Simulated example

Giovanni Saraceno

Contents

```
library(dplyr)
library(ggplot2)
```

```
## Warning: il pacchetto 'ggplot2' è stato creato con R versione 4.3.3
```

We have 50 subjects (25 with Alzheimer disease and 25 healthy individuals).
We analyze the response time (in milliseconds) of the Decoding Test VIPER-NAM: Images will appear on the screen for a short period of time and then disappear. Four letters will then appear, only one of which will correspond to the letter of the object. The user must choose the correct letter as quickly as possible. The response time is collected 10 times for each individual.
We also "collect" the Attention Control Scale (ATTC, i.e., self-report scale that is designed to measure attention focusing and attention shifting). The ATTC consists of 20 items (we will consider only one here) that are rated on a four-point Likert scale from 1 (almost never) to 4 (always). We also "collect" sex and age for each individual.

```
set.seed(1234)

Age = sample(c(15:60), 50, replace = TRUE)
Sex = sample(c(0, 1), 50, replace = TRUE)
Group = c(rep(0, 25),rep(1, 25))

generateData <- function(time){

  ATTC1 = ifelse(Group == 1, sample(c(3,4),1), sample(c(1:4),1))
  db <- data.frame(Age = Age,
                   Sex = Sex,
                   Group = Group,
                   ATTC1 = ATTC1,
                   Response_Time = log(Age) * rgamma(50, shape = 300) +
                        log(time) * rgamma(50, shape = 300) +
                        Sex * rgamma(50, shape = 300) +
                        Group *rgamma(50, shape = 300) +
                        log(ATTC1) * rgamma(50, shape = 300))

  return(db)
}

db <- sapply(c(1:10), function(x) generateData(x), simplify = FALSE)
db <- bind_rows(db)
db$Time <- rep(1:10, each = 50)
db$ID <- rep(1:50, 10)
```

a. Modify the `Sex` and `Group` variables such that: (i) for `Sex` 0 corresponds to "M" and 1 to "F"; (ii) for

Group 0 corresponds to "Control" while 1 to "Case.

```
db <- db %>%
  mutate(Sex = ifelse(Sex == 0, "M", "F"),
         Group = ifelse(Group == "0", "Control", "Case"))
```

b. Save the data set into a .csv file, remove everything from the environment and reload the data set.

```
write.csv(x = db, file = "first_dataframe.csv",row.names = FALSE)
rm(list=ls())
db <- read.csv("first_dataframe.csv")
```

c. Investigate the structure and summary of the data.

```
str(db)
```

```
## 'data.frame':    500 obs. of  7 variables:
##  $ Age          : int  42 30 36 51 58 23 19 52 30 18 ...
##  $ Sex          : chr  "F" "F" "M" "F" ...
##  $ Group        : chr  "Control" "Control" "Control" "Control" ...
##  $ ATTC1        : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ Response_Time: num  1655 1502 1206 1657 1453 ...
##  $ Time         : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ ID           : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
summary(db)
```

```
##       Age            Sex               Group               ATTC1
##  Min.   :16.00   Length:500         Length:500         Min.   :1.0
##  1st Qu.:22.00   Class :character   Class :character   1st Qu.:2.0
##  Median :35.50   Mode  :character   Mode  :character   Median :3.0
##  Mean   :35.94                                         Mean   :2.9
##  3rd Qu.:50.00                                         3rd Qu.:4.0
##  Max.   :58.00                                         Max.   :4.0
##  Response_Time       Time             ID
##  Min.   :1067   Min.   : 1.0   Min.   : 1.0
##  1st Qu.:1837   1st Qu.: 3.0   1st Qu.:13.0
##  Median :2107   Median : 5.5   Median :25.5
##  Mean   :2104   Mean   : 5.5   Mean   :25.5
##  3rd Qu.:2404   3rd Qu.: 8.0   3rd Qu.:38.0
##  Max.   :3015   Max.   :10.0   Max.   :50.0
```

d. Transform the character variables as factor, so R analyze them as categorical variables.

```
db$Group <- as.factor(db$Group)
levels(db$Group)
```

```
## [1] "Case"    "Control"
```

```
db$Sex <- as.factor(db$Sex)
levels(db$Sex)
```

```
## [1] "F" "M"
```

