tpthrt**, **pkthrt**, **dfdrtrt**, **time**, **rrt**, **frt**). Using this data file, implement the following tasks in **R**. You must include the **R** scripts in your answers.

T1: Show how to read this Excel datafile into **R** environment.

```
library(readxl)
```

```
SdA_HW <- read_excel("C:/Users/ozgur/Desktop/HW/SdA-HW.xls")
```

```
View(SdA_HW)
```

RStudio interface showing the SdA_HW Excel file loaded into the R environment. The Environment pane displays the data frame SdA_HW with 200 observations and 7 variables. The Data pane shows the structure of the data frame. The Console pane shows the R code used to load the file.

```
library(readxl)
SdA_HW <- read_excel("C:/Users/ozgur/Desktop/HW/SdA-HW.xls")
View(SdA_HW)
```

T2: This data file requires some preprocessing as it includes a column with no value, some cells with no numerical value (divide by 0 error, etc.), and some cells with zero. Write required script in **R** to remove the empty column and correct the cells with no numerical value and zero by using simple interpolation.

DELETE EMPTY COLUMN

```
delete_empty_column <- SdA_HW[!sapply(SdA_HW, function(x) all(is.na(x) | x == ""))]  
View(delete_empty_column)
```

The screenshot displays the RStudio interface with the following components:

- Environment Pane:** Shows two objects:
 - `delete_empty_column`: 200 obs. of 6 variables. Variables: `type` (num 1), `tpthrt` (num 14.6), `pkthrt` (num 7.16), `dfdrtr` (num 20.5), `rrt` (num 8.53), `frrt` (num 17.05).
 - `SdA_HW`: 200 obs. of 7 variables. Variables: `type` (num 1), `tpthrt` (num 14.6), `pkthrt` (num 7.16), `dfdrtr` (num 20.5), `time` (logi NA), `rrt` (num 8.53), `frrt` (num 17.05).
- Viewer Pane:** Displays a data frame with 23 rows and 6 columns:

	type	tpthrt	pkthrt	dfdrtr	rrt	frrt
1	1	14.611816	7.155748	20.501258	8.527265	17.054550
2	1	0.000000	0.000000	0.000000	NA	NA
3	1	10.387063	3.173866	21.038962	4.538626	7.564388
4	1	14.494864	5.891724	28.135395	8.425162	6.017975
5	1	6.899807	1.335272	25.239697	3.182401	4.773602
6	1	0.000000	0.000000	0.000000	NA	NA
7	1	14.929119	5.168394	16.150102	2.639675	9.238854
8	1	19.474133	7.977139	33.421967	3.001923	8.148078
9	1	27.051413	11.370301	12.655528	5.081104	0.967629
10	1	24.253508	10.279815	28.548077	2.722247	3.340940
11	1	6.834928	2.835058	1.557392	10.135340	10.135340
12	1	3.300711	0.332214	12.665563	2.375332	1.187666
13	1	15.991371	2.930654	21.597374	3.492362	0.952463
14	1	0.000000	0.000000	0.000000	NA	NA
15	1	0.000000	0.000000	0.000000	NA	NA
16	1	11.061486	4.428561	20.758657	15.832120	6.332840
17	1	28.863992	19.906871	26.068455	9.468942	10.166720
18	1	0.000000	0.000000	0.000000	NA	NA
19	1	0.000000	0.000000	0.000000	NA	NA
20	1	14.339388	6.110674	14.108686	7.281885	8.738260
21	1	0.000000	0.000000	0.000000	NA	NA
22	1	0.000000	0.000000	0.000000	NA	NA
23	1	14.979864	6.487209	19.371903	4.216685	15.461200
- Console:** Shows the execution of the following R code:


```
library(readxl)
SdA_HW <- read_excel("C:/Users/ozgur/Desktop/HW/SdA-HW.xls")
View(SdA_HW)
delete_empty_column <- SdA_HW[!sapply(SdA_HW, function(x) all(is.na(x) | x == ""))]
View(delete_empty_column)
```

SIMPLE INTERPOLATION

```
install.packages("zoo")
update.packages("zoo")
library(zoo)
interpolation <- delete_empty_column
interpolation$tpthrt <- na.approx(interpolation$tpthrt, method="linear")
interpolation$pkthrt <- na.approx(interpolation$pkthrt, method="linear")
interpolation$sdfdrrt <- na.approx(interpolation$sdfdrrt, method="linear")
interpolation$rrrt <- na.approx(interpolation$rrrt, method="linear")
interpolation$frrt <- na.approx(interpolation$frrt, method="linear")
interpolation <-na.approx(replace(interpolation, interpolation == 0, NA), method="linear")
View(interpolation)
```

The screenshot displays the RStudio interface. The main editor window shows a data frame with 200 rows and 6 columns. The columns are labeled 'type', 'tpthrt', 'pkthrt', 'dfdrrt', 'rrrt', and 'frrt'. The data is sorted by 'type' in ascending order. The console window shows the following output:

```
Content type 'application/zip' length 902441 bytes (881 KB)
downloaded 881 KB

package 'zoo' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
c:\Users\ozgur\AppData\Local\Temp\Rtmps7FmV6\downloaded_packages
> update.packages("zoo")
> library(zoo)

Attaching package: 'zoo'

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

    as.Date, as.Date.numeric

> library(zoo)
> interpolation <- delete_empty_column
> interpolation$tpthrt <- na.approx(interpolation$tpthrt, method="linear")
> interpolation$pkthrt <- na.approx(interpolation$pkthrt, method="linear")
> interpolation$sdfdrrt <- na.approx(interpolation$sdfdrrt, method="linear")
> interpolation$rrrt <- na.approx(interpolation$rrrt, method="linear")
> interpolation$frrt <- na.approx(interpolation$frrt, method="linear")
> interpolation <-na.approx(replace(interpolation, interpolation == 0, NA), method="linear")
> view(interpolation)
> |
```

The Environment pane on the right shows the following objects:

- delete_empty_column: 200 obs. of 6 variables
- interpolation: num [1:200, 1:6] 1 1 1 1 1 1 1 1 1 1 ...
- SdA_HW: 200 obs. of 7 variables

=====

T3: Find Five-number data summary of the variables for each data type in this dataset.

