Applied Research: Advance Statistics with R

6/10/2020

## WELCOME

Welcome to the workshop number 4: More about inferential stats with R.

### Learning outcomes:

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

* Know-how to perform t.test for paired data
* Know-how to conduct and use One-way Anova test
* Know-how to use chi-square for independent

### 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

Throughout the assignment(s) in this workshop, we will use several different datasets:

***Workshop Statistics\_ descriptives .xlsx.***

library(readxl)
mydat <- read\_excel("Workshop Statistics\_ descriptives .xlsx")

or download the excel through the following link

<https://docs.google.com/spreadsheets/d/1JchKI5u-I5IcWbnM_JpMkMJIZX1012oJVoIEZz2ra3c/edit#gid=0>

***Student Sleep data***

In order to import this dataset, you can either

1. download and import it in R using readxl() function here:

<https://docs.google.com/spreadsheets/d/1ngyhOJhkRFBco8l6AEhHogHlVquHcfxRvVWwe9Zr2x0/edit#gid=0>

1. or use the build-in sleep function to directly import into R. Please, run the following R command:

# import R data set called 'sleep'
sleep <- sleep

*Description*

A data from Cushny and Peebles (1905) who gave two different optical isomers of the same soporific drug to ten individuals. They then measured how much extra time the individuals slept, compared to a control, undrugged state.

*Format*

A data frame with 20 observations on 3 variables.

[variable 1] extra: numeric increase in hours of sleep

[variable 2] group: factor of drug given (1= drug one; 2 = drug two).

[variable 3] ID: patient ID

*Source*

Cushny, A. R. and Peebles, A. R. (1905) The action of optical isomers: II hyoscines. The Journal of Physiology 32, 501–510.

You can find more information about the data using help() function.

***Plant Growth Data***

In order to import this dataset, you can either

1. download and import it in R using readxl() function here:

<https://docs.google.com/spreadsheets/d/15NbP9_yd3C-uqhGPfsL1O3PEA5P9OWlSsfs2ofPmaUI/edit#gid=0>

\*note: if you decided to import in this way, you will need to convert the variable “group” as a factor with the following R command:

1. Import the data

my\_data <- read\_excel("PlantGrowth.xlsx")

2. Convert to factor variable “group”

my\_data$group <- factor(my\_data$group)

1. or use the built-in R data set named PlantGrowth. It contains the weight of plants obtained under a control and two different treatment conditions.

1. Import the data

my\_data <- PlantGrowth

*Description*

Results from an experiment to compare yields (as measured by dried weight of plants) obtained under a control and two different treatment conditions.

*Format*

A data frame of 30 cases on 3 variables.

|  |  |  |
| --- | --- | --- |
| [Variable 1] | id | numeric |
| [Variable 2] | weight | numeric |
| [Variable 3] | group | factor |

The levels of group are ‘ctrl’, ‘trt1’, and ‘trt2’.

*Source*

Dobson, A. J. (1983) An Introduction to Statistical Modelling. London: Chapman and Hall.

## Results from an Experiment on Plant Growth Dobson, A. J. (1983)

***Student Survey Data***

Let’s import the data as following:

1. download and import it in R using readxl() function here:

<https://docs.google.com/spreadsheets/d/14vo6sfIW64M3WJBX-8JPx-4CKtv-ag6S9U8HxfSvGYs/edit?usp=sharing>

\*note: if you import the excel file in this way, do not forget to convert the character variables into factor variable to allow R recognize them as categorical. You can see an example here:

# 1. Import the dataset

survey.data <- read\_excel("survey.xlsx")

# 1. Convert Exer and Smoke variables to categorical/factor

survey.data$Exer <- factor(survey.data1$Exer, levels = c("Freq","None","Some"))

survey.data$Smoke <- factor(survey.data1$Smoke, levels = c("Heavy","Never","Occas","Regul"))

1. or use the R library called MASS in order to import the dataset:

# 1. Import the library with the dataset survey:

library(MASS) # load the MASS package

# 2. Create an object (survey.data) to store the survey dataset

survey.data <- survey

*Description*

This data frame contains the responses of 237 Statistics I students at the University of Adelaide to a number of questions in relation to their habits.