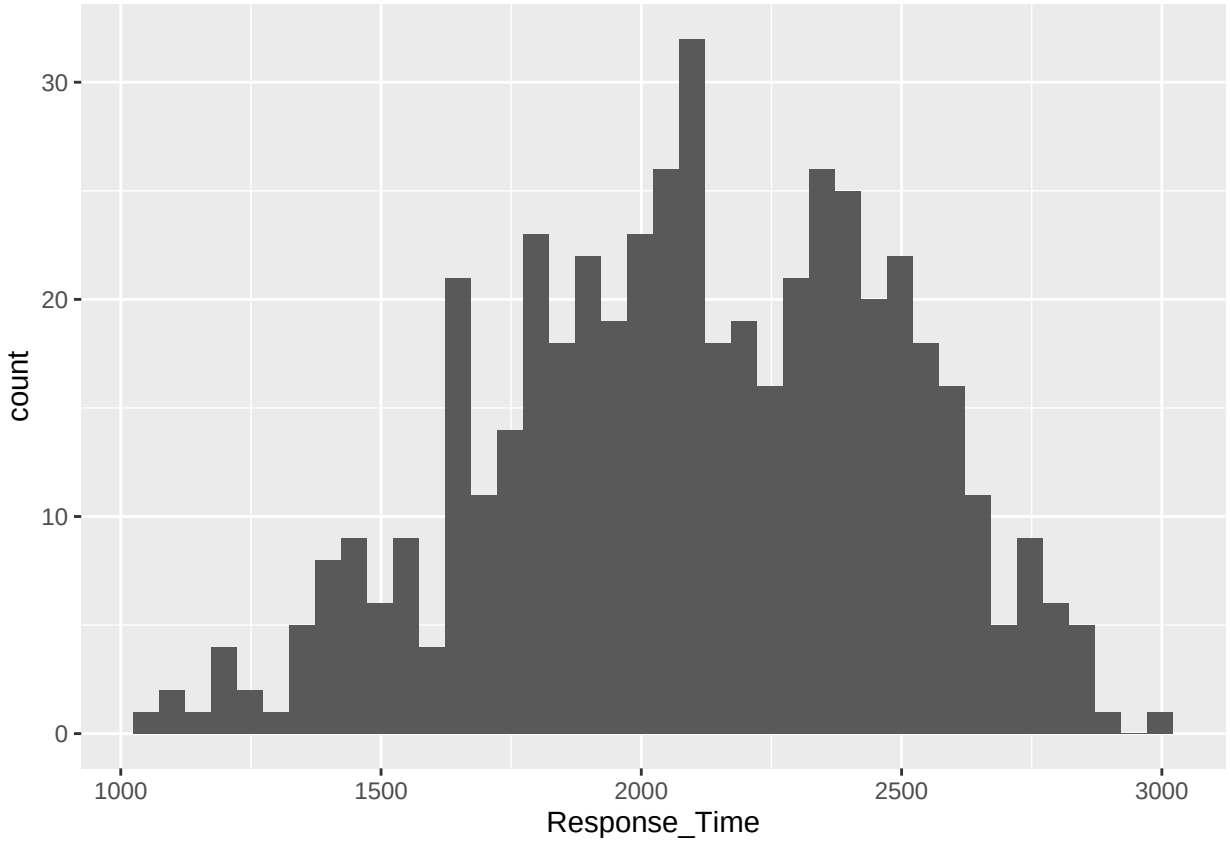
```
str(db)
```

```
## 'data.frame':    500 obs. of  7 variables:
##  $ Age          : int  42 30 36 51 58 23 19 52 30 18 ...
##  $ Sex          : Factor w/ 2 levels "F","M": 1 1 2 1 2 2 1 2 1 2 ...
##  $ Group        : Factor w/ 2 levels "Case","Control": 2 2 2 2 2 2 2 2 2 2 ...
```

```
##  $ ATTC1         : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ Response_Time: num  1655 1502 1206 1657 1453 ...
##  $ Time          : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ ID            : int  1 2 3 4 5 6 7 8 9 10 ...
```

e. Represent the variable `Response_Time` by an histogram.

```
ggplot(data = db,
       mapping = aes(x = Response_Time)) +
  geom_histogram(bins = 40)
```



