R-intro - Session 2 exercise solutions

João Gonçalves

19 de Julho de 2018

Contents

Missing data in R (NA values)

Exercise 1

1a)

```r
avg_temp <- c(10.5, NA, 12.3, 14.4, 15.6, 17.7, NA, 20.4, 20.3, 17.1, 13.3, 10.5)

# Calculate the sum of the logical vector to count the number of NA's
sum(is.na(avg_temp))
```

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

1b)

```r
# When NA values exist in a vector to calculate the mean or the std-dev the option
# na.rm must be set to TRUE

mean(avg_temp, na.rm= TRUE)
```

```
## [1] 15.21
```

```r
sd(avg_temp, na.rm= TRUE)
```

```
## [1] 3.650099
```

1c)

```r
# Don't forget to turn the option na.rm equal to TRUE to calculate the median
avg_temp[is.na(avg_temp)] <- median(avg_temp, na.rm = TRUE)

print(avg_temp)
```

```
##  [1] 10.5 15.0 12.3 14.4 15.6 17.7 15.0 20.4 20.3 17.1 13.3 10.5
```

1d)

```r
# Redefine the avg-temp again with NA values
avg_temp <- c(10.5, NA, 12.3, 14.4, 15.6, 17.7, NA, 20.4, 20.3, 17.1, 13.3, 10.5)


# Start by creating a vector for all vector indices
ind <- (1:length(avg_temp))

# Use vector ind to get the indices which are NA
ind_NA <- ind[is.na(avg_temp)]

# This is equal to using the which() function
ind_NA <- which(is.na(avg_temp))

# Replace values using vector positions
avg_temp[2] <- mean(avg_temp[c(1,3)])

avg_temp[7] <- mean(avg_temp[c(6,8)])


# More general solution using the ind_NA vector with NA indices
# Replace the first NA
avg_temp[ind_NA[1]] <- mean(avg_temp[c(ind_NA[1] - 1, ind_NA[1] + 1)])

# And the second NA
avg_temp[ind_NA[2]] <- mean(avg_temp[c(ind_NA[2] - 1, ind_NA[2] + 1)])

print(avg_temp)
```

```
##  [1] 10.50 11.40 12.30 14.40 15.60 17.70 19.05 20.40 20.30 17.10 13.30
## [12] 10.50
```

Dataframes in R

Quick exercise 1

QE1)

```r
precip <- data.frame(

  cities = c("Mobile", "Juneau", "Phoenix", "Little Rock", "Los Angeles",
             "Sacramento", "San Francisco", "Denver", "Hartford", "Wilmington"),

  precipitation = c( 67.0, 54.7, 7.0, 48.5, 14.0, 17.2, 20.7, 13.0, 43.4, 40.2),

  stringsAsFactors = FALSE
)

print(precip)
```

```
##           cities precipitation
## 1         Mobile          67.0
## 2         Juneau          54.7
## 3        Phoenix           7.0
## 4    Little Rock          48.5
## 5    Los Angeles          14.0
## 6     Sacramento          17.2
## 7  San Francisco          20.7
## 8         Denver          13.0
## 9       Hartford          43.4
## 10    Wilmington          40.2
```

```r
precip$cities
```

```
##  [1] "Mobile"        "Juneau"        "Phoenix"       "Little Rock"
##  [5] "Los Angeles"   "Sacramento"    "San Francisco" "Denver"
##  [9] "Hartford"      "Wilmington"
```

```r
class(precip$cities)
```

```
## [1] "character"
```

Exploratory data analysis - scatterplots and correlations

Exercise 2

2.

