R Notebook about statistic in the work of data analyst

The key idea of this notebook is to describe key statistical methods and practices, which are needed for the work of a data analyst. There is always no limit to studying, so this material can be updated.

Preparation work

First of all, we need to install the needed packages and prepare our environment (load the libraries).

```
#key package for data analysis
install.packages('tidyverse', repos = "http://cran.us.r-project.org")
library(tidyverse)
library(lubridate) #for the date format

#package for creating and showing in pdf graph diagrams
install.packages('DiagrammeR', repos = "http://cran.us.r-project.org")
library(DiagrammeR)
install.packages('webshot', repos = "http://cran.us.r-project.org")
webshot::install_phantomjs()

#package for statistical tests
install.packages('pwr', repos = "http://cran.us.r-project.org")
library(pwr)

#package for creating pivot-tables
install.packages('pivottabler', repos = "http://cran.us.r-project.org")
library(pivottabler)

#package for providing the summary statistics of df
install.packages('skimr', repos = "http://cran.us.r-project.org")
library(skimr)

#package for bootstraping
install.packages('boot',repos = "http://cran.us.r-project.org")
library(boot)

#package for some statistical (Levene's test) analysis
install.packages('car', repos = "http://cran.us.r-project.org")
library(car)
```

Why do I need to know statistics?

Definitely, why should I use methods to prove the difference or declare the correlation? Can I only compare figures mathematically to make my decision? The answer lays in the essence of sampling. To analyze the statistical population is always the most precise way to draw conclusions. But it's almost always very expensive, time-consuming, or even impossible. That's why analysts use samples, it's cheaper and easier.

But the sample isn't the whole population and there can be a difference compared to the "real" figures. Evaluating the difference between "real" and "sampled" data is very important, cause only in this case we can tell with some high probability (normally 95% or 99%), what is in our data and which conclusions we can make. This difference between "real" and "sampled" data is called as **sampling error**.

Confidence interval and confidence level

When we want to evaluate an indicator, we take its average and calculate **standard error** ($SE = \frac{\sigma}{\sqrt{n}}$)and **margin error** ($\Delta = t * SE$ for Student's test). Then we can calculate our **confidence interval** (a range of possible values for our parameter) with the chosen **confidence level** (normally 95%). So, for example, the average meaning in our sample is 5, margin error is 1. It means, that our "real" average is between 4 (=5-1) and 6 (=5+1) with the chosen confidence level = 95%, and there is only 5%, that the "real" average lays outside of this interval.

Let's try to calculate the confidence interval in R. You can do it manually, using the formulas above. Or you can go this way.

```
# For this exercise we can use the pre-installed data set - mtcars.
# Let's calculate the confidence interval for the Displacement (disp)
disp.line.regression <- lm(disp ~ 1, data = mtcars)
confint(disp.line.regression, level=0.95)
```

```
##                2.5 %   97.5 %
## (Intercept) 186.0372 275.4065
```

Intercept in this case means the confidence interval for the data. So we can tell, that there is 95% probability, that our "real" average lays in the interval from 186.0 till 275.4.

Null hypothesis

Null hypothesis (H0) is one more important term to explain our topic further. If we want to prove something (correlation or significant difference), we do it normally denying the opposite case. And this opposite case is our null hypothesis. For example, we have 2 groups of people and some indicator for each person. Our main hypothesis is, that there is no significant difference in this indicator between these 2 groups. Our null hypothesis is, that there is no significant difference between these groups. The null hypothesis is the unlucky one because we always want to decline it.

Sampling errors

There are two types of sampling errors: Type I and Type II errors. They can be easily introduced in the table below. **Horizontally** we are telling, if there is a difference between two parameters **statistically**. And **vertically** we are telling, if there is a difference between two parameters **really**.

	There **IS** diff (stat)	There is **NO** diff (stat)
There **IS** diff (real)	True Positive	False Negative (β)
There is **NO** diff (real)	False Positive (α)	True Negative

Alpha is a rest from this equation: 100% - Confidence level. Normally it equals 5%. It means, that there is a 5% probability, that we are wrong to decline the null hypothesis. Or in other words, there is only a 5% probability, that we detected a significant difference, where there was no actual difference.

Beta tells us about the opposite situation. It's the probability, that we stick to the null hypothesis, although there is an actual significant difference. Beta is directly connected to such term as a power of the test, it's calculated as 100% - Beta. Normally it's good if the power equals to or greater than 80%. Unfortunately, it's unreal to reduce type 1 and type 2 errors at the same time, because there is a negative correlation between them.

How to calculate the sample size

Let's imagine the situation. We want to conduct AB-Testing and we want to catch a small difference between the 2 groups. It seems obvious, that the smaller the difference we want to be able to catch, the bigger should be the sample. This ability to catch our difference is the power of the test. So to calculate the sample size, we can fix alpha as 0.05 and power as 0.8. Back to our example. Let's imagine, that we want to track some indicator, which normally has an average of 100 and has a standard deviation of 10. We hope, that our changes can improve our indicator to 101 (not too much, but maybe it's enough). So we can have an 80% probability, that we can catch this (1 Punkt) difference, keeping in mind, that we can catch it wrongly in 5%. Below is the code, how we can calculate the sample size.

```
mean_control = 100
sd_control = 10
mean_test = 101

#calculating the effect to detect
effect = (mean_test - mean_control) / sd_control

#calculating the sample
pwr.t.test(d=effect, sig.level=0.05, power=0.80, type="two.sample",
           alternative="greater")
```

```
##
##      Two-sample t test power calculation
##
##               n = 1237.188
##               d = 0.1
##       sig.level = 0.05
##           power = 0.8
##     alternative = greater
##
## NOTE: n is number in *each* group
```

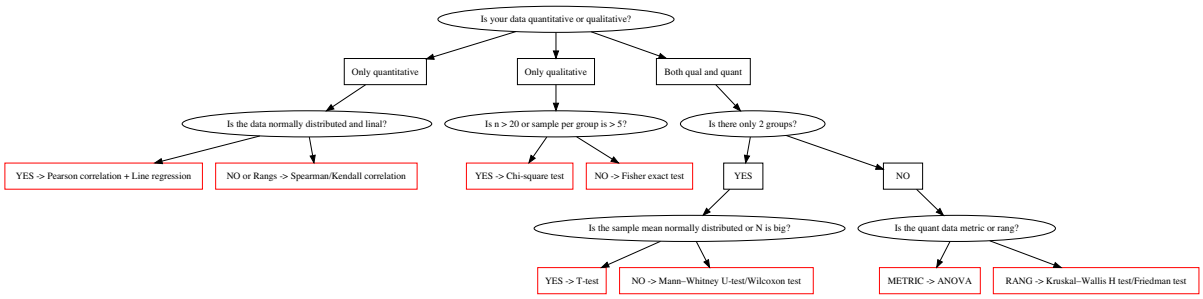
This result means, that we need to have at least 1238 users in each group.

Sometimes we don't know anything about our users and want to know something about them. In this case we can use the rule of thumb - to have at least 400 users in our sample. This figure is calculated based on this formula: $n = \frac{p*q*t^2}{\Delta^2}$, where $\Delta = 0.05$, p=0.5 (probability of our factor), q=1-p, t - Student's criteria (ca. 2).

Ok, now we are ready to move to statistical methods

Key schema of statistical methods

Here is the basic algorithm, how to choose the statistical method. It's not so strict and sometimes you have to choose another option - e.g.bootstrap or another.

