

Hands-on Biostatistics 2: Analysis of nonparametric data in R

Eliana Ibrahimi

Two related samples: Wilcoxon Signed rank test

Read the data

```
# Data in two numeric vectors
# ++++++
# Weight of the mice before treatment
weight_before <- c(200.1, 190.9, 192.7, 213, 241.4, 196.9, 172.2, 185.5, 205.2, 193.7)
# Weight of the mice after treatment
weight_after <- c(392.9, 393.2, 395.1, 393, 334, 327.9, 222, 383.9, 392.3, 352.2)
# Create a data frame
my_data <- data.frame(
  group = rep(c("before", "after"), each = 10),
  weight = c(weight_before, weight_after)
)
print(my_data)
```

```
##      group weight
## 1  before  200.1
## 2  before  190.9
## 3  before  192.7
## 4  before  213.0
## 5  before  241.4
## 6  before  196.9
## 7  before  172.2
## 8  before  185.5
## 9  before  205.2
## 10 before  193.7
## 11  after  392.9
## 12  after  393.2
## 13  after  395.1
## 14  after  393.0
## 15  after  334.0
## 16  after  327.9
## 17  after  222.0
## 18  after  383.9
## 19  after  392.3
## 20  after  352.2
```

Explore the data: Compute summary statistics

```
#install.packages("dplyr")
```

```
library(dplyr)
```

```
##  
## Attaching package: 'dplyr'  
## The following objects are masked from 'package:stats':  
##  
##   filter, lag  
## The following objects are masked from 'package:base':  
##  
##   intersect, setdiff, setequal, union
```

```
group_by(my_data, group) %>%  
  summarise(  
    count = n(),  
    median = median(weight, na.rm = TRUE),  
    IQR = IQR(weight, na.rm = TRUE)  
  )
```

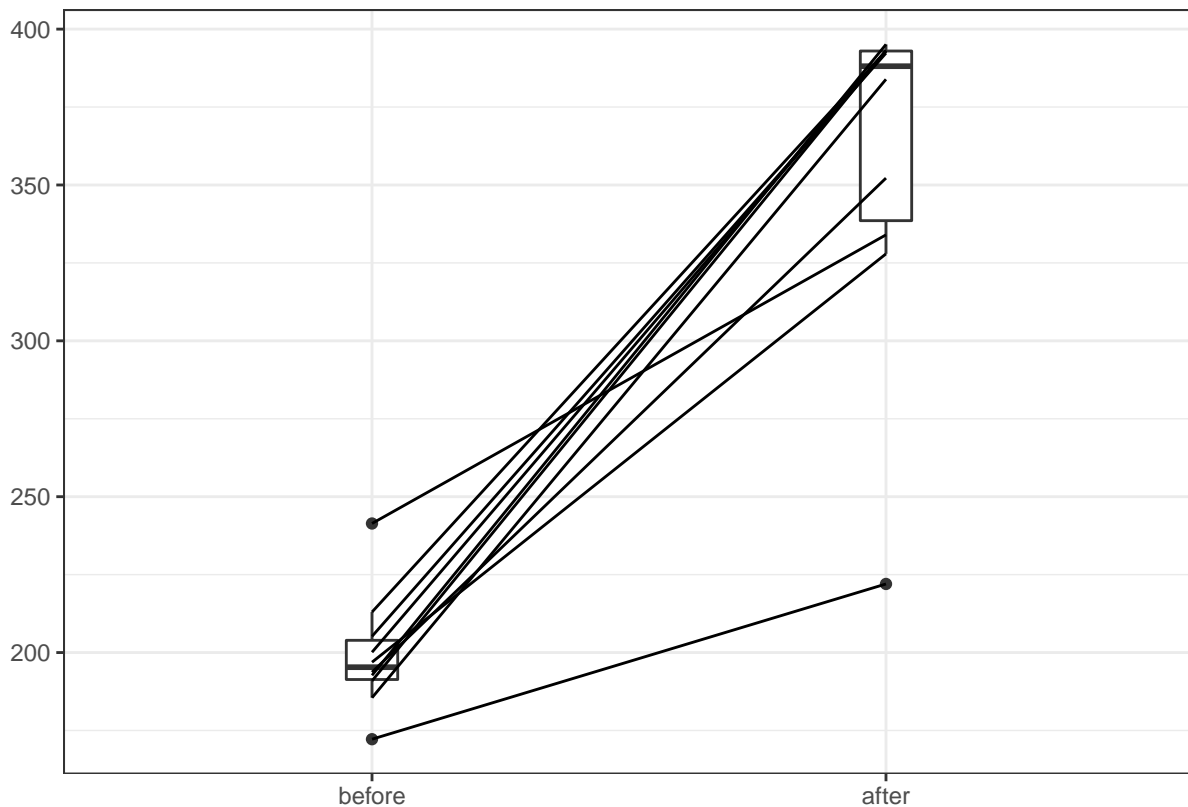
```
## 'summarise()' ungrouping output (override with '.groups' argument)  
## # A tibble: 2 x 4  
##   group count median  IQR  
##   <chr> <int> <dbl> <dbl>  
## 1 after     10   388.  54.4  
## 2 before    10   195.  12.6
```

```
# Subset weight data before treatment  
before <- subset(my_data, group == "before", weight,  
                 drop = TRUE)  
# subset weight data after treatment  
after <- subset(my_data, group == "after", weight,  
               drop = TRUE)  
# Plot paired data  
library(PairedData)
```

Visualize paired data

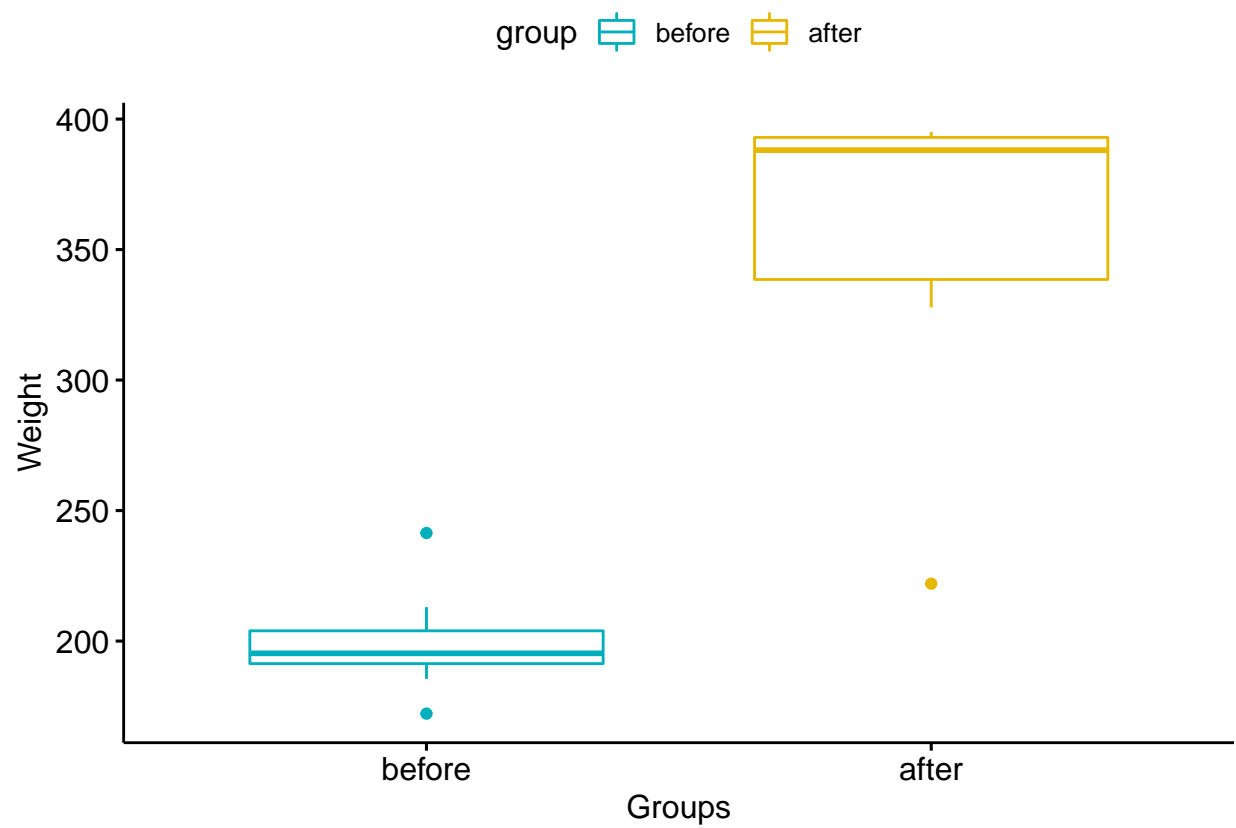
```
## Warning: package 'PairedData' was built under R version 4.0.5  
## Loading required package: MASS  
##  
## Attaching package: 'MASS'  
## The following object is masked from 'package:dplyr':  
##  
##   select  
## Loading required package: gld  
## Loading required package: mvtnorm  
## Loading required package: lattice  
## Loading required package: ggplot2  
##  
## Attaching package: 'PairedData'
```

```
## The following object is masked from 'package:base':
##
##      summary
pd <- paired(before, after)
plot(pd, type = "profile") + theme_bw()
```