2a)

```r
# Number of rows with missing values
sum(!complete.cases(airquality))
```

```
## [1] 42
```

```r
# Clean dataframe using complete.cases() function
aq <- airquality[complete.cases(airquality), ]

# This is equal to using the na.omit() function to suppress rows with NA's
aq <- na.omit(airquality)
```

```r
# Check which rows (by position) were omitted when function na.omit was used
attr(aq,"na.action")
```

```
##   5   6  10  11  25  26  27  32  33  34  35  36  37  39  42  43  45  46
##   5   6  10  11  25  26  27  32  33  34  35  36  37  39  42  43  45  46
##  52  53  54  55  56  57  58  59  60  61  65  72  75  83  84  96  97  98
##  52  53  54  55  56  57  58  59  60  61  65  72  75  83  84  96  97  98
## 102 103 107 115 119 150
## 102 103 107 115 119 150
## attr(,"class")
## [1] "omit"
```

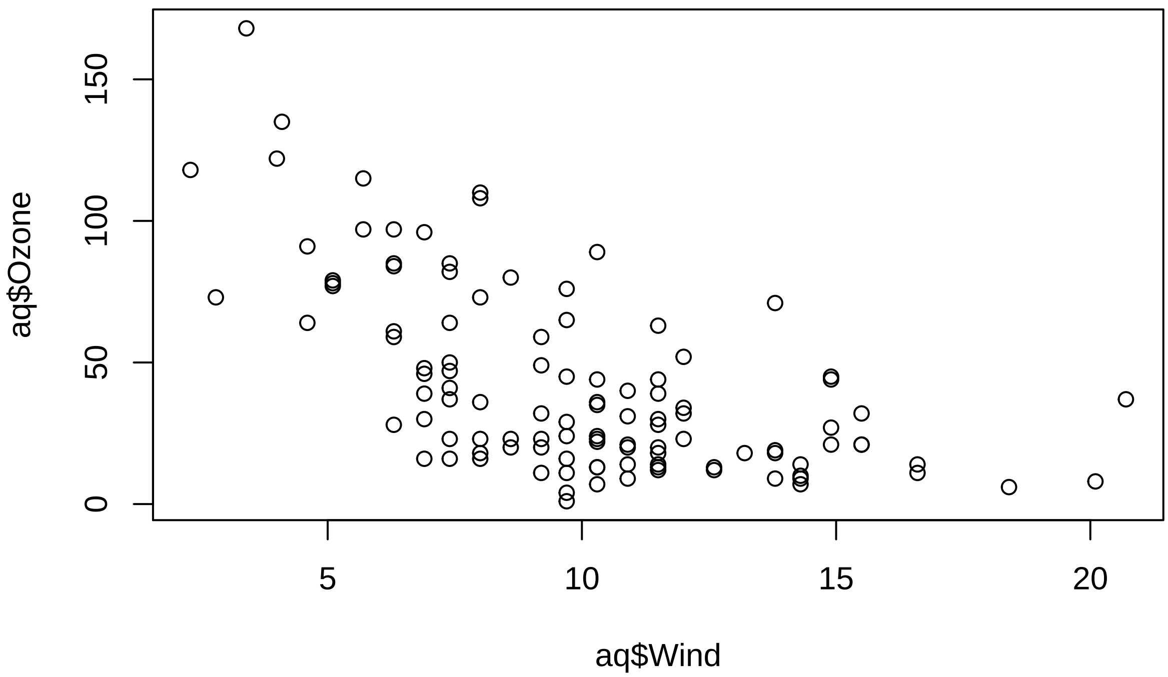
2b)

```r
summary(aq)
```

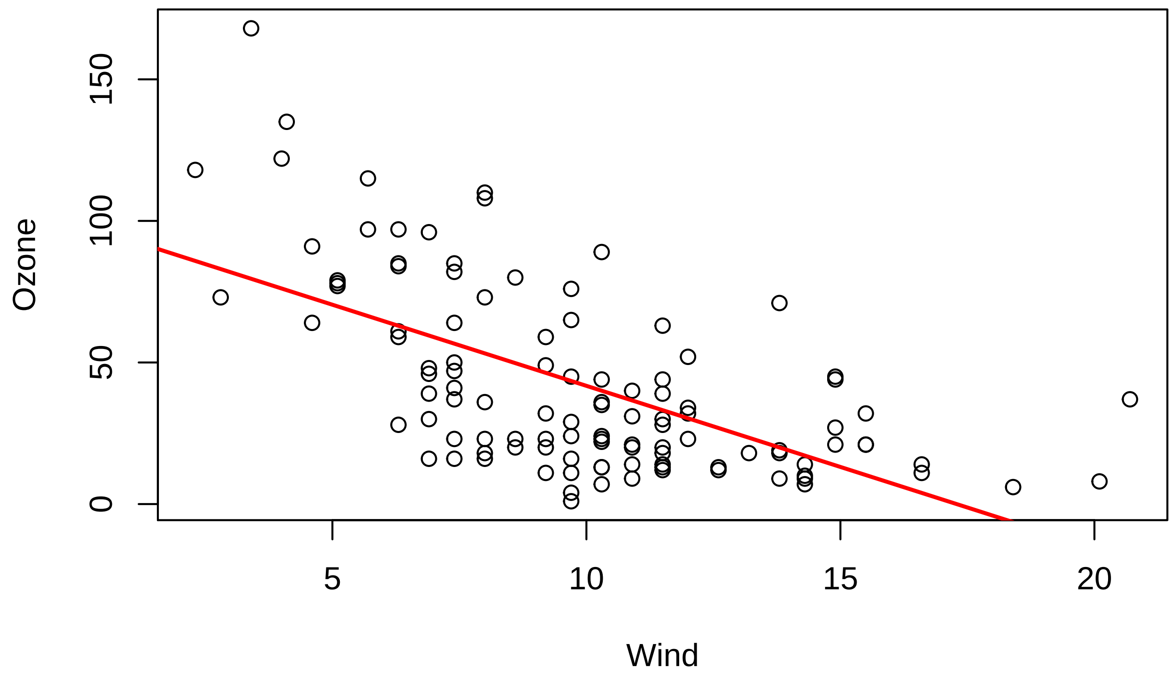
```