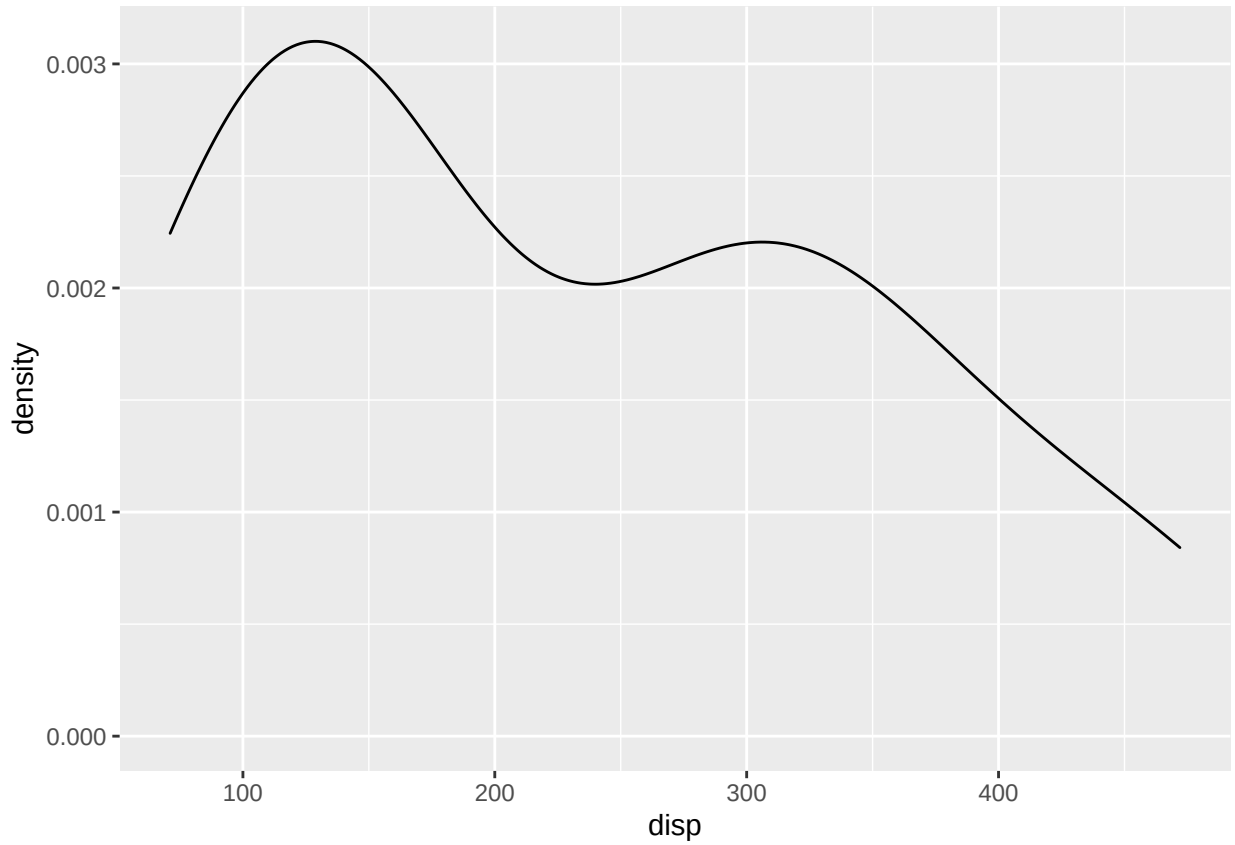


How to test normality?

Some statistical tests are very sensitive and it's needed to have normally distributed data to use them. Sometimes it's enough though, that data is close to the normal distribution. The first method is visualizing the data with the density plot. It's always a good idea to visualize your data to see the tendency and evaluate tails and outliers.

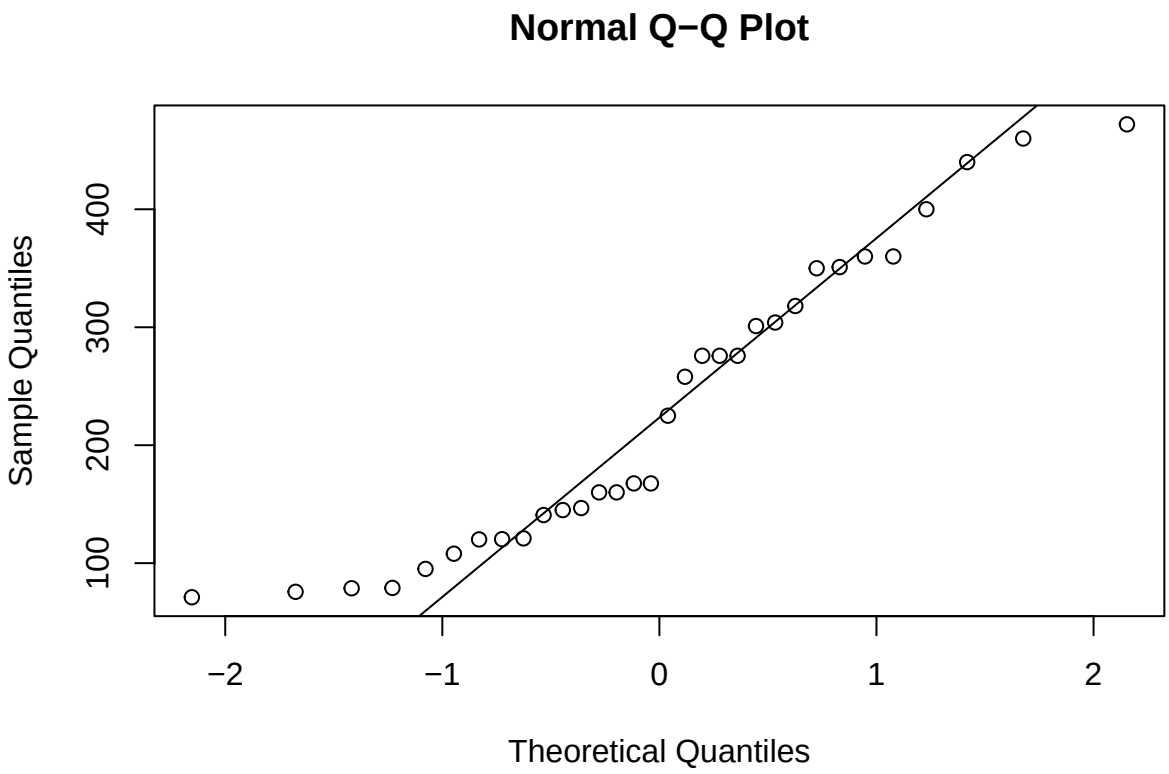
Let's evaluate our displacement (disp) data from the mtcars data set. To do it let's use ggplot.

```
ggplot(data=mtcars) +
  geom_density(mapping=aes(x=disp), kernel="gaussian")
```



It's obvious, that this data isn't normally distributed. Let's inspect, what the other tests can tell us. One more graphical method is qq-plot. It shows the difference to the normal distribution (line). If our distribution is normal, all of the dots are laying at the line.

```
qqnorm(mtcars$disp)
qqline(mtcars$disp)
```



And once again, our graphic shows us, that this distribution isn't normal. Now we can use the Shapiro-Wilk test to evaluate the normality