TYPE 1

```
type1 <-interpolation[ interpolation[, "type"]==1, ]
```

```
View(type1)
```

```
fivenum(type1[, "tpthrt"])
```

```
summary(type1[, "tpthrt"])
```

```
fivenum(type1[, "pkthrt"])
```

```
summary(type1[, "pkthrt"])
```

```
fivenum(type1[, "dfdr rt"])
```

```
summary(type1[, "dfdr rt"])
```

```
fivenum(type1[, "rrt"])
```

```
summary(type1[, "rrt"])
```

```
fivenum(type1[, "frt"])
```

```
summary(type1[, "frt"])
```

TPTHRT

0.33637	11.70441	14.73410	19.47500	28.86399
---------	----------	----------	----------	----------

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.3364	11.7300	14.7300	15.4800	19.4700	28.8600

PKTHRT

-8.358763	3.299621	5.949281	8.369141	19.906871
-----------	----------	----------	----------	-----------

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-8.359	3.320	5.949	5.793	8.319	19.910

DFDRRT

0.26008	12.06188	17.85649	22.45371	45.41403
---------	----------	----------	----------	----------

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.2601	12.2700	17.8600	17.5900	22.4100	45.4100

RRT

-59.765200	2.920039	5.654713	9.475557	20.609650
------------	----------	----------	----------	-----------

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-59.770	2.925	5.655	4.236	9.469	20.610

FRT

0.952463	4.544763	6.596013	10.113480	24.732330
----------	----------	----------	-----------	-----------

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.9525	4.5470	6.5960	7.5670	10.1000	24.7300

TYPE 2

```
type2 <-interpolation[ interpolation[, "type"]==2, ]
View(type2)
fivenum(type2[, "tpthrt"])
summary(type2[, "tpthrt"])
fivenum(type2[, "pkthrt"])
summary(type2[, "pkthrt"])
fivenum(type2[, "dfdr rt"])
summary(type2[, "dfdr rt"])
fivenum(type2[, "rrt"])
summary(type2[, "rrt"])
fivenum(type2[, "frt"])
summary(type2[, "frt"])
```

TPTHRT

0.008013	8.021390	11.451825	14.274517	20.963047	
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.008013	8.039000	11.450000	11.090000	14.270000	20.960000

PKTHRT

-9.622567	-1.020240	1.276692	4.162536	7.323946	
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-9.623	-1.003	1.277	1.301	4.161	7.324

DFDRRT

-3.475887	6.807376	13.798834	21.922938	40.545011	
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-3.476	7.022	13.800	14.420	21.880	40.550

RRT

-68.801400	-7.294725	2.210573	4.650650	26.127030	
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-68.800	-7.168	2.211	-3.131	4.616	26.130

FRT

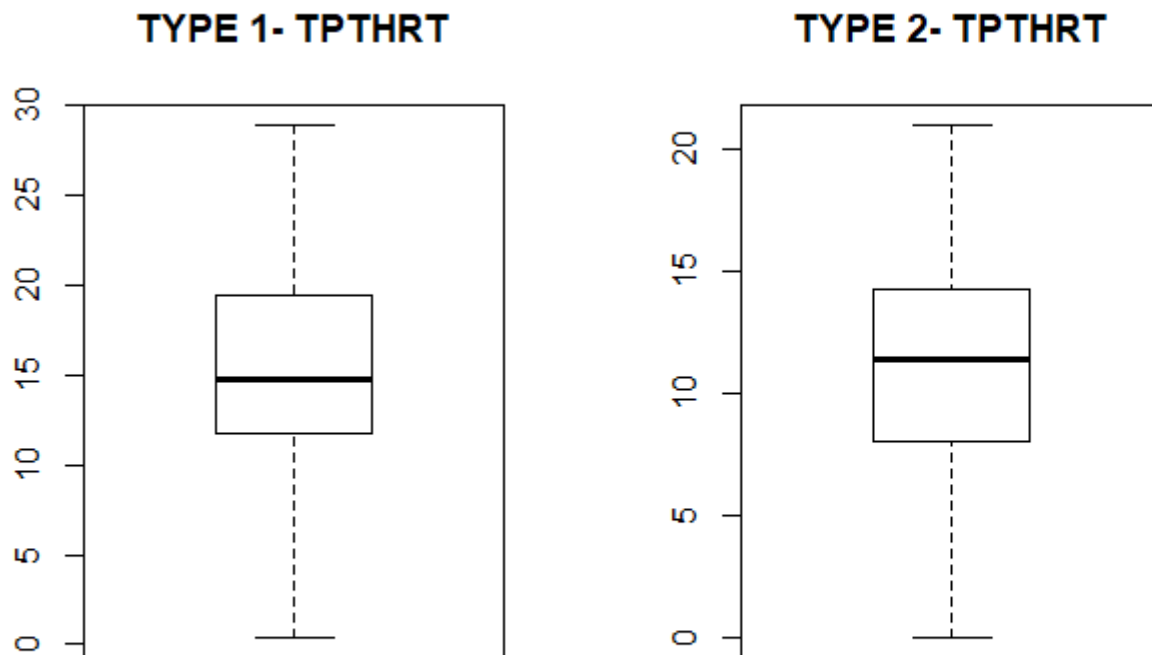
0.057293	2.127381	3.901223	6.326019	32.282380	
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.05729	2.16400	3.90100	4.93300	6.30500	32.28000

=====

T4: Plot **boxplots** of the **variables** for each **data type** and determine if there is any outlier in these variables.

```
boxplot(type1[, "tpthrt"], main="TYPE 1- TPTHRT")
boxplot(type1[, "pkthrt"], main="TYPE 1- PKTHRT")
boxplot(type1[, "dfdr rt"], main="TYPE 1- DFDRRT")
boxplot(type1[, "rrt"], main="TYPE 1- RRT")
boxplot(type1[, "frt"], main="TYPE 1- FRT")
```

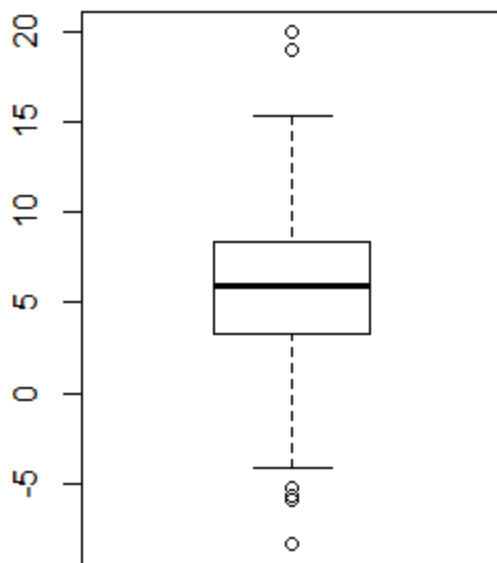
```
boxplot(type2[, "tpthrt"], main="TYPE 2- TPTHRT")
boxplot(type2[, "pkthrt"], main="TYPE 2- PKTHRT")
boxplot(type2[, "dfdr rt"], main="TYPE 2- DFDRRT")
boxplot(type2[, "rrt"], main="TYPE 2- RRT")
boxplot(type2[, "frt"], main="TYPE 2- FRT")
```



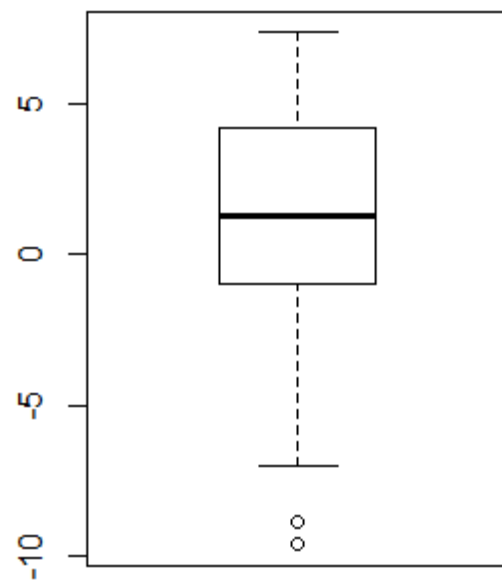
For "TYPE1-TPTHRT", the boxplot is almost symmetric, there is no outlier, central tendency is around 15.