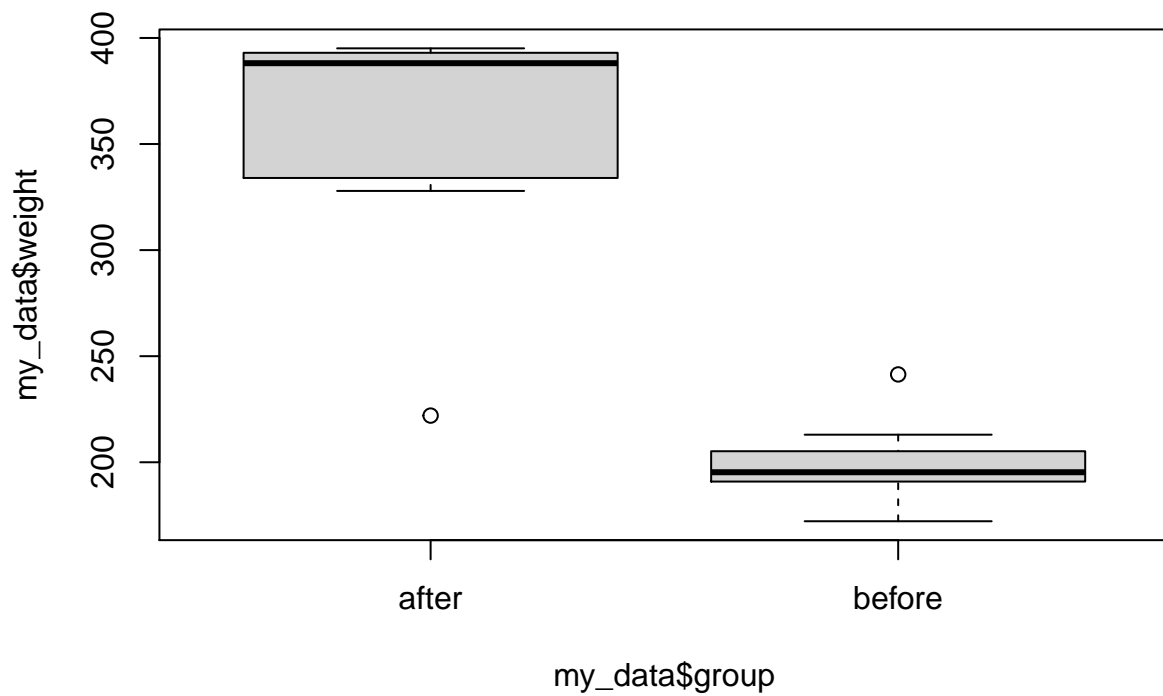
```
#install.packages("ggpubr")
```

```
# Plot weight by group and color by group
library("ggpubr")
ggboxplot(my_data, x = "group", y = "weight",
  color = "group", palette = c("#00AFBB", "#E7B800"),
  ylab = "Weight", xlab = "Groups")
```



#or you use simply

```
boxplot(my_data$weight~my_data$group)
```

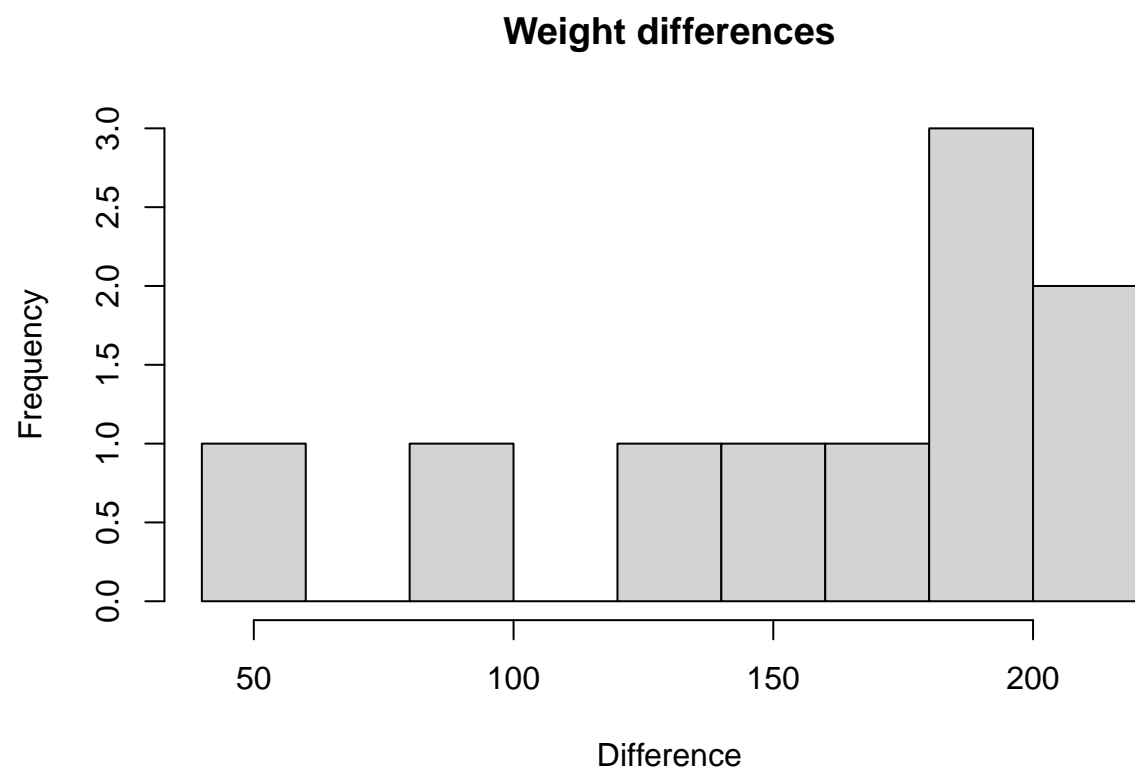


```
### Compute the difference between groups
```