Check more info here: <https://cran.r-project.org/web/packages/MASS/MASS.pdf>

### Two-sample t-test: Paired t-test

As you already know, t-tests are a great way of identifying if two group means are statistically different. This can be done by comparing a sample to the population (one-sample) or comparing two different samples (two-sample).

T-tests are further broken down into two categories: unpaired t-tests and paired t-tests (sometimes also referred to as independent vs dependent t-tests). In the previous workshop we looked at one-sample and unpaired/independent t-tests. The following section will focus on paired t-test.

***What is a paired t-test?***

A paired t-test is a statistical test that compares the means of two related groups to determine if there is a significant difference between the two groups (see example about the mine weight before and after treatment). Paired t-tests are considered more powerful than unpaired t-tests because using the same participants or subjects eliminate variation between the samples that could be caused by anything other than what’s being tested.

***Paired vs unpaired t-test***

The unpaired test compares means of independent groups. However, when utilizing paired t-tests, the observations are not independent because they stem from the same individual (e.g. same person measurement before and after lunch). So, the paired t-test compares sets of data from the same individuals at two different points whereas unpaired t-test compares sets of data that arises from different individuals.

As you have seen in the lecture, paired t-test analysis is performed as follow:

1. Calculate the difference (*d*) between each pair of value
2. Compute the mean (*m*) and the standard deviation (*s*) of *d*
3. Compare the average difference to 0.

## Dataset:

To illustrate the paired t-test, we will use the Student’s Sleep Data (see beginning of document on how to download and for dataset description).

### Check your data

First things first; let look at the data and check the variables that it contains

print(sleep)

## extra group ID
## 1 0.7 1 1
## 2 -1.6 1 2

str(sleep)

'data.frame': 20 obs. of 3 variables:

 $ extra: num 0.7 -1.6 -0.2 -1.2 -0.1 3.4 3.7 0.8 0 2 ...

 $ group: Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...

 $ ID : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...

Look at the ID Factor? What do you see? Each participant was observed twice. In this case, once when drug 1 was administered and once with drug 2. Hence, we need to run a paired t-test!

Let’s first visually have an intuition of what our data looks like. As a good starting point for out data analysis, it is always recommended to create some nice data visualization to get a sense of what are the summary statistics of our data. We will then create a boxplot with the two groups in order to their central measures of tendency. A barplot will illustrate us how is the data distribution per each group.

We first create boxplot for the two drug groups:

boxplot(sleep$extra ~ sleep$group,
 col = c("red", "blue"),
 ylab = 'weight',
 xlab = 'groups',
 main = 'formula = extra ~ groups')



**Exercise 1: What do you see in the boxplot? Is there any differences between both of them in terms of extra sleep?**

We want to assess if there is a statistically significant effect of a particular drug on sleep (two measurement groups with 10 patients per group). We need to test if there is an existing difference between means of the sleep hours for each (drug) group. Since the data are in ID order for each of the drugs the easiest way is to simply use the paired = TRUE option in the t.test() command. (take a look at the data to be sure!, e.g. wit the print() or str() function as we have done),

t.test(x = sleep$extra[sleep$group == 1],
 y = sleep$extra[sleep$group ==2],
 paired = TRUE)

##
## Paired t-test
##
## data: sleep$extra[sleep$group == 1] and sleep$extra[sleep$group == 2]
## t = -4.0621, df = 9, p-value = 0.002833
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -2.4598858 -0.7001142
## sample estimates:
## mean of the differences
## -1.58

**Interpretation**

With t = -4.06 and a 95% confidence interval on the difference in means that does not include zero (-2.46 to -0.7), the analysis suggests that drug 1 does indeed produce less extra sleep than drug 2 (you can see it in the negative t value – check means per groups to be sure what we are saying!). The fact that the p-value is 0.003 (rounded) reinforces this conclusion, telling us that a difference at least this large would be very unlikely if there was in fact no difference between the drugs. So, we can conclude that the drug 2 isomer is a more powerful soporific.