##      Ozone          Solar.R           Wind             Temp      
##  Min.   :  1.0   Min.   :  7.0   Min.   : 2.30   Min.   :57.00  
##  1st Qu.: 18.0   1st Qu.:113.5   1st Qu.: 7.40   1st Qu.:71.00  
##  Median : 31.0   Median :207.0   Median : 9.70   Median :79.00  
##  Mean   : 42.1   Mean   :184.8   Mean   : 9.94   Mean   :77.79  
##  3rd Qu.: 62.0   3rd Qu.:255.5   3rd Qu.:11.50   3rd Qu.:84.50  
##  Max.   :168.0   Max.   :334.0   Max.   :20.70   Max.   :97.00  
##      Month            Day       
##  Min.   :5.000   Min.   : 1.00  
##  1st Qu.:6.000   1st Qu.: 9.00  
##  Median :7.000   Median :16.00  
##  Mean   :7.216   Mean   :15.95  
##  3rd Qu.:9.000   3rd Qu.:22.50  
##  Max.   :9.000   Max.   :31.00
```

2c)

```r
# Use plot and specifically declare the x and y vector
plot(x = aq$Wind, y = aq$Ozone)
```




```r
# Or you can use the formula interface to get a similar effect
plot(Ozone ~ Wind, data = aq)

# EXTRA: add a regression line to the plot based on the abline() function
# Notice that a linear model (lm) is used as an input in abline
abline(lm(Ozone ~ Wind, data = aq), col="red", lwd=2)
```