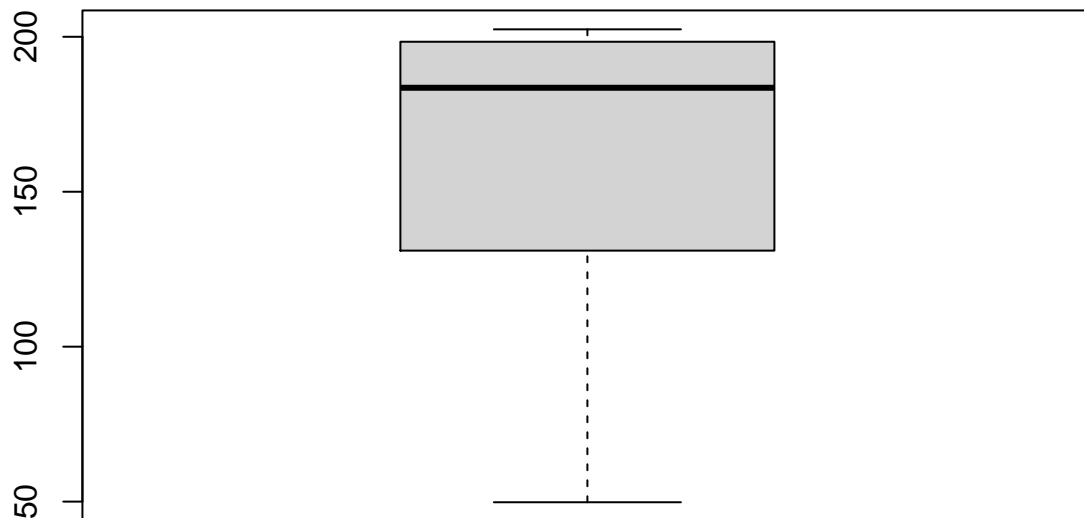
```
diff= weight_after - weight_before
diff
```

```
## [1] 192.8 202.3 202.4 180.0 92.6 131.0 49.8 198.4 187.1 158.5
```

```
hist(diff, main="Weight differences", xlab="Difference", breaks = 10)
```



```
boxplot(diff)
```



Run shapiro wilk test for differences

```
shapiro.test(diff)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  diff
## W = 0.81975, p-value = 0.02516
```

Run the Wilcoxon signed rank test

```
#install.packages("PairedData")
library(PairedData)

#In order to run the Wilcoxon signed rank test, use the command
#wilcoxon.test (variable1, variable2, Paired=TRUE, exact=FALSE)

#So for this example use
wilcox.test(weight_after, weight_before, paired = TRUE, exact=FALSE)
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  weight_after and weight_before
## V = 55, p-value = 0.005922
```

```
## alternative hypothesis: true location shift is not equal to 0
```

Two independent samples

Read the data

```
# Data in two numeric vectors
women_weight <- c(38.9, 61.2, 73.3, 21.8, 63.4, 64.6, 48.4, 48.8, 48.5)
men_weight <- c(67.8, 60, 63.4, 61, 89.4, 83.3, 67.3, 91.3, 62.4)
# Create a data frame
data <- data.frame(
  group = rep(c("Woman", "Man"), each = 9),
  weight = c(women_weight, men_weight)
)
print(data)
```

```
##   group weight
## 1  Woman   38.9
## 2  Woman   61.2
## 3  Woman   73.3
## 4  Woman   21.8
## 5  Woman   63.4
## 6  Woman   64.6
## 7  Woman   48.4
## 8  Woman   48.8
## 9  Woman   48.5
## 10 Man    67.8
## 11 Man    60.0
## 12 Man    63.4
## 13 Man    61.0
## 14 Man    89.4
## 15 Man    83.3
## 16 Man    67.3
## 17 Man    91.3
## 18 Man    62.4
```

```
attach(data)
```

Compute summary statistics

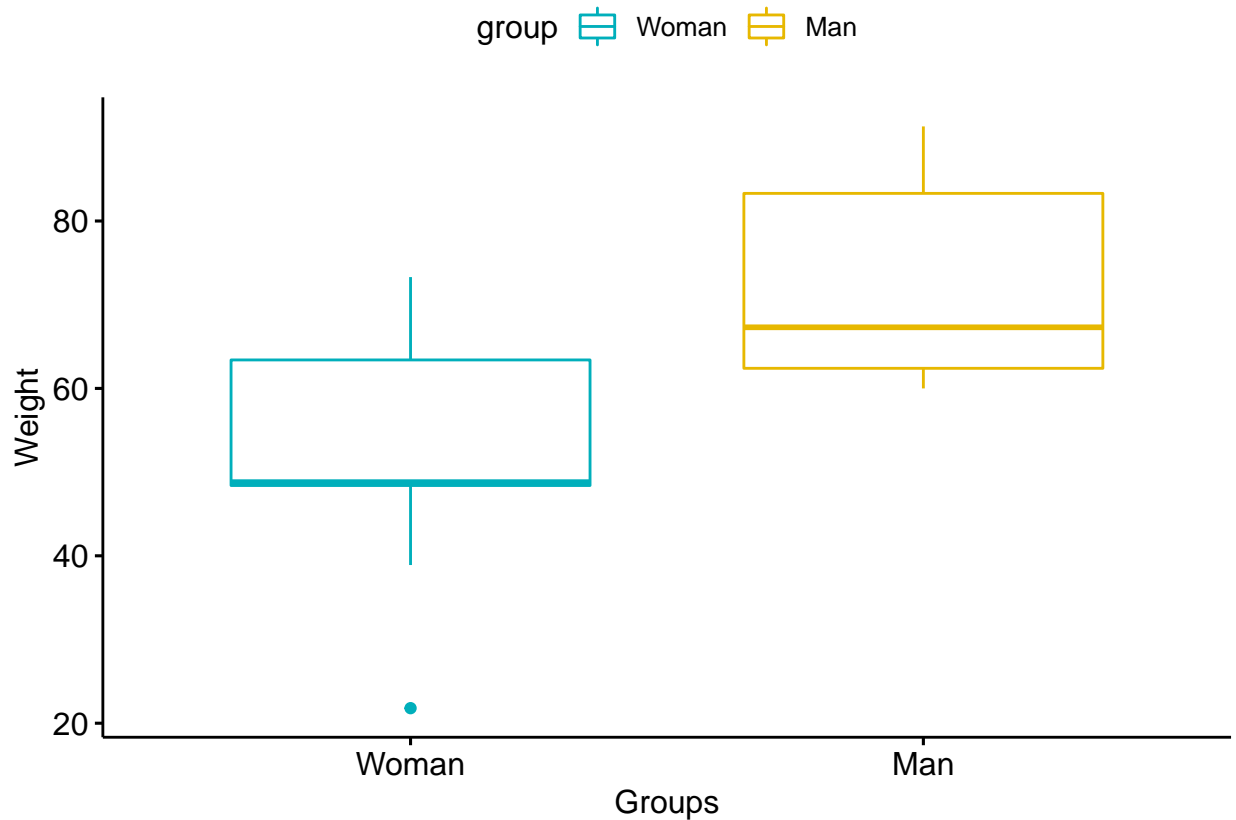
```
library(dplyr)
group_by(data, group) %>%
  summarise(
    count = n(),
    median = median(weight, na.rm = TRUE),
    IQR = IQR(weight, na.rm = TRUE)
  )
```

```
## 'summarise()' ungrouping output (override with '.groups' argument)
```

