Applied Research: Advance Statistics with R

Carlos Utrilla Guerrero

6/08/2020

## WELCOME

Welcome to the workshop number 3: Introduction to inferential stats with R.

### Learning outcomes:

By the end of this assignment(s), you should be able to:

* Know the difference between null and alternative hypothesis
* Formulate different hypothesis testing
* Calculate correlation and covariance
* Interpret Pearson correlation and scatter plots
* Loop functions in R (bonus exercise)

### Essential R assignment(s) document guidelines:

In the current document you will find the following color(s) highlight(s) and format(s). Please refer to this table for legend description

|  |  |
| --- | --- |
| # this comment: | This is a comment writing by you to describe what you intent to do. |
| print(‘the thing') | This is the thing that you want to run. |
| ## [1] "the output” | ## [1] This is the output of the thing that you run in R. |
| # insert your code here # | This is the expected answer of each question throughout the document. |

### Dataset

In this assignment(s), we will use the following dataset:

1. Workshop Statistics\_ descriptives .xlsx.
2. trial\_ACTG175.xlsx

## Brief Recap lecture: hypothesis testing

Statisticians use hypothesis testing to formally check whether the hypothesis is accepted or rejected. Hypothesis testing refers to the process of generating a clear and testable question, collecting and analyzing appropriate data, and drawing an inference that answers your question. Of course, this means that some steps come into play.

Generally speaking, hypothesis testing is conducted in the following manner:

|  |  |  |
| --- | --- | --- |
| Phase | Definition | Explanation |
| Phase 1 | State the hypotheses | Stating the null and alternative hypotheses |
| Phase 2 | Specify the level of significance | The selection of the probability to reject the null hypothesis when is true (α = 0.05) |
| Phase 3 | Compute t-statistics and p value | Calculation of the test statistics and p value with mean, standard deviation and sample size |
| Phase 4 | Interpret results | Application of the decision rules |

Remember from the lecture that hypothesis testing ultimately uses a p-value to weight the strength of the evidence.

The p-values ranges between 0 and 1. It can be interpreted in the following way:

* A small p-values (typically ≤ 0.05) indicates strong evidence against the null hypothesis, so you reject it.
* A large p-values (>0.05) indicates weak evidence against the null hypothesis, so you fail to reject it.

Also, recall the two type of errors that can occur from the hypothesis testing:

* Type I Error: it occurs when the researcher rejects the null hypothesis when it is in fact true. The term significance level is used to express the probability of Type I error.
* Type II Error: it occurs when the researcher accept the null hypothesis when is in fact false. The term power is used to express this probability to happen.

Also, remember the three ‘basic’ versions of t-test:

1. One sample t-test, which tests the mean of a single group against a known mean.

2. Independent samples t-test, which compares mean for two groups.

3. Paired sample t-test, which compares means from the same group at different times.

Today only t-test 1 and 2 will be covered. In the next lecture, we will deeply have a look on all of them and many others!

### Using the T-test function in R

R can handle various types of T-tests using the t.test() command. This function can be used to deal with one-sample tests as well as two-sample (un-)paired tests.

Listed below are the most common arguments used in the t.test and their explanation:

|  |  |
| --- | --- |
| Argument | Explanation |
| t.test(sample1,sample2) | The basic method of applying t.test as comparing means of numeric data. |
| var.equal = FALSE | If the var.equal instruction is set to TRUE, the variance is considered to be equal and the t-student test is carried out. If the instruction is set to FALSE (the default), the variance is considered unequal and the *Welch* two-sample test is carried out. |
| mu = 0 | If a one-sample test is carried out, mu indicates the mean against which the sample should be tested |
| alternative = “two.sided” | It sets the alternative hypothesis. The default value for this is “two.sided” but a greater or lesser value can also be assigned. |
| conf.level = 0.95 | It sets the confidence level of the interval (default = 0.95) (confidence level = 1-alpha level) |
| paired = FALSE | If set to TRUE, a matched *pair* T-test/ dependent t-test is carried out |

*EQUAL VARIANCE NOTES:*

By default, the R t.test() function for the two-sample t.test makes the assumption that the variances of the two groups of samples, being compared, are different. Therefore, Welch t-test is performed by default. Welch t-test is just an adaptation of t-test, and it is used when the two samples have possibly unequal variances. Therefore, we should always test our assumptions of whether the variances are equal using F test (var.test() in R).

More info: [here](http://www.sthda.com/english/wiki/wiki.php?id_contents=7600#:~:text=Remember%20that%20independent%20t%2Dtest,%2C%20being%20compared%2C%20are%20different.)