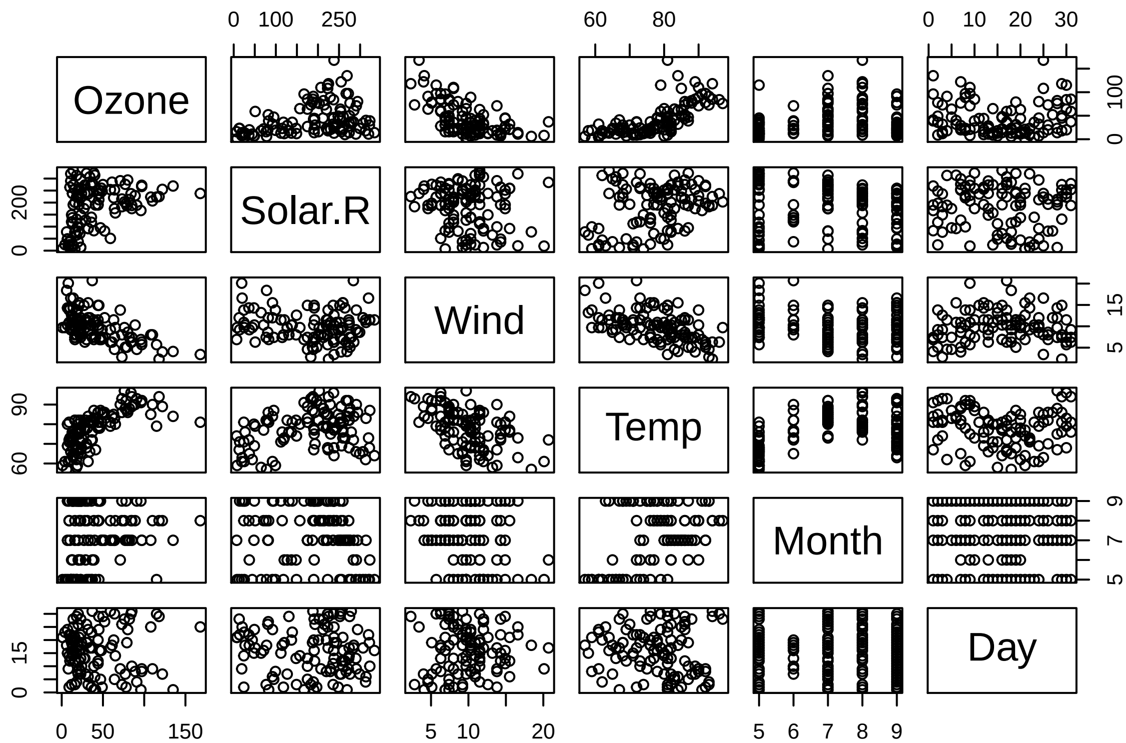



```r
# The correct option is...
print("iv")
```

```
## [1] "iv"
```

2d)

```r
plot(aq)
```




```r
# By visual inspection it seems that Temp (temperature), Wind (wind speed) and
# Solar.R (solar radiation) are more clearly correlated with Ozone concentration
```

2e)

```r
# (i)
cor.test(aq$Ozone, aq$Solar.R, method="spearman")
```

```
## Warning in cor.test.default(aq$Ozone, aq$Solar.R, method = "spearman"):
## Cannot compute exact p-value with ties

##
##  Spearman's rank correlation rho
##
## data:  aq$Ozone and aq$Solar.R
## S = 148560, p-value = 0.0001806
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.3481865
```

```r
## Rho = 0.3481865 p < 0.001

# (ii)
cor.test(aq$Ozone, aq$Wind, method="spearman")
```

```
## Warning in cor.test.default(aq$Ozone, aq$Wind, method = "spearman"): Cannot
## compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  aq$Ozone and aq$Wind
## S = 365840, p-value = 1.998e-12
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.6051364
```

```
## Rho = -0.6051364 p < 0.001

# (iii)
cor.test(aq$Ozone, aq$Temp, method="spearman")
```

```
## Warning in cor.test.default(aq$Ozone, aq$Temp, method = "spearman"): Cannot
## compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  aq$Ozone and aq$Temp
## S = 51753, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.7729319
```

```
## Rho = 0.7729319 p < 0.001
```

Dataframe indexation with [,]

Exercise 3

3.