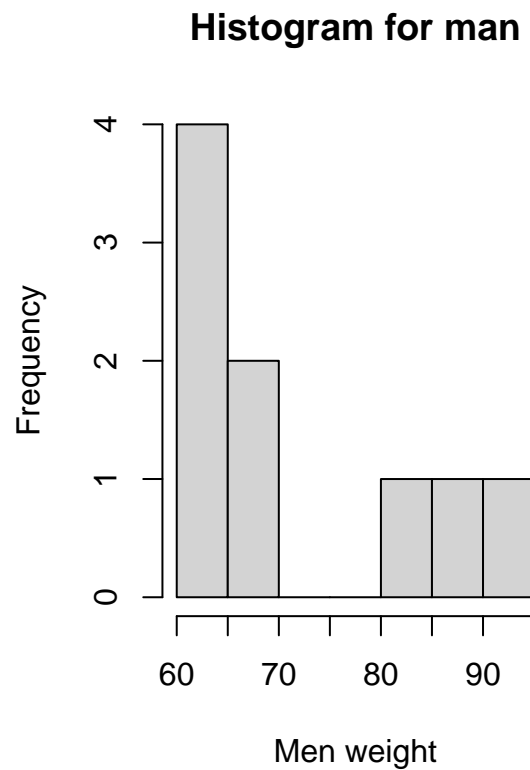
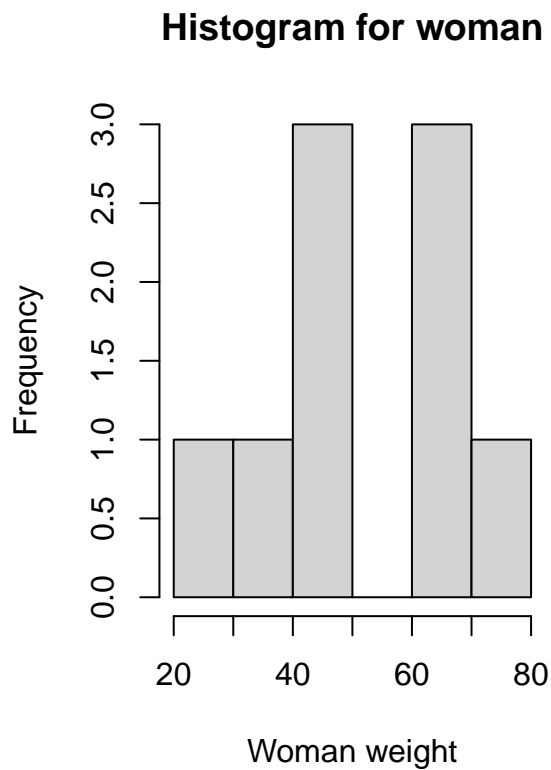
```
## # A tibble: 2 x 4
##   group count median  IQR
##   <chr> <int>   <dbl> <dbl>
## 1 Man      9    67.3   20.9
## 2 Woman    9    48.8    15
```


Visualize the data

```
# Plot weight by group and color by group
library("ggpubr")
ggboxplot(data, x = "group", y = "weight",
  color = "group", palette = c("#00AFBB", "#E7B800"),
  ylab = "Weight", xlab = "Groups")
```



```
#Plot histograms for the dependent ulcer free weeks by group
par(mfrow=c(1,2))
hist(data$weight[group=='Woman'],main='Histogram for woman',xlab='Woman weight')
hist(data$weight[group=='Man'],main='Histogram for man',xlab='Men weight')
```



Perform Shapiro Wilk test by group

```
shapiro.test(data$weight[group=="Woman"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data$weight[group == "Woman"]
## W = 0.94266, p-value = 0.6101
```

```
shapiro.test(data$weight[group=="Man"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data$weight[group == "Man"]
## W = 0.81403, p-value = 0.0295
```

Run Mann Whitney test using wilcox.test function

```
wilcox.test(weight~group, paired= FALSE, exact=FALSE)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  weight by group
## W = 65.5, p-value = 0.03042
```

```
## alternative hypothesis: true location shift is not equal to 0
```

Two or more independent samples

Import the data from csv file

```
library(readr)
dataplant <- read_csv("C:/Users/user/Desktop/HOB_2021/dataplant.csv",
  col_types = cols(height = col_number(),
    group = col_factor(levels = c("ctrl",
      "trt1", "trt2"))))

attach(dataplant)
```

```
## The following object is masked from data:
```

```
##
```

```
##      group
```

Compute summary statistics by group

```
library(dplyr)
group_by(dataplant, group) %>%
  summarise(
    count = n(),
    mean = mean(height, na.rm = TRUE),
    sd = sd(height, na.rm = TRUE),
    median = median(height, na.rm = TRUE),
    IQR = IQR(height, na.rm = TRUE)
  )
```

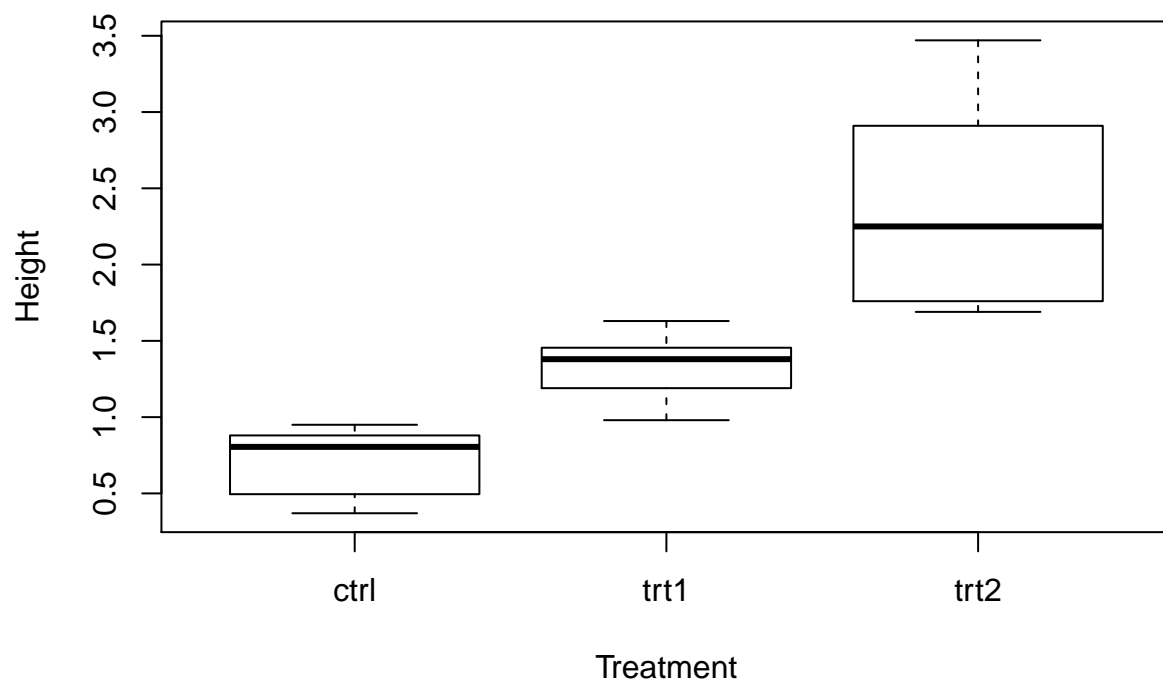
```
## 'summarise()' ungrouping output (override with '.groups' argument)
```

```
## # A tibble: 3 x 6
```

```
##   group count  mean    sd median  IQR
##   <fct> <int> <dbl> <dbl> <dbl> <dbl>
## 1 ctrl      8  0.71 0.230  0.805 0.318
## 2 trt1      8  1.33 0.210  1.38  0.222
## 3 trt2     14  2.37 0.634  2.25  1.07
```