Note that since we only the variable group has only two categories/groups, we could also consider creating an r command like this:

*t.test(extra ~ group,*

 *data = sleep,*

 *paired = TRUE)*

Run it into R console and check if you get same results!

Say you would have a categorical variable that has more than 2 groups/categories, but you want to compare specifically two. Then you have to use the first t.test command, where you use $ to select the two categories/groups that you want to compare.

Have a look on this tutorial (later on) for learning how other data structures and way to organize your data might help you out with your analysis:

<http://rstudio-pubs-static.s3.amazonaws.com/29006_84fdff6365a34ad7bcaaca73d3647853.html>

This above example for paired t-test has been executed with the assumption of different variance due to default method of var.equal = FALSE. If you notice, we didn’t specify the argument var.equal. If we believe that the variance between groups are equal then, we just need to explicitly mention the argument var.equal = TRUE. If you want to be sure that your variance between groups are equal, you should run a t.test for equal variance, called in R var.test(). It test the hypothesis of the equal variance between groups (Ho: variance group 1 = variance group 2). In order to conduct this test, R uses the var.test() function. The command will be as following:

var.test(sleep$extra[sleep$group == 1],

 sleep$extra[sleep$group == 2])

##
## F test to compare two variances
##
## data: sleep$extra[sleep$group == 1] and sleep$extra[sleep$group == 2]
## F = 0.79834, num df = 9, denom df = 9, p-value = 0.7427
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 0.198297 3.214123
## sample estimates:
## ratio of variances
## 0.7983426

The function var.test() returns a list containing the following components:

* **statistic**: the value of the F test statistic: (0.79834)
* **parameter**: the degrees of the freedom of the F distribution of the test statistic (9)
* **p.value**: the p-value of the test (0.7427)
* **conf.int**: a confidence interval for the ratio of the population variances (0.198297 3.214123)
* **estimate**: the ratio of the sample variances (0.7983)

**Interpretation**

The p-value of F-test is greater than the significant level of 0.05 (remember one minus confidence interval). In conclusion, there is no significant differences between the two variances, and we can assume equal variance between groups.

**Exercise 2: Now it is your turn, conduct the same paired t.test() under the assumption that variance between groups are equal. Remember to add to your t.test() the argument var.equal = TRUE. See previous workshop #3 if you need a hint on how to do this.**

## Other adjustments of t-test

This above exercise has demonstrated how to do paired t-tests. There are other adjustments we can make to these functions and this section will explain two of the available options and why we would maybe consider doing this. The following function adjustments will work for t.test:

### Alternative Hypothesis for one-tailed test:

When using a one-tailed test, you are testing for the possibility of the relationship in one direction and completely disregarding the possibility of a relationship in the other direction.

Let’s return to our example comparing the mean of the sleep hours for each (drug) group sample.  Our null hypothesis was that the mean differences are equal to *0*. The one-tailed test, will test either (1) if the mean differences are significantly greater than or (2) less than *0*, but not both.

Assume that, from the sleep survey, we hypothesize that the mean of group A is less than group B, perhaps because it is unreasonable to think the mean sleep in group A would be greater than the one in group B. If so, we only need to achieve a p value less than level of significance to achieve "statistical significance".

Now, we can adjust the alternative hypothesis easily within the t.test() function. In this case, we will set the argument alternative = “less” (H0≥0 and HA<0) as following:

t.test(extra~group, data = sleep, paired = TRUE, var.equal = TRUE, alternative = "less")

##
## Paired t-test
##
## data: extra by group
## t = -4.0621, df = 9, p-value = 0.001416
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
## -Inf -0.8669947
## sample estimates:
## mean of the differences
## -1.58

Since the p-value is less than our statistical significance .05 then we have achieved statistical significant results and reject the null hypothesis. We can then conclude that the true differences between means is less than 0.

