# Computational Inference with R - Assignment 3

Jannick Akkermans & Lauke Stoel

13 november 2020

## Part 1: empirical power calculation

You have the following data: two groups, an experimental and a control group, of 50 participants each, that were randomly assigned to two conditions. The values of interest are the test scores on an exam for both groups. The participants in the experimental grouped received a special training on learning strategies prior to an exam. Based on past research you expect that the control group will have a mean of 150 on the test with standard deviation of 15 and that the experimental group will have a mean on the test scores that is 10 points higher than the control group, with the same standard deviation. You plan to perform a t-test on the data and you would like to know the power of the test. You will estimate the power in two ways: using the in-built R function power.t.test and by simulation.

### 1. Assuming that the data are approximately normal, simulate one sample of data and perform a t-test using the R in-built t-test function. What is the p-value of the test? What is your conclusion?

Our null hypothesis is that there is no difference between the control group and the experimental group. Based on the past research mentioned above, we formulate our alternative hypothesis to be that the experimental group scores higher on average than the control group. Since this is a directional hypothesis, we will perform a one-sided t-test using the built-in `t.test()` function. we choose to test against an alpha level of 0.05.

```
set.seed(123) #set a seed so that the results will be the same every time
control_group <- rnorm(50, mean = 150, sd = 15) #sample 50 observations from
a normal distribution with a mean of 150 and a standard deviation of 15 to be
the control group
experimental_group <- rnorm(50, mean = 160, sd = 15) #sample 50 observations
from a normal distribution with a mean of 160 and a standard deviation of 15
to be the experimental group
t.test(control_group, experimental_group, alternative = "less") #perform t-
test with built-in t-test function

##
##  Welch Two Sample t-test
##
## data:  control_group and experimental_group
## t = -4.2517, df = 97.951, p-value = 2.425e-05
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
```

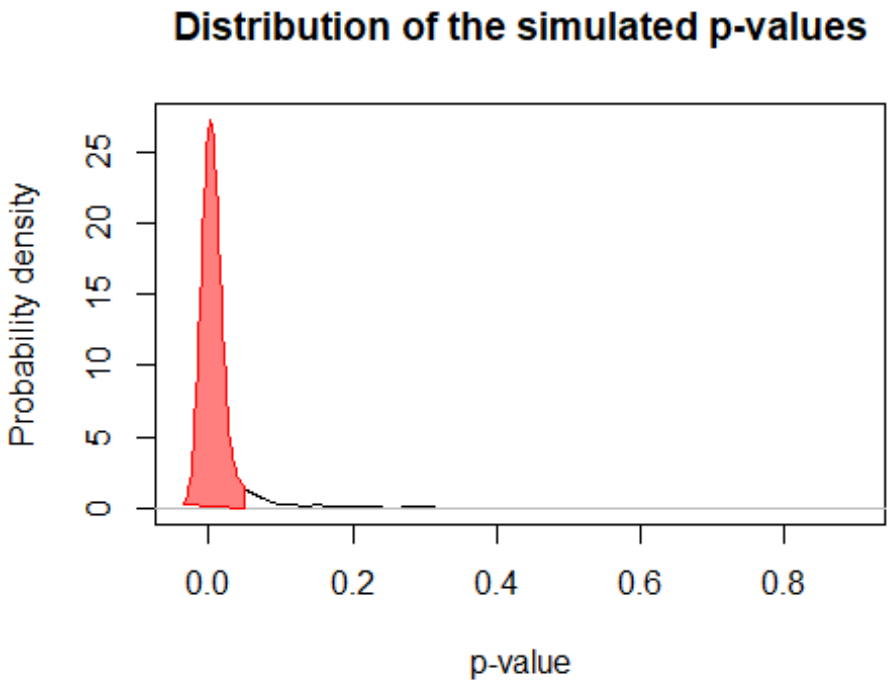
```
##       -Inf -7.118272
## sample estimates:
## mean of x mean of y
##  150.5161  162.1961
```

The p-value from this test is 2.425e-05. Since this is lower than the alpha-level of 0.05, we reject the null hypothesis. Therefore, there is statistical evidence supporting the alternative hypothesis that the experimental group scores higher on average than the control group. This evidence would support the claim that the special training on learning strategies is effective.