Visualize the data

```
boxplot(dataplant$height~dataplant$group, col="white", xlab = "Treatment", ylab = "Height")
```



```
library(car)
```

```
## Warning: package 'car' was built under R version 4.0.3
```

```
## Loading required package: carData
```

```
##
```

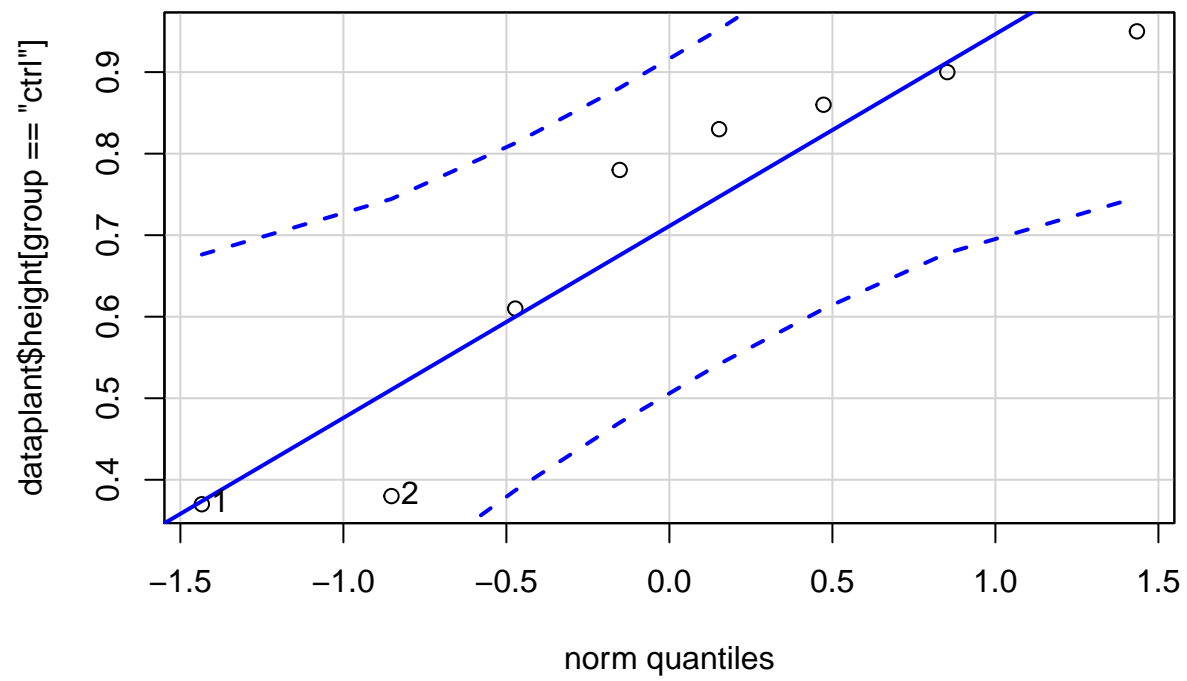
```
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
```

```
##
```

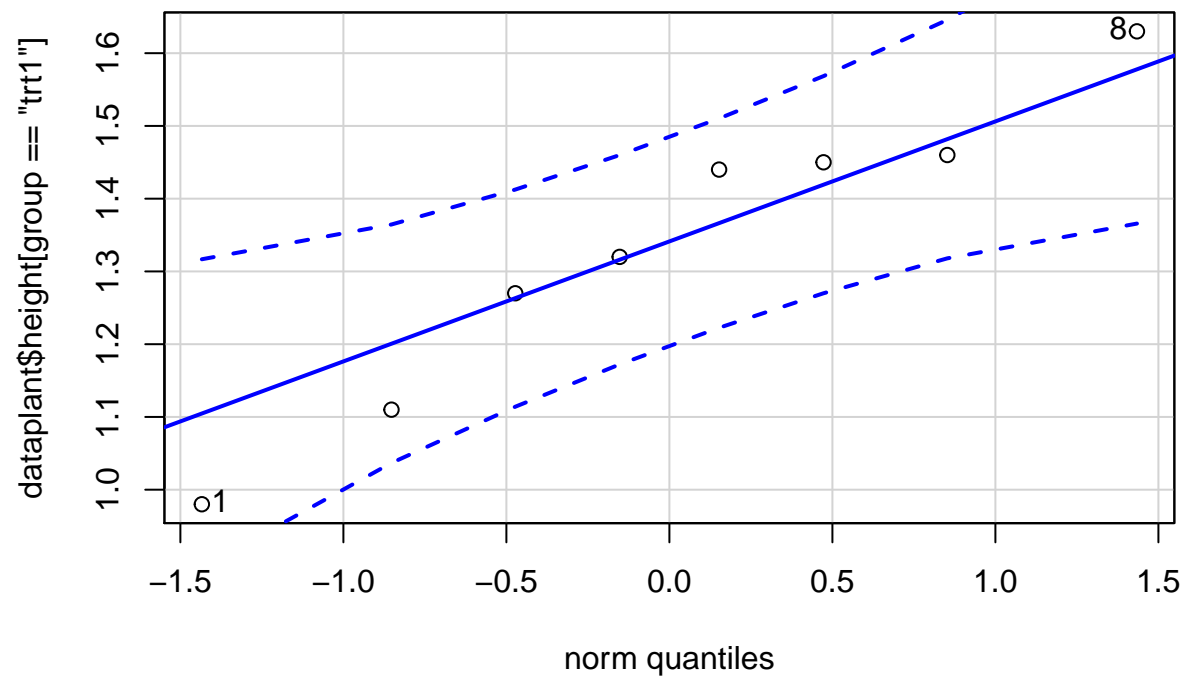
```
##      recode
```

```
qqPlot(dataplant$height[group=="ctrl"])
```