**However, for sake of simplicity, we will always run two-sample t.test with the predefined assumption of equal variances (var.equal = FALSE). What do we need to do? We will discuss this later when covering two-sample t-test.**

#### One-Sample T-test

All the tests in the t-test family compare differences in mean scores of continuous-level (i.e. numerical) and normally distributed data. Unlike the independent or dependent sample t-tests (also called unpaired or paired t-test), the one-sample t-test works with only one mean score.

The one-sample t-test compares the mean of a single sample to a predefined value. Some possible applications of the one-sample t-test include testing a sample against a predetermined or expected value, testing a sample against a certain benchmark, or testing the results of a replicated experiment against the original study. For example, a researcher may want to determine if the average age of retiring in a certain population is indeed 65 (as often defined by retirement law). The t-test determines whether the difference between the sample mean and predefined is larger than we would expect to see by chance.

In the lecture, we conducted an experiment in order to investigate whether or not ‘coffee lovers’ shops in Maastricht pounder 33 cl on the orange juice glass. In our experiment we collected data of 10 orange juices and measured the amount of orange cl per glass. The results are here:

|  |  |
| --- | --- |
| id | cl |
| 1 | 28 |
| 2 | 31 |
| 3 | 28 |
| 4 | 37 |
| 5 | 30 |
| 6 | 33 |
| 7 | 25 |
| 8 | 33 |
| 9 | 24 |
| 10 | 30 |

Let’s create the hypothesis null for ‘two-sides’ in R as following:

# Define orange sample  
orange <- c(28,31,28,37,30,  
 33,25,33,24,30)  
  
# Conduct one-sample t-test  
t.test(x = orange, # sample values  
 mu = 33) # Null Hypothesis

# Model output  
## One Sample t-test  
##   
## data: orange  
## t = -2.5135, df = 9, p-value = 0.03312  
## alternative hypothesis: true mean is not equal to 33  
## 95 percent confidence interval:  
## 27.11001 32.68999  
## sample estimates:  
## mean of x   
## 29.9

***Model output description***

|  |  |
| --- | --- |
| variable | description |
| title | Type of test performed. |
| t | t statistic. |
| df | degrees of freedom = sample size - 1. |
| p-value | probability of selecting a sample with mean not equal to 33 cl. |
| Alternative hypothesis (Ha) | Alternative hypothesis |
| C.I 95% | There is a 95% chance that the confidence interval (C.I) you calculated contains the true population mean. While C.I is usually expressed with 95% confidence, this is just a tradition. C.I can be computed for any desired degree of confidence |
| upper-lower | This is the boundary where the true population mean lies |

**Please find more information about confidence interval and its components in this article:** <https://www.simplypsychology.org/confidence-interval.html>

***Model output interpretation***

* the mean orange volume in cl for our sample is 29.9 cl.
* the two-sided 95% confidence interval tells you that the mean orange volume is likely to be between 27.11 and 32.6 cl. This already indicates that there isn’t statistical support that the amount of orange juice poured is equal to 33cl.
* This is further supported by the p-value of 0.033. This value tells you that if the true mean volume of orange juice would indeed be 33 cl, the probability to select a sample with a mean volume that DOES NOT EQUAL 33 cl would be approximately 3% (p-value = 0.033). Since the p-value is less than the significance level (0.05), we can reject the H0 which states that the true mean is equal to 33 cl.

#### Using Directional Hypothesis (one-sided)

In the previous experiment (orange), we are simply interested in testing if the means are different. In many other cases, however, you might want to know if the mean of a sample is lower or great, e.g. if the orange sample mean is lower or greater than 33 cl. We then use the ‘alternative’ option in R (see below) to switch from two-sided (default) to one-sided.

The choices you have are between ″two.sided″, ″less″, or ″greater″.

Suppose we want to use a one-sample t-test to determine whether the amount orange juice in coffee lovers are less than they say (33 cl). The test has the null hypothesis that the mean orange volume is equal to 33 cl, and the alternative hypothesis that the mean orange volume is less than 33 cl. A significant level of 0.05 is used.

t.test(orange,  
 mu = 33,  
 alternative = 'less') # the Ha is less than 33 cl

##   
## One Sample t-test  
##   
## data: orange  
## t = -2.5135, df = 9, p-value = 0.01656  
## alternative hypothesis: true mean is less than 33  
## 95 percent confidence interval:  
## -Inf 32.16084  
## sample estimates:  
## mean of x   
## 29.9

The output is similar to the previous one, except it contains a one-sided 95% confidence interval. This interval tells us that the mean orange volume is likely to be less than 32.16084 cl. The p-value of 0.01656 tell us that if the mean volume of the orange juice were 33 cl, the probability of selecting a sample with a mean less than or equal to this one would be approximately 1.6%. Since the value is less than the significant level 0.05, we can reject the null hypothesis that the mean orange volume is equal to 33 cl.