For " TYPE2-TPTHRT ", the boxplot is almost symmetric, there is no outlier, central tendency is around 10-15.

TYPE 1- PKTHRT



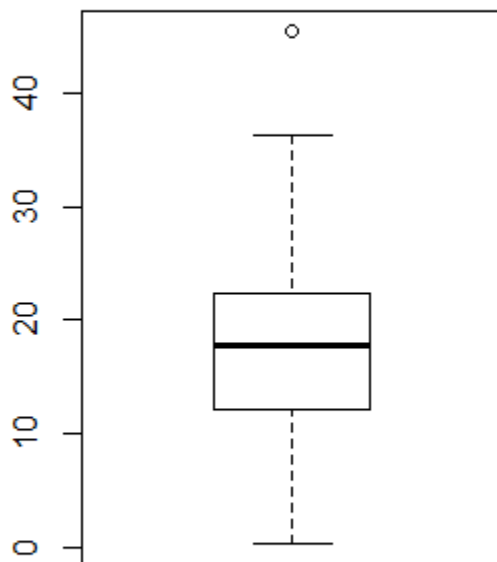
TYPE 2- PKTHRT



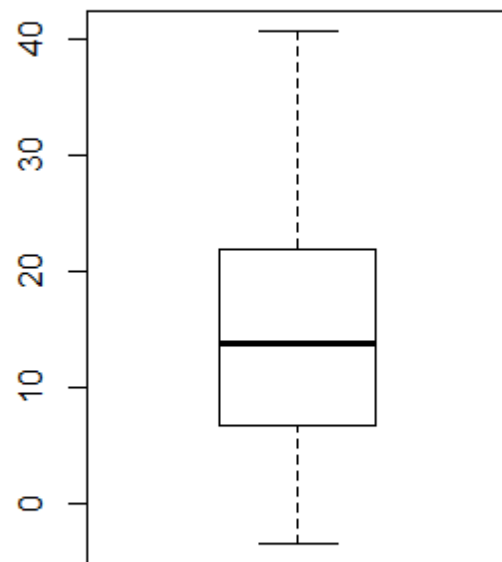
For "TYPE1-PKTHRT", the boxplot have outlier. The central tendency is around 5-10.

For " TYPE2- PKTHRT ", the boxplot have outlier. The central tendency is around 0-5.

TYPE 1- DFDRRT



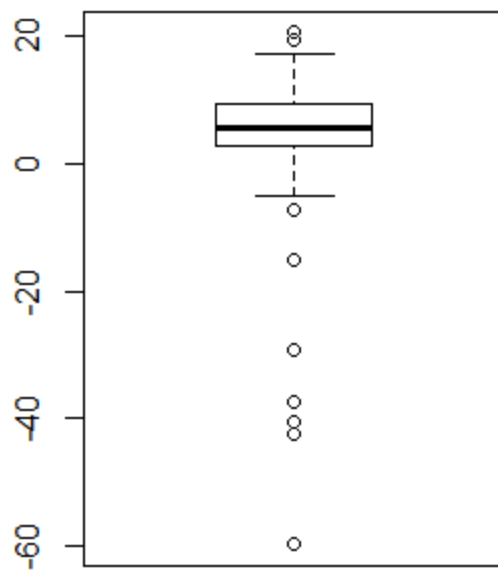
TYPE 2- DFDRRT



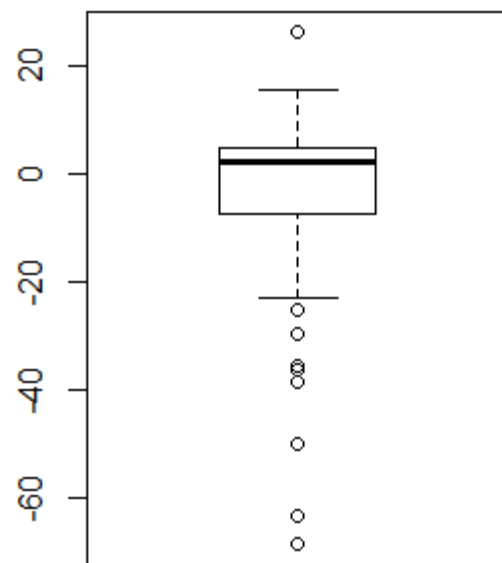
For "TYPE1-DFDRRT", the boxplot have outlier. The central tendency is around 10-20.

For " TYPE2- DFDRRT ", the boxplot is almost symmetric, there is no outlier, central tendency is around 10-20.

TYPE 1- RRT



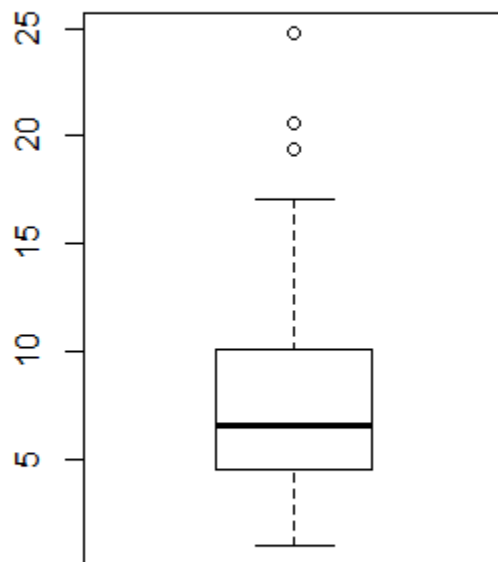
TYPE 2- RRT



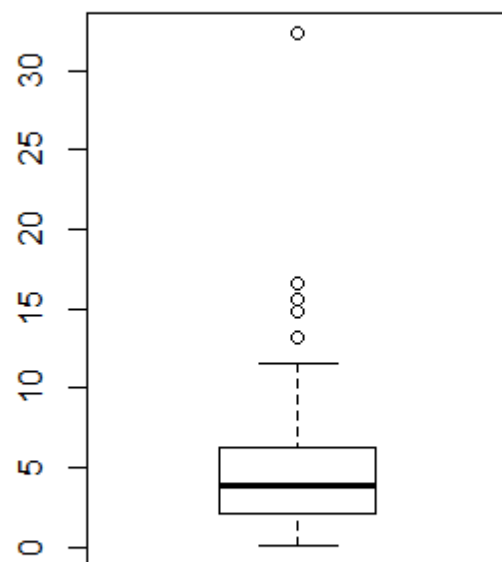
For "TYPE1-RRT", the boxplot have outlier. The central tendency is around 0-20.

For " TYPE2- RRT ", the boxplot have outlier. The central tendency is around 0-20.

TYPE 1- FRT



TYPE 2- FRT



For "TYPE1-FRT", the boxplot have outlier. The central tendency is around 5-10.