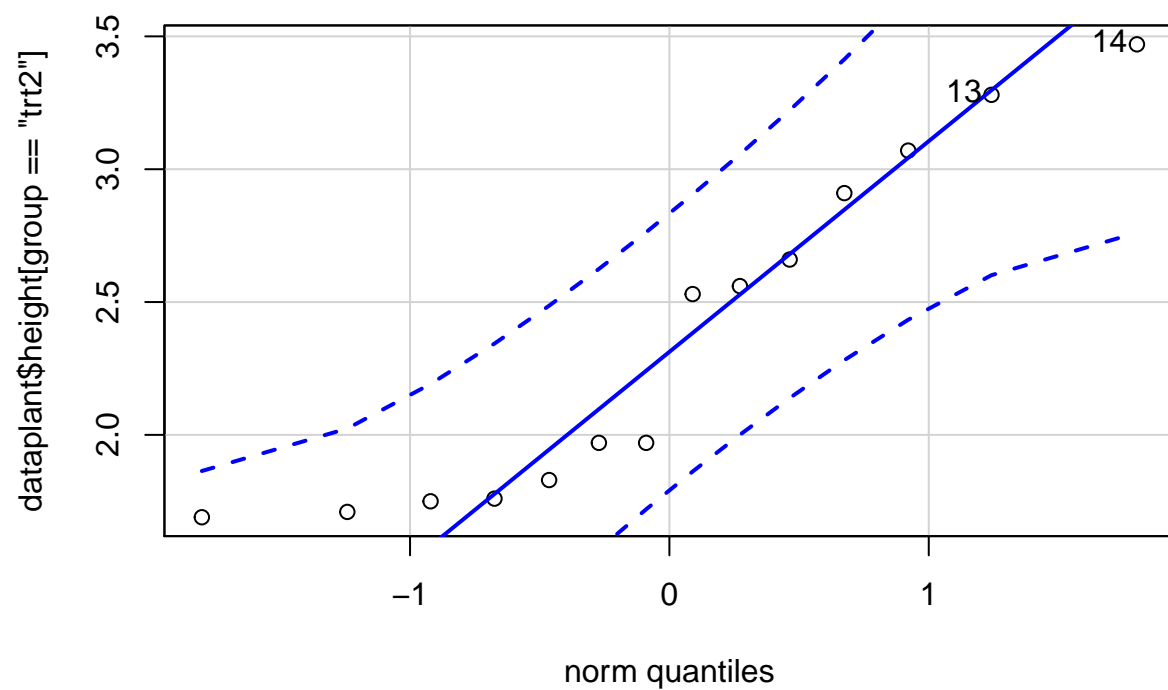
```
## [1] 1 2
```

```
qqPlot(dataplant$height[group=="trt1"])
```

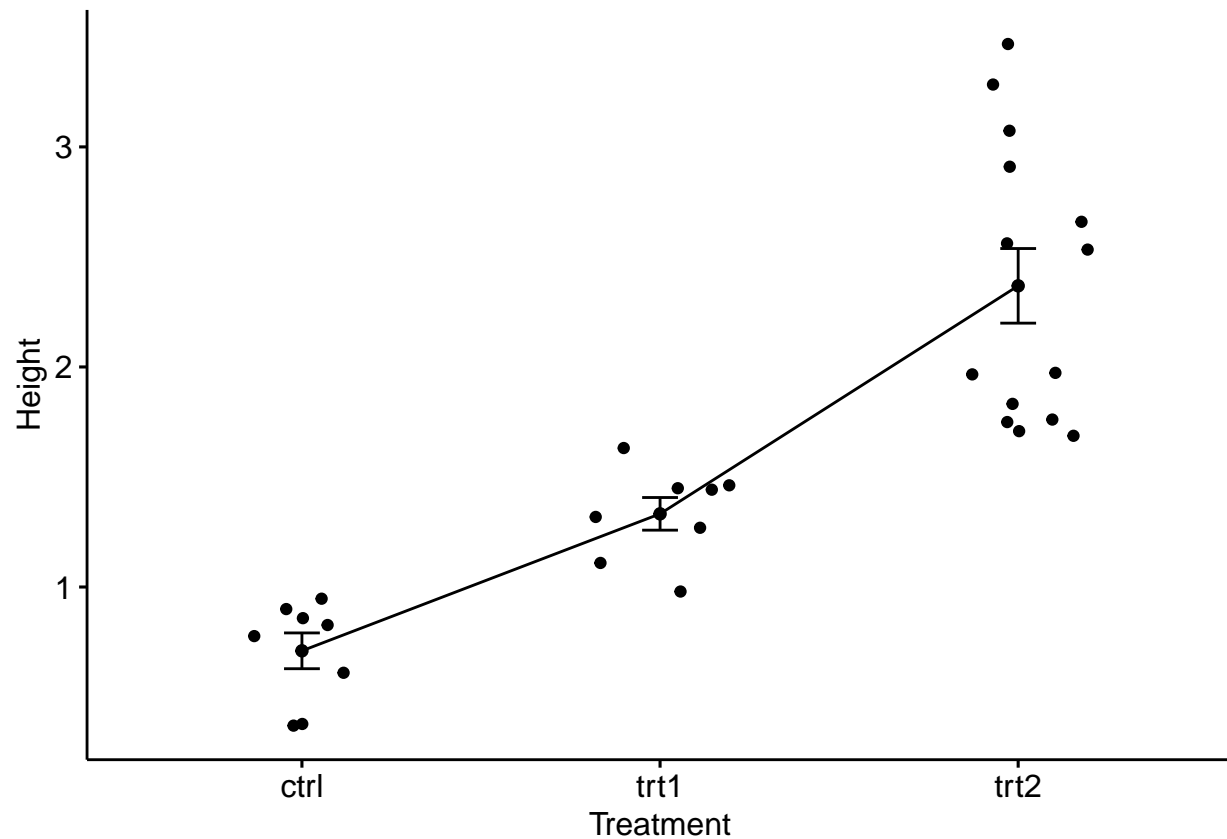


```
## [1] 1 8
```

```
qqPlot(dataplant$height[group=="trt2"])
```



```
## [1] 14 13
# Mean plots
# ++++++
# Plot weight by group
# Add error bars: mean_se
# (other values include: mean_sd, mean_ci, median_iqr, ....)
library("ggpubr")
ggline(dataplant, x = "group", y = "height",
  add = c("mean_se", "jitter"),
  order = c("ctrl", "trt1", "trt2"),
  ylab = "Height", xlab = "Treatment")
```



Perform the Kruskal-Wallis test

```
kruskal.test(height ~ group, data = dataplant)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  height by group
## Kruskal-Wallis chi-squared = 24.986, df = 2, p-value = 3.752e-06
```

Perform the pairwise Wilcoxon test

```
pairwise.wilcox.test(dataplant$height, dataplant$group, p.adj='bonferroni', exact=F)
```

```
##
## Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  dataplant$height and dataplant$group
##
##      ctrl      trt1
## trt1 0.00282 -
## trt2 0.00045 0.00045
##
## P value adjustment method: bonferroni
```