We could also make the same hypothesis test but with a different significant level.

For instance, let us do perform the same test (whether orange juice is less than 33 cl) but with a significant level of 0.01:

t.test(orange,  
 mu = 33,  
 alternative = 'less', # upper tail test  
 conf.level = 0.99) # conf level is 1 - significant level [alpha]

##   
## One Sample t-test model output orange  
##   
## data: orange  
## t = -2.5135, df = 9, p-value = 0.01656  
## alternative hypothesis: true mean is less than 33  
## 99 percent confidence interval:  
## -Inf 33.37977  
## sample estimates:  
## mean of x   
## 29.9

Want to already become more familiar with this t.test and the different applications in R? Use these sources:

+ [Performing a one-sample t-test in R](http://www.instantr.com/2012/12/29/performing-a-one-sample-t-test-in-r/)

+ [One-Sample t Test & Confidence Interval in R](https://www.youtube.com/watch?v=kvmSAXhX9Hs)

#### Exercise 1: Using t.test(), can you please determine whether the mean orange sample is greater than 33 cl? The significant level to be used is 0.05. Make sure you also interpret the output!

t.test(orange,  
 mu = 33,  
 alternative = 'greater')

##   
## One Sample t-test  
##   
## data: orange  
## t = -2.5135, df = 9, p-value = 0.9834  
## alternative hypothesis: true mean is greater than 33  
## 95 percent confidence interval:  
## 27.63916 Inf  
## sample estimates:  
## mean of x   
## 29.9

So far we have seen how to carry out the t-test on one vector of values. However, we can use specific columns from a dataset as well.

### 

### Application to a different data set

In this section, we will use the already known descriptive dataset (import into excel and call it *mydat*). You can download the data here: <https://docs.google.com/spreadsheets/d/1JchKI5u-I5IcWbnM_JpMkMJIZX1012oJVoIEZz2ra3c/edit#gid=0>

Let us use the variable age from this dataset. As you might remember, you can calculate the mean of the variable age as following:

mean(mydat$age)

## [1] 18.14493

Since we have missing data (e.g. -99), this statistic is not truly representative, and we have to make the necessary steps to clean the data. Here you have a brief summary of the steps that we have made last tutorial (please repeat):

# change the name cols  
names(mydat)[6] <- "favPet"  
names(mydat)[7:8] <- c('GoT','LotR')  
# missing data  
mydat[mydat == -99] <- NA  
# dummy variable  
mydat$favPet[mydat$favPet==1] <- 0  
mydat$favPet[mydat$favPet==2] <- 1  
# convert to factor  
mydat$favPet <- factor(mydat$favPet, labels = c('cat','dog'))

Now, we can again compute the mean of the variable age:

mean(mydat$age, na.rm = TRUE)

## [1] 21.64179

Now, we want to test whether or not the mean age of the population (e.g. all first- and second-year students from UCV) is equal to 20, at significance level of 5%.

t.test(mydat$age, #sample as vector  
 mu=20) # true mean is equal to 20 age

##   
## One Sample t-test  
##   
## data: mydat$age  
## t = 4.4081, df = 66, p-value = 3.927e-05  
## alternative hypothesis: true mean is not equal to 20  
## 95 percent confidence interval:  
## 20.89818 22.38540  
## sample estimates:  
## mean of x   
## 21.64179

***Model output interpretation***

p- value (0.00003927) is notoriously smaller than the significance level (=0.05) thus, we do have strong evidence to reject HO and we could draw a conclusion that true mean is equal NOT to 20. In addition, value of MU = 20 is not within the interval values of CI, thereof it can be rejected the Ho that true mean is equal to 20.

#### Exercise 2: Using t.test, can you please determine whether the mean length of the true population (i.e. venlo students year 1 and 2) is greater than 168 cm? The significant level to be used is 0.01.

t.test(mydat$length, # sample of length  
 mu = 168, # true mean in cm  
 alternative = 'greater', # ha as true mean greater than 168 cm  
 conf.level = 0.99) # significant level of 0.01

##   
## One Sample t-test  
##   
## data: mydat$length  
## t = 1.9855, df = 66, p-value = 0.02562  
## alternative hypothesis: true mean is greater than 168  
## 99 percent confidence interval:  
## 167.6194 Inf  
## sample estimates:  
## mean of x   
## 169.8955

### 

### (Unpaired) Independent samples t-test

The independent samples t-test (also called two samples unpaired t-test) compares the mean of one distinct group to the mean of another distinct group. An example research question for an independent sample t-test would be, “Do boys and girls differ in their age expectancy?”