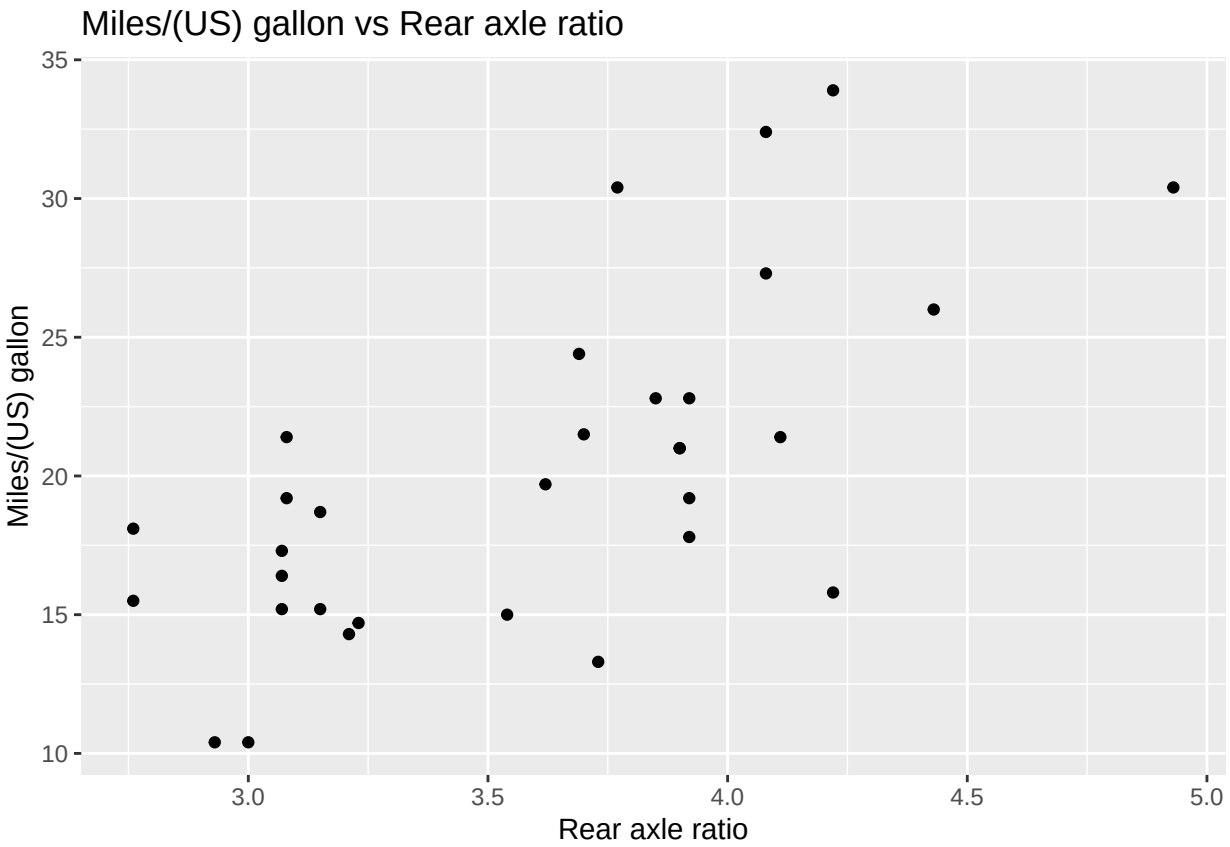
```
shapiro.test(mtcars$disp)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mtcars$disp
## W = 0.92001, p-value = 0.02081
```

We have a p-value < 0.05 and in this case, it's bad. Because our H0 is that there is NO difference between a normal distribution and our distribution. It means, that we can decline H0 and there IS this difference, so our distribution isn't normal. I recommend using not only the Shapiro-Wilk test but also visualizing data with the density plot. Now, let's move to statistical methods.

Pearson correlation and line regression

We can use Pearson correlation if our data is quantitative and normally distributed. Please keep in mind, that outliers can significantly change your results, that's why it's important to create a graphic of your data before the analysis. Let's analyze mtcars dataset and find the correlation between miles/gallon and rear axle ratio. First of all, draw the scatterplot.



As we can see from the chart, there is some correlation between two parameters and this correlation could be linear. Now let's test the normality.

```
shapiro.test(mtcars$drat)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mtcars$drat
## W = 0.94588, p-value = 0.1101
```

```
shapiro.test(mtcars$mpg)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mtcars$mpg
## W = 0.94756, p-value = 0.1229
```

Both of our parameters are distributed normally, so we can use Pearson correlation.

```
cor(mtcars$drat, mtcars$mpg, method="pearson")
```

```
## [1] 0.6811719
```

The result equals 0.68. It means, that there is a correlation, but it's quite moderate. Let's create a linear model with these 2 factors. Please note, that if the connection between factors isn't linear, but can be somehow transformed (using logarithm, etc) to it. We can steel use these methods, but it can be tricky to understand the idea behind such a connection.

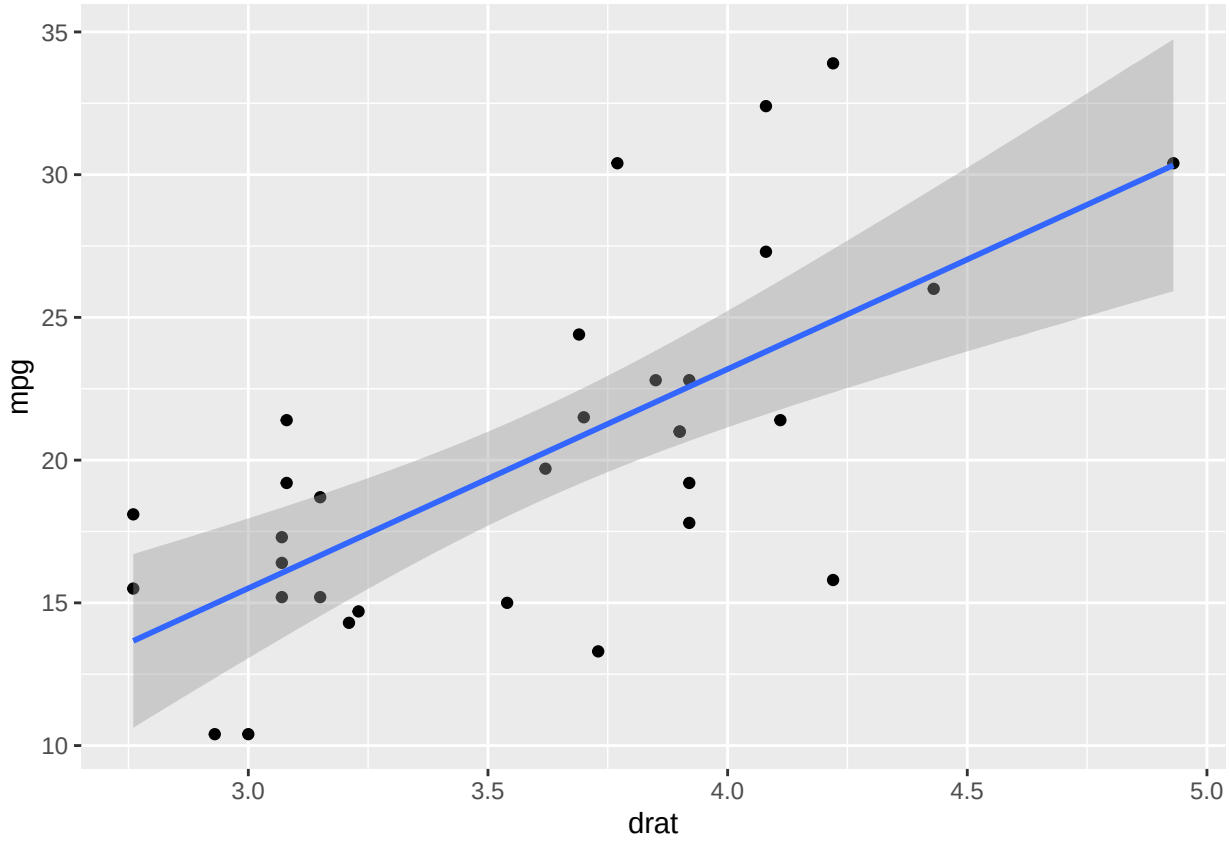
```
lr = lm(mpg~drat, data=mtcars)
summary(lr)
```

```
##
## Call:
## lm(formula = mpg ~ drat, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0775 -2.6803 -0.2095  2.2976  9.0225
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -7.525      5.477  -1.374     0.18
## drat           7.678      1.507   5.096 1.78e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.485 on 30 degrees of freedom
## Multiple R-squared:  0.464,  Adjusted R-squared:  0.4461
## F-statistic: 25.97 on 1 and 30 DF,  p-value: 1.776e-05
```

As we can see, this model isn't enough good. Why can we make such a conclusion? Let's go step by step.