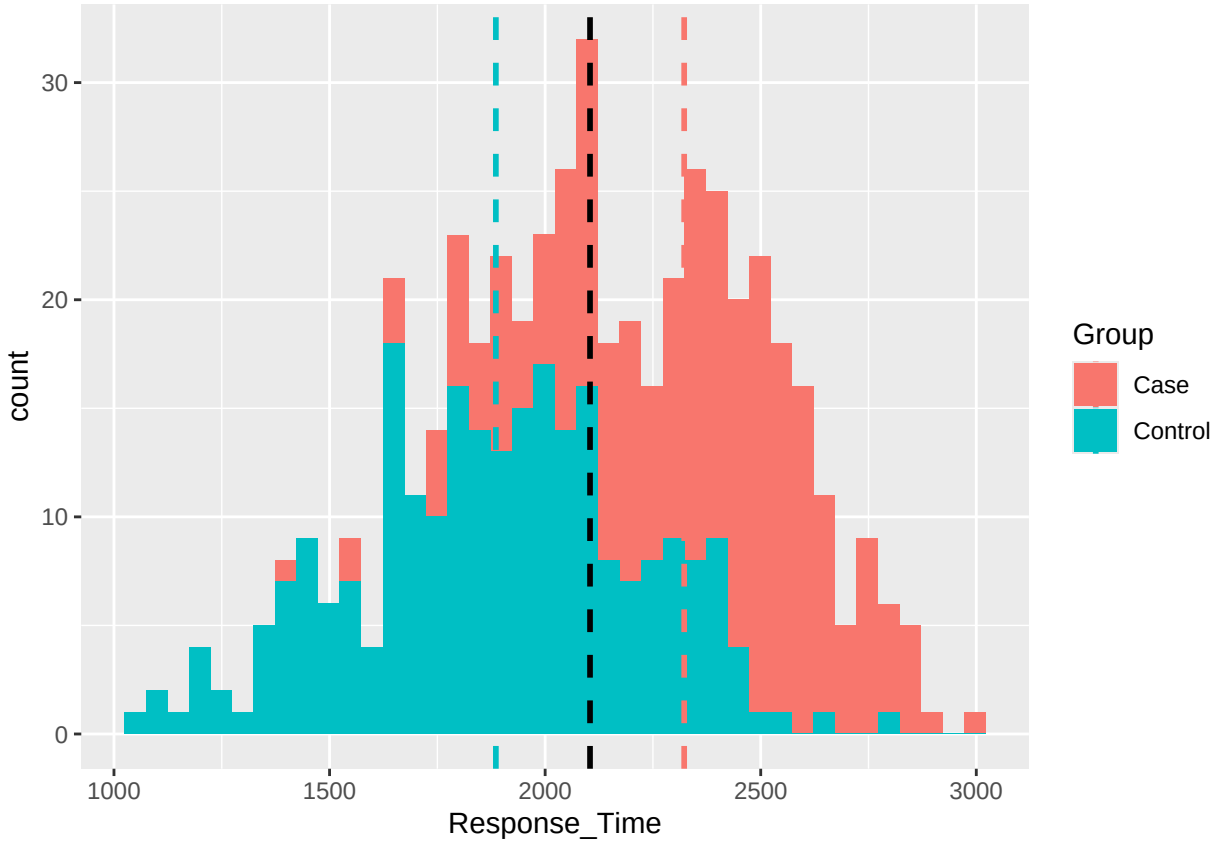
If you want to change the `stat_bin()` you can set the argument `bins = x` inside the `geom_histogram()` function where `x` is the number of bins. Note also that there are some visible outliers in the right part of the histogram.

f. Divide the observations by the `Group` variables and fill the histogram with two different colors.

```
ggplot() +
  geom_histogram(data = db, aes(x = Response_Time,
                     fill = Group), bins=40)+
  geom_vline(data = db, aes(xintercept = mean(Response_Time, na.rm = TRUE)),
              linetype="dashed", size=1)+
  geom_vline(data = db %>% filter(Group =="Case"),
             aes(xintercept= mean(Response_Time, na.rm = TRUE),
                 colour =Group),
             linetype="dashed",
             size=1)+
  geom_vline(data = db %>% filter(Group =="Control"),
             aes(xintercept=mean(Response_Time, na.rm = TRUE),
```