## Exercise 3: How would you write a t.test command in R in which you want to determine if the difference in average increased sleep of the two groups is ≤0 (thus HA>0), using the sleep dataset. Remember to add the argument alternative = “ ”:

t.test(x = sleep$extra[sleep$group==1],

y=sleep$extra[sleep$group==2],

paired=TRUE,

var.equal=TRUE,

alternative = "greater")

**Defining predefined “score” difference (in R, mu)**

Following the same example with the survey dataset, now, let’s presume that you don’t really have idea about what are their differences between groups. What is you arguably state: I do believe the sleep mean between groups are not equal to zero, indeed I believe they are quite different and thereby, I do expect their mean differences to be less than 1.

Suppose we want to test if the difference in average increased sleep of the two treatment groups is greater or equal to 1(Ho: differences are ≥1 and thus Ha: differences is < 1), using the sleep dataset:

t.test(extra~group,
 data = sleep,
 alternative = "less",
 paired = TRUE,
 mu = 1)

##
## Paired t-test
##
## data: extra by group
## t = -6.6331, df = 9, p-value = 4.779e-05
## alternative hypothesis: true difference in means is less than 1
## 95 percent confidence interval:
## -Inf -0.8669947
## sample estimates:
## mean of the differences
## -1.58

## Testing multiple groups ANOVA

So far we have covered one sample t-test, which is a one sample hypothesis test, and two-sample t-test, which is a two sample hypothesis tests. In each case you are comparing two things. You are either comparing 1 group to a hypothesized value (one-sample t-test), or comparing the relationship between two groups (either their means (two sample t-test) or correlation between two variables). In this section, you will cover how to analyse more complex experimental designs using ANOVAs.

### What is one-way ANOVA test?

The one-way analysis of variance (ANOVA), also known as one-factor ANOVA, is a statistical technique, commonly used to study differences between two or more group means. It is an extension of the *independent* two-samples **t-test** for comparing means in a situation where there are more than two groups.

In one-way ANOVA, the data is organized into several groups based on one single grouping variable (also called factor variable). The grouping variable is a nominal, categorical variable that has two or more categories/values. You are testing the effect of this nominal variable on a numerical (continuous) variable.

This section describes the basic principle of the one-way ANOVA test and provides practical anova test examples in R software.

### ANOVA test hypotheses:

Null hypothesis: the means of the different groups are the same.

Alternative hypothesis: At least one sample mean is not equal to the others.

### 2 General steps to conduct an ANOVA

Here are the 2 general steps you should follow to conduct a standard ANOVA in R:

1. Create an ANOVA object using the aov() function. In the aov() function, specify the independent and dependent variable(s) with a formula with the format y ~ x1 where y is the dependent variable, and x1, is the independent factor variable.

# Step 1: Create an object (you can call it mod.aov)
mod.aov <- aov(formula = y ~ x1,
 data = data)

1. Create a summary ANOVA table by applying the summary() function to the ANOVA object you created in Step 1.

# Step 2: Look at a summary of the aov object
summary(mod.aov)

## Dataset:

Let’s do an example by running a one-way ANOVA on the PlantGrowth data (see beginning of document on how to download and for dataset description).

### Check the dataset

To have an idea of what the data look like, we use the function head(). The head() function randomly picks a few of the observations in the data frame to print out. Of course you can also use str() or print() to get a ‘sense’ of what your data is about:

head(my\_data)

## weight group
## 1 4.17 ctrl
## 2 5.58 ctrl
## 3 5.18 ctrl
## 4 6.11 ctrl
## 5 4.50 ctrl
## 6 4.61 ctrl

Reminder: To use R terminology...the column “group” is called a factor and the different categories (“ctr”, “trt1”, “trt2”) are named factor levels. The levels are ordered alphabetically.