1. When assessing how well the model fits the data, you should look for symmetrical distribution across residuals on the mean value zero. Residuals aren't really small, but the median is close to zero and residuals are symmetric.
2. P-value of our drat factor is good (cause it's pretty small), but for the intercept, it's not so good.
3. R-squared is about 0.45-0.46, it's not enough to make a conclusion, that our model is good. Normally, the R-squared should be > 0.8 to conclude, that the model is good.
4. F-statistic > 1 and it's good. As a result, we can say, that these 2 factors are definitely connected with each other, but it's not enough to use only the rear axle ratio to forecast miles/gallon. Let's create a plot with this linear regression and then we can think over, what we can do to make our model better.

```
ggplot(data = mtcars, mapping=aes(y=mpg, x=drat)) +
  geom_point() +
  geom_smooth(method='lm')
```



This chart proves our conclusion, our model isn't perfect. Let's add all of the factors to our model and see if it can increase our R-squared.

```
lr = lm(mpg~cyl+disp+hp+drat+wt+qsec+vs+am+gear+carb, data=mtcars)
summary(lr)
```

```
##
## Call:
## lm(formula = mpg ~ cyl + disp + hp + drat + wt + qsec + vs +
##     am + gear + carb, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4506 -1.6044 -0.1196  1.2193  4.6271
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.30337   18.71788   0.657   0.5181
## cyl         -0.11144    1.04502  -0.107   0.9161
## disp         0.01334    0.01786   0.747   0.4635
## hp          -0.02148    0.02177  -0.987   0.3350
## drat         0.78711    1.63537   0.481   0.6353
## wt          -3.71530    1.89441  -1.961   0.0633 .
## qsec         0.82104    0.73084   1.123   0.2739
## vs           0.31776    2.10451   0.151   0.8814
## am           2.52023    2.05665   1.225   0.2340
```

```
## gear         0.65541    1.49326   0.439   0.6652
## carb        -0.19942    0.82875  -0.241   0.8122
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.65 on 21 degrees of freedom
## Multiple R-squared:  0.869,  Adjusted R-squared:  0.8066
## F-statistic: 13.93 on 10 and 21 DF,  p-value: 3.793e-07
```

Interesting. Our R-squared is now higher, but p-values are insignificant. Now we can create a correlation matrix, maybe we can find something interesting.

```
round(cor(mtcars), 2)
```

```
##        mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
## mpg   1.00 -0.85 -0.85 -0.78  0.68 -0.87  0.42  0.66  0.60  0.48 -0.55
## cyl  -0.85  1.00  0.90  0.83 -0.70  0.78 -0.59 -0.81 -0.52 -0.49  0.53
## disp -0.85  0.90  1.00  0.79 -0.71  0.89 -0.43 -0.71 -0.59 -0.56  0.39
## hp   -0.78  0.83  0.79  1.00 -0.45  0.66 -0.71 -0.72 -0.24 -0.13  0.75
## drat  0.68 -0.70 -0.71 -0.45  1.00 -0.71  0.09  0.44  0.71  0.70 -0.09
## wt   -0.87  0.78  0.89  0.66 -0.71  1.00 -0.17 -0.55 -0.69 -0.58  0.43
## qsec  0.42 -0.59 -0.43 -0.71  0.09 -0.17  1.00  0.74 -0.23 -0.21 -0.66
## vs    0.66 -0.81 -0.71 -0.72  0.44 -0.55  0.74  1.00  0.17  0.21 -0.57
## am    0.60 -0.52 -0.59 -0.24  0.71 -0.69 -0.23  0.17  1.00  0.79  0.06
## gear  0.48 -0.49 -0.56 -0.13  0.70 -0.58 -0.21  0.21  0.79  1.00  0.27
## carb -0.55  0.53  0.39  0.75 -0.09  0.43 -0.66 -0.57  0.06  0.27  1.00
```

There is a good correlation between mpg and the following factors: cyl, disp, hp, wt. Let's use only them to predict our mpg.

```
lr = lm(mpg~cyl+disp+hp+wt, data=mtcars)
summary(lr)
```

```
##
## Call:
## lm(formula = mpg ~ cyl + disp + hp + wt, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0562 -1.4636 -0.4281  1.2854  5.8269
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 40.82854    2.75747  14.807 1.76e-14 ***
## cyl         -1.29332    0.65588  -1.972 0.058947 .
## disp         0.01160    0.01173   0.989 0.331386
## hp          -0.02054    0.01215  -1.691 0.102379
## wt          -3.85390    1.01547  -3.795 0.000759 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.513 on 27 degrees of freedom
## Multiple R-squared:  0.8486, Adjusted R-squared:  0.8262
## F-statistic: 37.84 on 4 and 27 DF,  p-value: 1.061e-10
```

Now our results look better. R-squared is a little bit higher, F-statistic is bigger and 2 p-values are significant. Wt is a good predictor, but how it's connected with the other. There is a strong correlation between wt and disp (even better as between wt and mpg) and wt and cyl. Let's remove them from our model.

```
lr = lm(mpg~hp+wt, data=mtcars)
summary(lr)
```

```
##
## Call:
## lm(formula = mpg ~ hp + wt, data = mtcars)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.941 -1.600 -0.182  1.050  5.854
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 37.22727    1.59879  23.285  < 2e-16 ***
## hp          -0.03177    0.00903  -3.519  0.00145 **
## wt          -3.87783    0.63273  -6.129 1.12e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.593 on 29 degrees of freedom
## Multiple R-squared:  0.8268, Adjusted R-squared:  0.8148
## F-statistic: 69.21 on 2 and 29 DF,  p-value: 9.109e-12
```

Now we have a good result. Although R-squared is a little bit lower, as in the previous model, F-statistic is bigger and all of the coefficients are significant. I would recommend this model to predict. Important note! More isn't always better. **If you use strong-correlated factors as predictors for your model it can screw the final result. It's always a good idea to look at the correlation matrix, run your lm function a set of times with different predictors to decide, which factors are better for the final model.**

Spearman/Kendall correlation

Both Spearman and Kendall correlation coefficients are ranked. It means, that they compare not the real figures, but their ranks (like places in the score-table). They aren't so demanding as the Pearson coefficient, that's why we can use them if we have not so clear data or initial data is ranked. To show, how they work, we can take our previous dataset and correlation between disp and mpg. Disp isn't distributed normally.

```
cor(mtcars$disp, mtcars$mpg, method = "spearman")
```

```
## [1] -0.9088824
```

```
cor(mtcars$disp, mtcars$mpg, method = "kendall")
```

```
## [1] -0.7681311
```

The results are different, but both show a strong correlation between factors. Please note, that it's a common pattern, that the Spearman coefficient is higher than Kendall's. It's normal and it comes from the different calculation methods. So, we've finished the first part of our scheme, which describes methods for 2 quantitative variables. Let's move further and explore the next branch - 2 quantitative variables.

Chi-square test

The Chi-square test is used to show the correlation between categorical data such as gender/group etc. There is a good dataset for such an aim - the Titanic dataset. There are 2201 observations, so it's more than enough for our analysis. Please keep in mind, that this test works well if there are at least 5 observations in each cell. Let's check, if gender, age, or class was a factor for survival. Firstly, we need to create a contingency table of the two variables, so we need 3 tables.

```
class_surv <- apply(Titanic, c(1, 4), sum)
gender_surv <- apply(Titanic, c(2, 4), sum)
age_surv <- apply(Titanic, c(3, 4), sum)
```

Then we can use the chisq.test to analyze the correlation.

```
class <- chisq.test(class_surv)
gender <- chisq.test(gender_surv)
age <- chisq.test(age_surv)

class
```

```
##
##  Pearson's Chi-squared test
##
## data:  class_surv
## X-squared = 190.4, df = 3, p-value < 2.2e-16
```

```
gender
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  gender_surv
## X-squared = 454.5, df = 1, p-value < 2.2e-16
```

```
age
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  age_surv
## X-squared = 20.005, df = 1, p-value = 7.725e-06
```

As we can see, all of the factors were significant for the survival (p-value is less than 0.05), especially the gender (cause the X-squared is the biggest). Now let's see, which cell influences the result the most. Residuals analysis can show us, which group has the most effect.

```
round(class$residuals, 0)
```

```
##       Survived
## Class  No Yes
##   1st  -7  10
##   2nd  -2   3
##   3rd   2  -3
##   Crew  3  -4
```

```
round(gender$residuals, 0)
```

```
##         Survived
## Sex       No Yes
##   Male     6  -8
##   Female -11  16
```

```
round(age$residuals, 0)
```

```
##        Survived
## Age      No Yes
##   Child -3   4
##   Adult  1  -1
```

The results are the following: if you were a female or a passenger of the first class, then you would have a higher probability to survive.