The aim of this section is to show you how to calculate independent samples t test with R software. The t-test formula is described [here](http://www.sthda.com/english/wiki/t-test-formula).

Using the same dataset that you have imported previously (mydat), test whether group 1 student’s average age is significantly different from group 2 student’s average age. The number of individuals considered here is obviously low but fair enough to illustrate the usage of two-sample t-test.

Let’s first create a summary of the mean and standard deviation for each group:

library(dplyr) # Remember to import dplyr if you want to run among other

things summarise()

summarise(group\_by(mydat, group),   
 count = n(),  
 mean = mean(age, na.rm = TRUE))

## # A tibble: 8 x 3  
## group count mean  
## <dbl> <int> <dbl>  
## 1 0 3 21.7  
## 2 1 9 21.9  
## 3 2 10 22.2  
## 4 3 9 20.8  
## 5 4 12 21.9  
## 6 5 5 22.4  
## 7 6 11 21.1  
## 8 7 10 21.4

Since we got more than 2 categories within the group variable, the unpaired t-test can be performed as follow (though only for 2 categories; so we select group 1 and group 2). Remember that the default option of R is to assume that variances are unequal. We assume that variances are equal, thus we need to add var.equal = TRUE to the code. Since we are performing independent samples t-test we also need to specify paired = FALSE:

t.test(x = mydat$age[mydat$group == 1], y = mydat$age[mydat$group == 2], # sample 1 and sample 2  
 paired = FALSE, # different and independent observations from samples  
 var.equal = TRUE) # we assume variance are equal

##   
## Two Sample t-test  
##   
## data: mydat$age[mydat$group == 1] and mydat$age[mydat$group == 2]  
## t = -0.16052, df = 17, p-value = 0.8744  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -4.400235 3.778013  
## sample estimates:  
## mean of x mean of y   
## 21.88889 22.20000

In the result above:

* t is the t-test statistic value (t = -0.16052)
* df is the degrees of freedom (df= 17)
* p-value is the significance level of the t-test (p-value = 0.8744).
* conf.int is the confidence interval of the mean at 95% (conf.int = [-4.400235, 3.778013])
* sample estimates is the mean value of the sample (mean = 21.88889, 22.20000).

***Model Output Interpretation***

The t-test is determining whether there are differences between the means in group 1 and 2 observation. Remember that *H0* states there are no differences between the true means (mean.g1 = mean.g2) against a *Ha* in which we assume that the differences between group 1 and 2 means is not equal to 0 (mean.g1 <> mean.g2).

Since p-value is greater than the significance value 5%, we can accept (thus ‘fail to reject’) the *H0* of equal means. Based on this data, we conclude that there is not enough evidence of a difference between means of the group 1 and group 2.

#### 

#### Exercise 3: We are interested in determining whether or not the mean length of group 3 is equal or greater than the mean length of group 1 members at a significance level of 5%. Use the t.test to perform a unpaired two-sample tests assuming that the variances are equal. Interpret the output!

t.test(x = mydat$length[mydat$group == 3], y = mydat$length[mydat$group == 1], # samples filter  
 alternative = 'greater', # case Ha different is greater than  
 paired = FALSE, # different and independent observations from samples  
 var.equal = TRUE) # we assume variance are equal thus, we run Student t.test

##   
## Two Sample t-test  
##   
## data: mydat$length[mydat$group == 3] and mydat$length[mydat$group == 1]  
## t = -1.3613, df = 16, p-value = 0.9039  
## alternative hypothesis: true difference in means is greater than 0  
## 95 percent confidence interval:  
## -10.90528 Inf  
## sample estimates:  
## mean of x mean of y   
## 169.3333 174.1111

Another common scenario when analyzing your data is when you have a variable with only two possible categories of responses (like for ‘favPet’: dog, cat). Thus, we can write the following code:

t.test(age ~ favPet, # comparing mean differences between favPet respondents  
 data = mydat, # dataset  
 var.equal = TRUE) # assumption variances equal between two samples

##   
## Two Sample t-test  
##   
## data: age by favPet  
## t = -1.5249, df = 65, p-value = 0.1321  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -2.7998128 0.3753822  
## sample estimates:  
## mean in group cat mean in group dog   
## 20.80952 22.02174

#### 

#### Exercise 4: Can you please compare the length means between respondents who like dog vs those who like cats as a pets? Use a significant level of 0.01 to perform a two-sample tests assuming that the variances are equal. What conclusion can you draw?