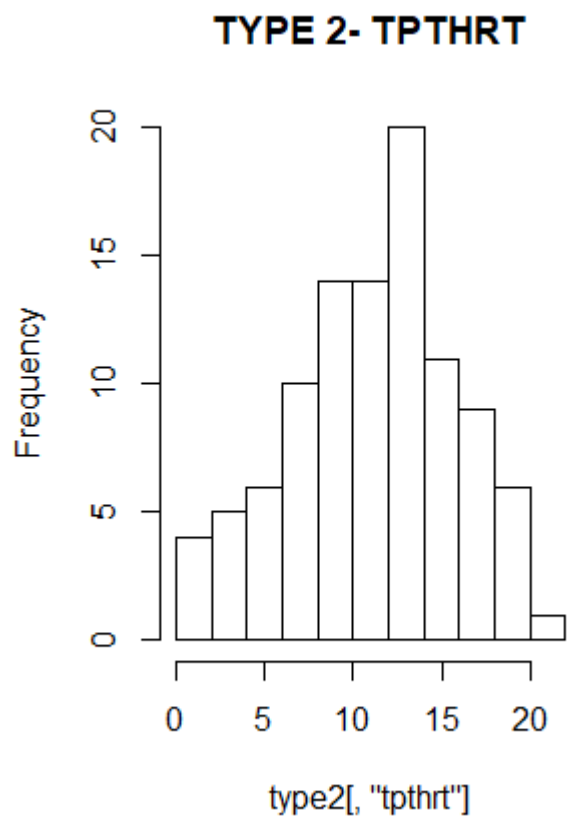
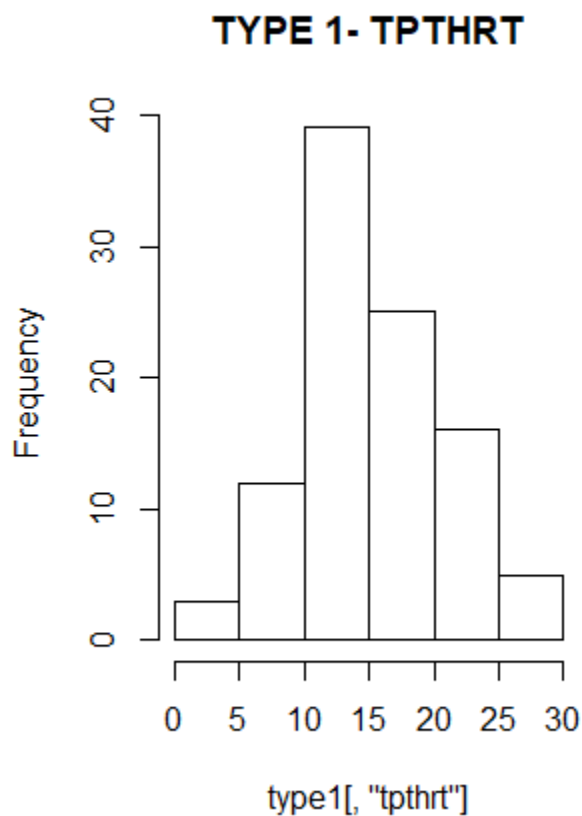
For " TYPE2- FRT ", the boxplot have outlier. The central tendency is around 0-5.

=====

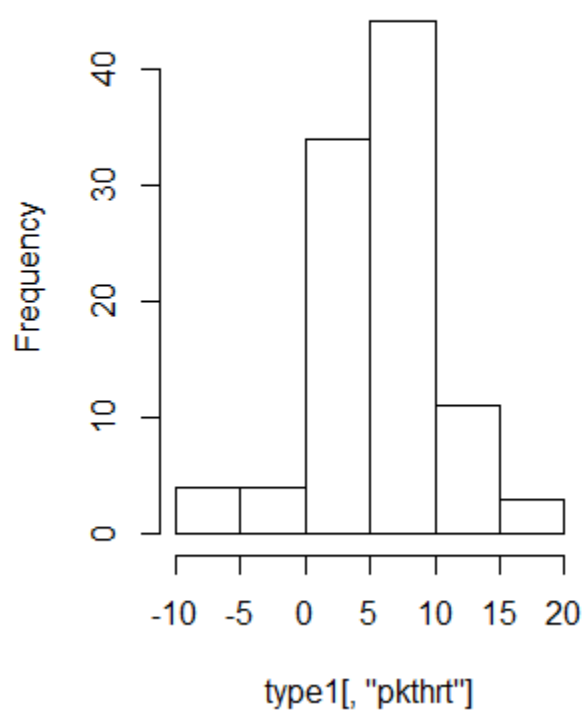
T5: Plot histograms of the **variables** for each **data type**, compare the histograms, and comment on the distributions.

```
hist(type1[, "tpthrt"], main="TYPE 1- TPTHRT")
hist(type1[, "pkthrt"], main="TYPE 1- PKTHRT")
hist(type1[, "dfdrtrt"], main="TYPE 1- DFDRRT")
hist(type1[, "rrt"], main="TYPE 1- RRT")
hist(type1[, "frt"], main="TYPE 1- FRT")
```

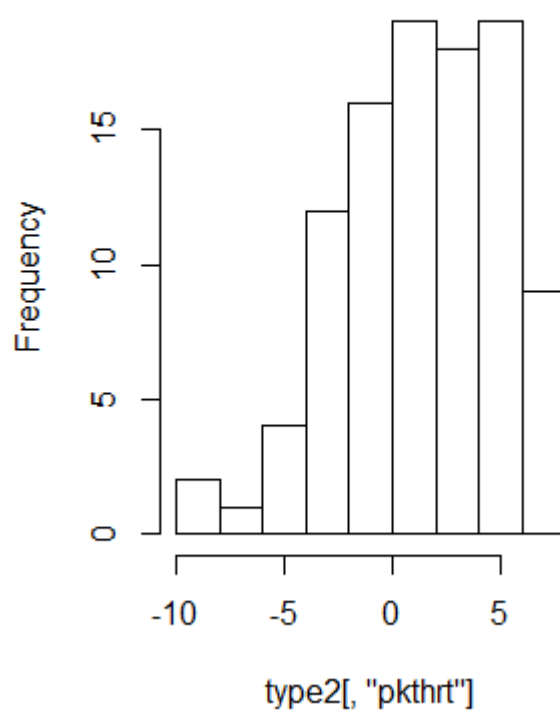
```
hist(type2[, "tpthrt"], main="TYPE 2- TPTHRT")
hist(type2[, "pkthrt"], main="TYPE 2- PKTHRT")
hist(type2[, "dfdrtrt"], main="TYPE 2- DFDRRT")
hist(type2[, "rrt"], main="TYPE 2- RRT")
hist(type2[, "frt"], main="TYPE 2- FRT")
```