Fisher exact test

The Fisher exact test allows us to analyze contingency tables, if there are not enough observations in each group (<5). Let's work with the data frame esoph and try to find out if there is a correlation between age and tobacco consumption among these patients (e.g. they are starting to smoke more, as they are getting older). Firstly we need to create a contingency table with the help of PivotTable.

```
pt <- PivotTable$new()
pt$addData(esoph)
pt$addColumnDataGroups("agegp",addTotal=FALSE)
pt$addRowDataGroups("tobgp", addTotal=FALSE)
pt$defineCalculation(calculationName="ncases", summariseExpression="sum(ncases)")
pt$evaluatePivot()
age_tob <- pt$asDataMatrix()
age_tob
```

```
##          25-34 35-44 45-54 55-64 65-74 75+
## 0-9g/day     0     2    14    25    31   6
## 10-19        1     4    13    23    12   5
## 20-29        0     3     8    12    10   0
## 30+          0     0    11    16     2   2
```

Pretty good, but now aren't able to use F-test for the whole matrix, cause it's quite big and figures aren't <5. To show this kind of analysis, let's cut our matrix and use only 2 age groups (25-34 and 35-44). Then we'll use F-test.

```
age_tob_cut <- age_tob[1:4, 1:2]
age_tob_cut
```

```
##          25-34 35-44
## 0-9g/day     0     2
## 10-19        1     4
## 20-29        0     3
## 30+          0     0
```

```
fisher.test(age_tob_cut)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  age_tob_cut
## p-value = 1
## alternative hypothesis: two.sided
```

Like any other statistical test, if the p-value is less than the significance level(0.05), we can reject the null hypothesis (there is no correlation). Our p-value equals 1, so we can't say, that there is any correlation between age and tobacco consumption. This example isn't especially meaningful in terms of data (honestly, the H1 hypothesis is really weak), but it can illustrate, how F-test works and how we can read the results.

Student's t-test

The t-test can be used to determine if the means of two sets of data are significantly different from each other. As it was mentioned previously, data should be distributed normally, or at least the sample size should be high. For this case let's use the Israel Covid19 DataSet from Kaggle. Our task is to find out if Covid is detected by women in Israel more often/rare than by men. In this case, we can use a statistical method, because we haven't tested the whole population of Israel and because our timeline is limited. For this exercise, we are using corona_age_and_gender.csv file as of 16/11/21. Now we have to import our data and prepare it for future analysis.

```
#this link should be updated, if you use another directory for the initials file.
covid <- read.csv("~/Develop/materials/corona_age_and_gender.csv")

skim_without_charts(covid) #providing key statistical summary
```

Table 2: Data summary

Name	covid
Number of rows	3759
Number of columns	7
Column type frequency:	
character	4
numeric	3
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
first_week_day	0	1	10	10	0	85	0
last_week_day	0	1	10	10	0	85	0
age_group	0	1	3	5	0	15	0

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
gender	0	1	3	10	0	3	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
weekly_tests_num	0	1	7755.53	17945.19	0	213	2931	8414.0	247689
weekly_cases	0	1	352.61	1126.44	0	0	36	263.5	17049
weekly_deceased	0	1	1.24	7.51	0	0	0	0.0	112

```
covid$first_week_day <- dmy(covid$first_week_day) #changing the format
covid$last_week_day <- dmy(covid$last_week_day) #changing the format

head(covid)
```

```
##   first_week_day last_week_day age_group     gender weekly_tests_num
## 1     2020-03-15    2020-03-21      0-19        Men              1115
## 2     2020-03-15    2020-03-21      0-19 Not Binary                 0
## 3     2020-03-15    2020-03-21      0-19      Women              1065
## 4     2020-03-15    2020-03-21     20-24        Men               613
## 5     2020-03-15    2020-03-21     20-24 Not Binary                 0
## 6     2020-03-15    2020-03-21     20-24      Women               710
##   weekly_cases weekly_deceased
## 1           44               0
## 2            0               0
## 3           39               0
## 4           92               0
## 5            0               0
## 6           50               0
```

Our data have no missing values, so we can transform it to the needed format.

```
covid_new <- covid %>%
  select(-weekly_deceased) %>%
  pivot_wider(names_from = gender, values_from = c(weekly_tests_num,
                                                   weekly_cases)) %>%
  group_by(first_week_day, last_week_day) %>%
  summarise(tested_men=sum(weekly_tests_num_Men),
            tested_women=sum(weekly_tests_num_Women),
            cases_men=sum(weekly_cases_Men),
            cases_women=sum(weekly_cases_Women)) %>%
  mutate(cov_freq_percent_w=round(cases_women/tested_women*100, 2)) %>%
  mutate(cov_freq_percent_m=round(cases_men/tested_men*100, 2)) %>%
  select(-c(tested_men, tested_women, cases_men, cases_women)) %>%
  arrange(first_week_day)

head(covid_new)
```

```
## # A tibble: 6 x 4
## # Groups:   first_week_day [6]
```

```
##   first_week_day last_week_day cov_freq_percent_w cov_freq_percent_m
##   <date>         <date>                    <dbl>              <dbl>
## 1 2020-03-15     2020-03-21                 4.99               8.36
## 2 2020-03-22     2020-03-28                 7.08              10.1
## 3 2020-03-29     2020-04-04                 6.79               8.82
## 4 2020-04-05     2020-04-11                 6.2                6.72
## 5 2020-04-12     2020-04-18                 3.83               4.08
## 6 2020-04-19     2020-04-25                 2.22               2.36
```

Now we want to decide if this data is good enough for the t-test. According to the wiki, the following assumption should be met:

The means of the two populations being compared should follow normal distributions. Under weak assumptions, this follows in large samples from the central limit theorem, even when the distribution of observations in each group is non-normal.

Our sample sizes are equal and enough large to use the T-test. But to prove the assumptions we can use the bootstrap method.

Bootstrap

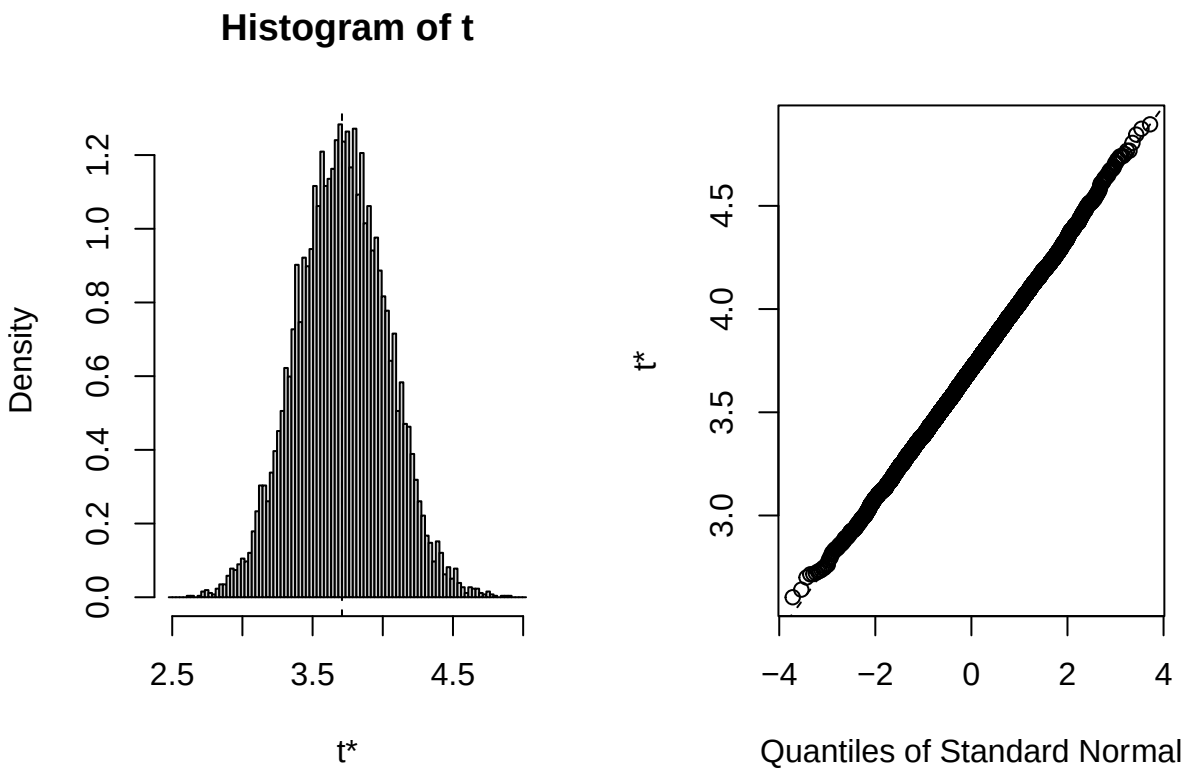
Bootstrapping is a method that estimates the sampling distribution by taking multiple samples with replacement from a single random sample. These repeated samples are called resamples. Let's create 10 000 samples of cov_freq_percent_w variable and for each sample find the mean. Then evaluate the distribution of this mean.

```
set.seed(1) #to reproduce results in the future to be at the same page.

meanfunc_w <- function(data, i){
  d <- data[i, ]
  return(mean(d$cov_freq_percent_w))
}

bootstrap_freq_w <- boot(covid_new,meanfunc_w,R=10000)

plot(bootstrap_freq_w)
```