You can instruct R to directly to show you the different factor levels that exist:

# Show the levels
levels(my\_data$group)

## [1] "ctrl" "trt1" "trt2"

**Exercise 4: Create a summary statistic to see what the central measures by group are.**

Note: As you might notice, it’s possible to compute summary statistics (mean and sd) by groups using the dplyr package. You have to import this package again (see workshop 2 for the code).

library(dplyr) # import library

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
## filter, lag

## The following objects are masked from 'package:base':
##
## intersect, setdiff, setequal, union

summarise(group\_by(my\_data1, group),
 count = n(),
 mean = mean(weight, na.rm = TRUE),
 sd = sd(weight, na.rm = TRUE)
 )

## # A tibble: 3 x 4
## group count mean sd
## <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

### Visualise your data

**Exercise 5: Create a plot where you can see the distribution of the data by group.**

boxplot(my\_data$weight ~ my\_data$group,
 col = c("red", "blue","yellow"),
 ylab = 'weight',
 xlab = 'groups',
 main = 'formula = weight ~ groups')



If you have created a boxplot, you might notice that the groups are quite different from each other. But how much different from each other? Can we quantify it? Is there any significant difference between the average weights of plants in the 3 experimental conditions?

###

### Compute one-way ANOVA test

To answer this question, we’ll have to take as briefly discussed two general steps.

1. Create an ANOVA object with aov. (see below). Because weight is the dependent variable and group is the independent variable, we’ll set the formula to = weight ~ group.

# Step 1: aov object (plant.aov) that stores the results of aov

plant.aov <- aov(formula = weight ~ group,
 data = my\_data)

1. The function summary.aov() is used to summarize the analysis of variance (ANOVA) model. To see a full ANOVA summary table of the ANOVA object, apply the summary() to the ANOVA object from Step 1.

# Step 2: Look at the summary of the anova object
summary(plant.aov)

## 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

The output includes the columns F value and Pr(>F) corresponding to the p-value of the test.

**Interpretation**

The main result from our table is that we have a significant effect of groups on plant weight (F(2, 27) = 4.846, p = 0.016 < 0.05\*). However, the ANOVA table does not tell us which levels of the independent variable differ. In other words, we do not know which group does better or worse than which.

To answer this, we need to conduct a post-hoc test (something we won’t cover today). To learn more about the logic behind different post-hoc tests and how to do it in R, check out the website page here: <https://rpubs.com/aaronsc32/post-hoc-analysis-tukey>

**Exercise 6: With our descriptive dataset, can you run a ANOVA test to determine whether or not there are age mean differences between the tutor variable? Remember our steps:**

1. Create an ANOVA object with aov. (see below).

# Step 1: aov object (plant.aov) that stores the results of aov

age.aov <- aov(formula = age ~ tutor,
 data = my\_data)

1. The function summary.aov() is used to summarize the analysis of variance (ANOVA) model. To see a full ANOVA summary table of the ANOVA object, apply the summary() to the ANOVA object from Step 1.

# Step 2: Look at the summary of the anova object
summary(age.aov)

##

mydata.aov<- aov(formula =tutor~age,
                 data = mydata)
summarise(mydata.aov)

## Chi-square: chsq.test()

In workshop 3 you have been introduced to correlation test. Contrary to correlation, the Chi-Square test of independence is used to determine if there is a significant relationship between two nominal (categorical) variables.

To learn more about how the test works and how to do it by hand, I invite you to read the article [“Chi-square test of independence by hand”](https://www.statsandr.com/blog/chi-square-test-of-independence-by-hand/).

To briefly recap what has been said in that article, the Chi-square test of independence tests whether there is a relationship between two categorical variables.

## Hypotheses:

Let’s say that a researcher wants to examine the relationship between gender (male vs. female) and empathy (high vs. low). The chi-square test of independence can be used to examine this hypothetical relationship.