Spearman correlation

Read the data and create the dataframe

```
#Use the weight dataset from example 1
# Data in two numeric vectors
# ++++++
# Weight of the mice before treatment
x <-c(200.1, 190.9, 192.7, 213, 241.4, 196.9, 172.2, 185.5, 205.2, 193.7)
# Weight of the mice after treatment
y <-c(392.9, 393.2, 395.1, 393, 410, 327.9, 222, 383.9, 392.3, 352.2)

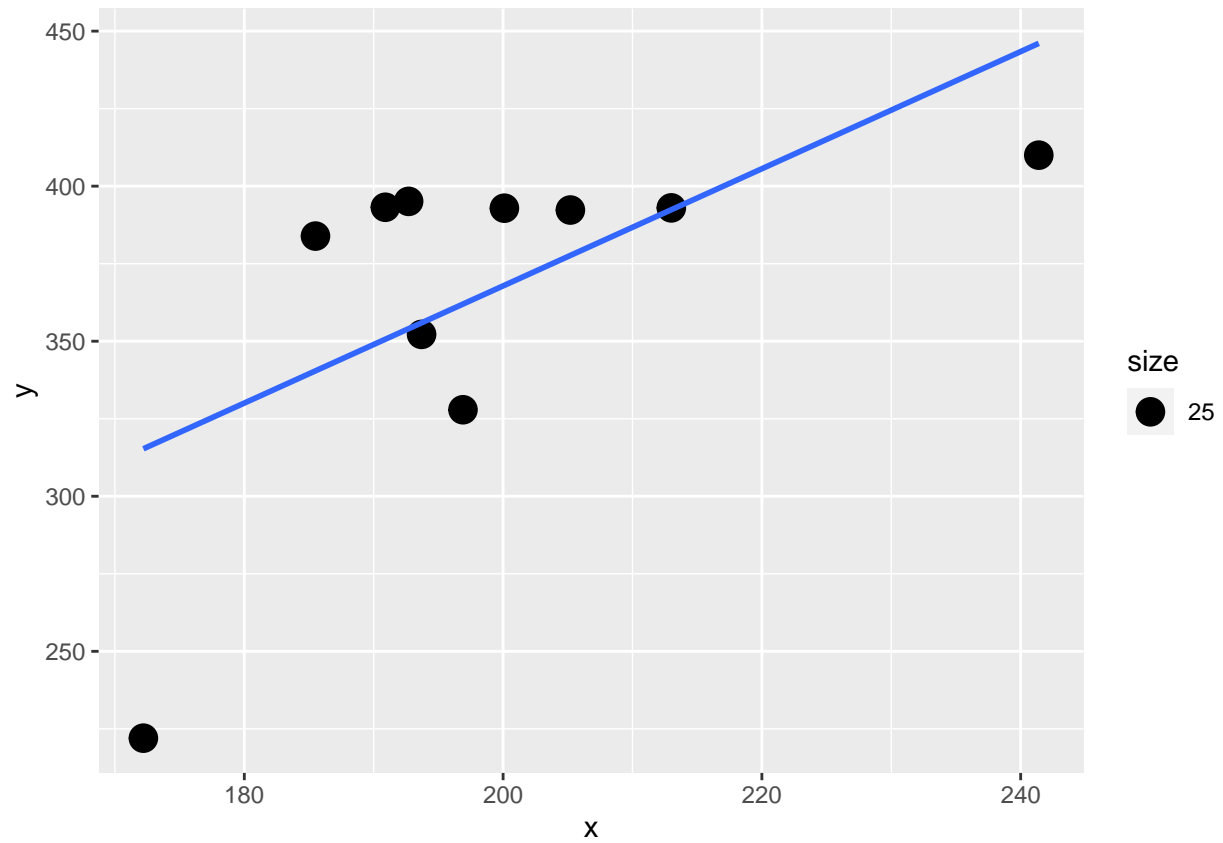
my_data <- data.frame(x,y)
print(my_data)
```

```
##           x           y
## 1  200.1  392.9
## 2  190.9  393.2
## 3  192.7  395.1
## 4  213.0  393.0
## 5  241.4  410.0
## 6  196.9  327.9
## 7  172.2  222.0
## 8  185.5  383.9
## 9  205.2  392.3
## 10 193.7  352.2
```

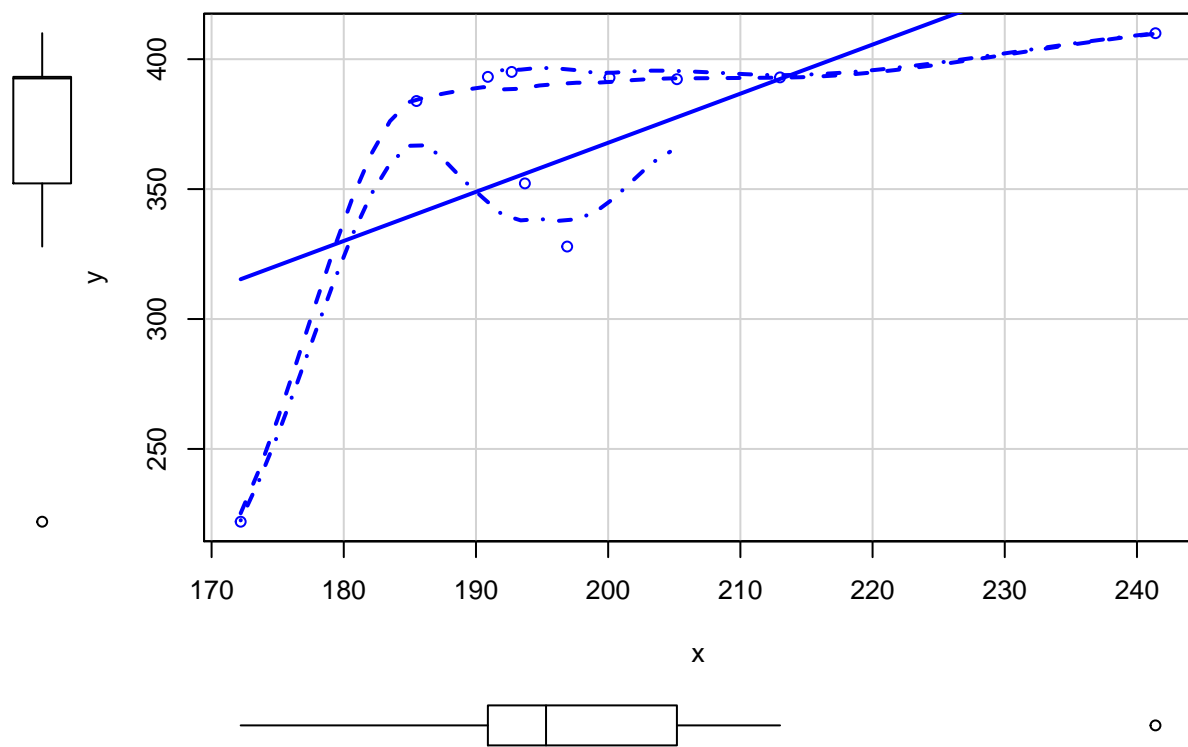
Visualize the data using scatter plot

```
library(ggplot2)
ggplot(my_data, aes(x=x, y=y)) +
  geom_point(aes(size=25)) +
  geom_smooth(method=lm, se=FALSE)

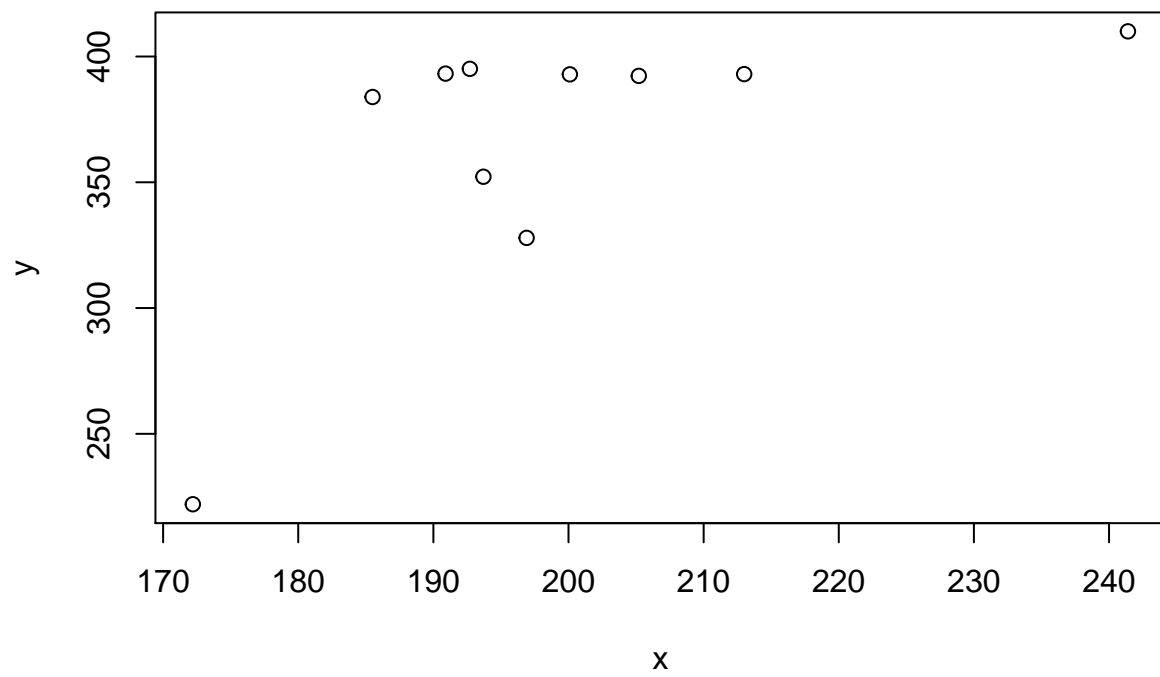
## 'geom_smooth()' using formula 'y ~ x'
```



```
library(car)  
scatterplot(x,y)
```