t.test(length ~ favPet,  
 data = mydat,  
 conf.level = 0.99,  
 var.equal = TRUE)

##   
## Two Sample t-test  
##   
## data: length by favPet  
## t = -1.0732, df = 65, p-value = 0.2872  
## alternative hypothesis: true difference in means is not equal to 0  
## 99 percent confidence interval:  
## -7.660840 3.248832  
## sample estimates:  
## mean in group cat mean in group dog   
## 168.381 170.587

## 

## Scatter plot, correlation and covariance

In statistics, you deal with a lot of data. The hard part is finding patterns that fit the data. As you learned, you can identify basic patterns using a scatter plot and correlation.

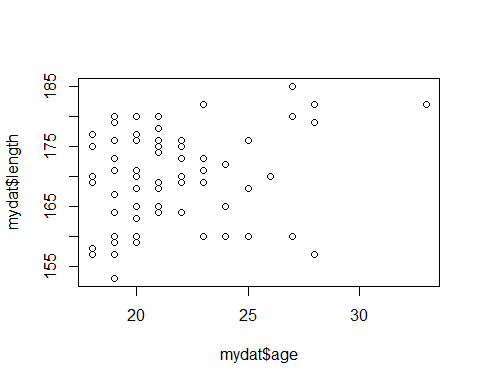
The most useful graph for displaying the relationship between two numerical variables is a scatterplot. As a reminder, a scatterplot shows the relationship between two numerical variables measured for the same individuals. The values of one variable appear on the horizontal axis, and the values of the other variable appear on the vertical axis. Each individual in the data appears as a point on the graph.

A correlation is a mathematical relationship between two variables. There are three ways that data can correlate:

* Positive correlation is when the scatter plot takes a generally upward trend. It also means that the line of best fit has a positive slope.
* Negative correlations, you guessed it, have a generally downward trend in the scatter plot. This means that the slope of the line of best fit has a negative slope.
* Zero correlation is also referred to as no correlation. This means that the pattern has no discernible pattern. This usually implies that the two variables are unrelated.

Create as was done in workshop 2 a simple scatter plot:

plot(mydat$age,mydat$length)



According to the above figure, it is quite difficult to make a solid argument about age and length relationship. Perhaps not even longer consider linear relationship right?

### Using visualization to create a more meaningful scatter plot I

In order to create a meaningful scatter plot for correlation analysis, we could consider to add into the graph a line that represent the direction of the relationship. For this purpose, we can use this command and run it together (select both code lines and run in R):

plot(mydat$age,mydat$length)  
abline(lm(length~age,mydat)) # add the estimated line

A screenshot of a cell phone

Description automatically generated

You could see from the scatter plot that including this line, add more information, at very least, about the slope/direction of the relationship.

Still, relying on the interpretation of a scatterplot is too subjective. More precise evidence is needed, and this evidence is obtained by computing a coefficient that measures the strength of the relationship under investigation.

### 

### Calculate coefficient correlation

Now, I have been using the word ‘slope’ to refer to the line of best fit, but that does not really tell you the strength of the correlation. To determine the strength of a correlation, the correlation coefficient is best. As shown in the lecture, the correlation coefficient is comprised between -1 and 1:

* -1 indicates a strong negative correlation : this means that every time age increases, length decreases
* 0 means that there is no association between the two variables (age and length)
* +1 indicates a strong positive correlation : this means that length increases with age

There are different methods to compute the correlation, the most famous ones being Pearson correlation and Spearman rank-order correlation. The Pearson correlation is the one which you use when you want to calculate the correlation between two continuous variables. Both variables should have a normal distribution and be linearly related! (More on this when we talk about ‘correlation testing’ later). Spearman correlation is a non-parametric alternative for pearson correlation. We won’t cover the latter in more detail here, but if you want to know more about it you can have a look at the following sources:

<http://www.biostathandbook.com/spearman.html>

<https://www.youtube.com/watch?v=F0lvYZmxib8>

In R, for bivariate/pearson correlation we use the cor() R function to obtain a correlation coefficient, as shown in the following command:

# correlation between age and length  
cor(mydat$age, mydat$length)

## [1] NA

It returns NA. As it happens with other descriptive statistics (e.g. mean, variance and standard deviation), the function will return NA in case there are existing missing values. Inconveniently, the developers of the cor function did not use the na.rm = TRUE to circumvent this issue. Instead, cor() function deals with missing data adding an argument called ‘use’. The most common useful are use = ‘complete.obs’ and use=’pairwise.complete.use’. The former deletes all cases with missing values before calculating the correlation. The former applies when calculating a correlation matrix (e.g. correlations between more than two variables). If you want to know more about how to handle with missing data, a good tutorial is here: <https://thomasleeper.com/Rcourse/Tutorials/NAhandling.html>