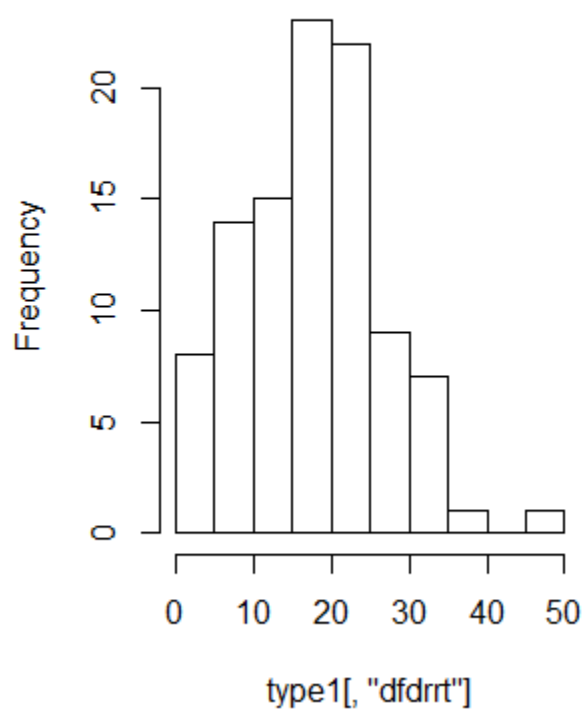
TYPE 1- PKTHRT



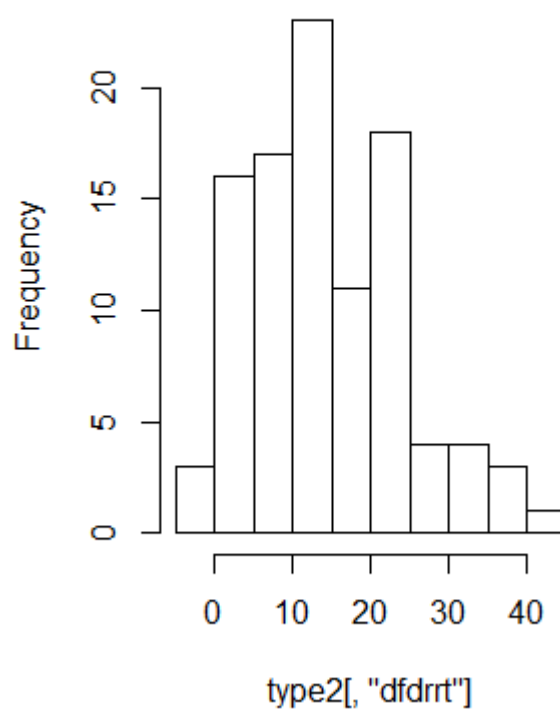
TYPE 2- PKTHRT



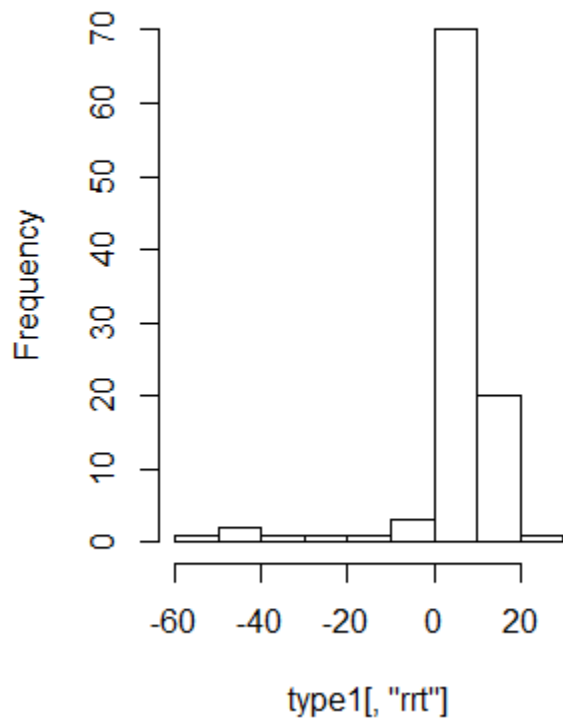
TYPE 1- DFDRRT



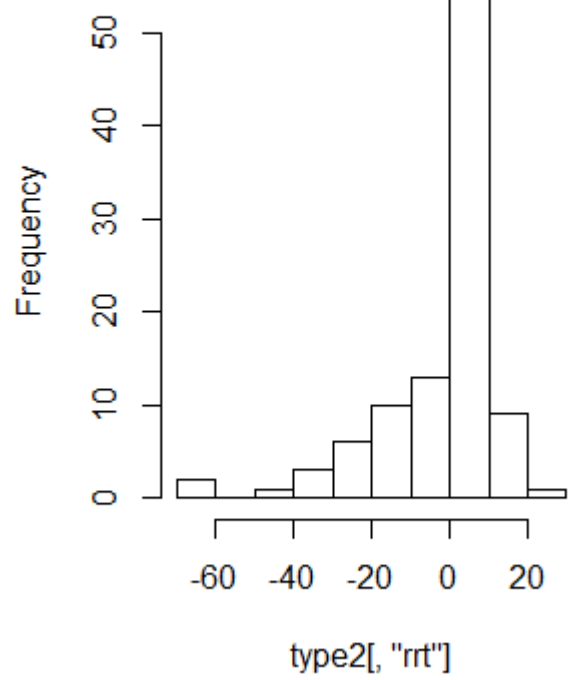
TYPE 2- DFDRRT



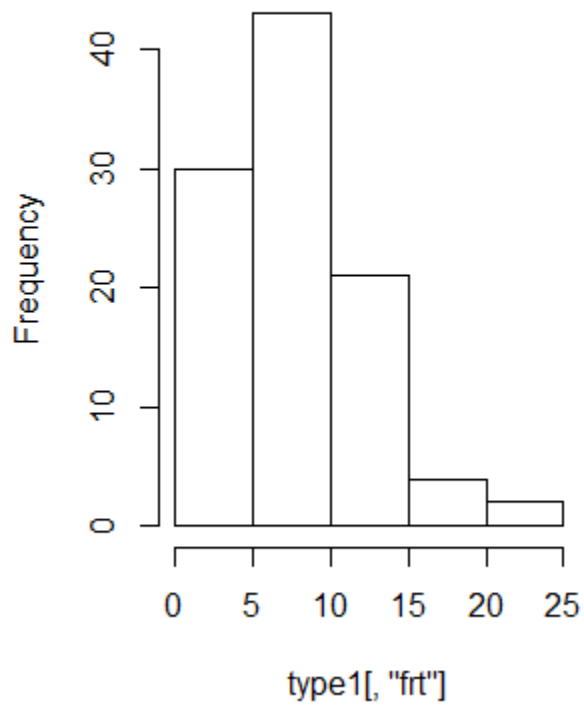
TYPE 1- RRT



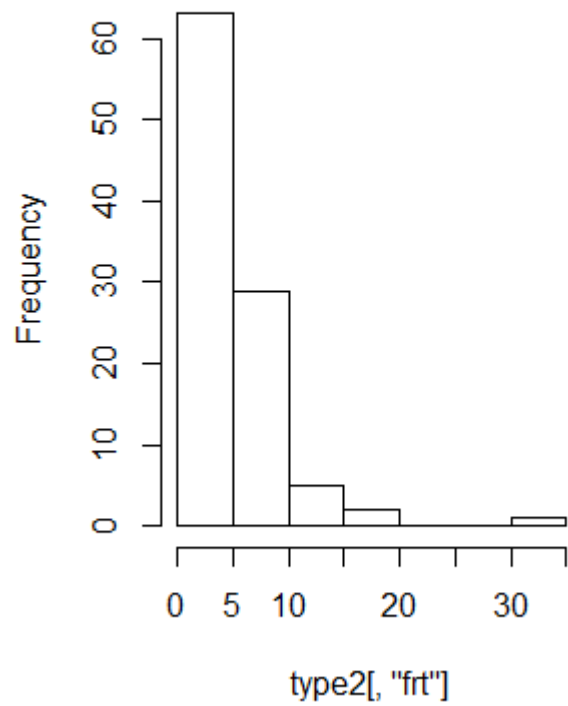
TYPE 2- RRT



TYPE 1- FRT



TYPE 2- FRT



=====

T6: First, normalize the **variables** for each **data type** so that the values of these variables range between **0** and **1**, and then line-plot (using different colors) each variables for both data types in one figure (total 5 figures). Comment on the similarities of the variables for each plot.

```
normalizedtype1 <-type1
normalizedtype1 <- (type1[, c(2, 3, 4, 5,6)]-min(type1[, c(2, 3, 4, 5,6)]))/(max(type1[, c(2, 3, 4, 5,6)])-
min(type1[, c(2, 3, 4, 5,6)]))
View(normalizedtype1)
```

```
normalizedtype2 <-type2
normalizedtype2 <- (type2[, c(2, 3, 4, 5,6)]-min(type2[, c(2, 3, 4, 5,6)]))/(max(type2[, c(2, 3, 4, 5,6)])-