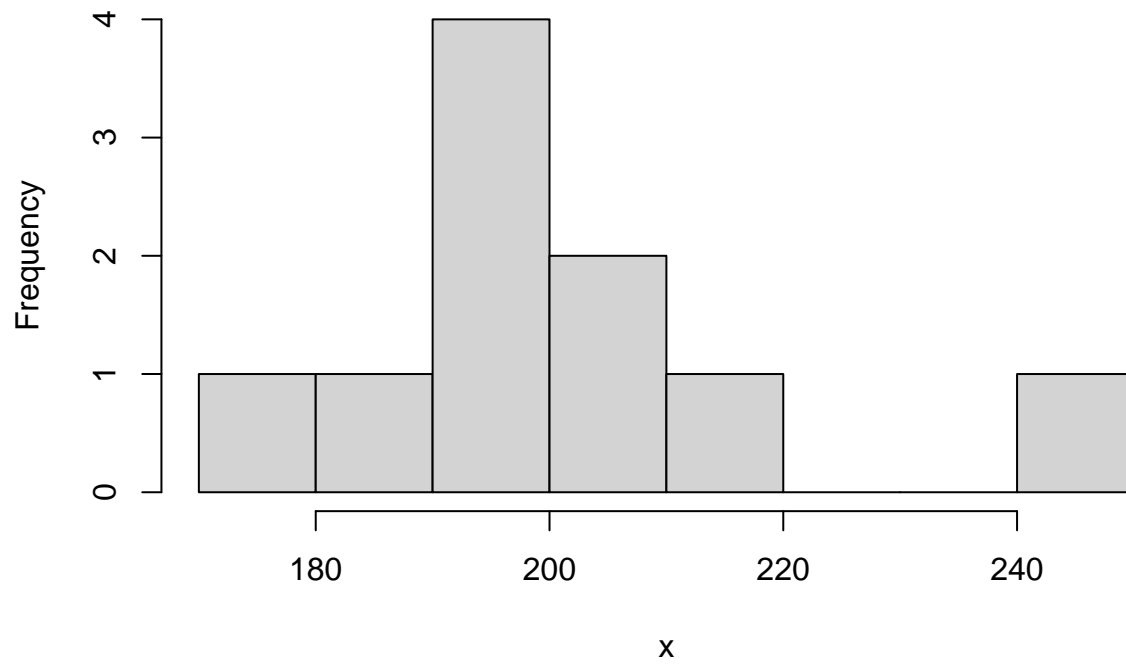


```
plot(x,y)
```

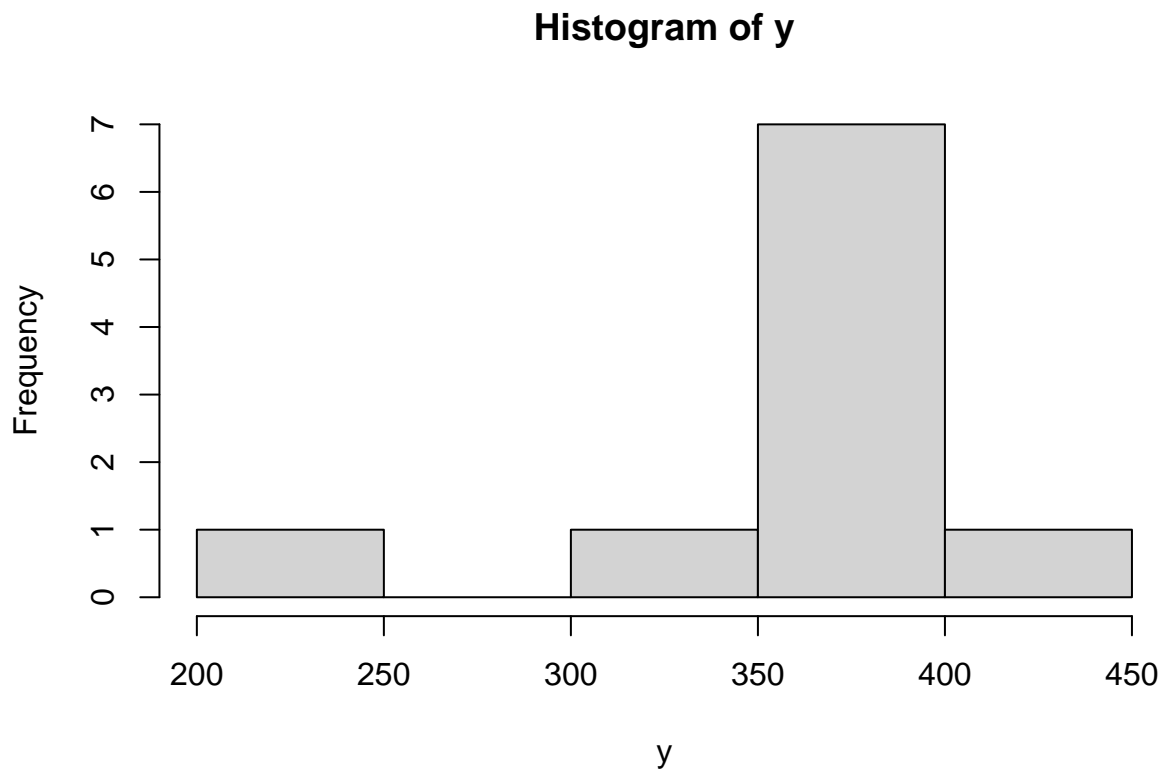


```
hist(x)
```

Histogram of x



```
hist(y)
```



Calculate the Spearman correlation

```
cor(x, y, method = c("spearman"))
```

```
## [1] 0.4666667
```

```
cor.test(x, y, method=c("spearman"))
```

```
##  
## Spearman's rank correlation rho  
##  
## data: x and y  
## S = 88, p-value = 0.1782  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho  
## 0.4666667
```

References

https://rcompanion.org/rcompanion/e_02.html <http://www.sthda.com> <http://www.statstutor.ac.uk/> https://www.sheffield.ac.uk/polopoly_fs/1.885209!/file/101_KruskalWallis.pdf <http://www.sthda.com/english/wiki/ggplot2-scatter-plots-quick-start-guide-r-software-and-data-visualization>