3a)

```
aq[,-c(3:ncol(aq))]
```

3b)

```
aq[,c("Day","Month")]
```

3c)

```
aq[1:3, ]
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
```

3d)

```
nr <- nrow(aq)
aq[(nr-10):nr, ]
```

```
##     Ozone Solar.R Wind Temp Month Day
## 142    24     238 10.3   68     9  19
## 143    16     201  8.0   82     9  20
## 144    13     238 12.6   64     9  21
## 145    23      14  9.2   71     9  22
## 146    36     139 10.3   81     9  23
## 147     7      49 10.3   69     9  24
## 148    14      20 16.6   63     9  25
## 149    30     193  6.9   70     9  26
## 151    14     191 14.3   75     9  28
## 152    18     131  8.0   76     9  29
## 153    20     223 11.5   68     9  30
```

```r
# Or more simply
tail(aq, n = 10)
```

```
##     Ozone Solar.R Wind Temp Month Day
## 143    16     201  8.0   82     9  20
## 144    13     238 12.6   64     9  21
## 145    23      14  9.2   71     9  22
## 146    36     139 10.3   81     9  23
## 147     7      49 10.3   69     9  24
## 148    14      20 16.6   63     9  25
## 149    30     193  6.9   70     9  26
## 151    14     191 14.3   75     9  28
## 152    18     131  8.0   76     9  29
## 153    20     223 11.5   68     9  30
```

3e)

```r
aq[aq$Ozone > mean(aq$Ozone), ]
```

```
##     Ozone Solar.R Wind Temp Month Day
## 29     45     252 14.9   81     5  29
## 30    115     223  5.7   79     5  30
## 40     71     291 13.8   90     6   9
## 62    135     269  4.1   84     7   1
## 63     49     248  9.2   85     7   2
## 66     64     175  4.6   83     7   5
## 68     77     276  5.1   88     7   7
## 69     97     267  6.3   92     7   8
## 70     97     272  5.7   92     7   9
## 71     85     175  7.4   89     7  10
## 77     48     260  6.9   81     7  16
## 79     61     285  6.3   84     7  18
## 80     79     187  5.1   87     7  19
## 81     63     220 11.5   85     7  20
## 85     80     294  8.6   86     7  24
## 86    108     223  8.0   85     7  25
## 88     52      82 12.0   86     7  27
## 89     82     213  7.4   88     7  28
## 90     50     275  7.4   86     7  29
## 91     64     253  7.4   83     7  30
## 92     59     254  9.2   81     7  31
## 99    122     255  4.0   89     8   7
## 100    89     229 10.3   90     8   8
```

```
## 101   110      207  8.0   90     8   9
## 104    44      192 11.5   86     8  12
## 106    65      157  9.7   80     8  14
## 109    59       51  6.3   79     8  17
## 112    44      190 10.3   78     8  20
## 116    45      212  9.7   79     8  24
## 117   168      238  3.4   81     8  25
## 118    73      215  8.0   86     8  26
## 120    76      203  9.7   97     8  28
## 121   118      225  2.3   94     8  29
## 122    84      237  6.3   96     8  30
## 123    85      188  6.3   94     8  31
## 124    96      167  6.9   91     9   1
## 125    78      197  5.1   92     9   2
## 126    73      183  2.8   93     9   3
## 127    91      189  4.6   93     9   4
## 128    47       95  7.4   87     9   5
## 134    44      236 14.9   81     9  11
## 139    46      237  6.9   78     9  16
```

3f)

```r
aq[aq$Ozone < quantile(aq$Ozone, probs=0.1), ]
```

```
##     Ozone Solar.R Wind Temp Month Day
## 9       8      19 20.1   61     5   9
## 18      6      78 18.4   57     5  18
## 21      1       8  9.7   59     5  21
## 23      4      25  9.7   61     5  23
## 73     10     264 14.3   73     7  12
## 76      7      48 14.3   80     7  15
## 94      9      24 13.8   81     8   2
## 114     9      36 14.3   72     8  22
## 137     9      24 10.9   71     9  14
## 147     7      49 10.3   69     9  24
```

Slicing with subset()

Quick exercise 2

QE2)

```r
# Combine each condition using the & (AND) operator

subset(aq,
       Ozone > 80 &
       Temp > 90 &
       Wind < 5)
```

```
##     Ozone Solar.R Wind Temp Month Day
## 121   118     225  2.3   94     8  29
## 127    91     189  4.6   93     9   4
```

Factor variables

Exercise 4

4.