Then, the formal null and alternative hypotheses would be:

* ***null hypothesis***: there is **no** relationship between gender and empathy.
* ***alternative hypothesis***: there is a relationship between gender and empathy (e.g. there are more high-empathy females than high-empathy males).

In other words, the *Null hypothesis* assumes that there is no association between the two variables whereas, the *alternative hypothesis* assumes that there is an association between the two variables.

A key difference between the chisq.test() and the other hypothesis tests that we’ve covered so far, is that chisq.test() requires a table to be created as its main argument, using the table() function. Basically, the table() creates a **contingency table** where each row represents a category for one variable and each column represents a category for the other variable.

Simply speaking, the Chi-square test of independence works by comparing the observed frequencies (so the frequencies observed in your sample) to the expected frequencies if there was no relationship between the two categorical variables (so the expected frequencies if the null hypothesis was true).

## Example

In this section, we will use the descriptive dataset that we used in previous workshops as well. Import it into R and call it mydat. See beginning of this document on where to download.

**Preprocessing steps**

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'))

Let’s check again the type of data:

# checking data

str(mydat)

## tibble [69 x 8] (S3: tbl\_df/tbl/data.frame)
## $ group : num [1:69] 1 1 1 2 2 2 2 2 3 3 ...
## $ tutor : chr [1:69] "unknown" "unknown" "unknown" "unknown" ...
## $ year : num [1:69] 2019 2019 2019 2019 2019 ...
## $ age : num [1:69] 19 23 24 18 20 26 27 33 18 19 ...
## $ length: num [1:69] 171 173 165 170 160 170 180 182 157 153 ...
## $ favPet: Factor w/ 2 levels "cat","dog": 2 2 2 2 2 2 2 2 1 1 ...
## $ GoT : num [1:69] NA 8 NA NA 7 8 NA NA NA 3 ...
## $ LotR : num [1:69] 6 7 1 NA NA 7 8 9 NA 1 ...

Since we only have one categorical data (factor) and Chi-square test requires two categorical variables, lets convert for instance tutor variable into a categorical one:

# converting to factor tutor

mydat$tutor <- factor(mydat$tutor, levels = c('unknown',
 'Koen',
 'Khrystyna',
 'Britt',
 'Anouk',
 'Daniëlle',
 'Hidde',
 'PEERS'))

# Exercise 6: Can you please check the levels of the tutor variable?

levels(mydat$tutor)

## [1] "unknown" "Koen" "Khrystyna" "Britt" "Anouk" "Daniëlle"
## [7] "Hidde" "PEERS"

We now create a contingency table of the two variables e.g. favpet and tutor (remember, we have done this before in workshop 2):

# create the contingency table

table(mydat$tutor, # first variable
 mydat$favPet) # second variable

##
## cat dog
## unknown 8 19
## Koen 3 3
## Khrystyna 0 5
## Britt 1 5
## Anouk 4 2
## Daniëlle 1 3
## Hidde 3 7
## PEERS 1 2

The contingency table gives the observed number of cases in each subgroup.

It is also a good practice to draw a barplot to visually represent the data. You can use the barplot function learned in workshop 2. But if you wish to make a more advanced one, then use ggplot2 library. It has more advance functionalities. See below:

library(ggplot2) # import library

# note...if you haven’t installed the ggplot2 package before, then you first have to run install.packages(“ggplot2”) before running library(ggplot2).

ggplot(mydat) +
 aes(x = favPet, fill = tutor) +
 geom\_bar() +
 scale\_fill\_hue() +
 theme\_minimal()



It is worth to notice that we have missing data (NA) for FavPet. Although R does not include them in the analysis, the barplot() function does, namely as a separate bar. In order to solve it and create a meaningful plot, we can use na.omit to remove them from your graph and create the plot again (note this only removes them from the barplot; NOT the entire dataset):

library(ggplot2)
ggplot(na.omit(mydat)) +
 aes(x = favPet, fill = tutor) +
 geom\_bar() +
 scale\_fill\_hue() +
 theme\_minimal()



<https://www.dummies.com/programming/r/how-to-remove-rows-with-missing-data-in-r/>

## Chi-square test

If you want to see if the frequency of one nominal variable depends significantly on a second nominal variable, you’d conduct a chi-square test. For our example, we want to determine whether is a statistically significant association between being a student from one tutor group and chosen favorite pet.