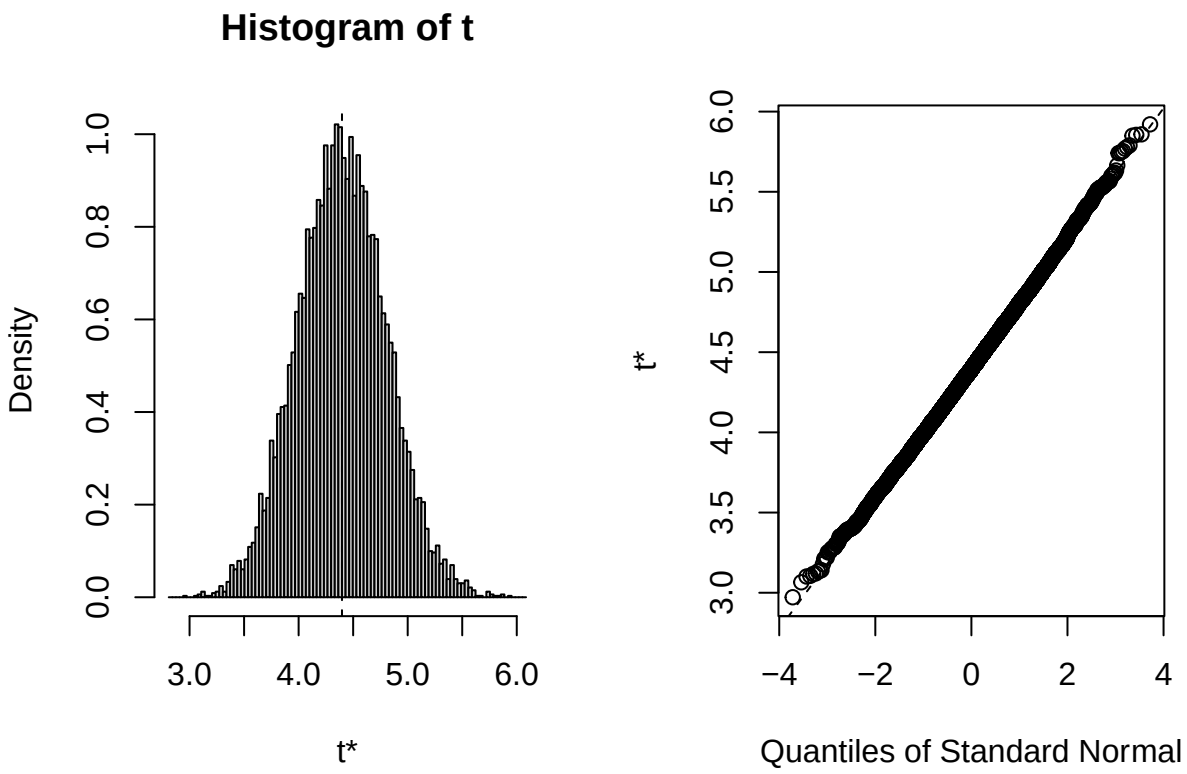
Perfect! Our mean is distributed normally. Then we continue this exercise with the same indicator for men.

```
set.seed(1) #to reproduce results in the future to be at the same page.

meanfunc_m <- function(data, i){
  d <- data[i, ]
  return(mean(d$cov_freq_percent_m))
}

bootstrap_freq_m <- boot(covid_new,meanfunc_m,R=10000)

plot(bootstrap_freq_m)
```



Also great! Now we can be sure, that means are distributed normally and we can use the T-test.

T-test, results, visualization

```
t.test(covid_new$cov_freq_percent_w, covid_new$cov_freq_percent_m)
```

```
##
##  Welch Two Sample t-test
##
## data:  covid_new$cov_freq_percent_w and covid_new$cov_freq_percent_m
## t = -1.3256, df = 158.9, p-value = 0.1869
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.7136379  0.3371673
## sample estimates:
## mean of x mean of y
##  3.709294  4.397529
```

Unfortunately, the p-value is more than 0.05, so we can't reject the Ho. There is no significant difference between men and women in terms of diagnosed Covid through tests ratio. What is also interesting, that means of our factors are the most popular values in bootstraps distributions of means.

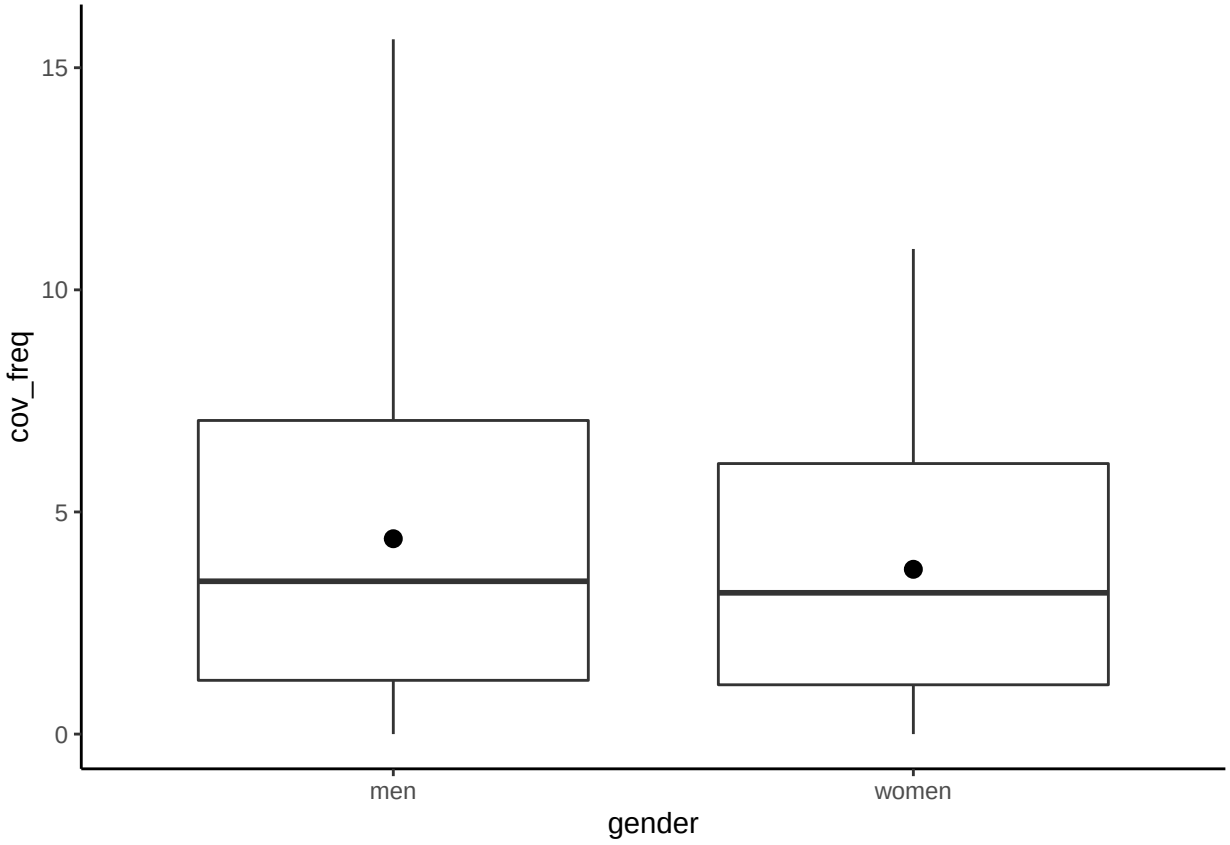
Sometimes it's also good to show our results as a boxplot. Boxplot shows five summary statistics: the minimum, the maximum, the median, and the first and third quartiles of the data. Sometimes, you might

want to add other statistical summary values on the boxplot (like mean). If two boxes do not overlap with one another, say, box A is completely above or below box B, then there is a difference between the two groups. Let's do the needed calculations and visualize it.

```
#preparing the data frame
covid_new_long <- pivot_longer(covid_new, cols=3:4, values_to = "cov_freq",
                               names_to = "gender")

xlabs <- c("men", "women")

ggplot(data = covid_new_long, mapping=aes(y=cov_freq, x=gender)) +
  geom_boxplot() +
  theme_classic() +
  scale_x_discrete(labels= xlabs) +
  stat_summary(fun="mean")
```