4a)

```r
# Use the line below to call the data
data(iris)
head(iris) # This is used to make a first check of the data
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```r
# Check object class
class(iris$Species)
```

```
## [1] "factor"
```

```r
# Check if it is a factor variable
is.factor(iris$Species)
```

```
## [1] TRUE
```

4b)

```r
levels(iris$Species)
```

```
## [1] "setosa"     "versicolor" "virginica"
```

4c)

```r
table(iris$Species)
```

```
## 
##     setosa versicolor  virginica 
##         50         50         50
```

4d)

```r
# The function by() can be used for calculating a function for each level in INDICES
by(iris$Petal.Length, INDICES = iris$Species, FUN = mean)
```

```
## iris$Species: setosa
## [1] 1.462
## ------------------------------------------------------------
## iris$Species: versicolor
## [1] 4.26
## ------------------------------------------------------------
## iris$Species: virginica
## [1] 5.552
```

```r
# Or using indexation the mean can be calculated this way for each species
mean(iris$Petal.Length[iris$Species == "setosa"])
```

```
## [1] 1.462
```

```r
mean(iris$Petal.Length[iris$Species == "versicolor"])
```

```
## [1] 4.26
```

```r
mean(iris$Petal.Length[iris$Species == "virginica"])
```

```
## [1] 5.552
```

4e)

```r
# The function by() can be used for calculating a function for each level in INDICES
by(iris$Petal.Length, INDICES = iris$Species, FUN = sd)
```

```
## iris$Species: setosa
## [1] 0.173664
## --------------------------------------------------------
## iris$Species: versicolor
## [1] 0.469911
## --------------------------------------------------------
## iris$Species: virginica
## [1] 0.5518947
```

```r
# Or using indexation the std-dev can be calculated this way for each species
sd(iris$Petal.Length[iris$Species == "setosa"])
```

```
## [1] 0.173664
```

```r
sd(iris$Petal.Length[iris$Species == "versicolor"])
```

```
## [1] 0.469911
```

```r
sd(iris$Petal.Length[iris$Species == "virginica"])
```

```
## [1] 0.5518947
```