To conduct a chi-square test on these data, we will create a contingency table (observed frequencies) with two data vectors as we have carried out before (tutor and favPet):

# create the contingency table – observed frequencies

table(mydat$tutor, # first variable
 mydat$favPet) # second variable

##
## cat dog
## unknown 8 19
## Koen 3 3
## Khrystyna 0 5
## Britt 1 5
## Anouk 4 2
## Daniëlle 1 3
## Hidde 3 7
## PEERS 1 2

The contingency table gives the observed number of cases in each subgroup.

Then, we are going to run the chisq.test() as following:

# Create an object called mydat.cstest -> chisq.test()

mydat.cstest <- chisq.test(x = table(mydat$tutor, # first col
 mydat$favPet)) # second col

## Warning in chisq.test(x = table(mydat$tutor, mydat$favPet)): Chi-squared
## approximation may be incorrect

\*Note that the warning message found is due to the small number of observed frequencies (e.g. only 4 observation for Danniele tutor).

Due to this warning message, here we have a break. We start again, there exists two different types of independence test:

* the Chi-square test
* the Fisher’s exact test

The Chi-square test is used when the sample is large enough. On the other hand, the Fisher’s exact test is used when the sample is small. Some authors suggest that the *“usual rule for deciding whether the χ2 approximation is good enough is that the Chi-square test is not appropriate when the* ***expected*** *values in one of the cells of the contingency table is less than 5, and in this case the Fisher’s exact test is preferred”* (McCrum-Gardner 2008; Bower 2003).

Let’s then try to get the expected values from R. You can do it with the following R command:

mydat.cstest$expected # get frequency table

##
## cat dog
## unknown 8.4626866 18.537313
## Koen 1.8805970 4.119403
## Khrystyna 1.5671642 3.432836
## Britt 1.8805970 4.119403
## Anouk 1.8805970 4.119403
## Daniëlle 1.2537313 2.746269
## Hidde 3.1343284 6.865672
## PEERS 0.9402985 2.059701

The expected frequency table above confirms that we should use the Fisher’s exact test instead of the Chi-square test because there is at least one cell below 5.

Note: although it is a good practice to check the expected frequencies **before** deciding between the Chi-square and the Fisher test, it is not a big issue if you forget. As you might notice above, when performing the Chi-square test in R (with chisq.test()), a warning message as “Chi-squared approximation may be incorrect” has appeared. This warning means that the smallest expected frequencies is lower than 5. Therefore, do not worry if you forgot to check the expected frequencies before applying the appropriate test to your data, R will warn you that you should use the Fisher’s exact test instead of the Chi-square test if that is the case.

Fisher’s exact test in R:

To perform the Fisher’s exact test in R, use the fisher.test() function as you would do for the Chi-square test:

# Create an object called mydat.ftest -> fisher.test()
mydat.ftest <- fisher.test(x = table(mydat$tutor,
 mydat$favPet))

# Call the object to see results
mydat.ftest

##
## Fisher's Exact Test for Count Data
##
## data: table(mydat$tutor, mydat$favPet)
## p-value = 0.4079
## alternative hypothesis: two.sided

**Interpretation**

We got a p-value of 0.4079. At the traditional p = .05 threshold for significance, we would conclude that we fail to reject the null hypothesis and state that we do not have enough information to determine if students from different tutor groups differ in how likely they are to prefer cats or pets.