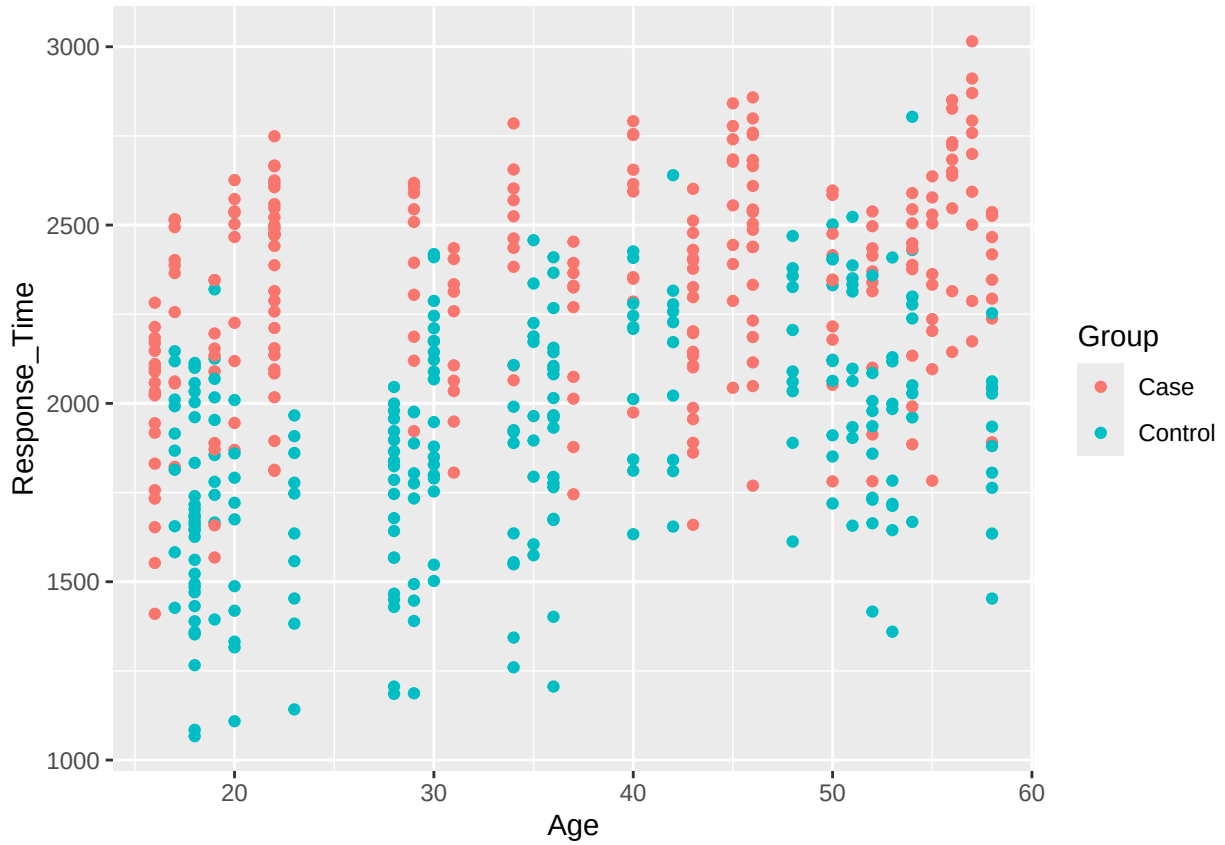
```
                colour = Group),
            linetype="dashed",
            size=1)
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



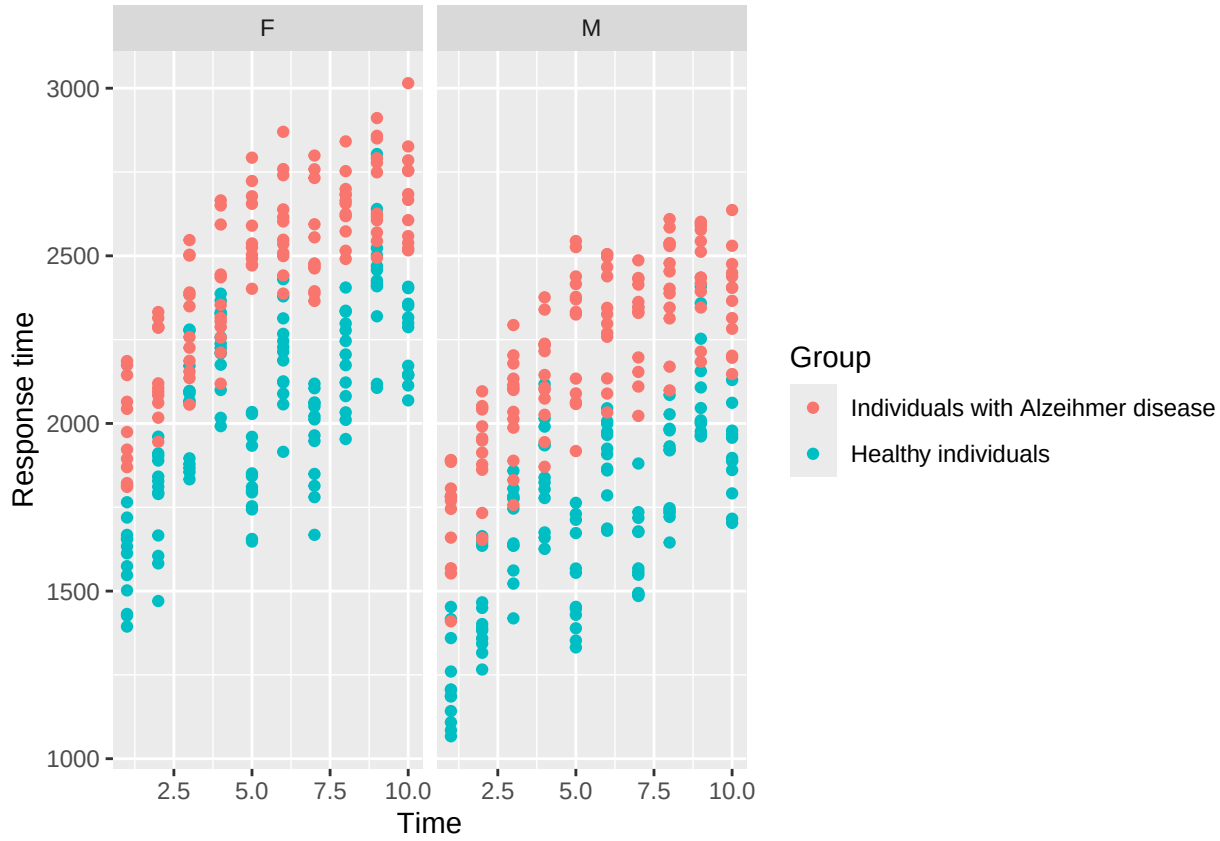
g. Generate the scatter plot between `Response_Time` and `Age`, colored by Group.