Analyzing distributions

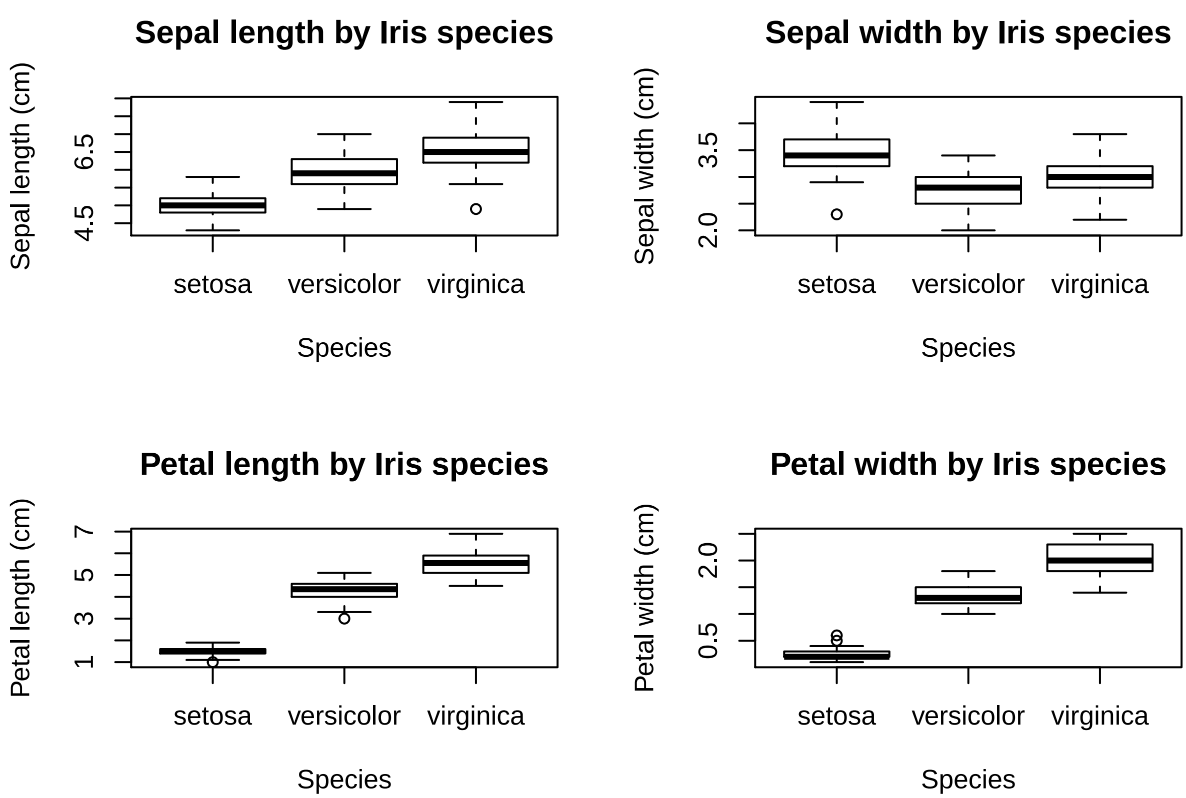
Exercise 5

5.

5a)

```r
# This command will allow to make a 2 x 2 plot i.e. with the 4 boxplots
par(mfrow=c(2,2))


# Now let's make the boxplots
# In boxplot it's possible to modify the title (main),
# the y-axis label (ylab) and x-axis label (xlab)
```

```r
boxplot(Sepal.Length ~ Species, data = iris, main = "Sepal length by Iris species",
        ylab = "Sepal length (cm)", xlab = "Species")

boxplot(Sepal.Width ~ Species, data = iris, main = "Sepal width by Iris species",
        ylab = "Sepal width (cm)", xlab = "Species")

boxplot(Petal.Length ~ Species, data = iris, main = "Petal length by Iris species",
        ylab = "Petal length (cm)", xlab = "Species")

boxplot(Petal.Width ~ Species, data = iris, main = "Petal width by Iris species",
        ylab = "Petal width (cm)", xlab = "Species")
```



5b)

i) Overall, petal dimensions seems to provide better differentiation between species with lower overlap between boxes/distributions.

ii) Both petal length and width provide very good differentiation.

iii) Sepal width seems to perform worst due to the higher overlap between boxes/distributions.

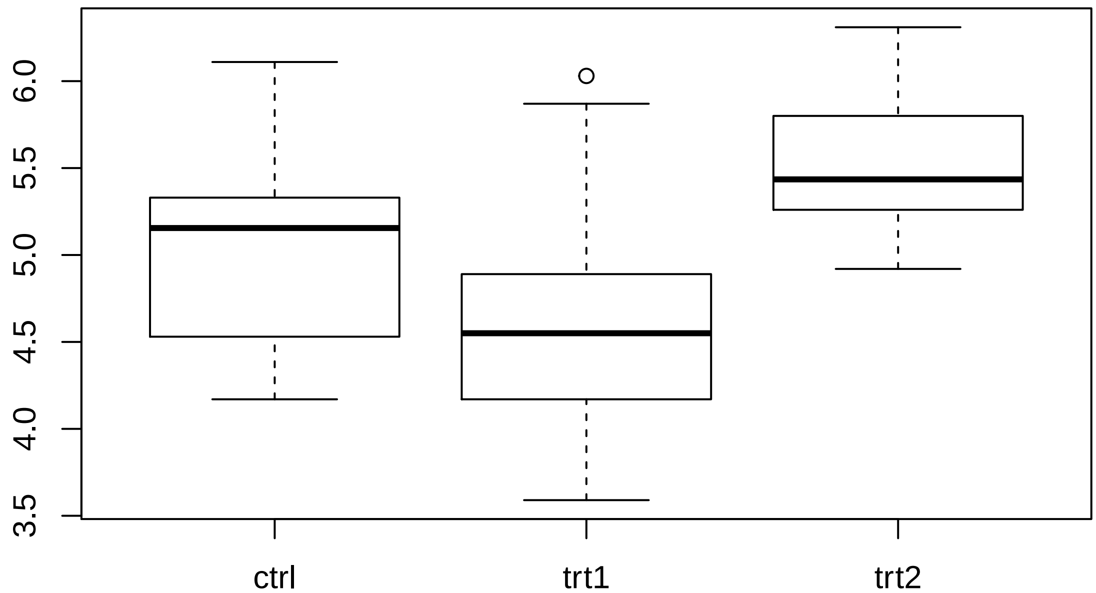
Hypothesis testing

Exercise 6

6.