**2. Now you will approximate the power by simulation. Assume that the data are normally distributed, write a function that generates data and performs a t-test 1000 times and that stores the values of the t-statistic (or the p-value) in a vector. Thinking about the definition of power in terms of rejecting the null hypothesis, how would you obtain an estimate of the power in this situation? Compare your results with those given by power.t.test.**

```r
library(ggplot2)
set.seed(123) #set a seed so that the results will be the same every time the code block is run
N <- 1000 #set the number of bootstrap replications

simulateTtest <- function(N, n1, n2, mu1, mu2, sd1, sd2) {
  p_values <- numeric(N) #create a vector of 0's whose length is equal to N

  for (i in 1:N) { #simulate 1000 t-tests
    control_group <- rnorm(n1, mean = mu1, sd = sd1) #sample the control group
    experimental_group <- rnorm(n2, mean = mu2, sd = sd2) #sample the experimental group
    p_values[i] <- t.test(control_group, experimental_group)$p.value #perform a t-test for each iteration and save the p-value
  }

  dx <- density(p_values, adjust = 10) #estimate the probability density function of our p-values
  plot(dx, xlab = "p-value", ylab = "Probability density", main = "Distribution of the simulated p-values"); polygon(c(dx$x[dx$x < 0.05], 0.05), c(dx$y[dx$x < 0.05], 0.05), col = rgb(1, 0, 0, alpha = 0.5), border = "red", main = "") #create a density plot that shows the distribution of the p-values and color the area that is lower than 0.05

  Sim_power <- mean(p_values < 0.05) #calculate the empirical power based on all the bootstrapped p-values
  True_power = power.t.test(50, delta = 10, sd = 15, sig.level = 0.05, type = "two.sample", alternative = "one.sided")$power
  mean1 <- mean(control_group) #compute the mean of both groups
```

```
  mean2 <- mean(experimental_group)

  bias <- Sim_power - True_power #calculate the bias of our power estimate
with respect to  the true value

  list("Mean control group" = mean1, "Mean experimental group" = mean2,
"Simulated power" = Sim_power, "True power" = True_power, "Bias" = bias)
#return a list that contains the means of both groups, simulated power and
the true power
}


simulateTtest(N, 50, 50, 150, 160, 15, 15)
```




```
## $`Mean control group`
## [1] 150.2909
##
## $`Mean experimental group`
## [1] 162.7105
##
## $`Simulated power`
## [1] 0.929
##
## $`True power`
## [1] 0.952077
##
```

```
## $Bias
## [1] -0.02307696
```

Power is defined as the probability of correctly rejecting the null hypothesis. In this exercise, a simulated value for power is calculated as the proportion of p-values lower than 0.05. The area that is associated with this proportion can be seen in the plot above, where the area with p-values lower than 0.05 is coloured red. The simulation returned a power value of 0.929. The function `power.t.test()` returns a power value of 0.952 which is about two hundredths higher than our simulated power value. This is most likely due to some correction applied in the `power.t.test()` function to account for type II errors. In this question, we are asked to define power in terms of rejecting the null-hypothesis, i.e. 1-alpha. This means that in our simulation, every p-value lower than 0.05 contributes to the value of the simulated power. However, in actuality, power is defined as `correctly` rejecting, i.e. 1-beta. This means that in our current estimate, there are likely also be some Type-II error cases included in this simulated power value. We think the difference between our function and the `power.t.test()` function arises because the `power.t.test()` function accounts for this.

## Part 2: Resampling techniques

In this exercise you will use resampling techniques on the Flight instruction data, used in Assignment 2. In Assignment 2 you programmed your own t-test function. The significance of the t-test was evaluated by using the theoretical t-distribution. In this exercise you are asked to use resampling techniques to perform statistical inference.

### 1. Choose an appropriate resampling technique and make a function that performs a hypothesis test based on the t-statistic produced by the standard t-test function. Use the in-built `sample()` function to perform the resampling of the data. Include relevant statistics in your function and give the function a clear structure (choose sensible input arguments, organize the output in an efficient way, add comments, etc). Show the results of your resampling technique on the Flight instruction data.

The null-hypothesis we intend to test is “there is no difference in the means between the tfi and csfi groups”. The alternative hypothesis is formulated as “the means of the tfi and csfi groups differ from each other”.