```
ggplot(data = db,
       mapping = aes(x = Age,
                     y = Response_Time,
                     color = Group)) +
  geom_point()
```



Generate another scatter dividing also by `Sex`.

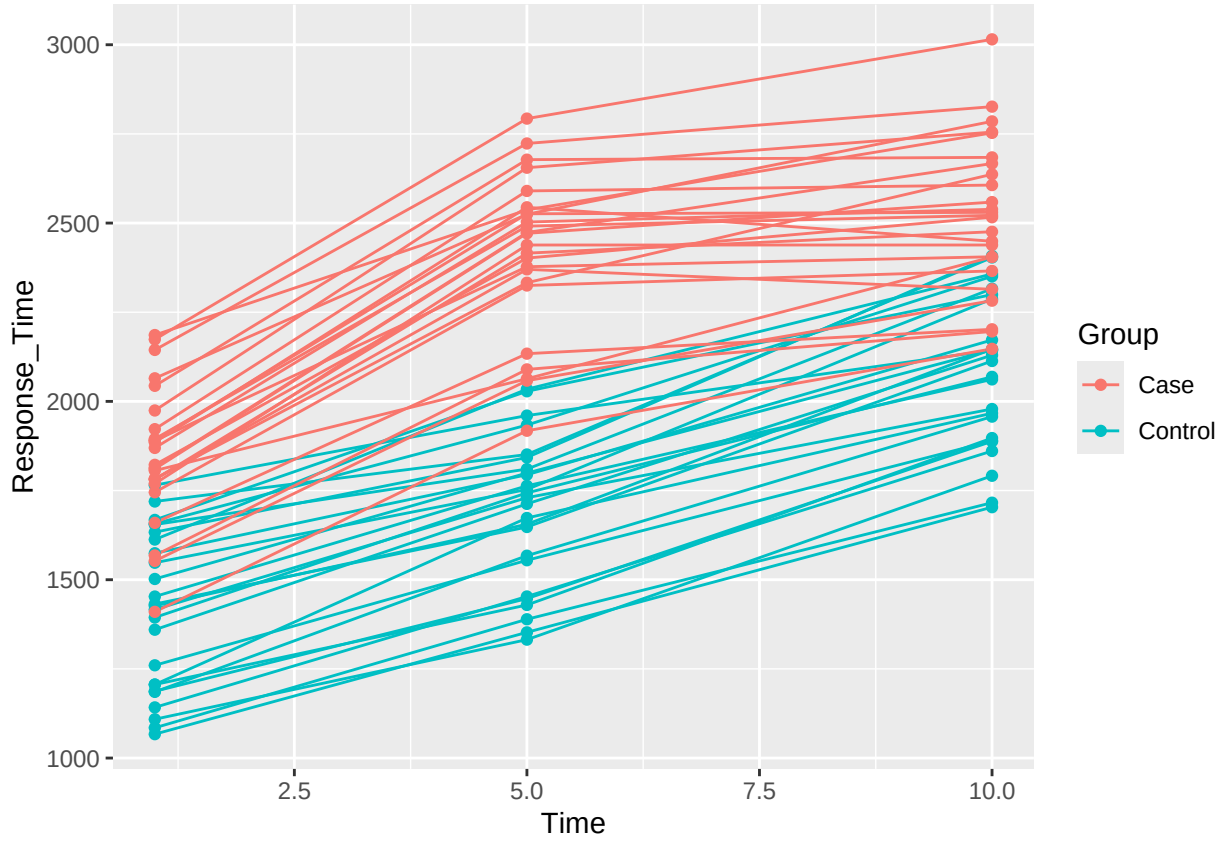
```
ggplot(data = db,
       mapping = aes(x = Time, y = Response_Time, color = Group)) +
  geom_point() +
  facet_wrap(. ~Sex)+
       scale_colour_discrete(name = "Group", labels = c("Individuals with Alzeihmer disease", "Healthy
  ylab("Response time")
```



- `facet_wrap(. ~Sex)` - Here we put the variable to divide the scatter plots, i.e., the Sex object. R directly creates two plots one for each level of the Sex factor object. - `scale_colour_discrete` - Here