min(type2[, c(2, 3, 4, 5,6)]))
View(normalizedtype2)
```

```
plot(x=1:100,y=normalizedtype1[, "tpthrt"],type = 'l',col = "red",main="TPTHRT")
lines(x=1:100, y=normalizedtype2[, "tpthrt"],type = 'l',col = "blue")
# Add a legend to the plot
legend("topright", legend=c("TYPE 1", "TYPE 2"),
      col=c("red", "blue"), lty = 1:2, cex=0.8)
```

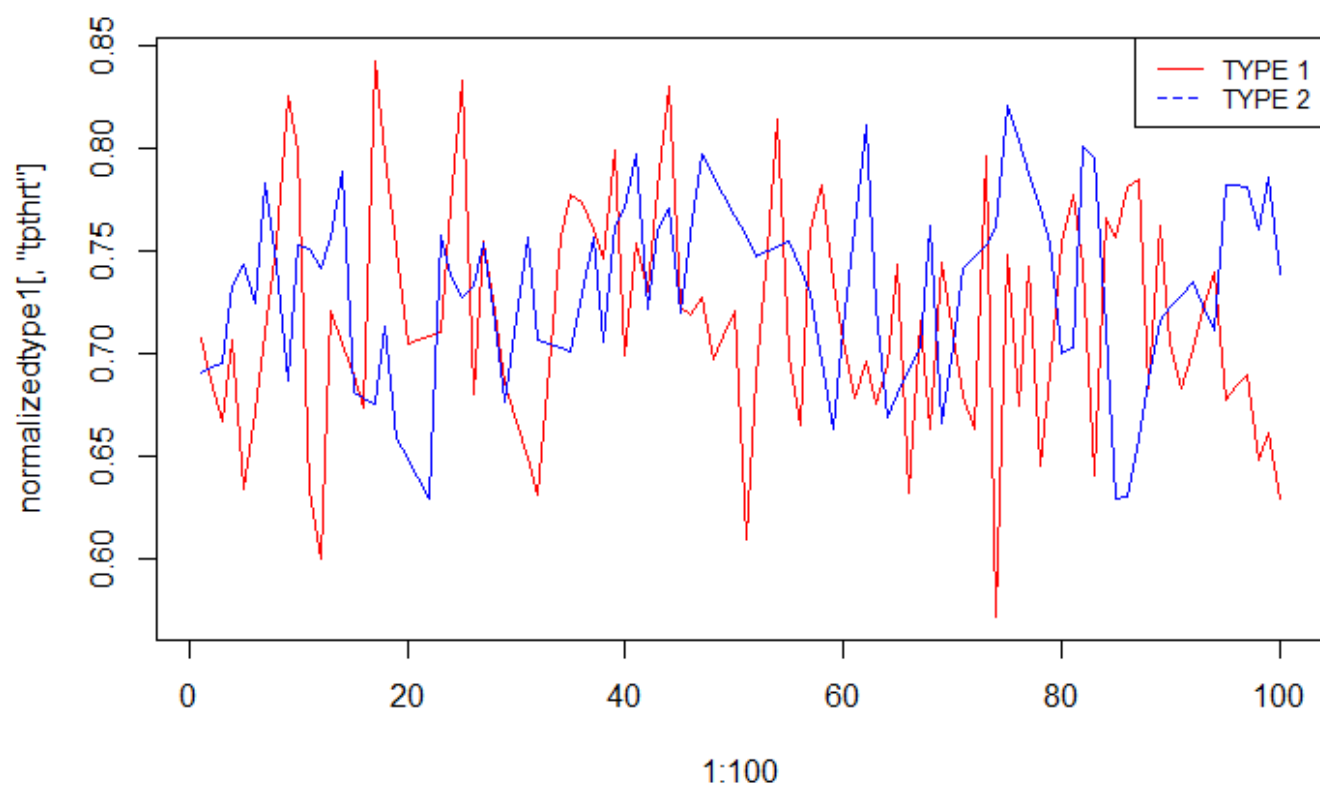
```
plot(x=1:100,y=normalizedtype1[, "pkthrt"],type = 'l',col = "red",main="PKTHRT")
lines(x=1:100, y=normalizedtype2[, "pkthrt"],type = 'l',col = "blue")
# Add a legend to the plot
legend("topright", legend=c("TYPE 1", "TYPE 2"),
      col=c("red", "blue"), lty = 1:2, cex=0.8)
```

```
plot(x=1:100,y=normalizedtype1[, "dfdrrt"],type = 'l',col = "red",main="DFDRRT")
lines(x=1:100, y=normalizedtype2[, "dfdrrt"],type = 'l',col = "blue")
# Add a legend to the plot
legend("topright", legend=c("TYPE 1", "TYPE 2"),
      col=c("red", "blue"), lty = 1:2, cex=0.8)
```

