Statistical Inference - Inferential Data Analysis

Silva, RAFAEL

December 4, 2017

Setup

This section presents all the libraries used during the project.

```
library(datasets)
library(ggplot2)
library(dplyr)
library(tidyr)
library(knitr)
```

Introduction

The objective of this project is to make some inferences about the length variation on each supplement and dose on the ToothGrowth data from the R datasets package. The análysis is going to follow the four steps below:

1. Exploratory data analysis and summary;
2. Hypothesis testing;
3. Conclusions and assumptions.

The chunk bellow is used to read the data set into the workspace.

```
data("ToothGrowth")
```

For more information on the data set (description, dimensions, variables and source) enter ?ToothGrowth on the console to check its documentation.

Exploratory analysis

To start the analysis, a brief verification on how the dataset looks like is necessary.

```
dim(ToothGrowth)
```

```
## [1] 60  3
```

```
head(ToothGrowth)
```

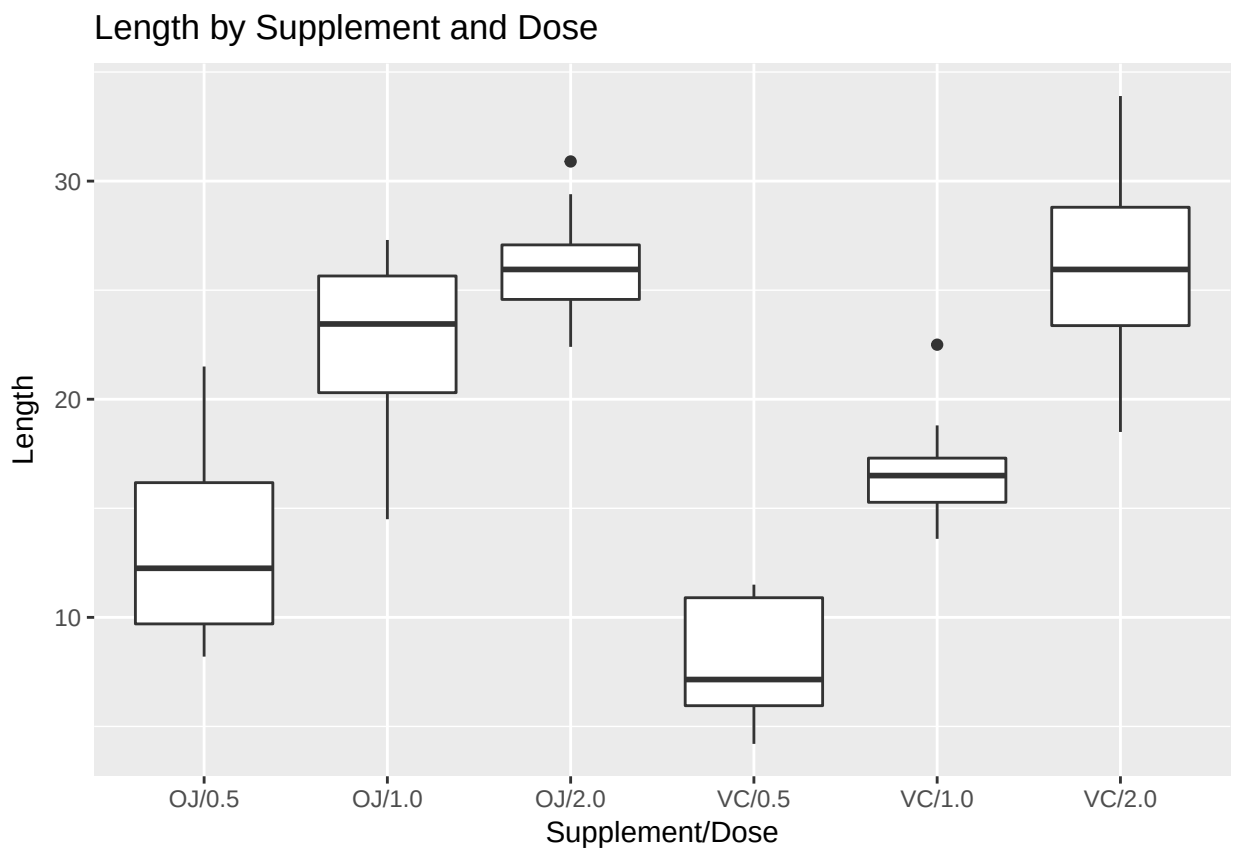
```
##    len supp dose
## 1  4.2   VC  0.5
## 2 11.5   VC  0.5
## 3  7.3   VC  0.5
## 4  5.8   VC  0.5
## 5  6.4   VC  0.5
## 6 10.0   VC  0.5
```