We can change the labels of the Group legend. We must specify the name of the object as a string (i.e., Group) and the labels desired as vector. - `ylab` - Here We can change the y axis.

h. Generate line plots to understand individual data

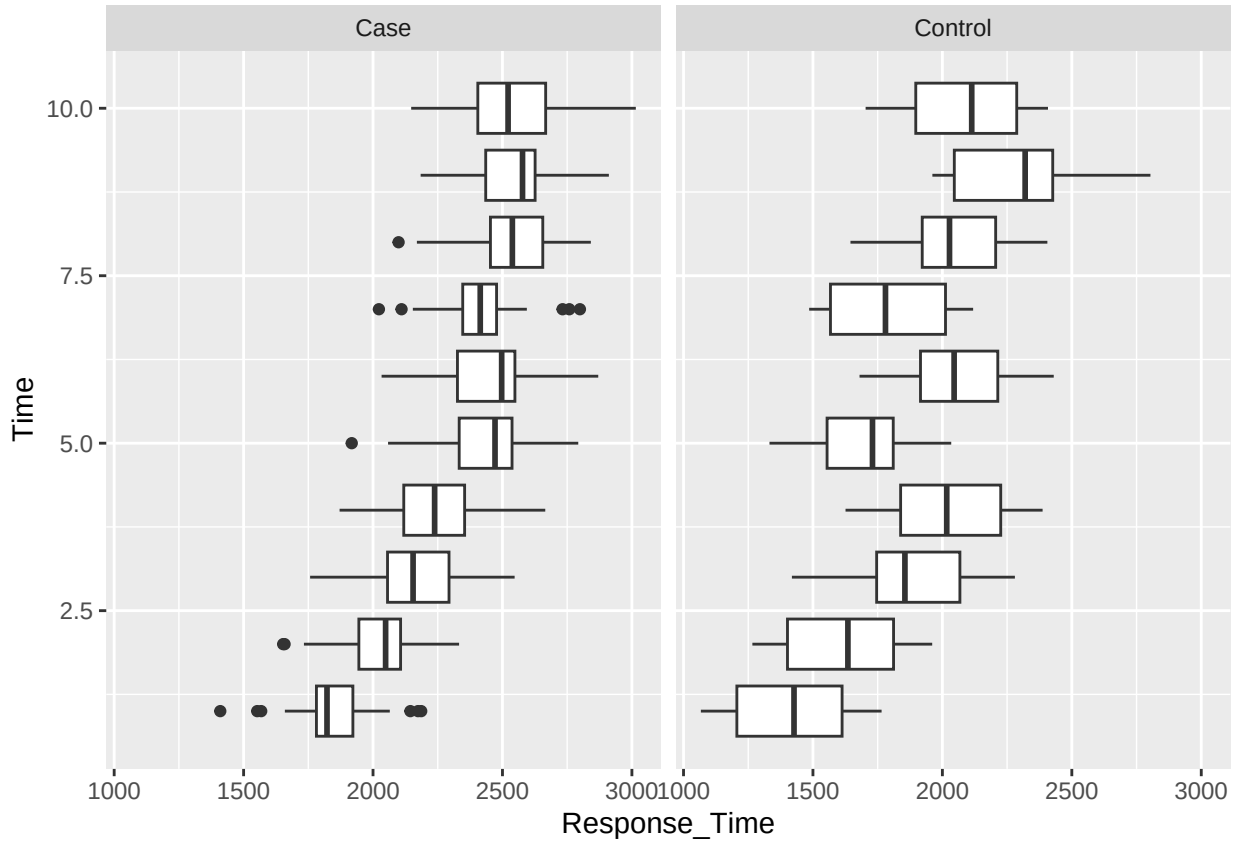
```
ggplot(data = db %>% filter(Time %in% c(1,5,10)),
       mapping = aes(x = Time, y = Response_Time, group = ID, color = Group)) +
  geom_line() +
   geom_point(aes(fill=Group),
              shape=21)
```



- We consider only observations at time 1, 5 and 10 - `group = ID, color = Group` - Here We group the observations by ID (i.e., one line for each individual) and We color them by `Group`. - shape = 21 specify the type of mark (i.e., point) and fill=Group colors the points by Group.