First, to determine which resampling technique for hypothesis testing we should use, we should check the data for outliers. Bootstrapping does not rely on assumptions of an underlying distribution, but is more sensitive to outliers than the permutation test. Additionally, it is less exact than the permutation test when you have small samples like in this question. Nevertheless, bootstrapping does have better generalisability than the permutation test.

```
par(mfrow = c(1,2)) #define parameter for plots
```

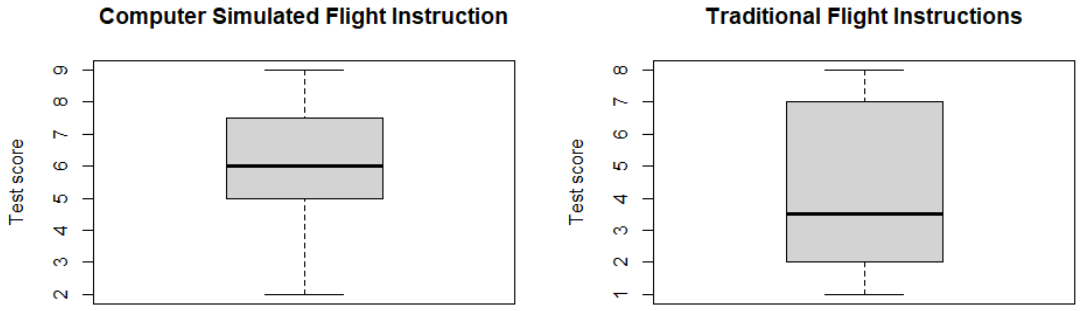
```r
csfi <- c(2,5,5,6,6,7,8,9) #data from the computer-simulated flight instructions group
tfi <- c(1,1,2,3,3,4,5,7,7,8) #data from the traditional flight instructions group

var(csfi) #calculate the variance of the computer-simulated flight instructions group -> 4.57
```

```
## [1] 4.571429
```

```r
var(tfi) #calculate the variance of the traditional flight instructions group -> 6.54
```

```
## [1] 6.544444
```

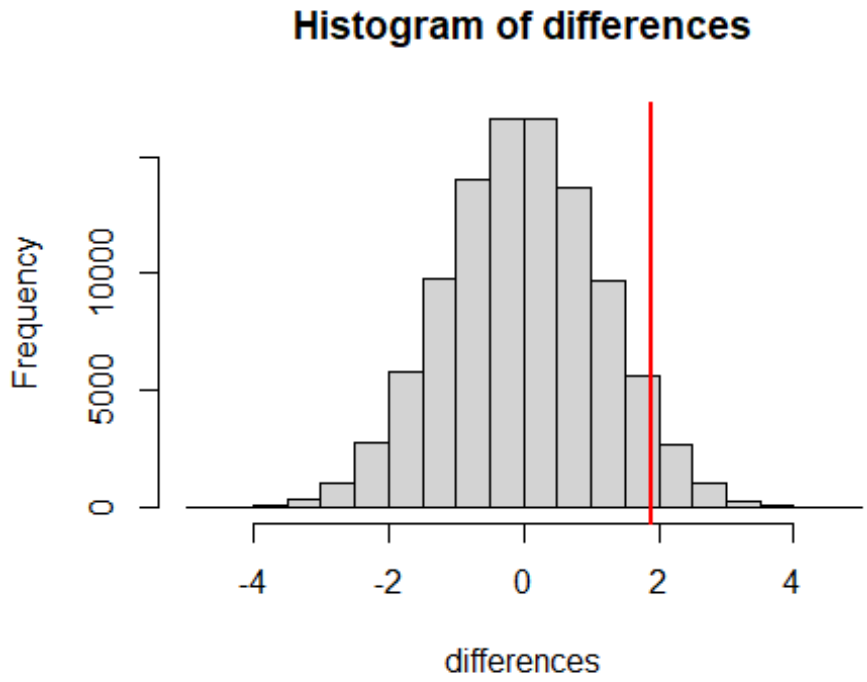
```r
boxplot(csfi, main = "Computer Simulated Flight Instruction", ylab = "Test score"); boxplot(tfi, main = "Traditional Flight Instructions", ylab = "Test score") #create boxplots for both groups. These are used to check outliers
```