6a)

```
boxplot(weight ~ group, data = PlantGrowth)
```



6b)

```
# Make a variable with weight measurements for the control group
weight_control <- PlantGrowth$weight[PlantGrowth$group == 'ctrl']

# Make a variable with weight measurements for the treatment-1 group
weight_trt1 <- PlantGrowth$weight[PlantGrowth$group == 'trt1']

# Now the t.test
t.test(x = weight_control, y = weight_trt1)
```

```
##
##  Welch Two Sample t-test
##
## data:  weight_control and weight_trt1
## t = 1.1913, df = 16.524, p-value = 0.2504
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.2875162  1.0295162
## sample estimates:
## mean of x mean of y
##     5.032     4.661
```

According to the t-Test there is not sufficient evidence to reject the null-hypothesis, H0: Avg(control) =

Avg(trt1), given that the p-value > 0.05

6c)

```r
# Make a variable with weight measurements for the control group
weight_control <- PlantGrowth$weight[PlantGrowth$group == 'ctrl']

# Make a variable with weight measurements for the treatment-2 group
weight_trt2 <- PlantGrowth$weight[PlantGrowth$group == 'trt2']

# Now the t.test
t.test(x = weight_control, y = weight_trt2)
```

```
##
##  Welch Two Sample t-test
##
## data:  weight_control and weight_trt2
## t = -2.134, df = 16.786, p-value = 0.0479
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.98287213 -0.00512787
## sample estimates:
## mean of x mean of y
##     5.032     5.526
```

According to the t-Test there is sufficient evidence to reject the null-hypothesis, H0: Avg(control) = Avg(trt2), given that the p-value < 0.05

Analysis of variance (one-way)

Exercise 7

7.

7a)

```r
aov_iris <- aov(Sepal.Width ~ Species, data = iris)

summary(aov_iris)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Species       2  11.35   5.672   49.16 <2e-16 ***
## Residuals   147  16.96   0.115
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

According to the ANOVA test there is sufficient evidence to reject the null-hypothesis, H0: Avg(setosa) = Avg(versicolor) = Avg(virginica), given that the p-value < 0.05

7b)

```r
aov_iris <- aov(Petal.Width ~ Species, data = iris)

summary(aov_iris)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Species       2  80.41   40.21     960 <2e-16 ***
## Residuals   147   6.16    0.04
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

According to the ANOVA test there is sufficient evidence to reject the null-hypothesis, H0: Avg(setosa) = Avg(versicolor) = Avg(virginica), given that the p-value < 0.05

Exercise 8

8.

8a)

```
# Read the csv data file
# NOTE: depending on the working directory this path may not work and in that case you have to adjust i
diet <- read.csv("../Session-02/diet.csv")
```

8b)

```
# Use na.omit() to quickly exclude rows with NA values
diet <- na.omit(diet)
```

8c)

```
# To add a new column to the dataset we can use either the function data.frame() or cbind()
diet <- data.frame(diet, Treatment = factor(diet$Diet, levels = c(1,2,3),
                                             labels = c("Trt-A", "Trt-B", "Trt-C")))

# Solution with cbind()
# diet <- cbind(diet, Treatment = factor(diet$Diet, levels = c(1,2,3), labels = c("Trt-A", "Trt-B", "Tr
```

8d)

```
diet <- data.frame(diet, Weight.Loss = diet$pre.weight - diet$weight6weeks)

#print(diet)
```

8e)

```
aov_diet <- aov(Weight.Loss ~ Treatment, data = diet)

summary(aov_diet)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## Treatment    2   60.5  30.264   5.383 0.0066 **
## Residuals   73  410.4   5.622
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

According to the ANOVA test there is sufficient evidence to reject the null-hypothesis, H0: Avg(Trt-A) = Avg(Trt-B) = Avg(Trt-C), given that the p-value < 0.05