i. Plot the distribution of the `Response_Time` variable across different time points (using boxplots).

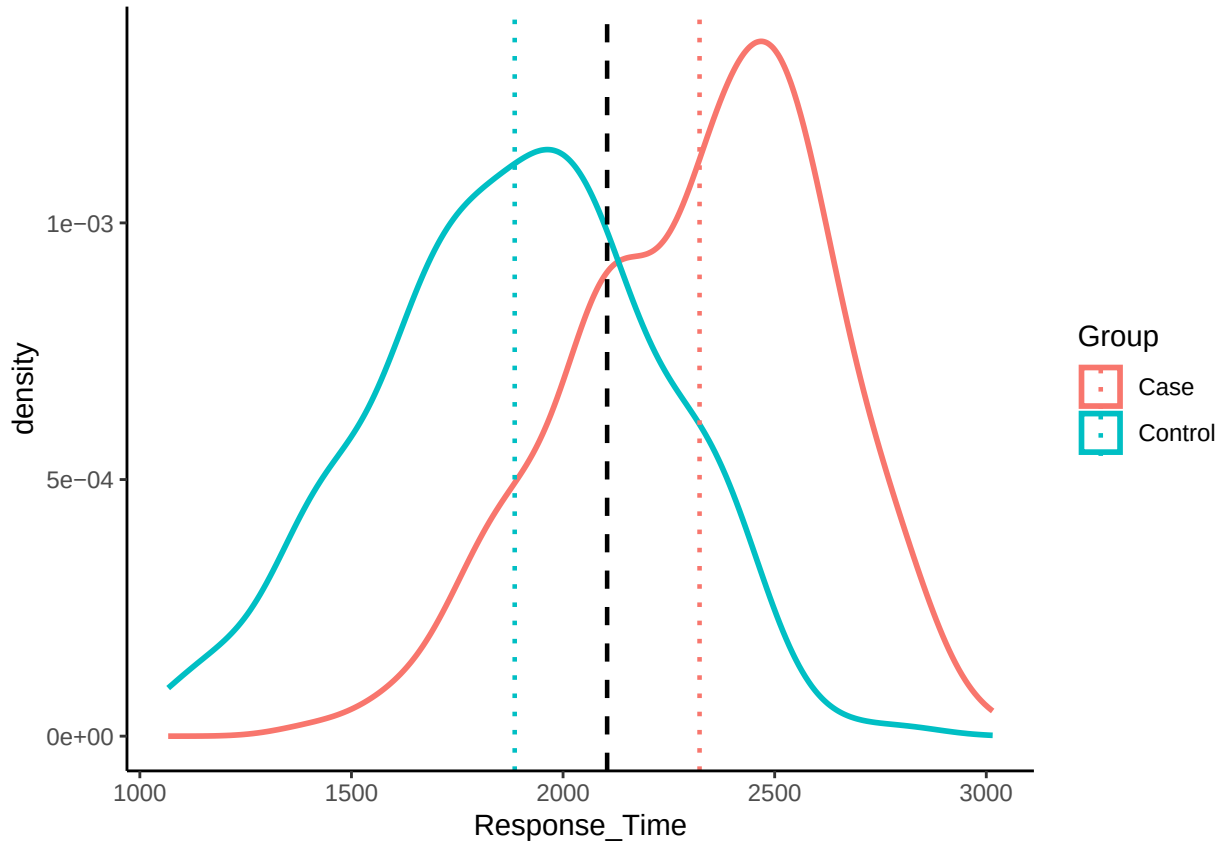
```
ggplot(data = db,
       mapping = aes(x = Response_Time, y = Time, group = Time)) +
  geom_boxplot() +
  facet_grid(. ~Group)
```



- `group = Time` - Here We group the observations by Time,
- `facet_grid(. ~Group)` - We divide the plots into two graphs, one for the case group and one for the control group.
  Note the outliers in the right part of the graph.

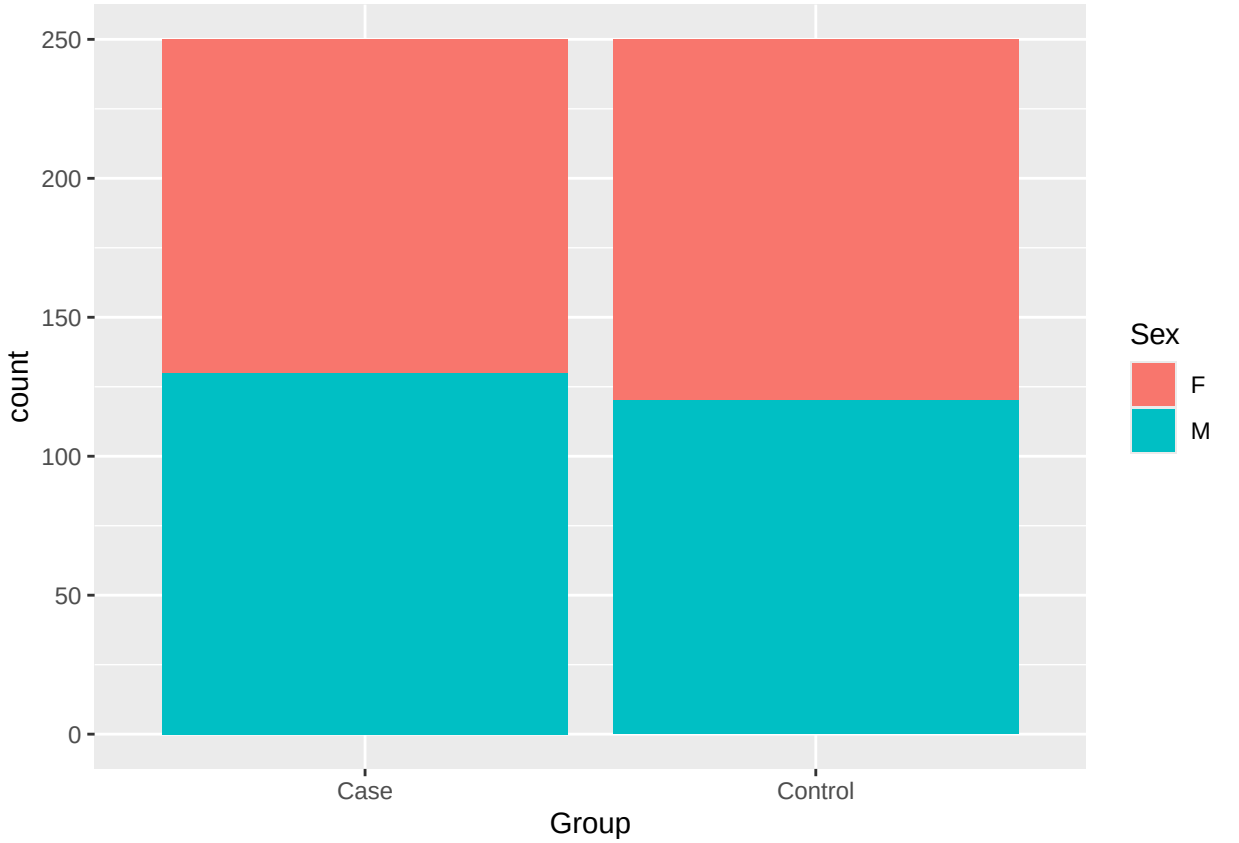
l. Display the density plot of `Response_Time` colored with respect to the `Group` variable.