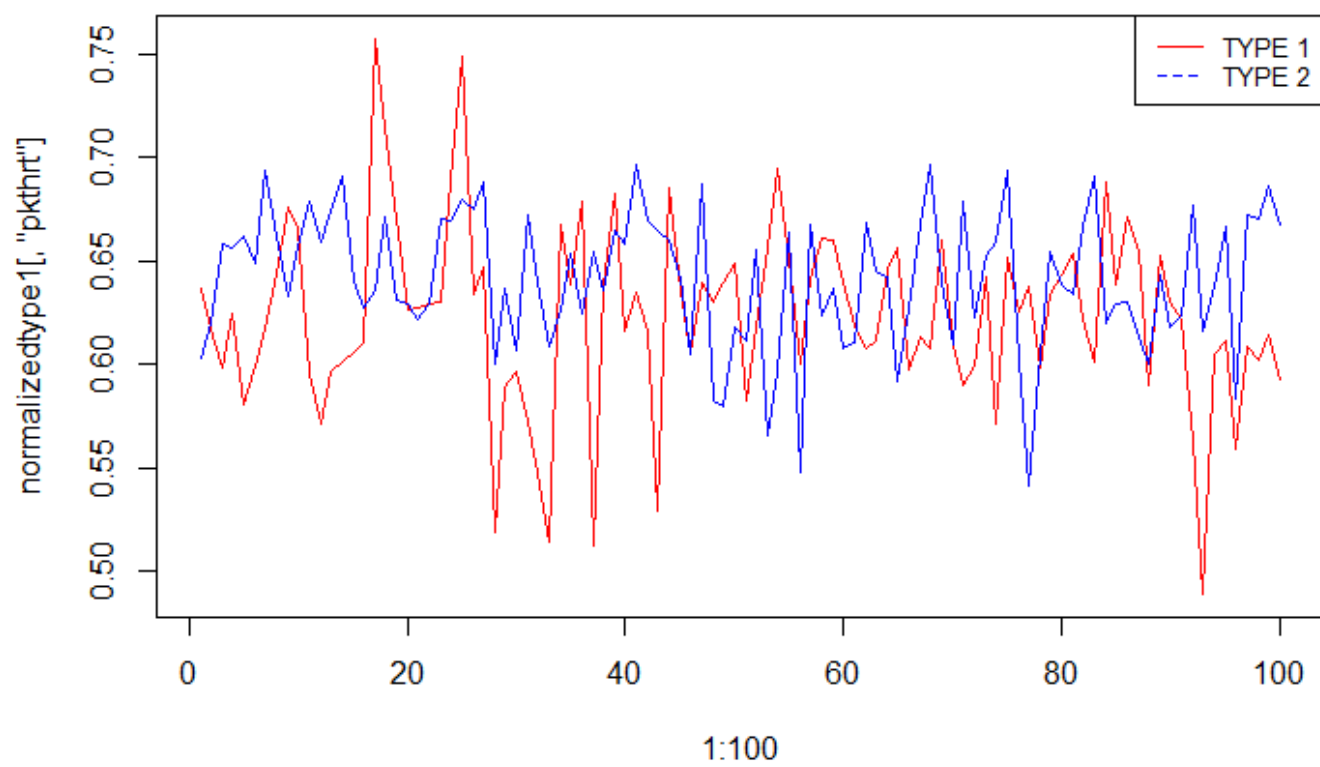
```
plot(x=1:100,y=normalizedtype1[, "rrt"],type = 'l',col = "red",main="RRT")
lines(x=1:100, y=normalizedtype2[, "rrt"],type = 'l',col = "blue")
# Add a legend to the plot
legend("topright", legend=c("TYPE 1", "TYPE 2"),
      col=c("red", "blue"), lty = 1:2, cex=0.8)
```

```
plot(x=1:100,y=normalizedtype1[, "frt"],type = 'l',col = "red",main="FRT")
lines(x=1:100, y=normalizedtype2[, "frt"],type = 'l',col = "blue")
# Add a legend to the plot
legend("topright", legend=c("TYPE 1", "TYPE 2"),
      col=c("red", "blue"), lty = 1:2, cex=0.8)
```