So we can see the same result, as it was by the T-test. There is no significant difference between these groups.

Mann–Whitney U-test/Wilcoxon test

These tests are non-parametric analogs to the T-test. If our sample size isn't enough large or we are working with ranks, then we can use these tests. For this purpose, we can use data set frets. It contains only 25 observations, which isn't enough for the student test. Let's see, if there is a significant difference between head length and head breadth of brothers.

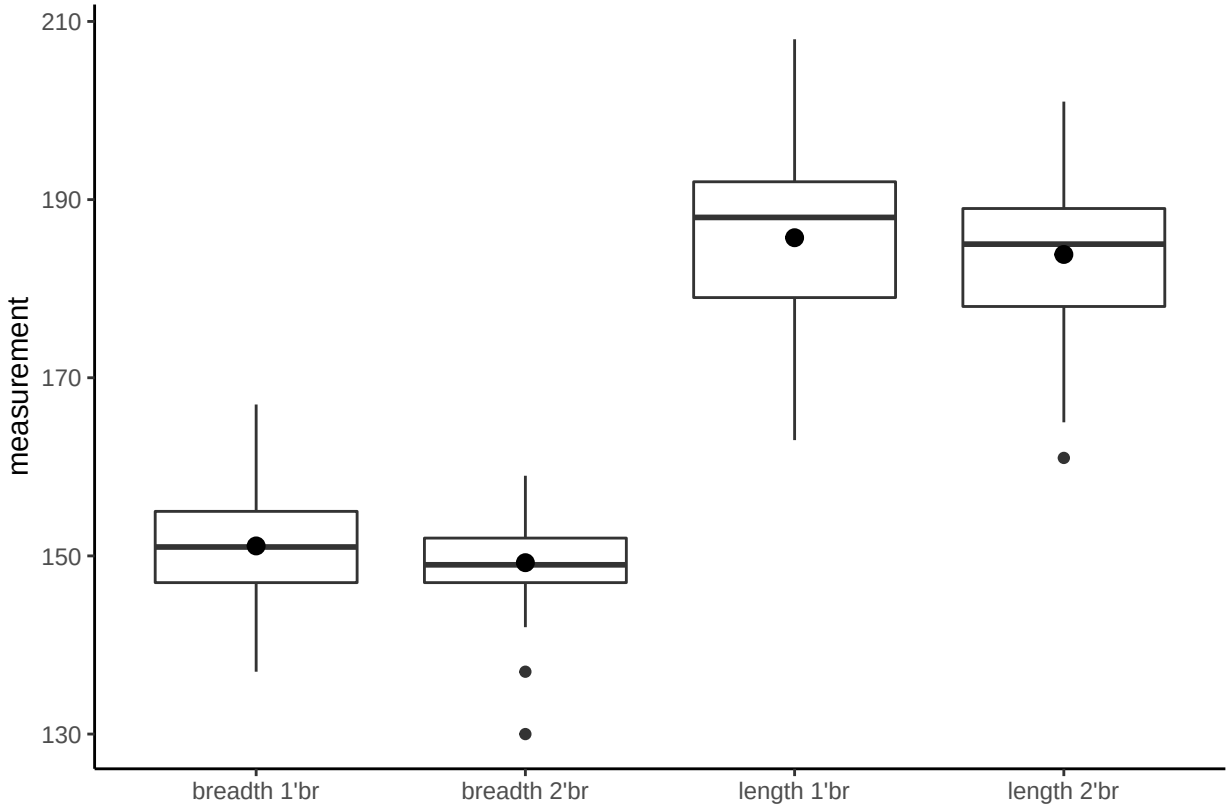
```
wilcox.test(frets$l1,frets$l2, exact = FALSE)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  frets$l1 and frets$l2
## W = 348.5, p-value = 0.4905
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(frets$b1,frets$b2, exact = FALSE)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  frets$b1 and frets$b2
## W = 359.5, p-value = 0.366
## alternative hypothesis: true location shift is not equal to 0
```

P-values in both cases are greater than 0.05, so there is no significant difference in the brother's head's dimensions. And in this case, we can also show it on the boxplot.



The same picture. Visually there is also no significant difference between these groups.

ANOVA

Analysis of variance (ANOVA) is based on the analysis of variation among and between groups. To use ANOVA, the equality of variances (homoscedasticity) should be met.

The important note about this method! Just because it allows to compare a lot of groups and test a lot of hypotheses, there is such effect as **multiple comparisons problem**. In easy words, it means, that as more comparisons we are making, as higher is the probability of type I error. To handle such an issue, some corrections to the p-value should be done. There are some methods, the most popular are Bonferroni correction and Tukey's range test. The first one is too conservative and increases type II error, that's why it's not so popular in some areas of analysis. There are different types of ANOVA, depending on the number of factors, independence of samples, and so on.

The algorithm of the analysis is the following:

1. Levene's test to prove the homoscedasticity,
2. ANOVA (+ choosing the best model if applicable),
3. Tukey's HSD (if we find significant results)

This time let's work with the PlantGrowth database.

```
plant <- PlantGrowth

plant_stat <- plant %>%
  group_by(group) %>%
  summarise(
    number_observ = n(),
    mean_value = mean(weight),
    sd_value = sd(weight)
  )

plant_stat
```

```
## # A tibble: 3 x 4
##   group number_observ mean_value sd_value
##   <fct>         <int>      <dbl>    <dbl>
## 1 ctrl             10       5.03    0.583
## 2 trt1             10       4.66    0.794
## 3 trt2             10       5.53    0.443
```

```
leveneTest(weight ~ group, data = plant)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  2  1.1192 0.3412
##       27
```

P-value is 0.34, so it's bigger than 0.05. It means, that we can't decline H0 about homoscedasticity, so as with the Shapiro-Wilk test, it's a good sign for us.

```r
anova <- aov(weight ~ group, data = plant)
summary(anova)
```