```
ggplot()  +
  geom_density(data = db,
               aes(x = Response_Time, color = Group), size=1) +
  geom_vline(data = db %>% filter(Group == "Case"),
             aes(xintercept = mean(Response_Time, na.rm = TRUE),
                 color = Group), size=.8, linetype = "dotted")+
  geom_vline(data = db %>% filter(Group == "Control"),
             aes(xintercept = mean(Response_Time, na.rm = TRUE),
                 color = Group), size=.8, linetype = "dotted")+
  geom_vline(data = db,
             aes(xintercept = mean(Response_Time, na.rm = TRUE)), size=.8, linetype = "dashed") +
    theme_classic()
```



m. Display the distribution of people with respect to `Sex` and `Group`.

```
ggplot(data = db,
       mapping = aes(x = Group, fill = Sex)) +
  geom_bar()
```



n. check the correlation between the quantitative variables and Plot the correlation matrix.

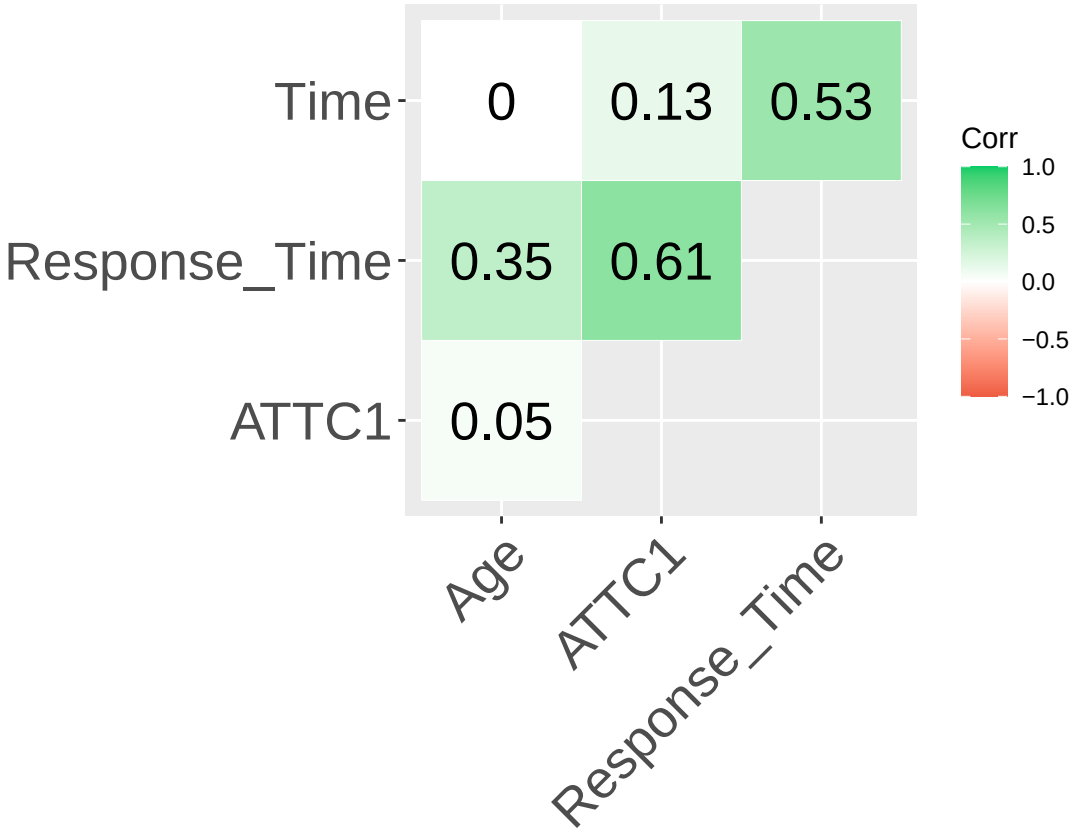
```
corr_matrix <- round(cor(db[,!(colnames(db) %in% c("Sex", "Group", "ID"))], use = "complete.obs"), 2)
corr_matrix[lower.tri(corr_matrix)] <- 0
corr_matrix
```

```
##                Age ATTC1 Response_Time Time
## Age              1  0.05          0.35 0.00
## ATTC1            0  1.00          0.61 0.13
## Response_Time    0  0.00          1.00 0.53
## Time             0  0.00          0.00 1.00
```

```
library(ggcorrplot)
```

```
## Warning: il pacchetto 'ggcorrplot' è stato creato con R versione 4.3.3
```

```
ggcorrplot(corr_matrix,
           type = "upper",
           lab = T,
           lab_size = 7,
           outline.col = "white",
           colors = c("tomato2", "white", "springgreen3"),
           title = "",
           ggtheme = theme_gray,
           pch.cex = 30,
           tl.cex = 20)
```



o. Remove observations with NA

```
db <- db %>% filter(!is.na(Response_Time))
# Alternatively, we can use the basic commands of R:
#db <- db[!is.na(db$Response_Time),]
```

p. Ideas for other plots?