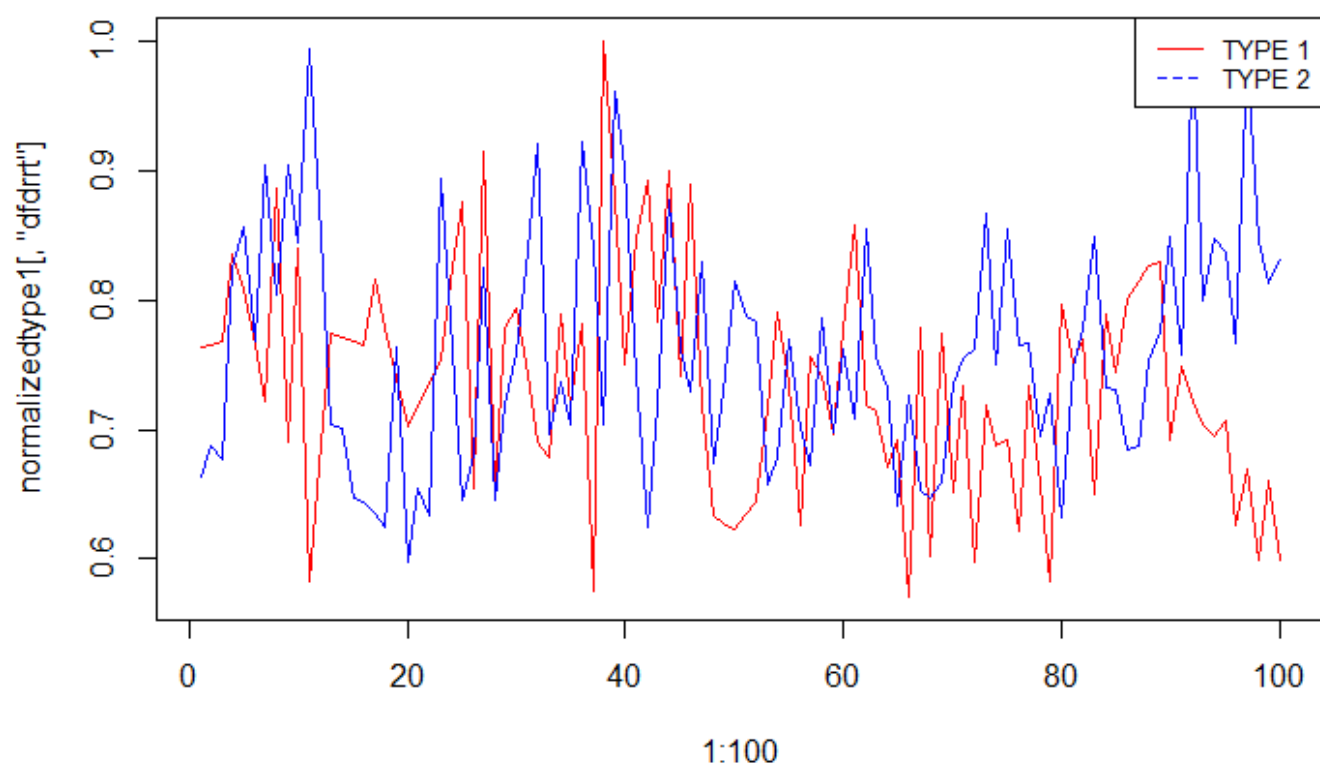
TPTHRT



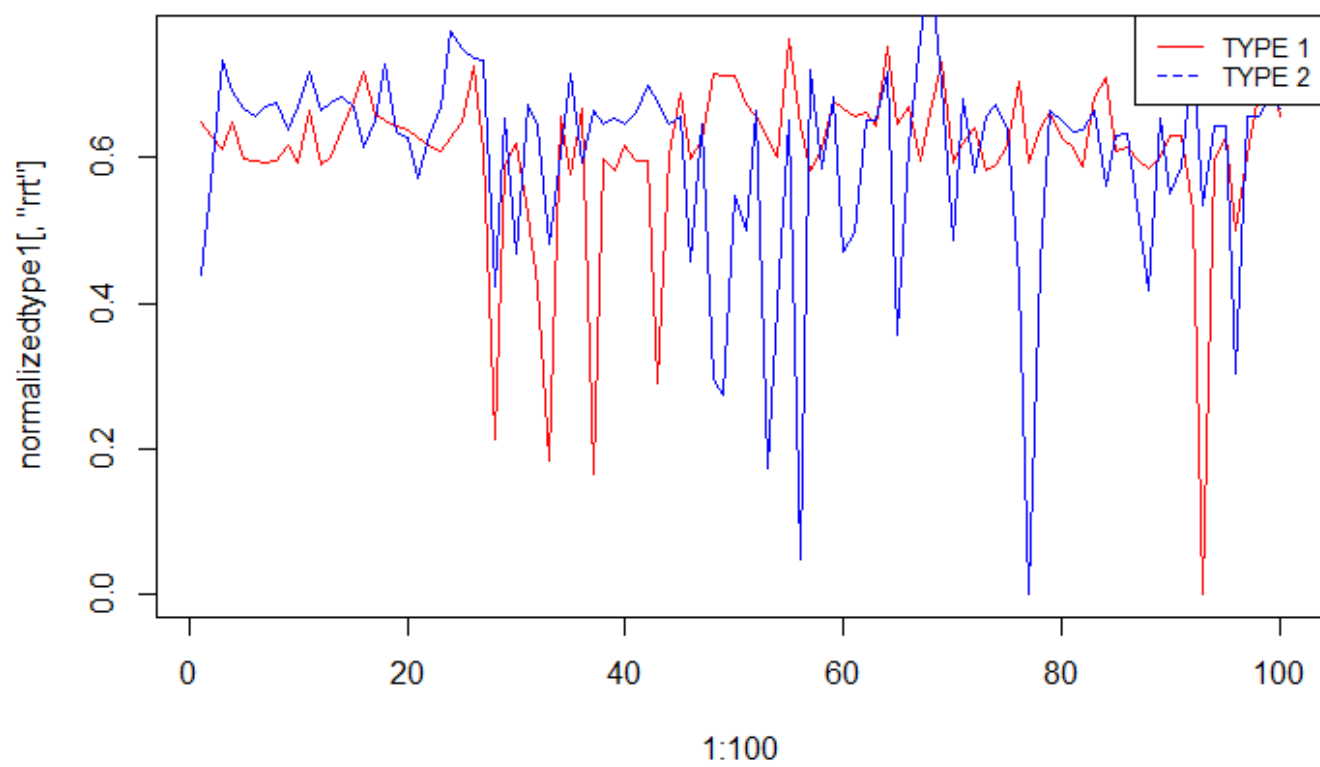
PKTHRT

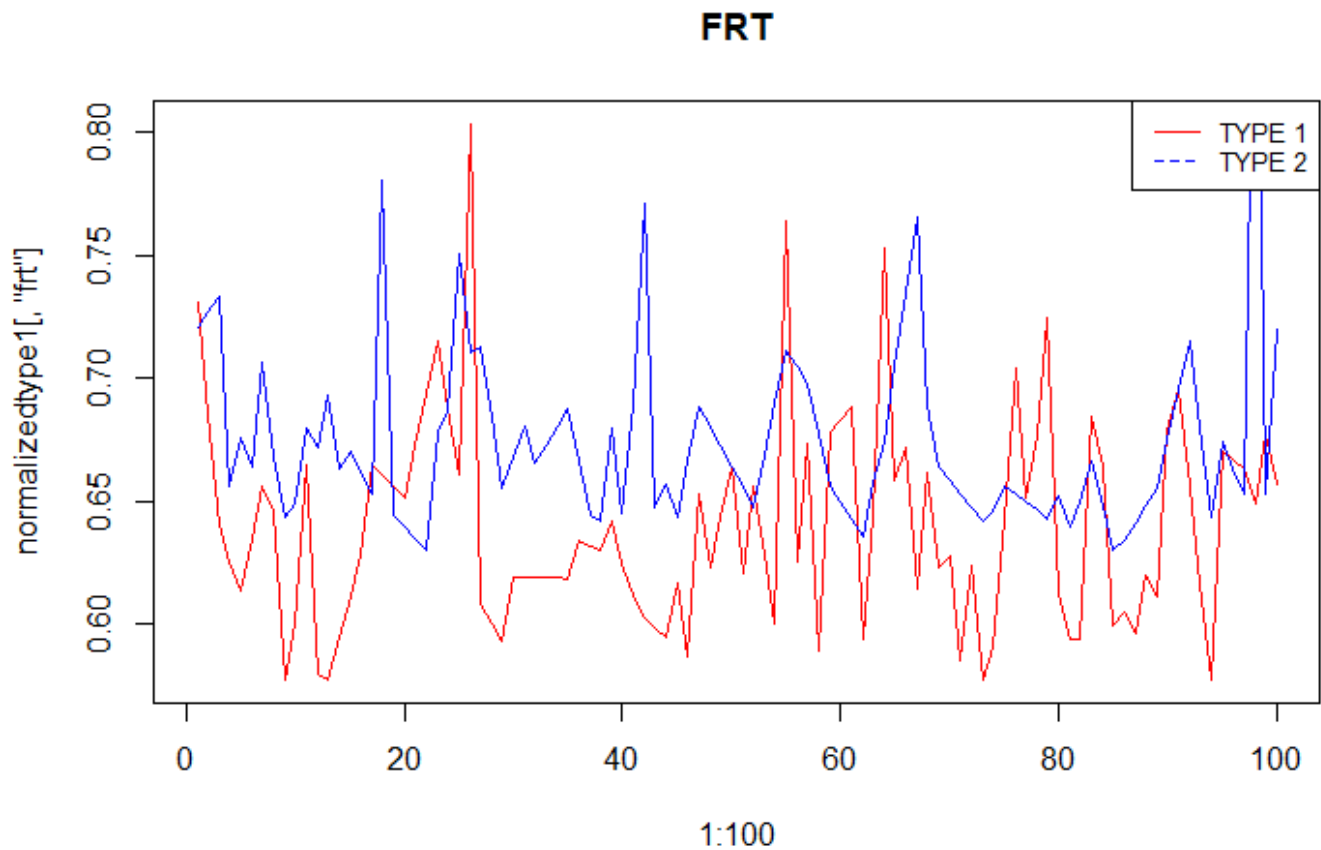


DFDRRT



RRT





- When we look at the first, second, third and fifth figures, type1 and type2 are not alike. While the values in the variable type1 decrease, the values in the type2 variable increase. Therefore, we can say that there is no similarity between the two types.

- When we look at the RRT variable in the fourth figure, it is seen that type1 and type2 behave similarly.

=====

T7: First, determine how similar the variables **tpthrt** and **pkthrt** are for each **data type**, and then determine how similar **tpthrt of data type 1** and **tpthrt of data type 2** by using similarity metric (correlation).

```
cor(normalizedtype1["tpthrt"],normalizedtype1["pkthrt"],method = c("pearson"))  
cor(normalizedtype2["tpthrt"],normalizedtype2["pkthrt"],method = c("pearson"))  
cor(normalizedtype1["tpthrt"],normalizedtype2["tpthrt"],method = c("pearson"))
```

Type 1(tpthrt and pkthrt)= 0.6417237

Type 2(tpthrt and pkthrt)= 0.2726601

TPTHRT(Type1 and Type2)= -0.1859388

- When we examined the results, it was observed that the variables in type 1 were a little more similar to each other, but it was observed that the variables in type 2 were not alike.
- (+) correlation coefficient indicates that the two variables are in the same direction, while a negative (-) relationship indicates an inverse relationship between the two variables. Therefore, we can say that TPTHRT variable type1 and type2 also showed an inverse relationship.