For the moment, we can choose ‘complete.obs’ as a way to deal with missing data. Then, we add the use argument in our R code:

cor(mydat$age, mydat$length,   
 use= 'complete.obs') # deal with NA

## [1] 0.259803

The correlation coefficient is 0.259. Using the [rule of thumb](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3576830/#:~:text=The%20Pearson's%20correlation%20coefficient%20for%20these%20variables%20is%200.80.&text=For%20example%2C%20a%20correlation%20coefficient,use%20the%20most%20appropriate%20one.) for interpreting the size of the correlation coefficient introduced in the lecture (click on link to get access to an article on coefficient interpretation), we could consider that there is not a clearly, linear association between both variables.

Learn more about how to interpret a correlation coefficient here:

<https://www.dummies.com/education/math/statistics/how-to-interpret-a-correlation-coefficient-r/>

### 

**Exercise 5: Download and import into R the dataset ‘trial\_ACTG175’ (accessible through *Student Portal 🡪 Course Materials 🡪 R Workshops 🡪 R Workshop 3*). Determine the strength of the correlation between the two continuous variables: cd4\_baseline and cd8\_baseline. Briefly interpret the outcome.**

### Calculate covariance

The cov() command uses a syntax similar to the cor() command to examine covariance.

We now check the relationship between variable age and length using cov().

cov(mydat$age, mydat$length, use = 'complete.obs')

## [1] 6.189281

The covariance is 6.189 an indication of a positive linear relationship between two variables.

*What are the differences between covariance and correlation?*

In simple words, both the terms (correlation and covariance) measure the relationship and the dependency between two variables. “Covariance” indicates the direction of the linear relationship between variables. “Correlation” on the other hand measures both the strength and direction of the linear relationship between two variables. Correlation is a function of the covariance. What sets them apart is the fact that correlation values are standardized whereas, covariance values are not. You can obtain the correlation coefficient of two variables by dividing the covariance of these variables by the product of the standard deviations of the same values.

Check this article to get more insight about its differences: [https://towardsdatascience.com/let-us-understand-the-correlation-matrix-and-covariance-matrix-d42e6b643c22#](https://towardsdatascience.com/let-us-understand-the-correlation-matrix-and-covariance-matrix-d42e6b643c22)

### Correlation testing

Getting a correlation coefficient is generally only half the story, and you might want to know if the relationship is statistically significant different from zero. Similar to hypothesis testing, the association between two numerical variables can be evaluated using the cor.test command.

The cor.test is an inferential statistical test that examines whether there is a linear relationship between two variables. In this case, our null hypothesis is *“there is no correlation between the two variables”* and alternative hypothesis *is “there is a nonzero (significant different from zero) correlation between two variables”*. This R function returns both the correlation coefficient and the significance level (or p-value) of the correlation.

Note that the cor.test in R runs by default the pearson correlation test. This test has two assumptions:

1. The two variables are normally distributed. We can test/check this assumption using:
   * Shapiro-Wilk statistical test
   * Looking at the plots (remember workshop 2?):
     + Histogram
     + A boxplot
2. The relationship between two variables is linear. We can test this assumption by examining the two variables using the Shapiro-wilk test OR scatterplots.

If these assumptions aren’t met, alternative methods have to be consulted (e.g. spearman, kendall).

Check this tutorial to know more about “preliminary test to check corr assumptions in R”:

<http://www.sthda.com/english/wiki/correlation-test-between-two-variables-in-r#preleminary-test-to-check-the-test-assumptions>

Let’s presume that there is a linear relationship between age and length and focus today on the ‘pearson’ method. We then run a pearson correlation test between age and length variables:

cor.test(mydat$age, mydat$length)

##   
## Pearson's product-moment correlation  
##   
## data: mydat$age and mydat$length  
## t = 2.1691, df = 65, p-value = 0.03374  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.02089861 0.47064042  
## sample estimates:  
## cor   
## 0.259803

***Model Output Description***

t: is the t-test statistic value (t = 2.1691)

df: is the degrees of freedom (df= 65)

p-value: is the significance level of the t-test (p-value = 0.03374).

conf.int: is the confidence interval of the correlation coefficient at 95% (conf.int = [0.02089861, 0.47064042])

cor: sample estimates is the correlation coefficient (Cor.coef = 0.259803)

***Model Output Interpretation***

When testing the null hypothesis that there is no correlation between age and length, we reject the null hypothesis (r = 0.2598, t = 2.1691, with a p-value = 0.03374). The p-value of the test is 0.03374, which is less than the significance level alpha = 0.05.