As the outputs above show, the data set has 60 observations on 3 variables. As the instroduction chapter states, the objective is to make some inferences about the length variation on each supplement and dose. The boxplot below gives a feeling about how each combination affects the length.

```
ToothGrowth <- ToothGrowth %>%
      mutate(suppDose = paste(supp, format(dose, digits = 2), sep = "/"))

g1 <- ggplot(ToothGrowth, aes(x = suppDose, y = len)) +
      geom_boxplot() +
      labs(title = "Length by Supplement and Dose",
           x = "Supplement/Dose", y = "Length")

print(g1)
```



It is important to notice that, in the graph above, the x axis does not present a continuous scale. This means that when analysing the graph you should not look at how the length increase "evolve" through the doses but at how each group behave in the y scale.

In the table below, the main statistical attributes of each group are summarized.

```
summ <- ToothGrowth %>%
      group_by(suppDose) %>%
      summarise(min = min(len),
                q1 = quantile(len, .25),
                mean = mean(len),
                median = median(len),
                q3 = quantile(len, .75),
                max = max(len),
                variance = round(var(len),3),
                n = length(len))
```

```r
kable(summ, caption = "Data Summary by Supplement and Dose",
      col.names = c("Supp/Dose", "Min", "1st Qu.", "Mean", "Median", "3rd Qu.",
                    "Max", "Variance", "n"))
```

Table 1: Data Summary by Supplement and Dose

Supp/Dose	Min	1st Qu.	Mean	Median	3rd Qu.	Max	Variance	n
OJ/0.5	8.2	9.700	13.23	12.25	16.175	21.5	19.889	10
OJ/1.0	14.5	20.300	22.70	23.45	25.650	27.3	15.296	10
OJ/2.0	22.4	24.575	26.06	25.95	27.075	30.9	7.049	10
VC/0.5	4.2	5.950	7.98	7.15	10.900	11.5	7.544	10
VC/1.0	13.6	15.275	16.77	16.50	17.300	22.5	6.327	10
VC/2.0	18.5	23.375	26.14	25.95	28.800	33.9	23.018	10

With the summary presented above, it is possible to manualy calculate the confidence interval for all the groups, besides that, it is also possible to perform an equivalence test on each combination. However, R provides a usefull tool to test the equivalence of each group combination, this is what will be covered in the next section.

Hypothesis test

In this section, each group combination will go through a t.test. The aim is to test for H*subscript*0*subscript*: *musubscript*X*subscript* = *musubscript*Y*subscript*. The code below performs a two sided test on the differences of each group combination, the 95% confidence interval of each test is extracted and organized in a table for further analysis.

```r
suppDose <- unique(ToothGrowth$suppDose)
testTable <- data.frame(suppDose1 = rep(suppDose, each = length(suppDose)),
                        suppDose2 = rep(suppDose, times = length(suppDose)))
testTable$confInt <- NA

for(i in 1:nrow(testTable)) {
      x = ToothGrowth$len[ToothGrowth$suppDose == testTable$suppDose1[i]]
      y = ToothGrowth$len[ToothGrowth$suppDose == testTable$suppDose2[i]]
      confInt <- round(t.test(x, y, var.equal = TRUE)$conf, 2)
      testTable$confInt[i] <- paste(confInt[1], confInt[2], sep = "  _  ")
}

testTable <- testTable %>%
      spread(suppDose2, confInt)

row.names(testTable) <- testTable$suppDose1

testTable <- testTable[, 2:7]
```