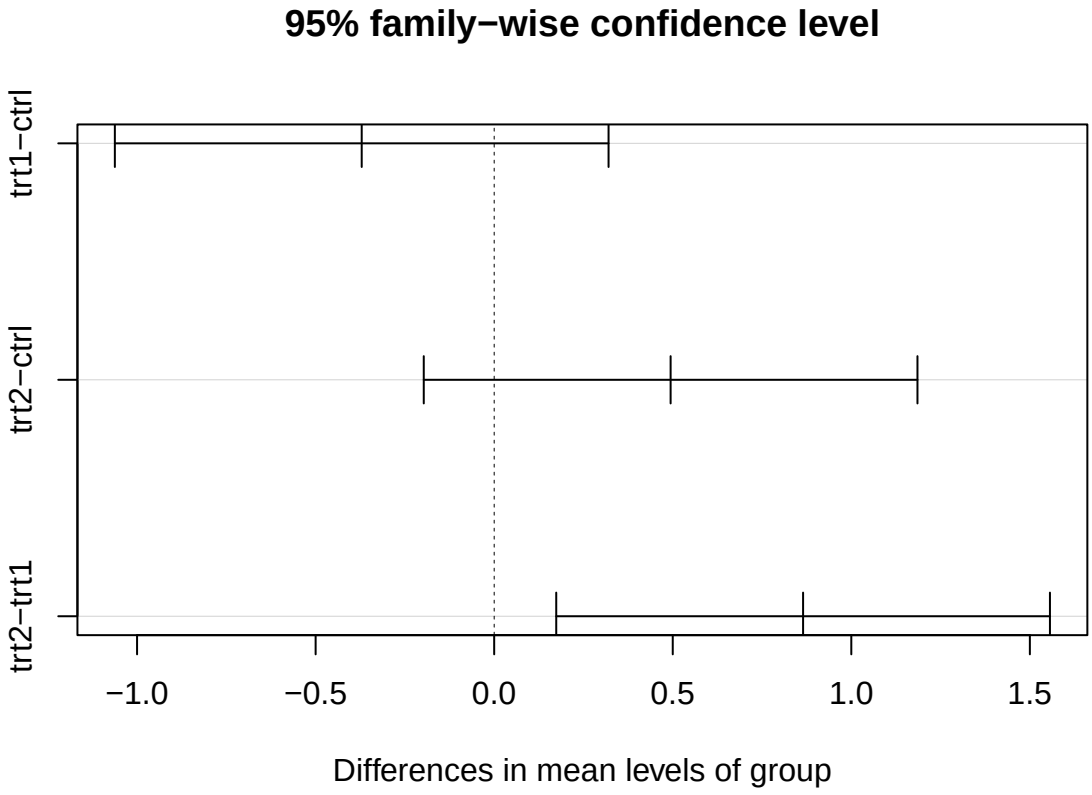
```
##             Df Sum Sq Mean Sq F value Pr(>F)
## group        2  3.766  1.8832   4.846 0.0159 *
## Residuals   27 10.492  0.3886
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

P-value is less than 0.05, so there is at least one significant difference between our groups. Let's check, where there is this difference.

```r
tukey_anova <- TukeyHSD(anova)
tukey_anova
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = weight ~ group, data = plant)
##
## $group
##             diff        lwr       upr     p adj
## trt1-ctrl -0.371 -1.0622161 0.3202161 0.3908711
## trt2-ctrl  0.494 -0.1972161 1.1852161 0.1979960
## trt2-trt1  0.865  0.1737839 1.5562161 0.0120064
```

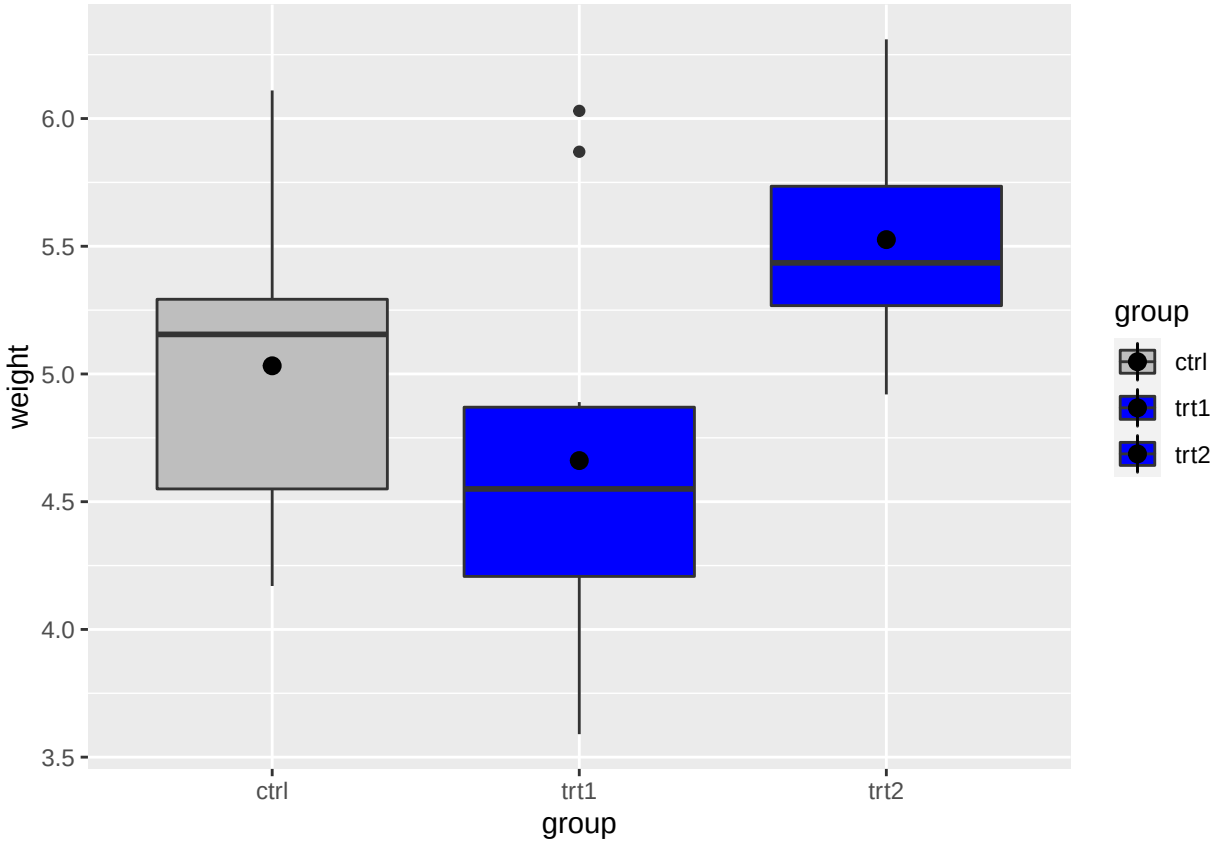
```r
plot(tukey_anova)
```



There is only one significant difference (p adj < 0.05) - between trt2 and trt1. Visualization can also help us to notice this difference - the interval shouldn't include 0 - it also means, that there is a significant difference here.

Finally, we can show our results as a box plot.

```
ggplot(data = plant, mapping=aes(y=weight, x=group, fill=group)) +
  scale_fill_manual(values=c("grey", "blue", "blue")) +
  geom_boxplot() +
  stat_summary(fun="mean")
```



Sometimes if we have 2 or more factors, it can be useful to compare our models. For this case, we can use aictab function from the AICcmodavg library. The model with the lowest AIC score is the best fit for the data and AICcwt shows the percentage, which this model describes. We don't have another factor here, so we don't have to follow this step now.

```
install.packages('AICcmodavg', repos = "http://cran.us.r-project.org")
library(AICcmodavg)
aictab()
```

Kruskal–Wallis H test and Friedman test

To save time, let's use the same dataset, that we've used in the previous section. The Kruskal–Wallis H test should be used, if our observations are independent, but there is no homoscedasticity between the groups. To find out, where exactly is the difference we can use the Wilcoxon test, adjusted for the multiple comparisons.

```
kruskal.test(weight ~ group, data = plant)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  weight by group
## Kruskal-Wallis chi-squared = 7.9882, df = 2, p-value = 0.01842
```

```
pairwise.wilcox.test(plant$weight, plant$group,
                 p.adjust.method = "BH")
```

```
## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot compute
## exact p-value with ties
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  plant$weight and plant$group
##
##      ctrl  trt1
## trt1 0.199 -
## trt2 0.095 0.027
##
## P value adjustment method: BH
```

According to these tests, there is a significant difference, cause the p-value in the first case is less than 0.05. And this difference is only between trt1 and trt2 groups, according to the p-value matrix in the second case. So we have the same results, as we had using the ANOVA method.

The Friedman test is a non-parametric alternative to the one-way repeated measures ANOVA test (if our observations aren't independent). This test is used very rarely, so let's just briefly mention, what exactly should we do. The algorithm is the same, as for the H test. Firstly we need to run friedman.test(), then we can find out the effect size, using friedman_effsize() - if it's close to 1, the result is trustworthy. Then we can run pairwise.wilcox.test() to adjust p-values and find the differences.

```
friedman.test()
friedman_effsize()
pairwise.wilcox.test()
```