Note that this 95% confidence interval [0.02089861, 0.47064042]) does not contain 0, which is consistent with our decision to reject the null hypothesis. The p-value of the test is 0.03374, which is less than the significance level alpha = 0.05. We then can conclude that age and length variables are significantly correlated.

Note that the conclusion that age and length are related (thus accepting the Ha) is not supported by the slightly moderate correlation coefficient of 0.259. How can this be explained? Could it be that not meeting the Pearson correlation assumptions has influenced our results? **We asked you to presume for exercise sake that the assumptions were met, but actually they were not.** Might we draw a wrong conclusion because of low sample size? Are we therefore making a Type I Error? Can a weak correlation be significant? Check this article out if you want to know more about correlation and its interpretations:

Eight things you need to know about interpreting correlations (Research Skills One):

<http://users.sussex.ac.uk/~grahamh/RM1web/Eight%20things%20you%20need%20to%20know%20about%20interpreting%20correlations.pdf>

**Exercise 6: In exercise 5 you have assessed the strength of the correlation between the two continuous variables: cd4\_baseline and cd8\_baseline (assume that the Pearson assumptions are met!). But are these two variables significantly correlated? Run a correlation test to examine this.**

### Using visualization to create a more meaningful scatter plot II

Let’s create together an informative scatter plot that can be undoubtly used in a scientific report with the following inputs:

- Line representing the relationship between variables (linear regression)

- Shape of the confidence intervals for regression line

- r as coefficient value

- p value to make conclusion

install.packages(“ggpubr”) # install packages ggpubr first

library(ggpubr) # import library

## Loading required package: ggplot2

ggscatter(mydat, x = 'age', y = 'length', # sample data  
 add = 'reg.line', # add regression line  
 conf.int = TRUE, # add its conf interval  
 add.params = list(color = "blue",  
 fill = "lightgray"),  
 cor.coef = TRUE, # add r coef  
 cor.method = 'pearson',  
 xlab = "age", ylab = "length(cm)")

A screenshot of a cell phone

Description automatically generated

As you noted, we have all needed information in one graph. Indeed, R packages are amazingly powerful and can help you out to create data visualisations. Check this articles and learn more about R packages for graphs:

1. <https://towardsdatascience.com/a-guide-to-data-visualisation-in-r-for-beginners-ef6d41a34174>
2. <https://mode.com/blog/r-data-visualization-packages/>

**Exercise 7: Create an informative scatter plot for the continuous variables: cd4\_baseline and cd8\_baseline (dataset Trial\_ACTG175) . Insert your plot.**

### 

### GOING FORWARD

You learned today the basics of data exploration and inferential statistics in R and there’s so much more that you can do to make sure that you get a good feel for your data before you start analyzing and modeling it.

You want to practice more? Head on over to [Kaggle](https://www.kaggle.com/datasets) and find a rich, fascinating data set of your own to explore ☺.

## Bonus section: Standard deviation, Standard Error and Loop function

1. What is the age mean within each group and over all groups?

# Step 1: Compute the mean and standard deviation of the entire dataset (= over all groups)  
mydat.original <- mydat  
paste0( 'The mean of the age for the entire data set is ', mean(mydat$age, na.rm = TRUE))

## [1] "The mean of the age for the entire data set is 21.6417910447761"

# Step 2: compute the mean within each group (=col 1 and has seven categories)  
  
# create two objects in which the mean (or length) is stored  
  
ageGrouped <- rep(NA, times =7)   
lengthGrouped <- rep(NA, times =7)  
  
# For each group, compute the age mean, allocate the information into the desired object  
ageGrouped[1] <- mean(mydat$age[mydat$group ==1], na.rm = TRUE)   
ageGrouped[2] <- mean(mydat$age[mydat$group ==2], na.rm = TRUE)  
ageGrouped[3] <- mean(mydat$age[mydat$group==3], na.rm = TRUE)  
ageGrouped[4] <- mean(mydat$age[mydat$group==4], na.rm = TRUE)  
ageGrouped[5] <- mean(mydat$age[mydat$group==5], na.rm = TRUE)  
ageGrouped[6] <- mean(mydat$age[mydat$group==6], na.rm = TRUE)  
ageGrouped[7] <- mean(mydat$age[mydat$group==7], na.rm = TRUE)  
  