You have seen only one way to handle small dataset for chi square using Fisher exact t test. There is a lot of statistical solution to sort it out (i.e. MonteCarlo Simulation to estimate p-values: <https://astrostatistics.psu.edu/su07/R/html/stats/html/chisq.test.html>) or easier: combine some columns of the contingency table and rerun the test with chisq.test function(<http://www.r-tutor.com/elementary-statistics/goodness-fit/chi-squared-test-independence>)

### Example with Student Survey Data

Import the dataset (survey.data). This data contains 237 obs and 12 variables (+1 for id in case you import the excel file). We want to determine whether or not there is any existing relationship between student smoke and exercise habit. In other words, is there any association between smoke and exercise habit between the student survey?

After you imported the data (in case you used import excel file, be sure to convert to categorical – factor – the variables ‘smoke’ and ‘exercise’), you can quickly check the structure of your data. Remember, we are only interested on the variable ‘Exer’ and ‘Smoke’. You can already start thing about this code: ` survey.data$Exer` and ` survey.data$Smoke`

### Exercise 7: Can you check if the variables Smoke and Exer from the survey.data are categorical (factor) and if yes, which values/categories do they have?

str(survey.data)

## 'data.frame': 237 obs. of 3 variables:
## $ gender: Factor w/ 2 levels "Female","Male": 1 2 2 2 2 1 2 1 2 2 ...
## $ Exer : Factor w/ 3 levels "Freq","None",..: 3 2 2 2 3 3 1 1 3 3 ...
## $ Smoke : Factor w/ 4 levels "Heavy","Never",..: 2 4 3 2 2 2 2 2 2 2 ...

levels(survey.data$Exer)

levels(survey.data$Smoke)

Great, Exer and Smoke are the only variables that we are interested. We can see that the allowed values in Smoke are “Heavy”, “Regul” (regularly), “Occas” (occasionally) and “Never”. As for Exer, they are “Freq” (frequently), “Some” and “None”.

**Exercise 8: Create (a) a contingency table for these two variables and (b) a barplot that displays both variables. If you find NA values, remember you can use na.omit()**

table(survey.data$Exer,
 survey.data$Smoke)

##
## Heavy Never Occas Regul
## Freq 7 87 12 9
## None 1 18 3 1
## Some 3 84 4 7

ggplot(survey.data) +
 aes(x = Exer, fill = Smoke) +
 geom\_bar() +
 scale\_fill\_hue() +
 theme\_minimal()



**Exercise 9: Test the hypothesis whether the students smoking habit is associated with their exercise level at .05 significance level. Create a chisq.test() with the previous contingency table. Be aware that records within the classes are so small and it might return a warning message as explained before. You can fix it by using fisher.test() instead. Interpret the output**

chisq.test(x = table(survey.data$Exer,
 survey.data$Smoke))

## Warning in chisq.test(x = table(smoke$Exer, smoke$Smoke)): Chi-squared
## approximation may be incorrect

##
## Pearson's Chi-squared test
##
## data: table(smoke$Exer, smoke$Smoke)
## X-squared = 5.4885, df = 6, p-value = 0.4828

chisq.test(x = table(smoke$Exer,
 smoke$Smoke),
 simulate.p.value = TRUE)

##
## Pearson's Chi-squared test with simulated p-value (based on 2000
## replicates)
##
## data: table(smoke$Exer, smoke$Smoke)
## X-squared = 5.4885, df = NA, p-value = 0.4838

**Enhance the solution**

Coming back to the warning message of approximate p value result, we have practiced how to solve it using the Fisher test. As we mentioned previously, there are (you can’t imagine) a lot of ways to do it but you are going to learn a new way: combining the columns. This solution will help you a lot to be much more familiar with R and its function to properly manipulate and organize your data:

# create a contigency table with Smoke and Exer
table = table(survey.data1$Smoke, survey.data1$Exer)

# combine rows and columns with cbind() function
ctable = cbind(table[,"Freq"], # first column
 table[,"None"] + table[,"Some"]) # second column combine

chisq.test(ctable)

##
## Pearson's Chi-squared test
##
## data: ctable
## X-squared = 3.2328, df = 3, p-value = 0.3571