# Repeat same logic to compute length mean  
lengthGrouped[1] <- mean(mydat$length[mydat$group ==1], na.rm = TRUE)   
lengthGrouped[2] <- mean(mydat$length[mydat$group ==2], na.rm = TRUE)  
lengthGrouped[3] <- mean(mydat$length[mydat$group==3], na.rm = TRUE)  
lengthGrouped[4] <- mean(mydat$length[mydat$group==4], na.rm = TRUE)  
lengthGrouped[5] <- mean(mydat$length[mydat$group==5], na.rm = TRUE)  
lengthGrouped[6] <- mean(mydat$length[mydat$group==6], na.rm = TRUE)  
lengthGrouped[7] <- mean(mydat$length[mydat$group==7], na.rm = TRUE)

***Remark: We are doing essentially a filter of our data set. In the script we tell to R to allocate in the new object, the results of the mean for the entire dataset, select age and filter per group***

## Parenthesis:

Now if this feels tedious to you, you are absolutely right! there is a much quicker way to do it,

Like build our loop function, which will save you countless hours by automating your mean calculation.

You can use what ever letter you like here as long as you keep using the same one the i in 1: 7 will give i first the value 1, execute everything between the curly brackets, then i becomes 2 etc. until 7

**You can compare the old and new objects and see they are identical, but now it with only 4 lines of code!**

# first create a two new objects in which the results of loop function will be stored  
ageGroupedWithForLoop <- rep(NA, times =7)  
lengthGroupedWithForLoop <- rep(NA, times =7)  
  
for(i in 1:7){  
 ageGroupedWithForLoop[i] <- mean(mydat$age[mydat$group == i], na.rm = TRUE)  
 lengthGroupedWithForLoop[i] <- mean(mydat$length[mydat$group == i], na.rm = TRUE)  
}  
  
# Compare means for age  
mean(mydat$age)

## [1] NA

mean(ageGrouped, na.rm = TRUE)

## [1] 21.67063

mean(ageGroupedWithForLoop, na.rm = TRUE)

## [1] 21.67063

# compare means for length  
sd(mydat$length)

## [1] NA

sd(lengthGrouped, na.rm = TRUE)

## [1] 2.977935

sd(lengthGroupedWithForLoop, na.rm = TRUE)

## [1] 2.977935

1. Now, it is time to replicate previous steps to compute the standard deviation. Can you make it with For Loop method?

paste0('The standard deviation of age for the entire data set is ', round(sd(mydat$age),digits = 2))

## [1] "The standard deviation of age for the entire data set is NA"

paste0('The standard deviation of length for the entire data set is ', sd(mydat$length))

## [1] "The standard deviation of length for the entire data set is NA"

# Create the new object that will store result of standard deviation for age and length  
SDageGroupedwithForLoop <- rep(NA, times = 7)  
SDlengthGroupedwithForLoop <- rep(NA, times = 7)  
  
# Create for loop function  
  
for (i in 1:7) {  
 SDageGroupedwithForLoop[i] <- sd(mydat$age[mydat$group == i], na.rm = TRUE)  
 SDlengthGroupedwithForLoop[i] <- sd(mydat$length[mydat$group == i], na.rm = TRUE)  
}  
  
# Compare standard deviations  
sd(mydat$age)

## [1] NA

sd(ageGrouped, na.rm = TRUE)

## [1] 0.592153

sd(ageGroupedWithForLoop, na.rm = TRUE)

## [1] 0.592153

## What do you notice?

The standard deviation over the means (sd(ageGroupedWithForLoop, na.rm = TRUE)) is smaller than the standard deviation in each of the groups (elements on SDageGroupedWithForLoop() object.

## 

## APPLYING YOUR KNOWLEDGES

### Research Question: Is there a correlation between a patient’s age and their CD4 T cell count at baseline?

*1. Create a plot showing the relationship between age and CD4 T cell count at baseline. Based on this plot, do you expect to find a relationship between the two variables?*

*2. Using the cor.test() function which conducts a correlation test between a patient’s age at baseline (age) and their CD4 T cell count at baseline (cd4\_baseline).*

### Research Question: Do patients with a history of intravenous drug use have higher CD4 T cell counts at baseline compared to patients without a history of intravenous drug use?

*1. create a plot showing the distribution of CD4 T cell counts at baseline (cd4\_baseline) depending on whether or not patients were symptomatic (symptomatic variable). Based on this plot, do you expect to find that symptomatic patients generally have higher (or lower) CD4 T cell counts?*

*2.Use t.test() function to conduct a two-sample t-test comparing patients’ CD4 T cell count at baseline (cd4\_baseline) between patients that are symptomatic and those that are not (symptomatic)*