There are no outliers present in the data as can be seen in the boxplots. Therefore, we decided to use bootstrapping in light of the better generalisability.

```r
bootstrapTest <- function(group1, group2, B) {
  sample_difference <- mean(group1) - mean(group2) #The difference between the two groups as they are now. This difference is used to calculate the empirical p-value
  N <- length(group1) #size of group 1
  M <- length(group2) #size of group 2
  differences <- numeric(B) #create a vector with 0's with a length equal to the number of repetitions

  for (i in 1:B) {
    bootstrap_sample <- sample(c(group1, group2), replace = TRUE) #sample WITH replacement from the original sample
    tgroup1 <- bootstrap_sample[1:N] #assign the first N sampled observations to the first group
    tgroup2 <- bootstrap_sample[(N+1):(N+M)] #assign the remaining sampled
```

```
observations to the second group
    differences[i] <- mean(tgroup1) - mean(tgroup2) #calculate the mean
difference between the groups and assign it to the differences vector
  }

  emp_p <- mean(abs(differences) > abs(sample_difference)) #calculate the
proportion of absolute mean differences that is greater than the sample mean
differences. We use the absolute value since we want to calculate a two-sided
p-value.
  hist(differences); abline(v = sample_difference, lwd = 2, col = "red")
#create a histogram of the distribution of the mean differences. Draw a red
vertical line at the position of the observed mean difference
  list("Sample difference" = sample_difference, "Distribution mean" =
mean(differences), "Distribution sd"= sd(differences), "P-value" = emp_p)
#return a list with the observed mean difference, the mean of the bootstrap
distribution, the standard deviation of the bootstrap distribution and the
empirical p-value
}

bootstrapTest(csfi, tfi, 100000)
```




```
## $`Sample difference`
## [1] 1.9
##
## $`Distribution mean`
## [1] 0.0019235
##
```

```
## $`Distribution sd`
## [1] 1.154938
##
## $`P-value`
## [1] 0.09818
```

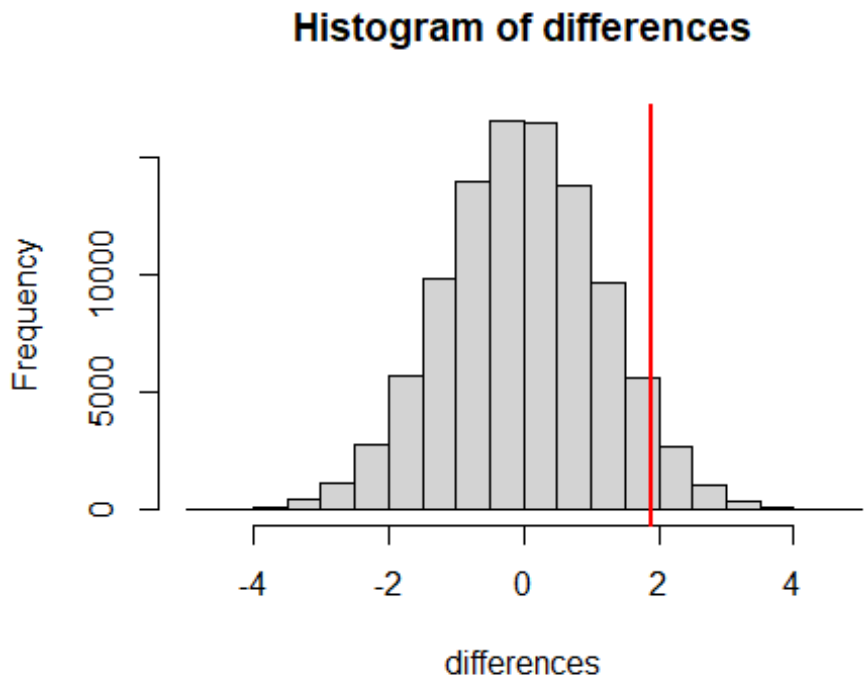
The difference between the two sample means is 1.9. After bootstrapped resampling, the difference in means is -0.005, which is very close to 0, and a standard deviation of 1.155. These data tell us that this distribution mimics the standard normal distribution. The p-value reported is 0.099, indicating to us that there is no statistical evidence to support the alternative hypothesis that the means of these two groups differ from each other significantly. In the context of the example data, our conclusion would be that the computer simulated flight instructions are not a significant improvement upon the traditional flight instruction method.

### 2. How would you program the drawing of random samples when you do not want to use the `sample()` function? Modify the function you have programmed in (1) in such a way that it does not use the `sample()` function anymore (hint: you will have to work with indices). Run your new function and show that you can obtain the same results as in (1).

When you do not want to use the `sample()` function to draw random samples, you could instead generate n random integers. These integers represent the cases from the combined sample that need to be in the random sample. This can be achieved by using the function `runif()` combined with the `floor()` function. This will generate n random integers, possibly containing some duplicates. Therefore, this procedure essentially mimics the bootstrapping approach.

```
bootstrapTestnosample <- function(group1, group2, B) {
  sample_difference <- mean(group1) - mean(group2) #The difference between
the two groups as they are now. This difference is used to calculate the
empirical p-value
  N <- length(group1) #size of group 1
  M <- length(group2) #size of group 2
  differences <- numeric(B) #create a vector with 0's with a length equal to
the number of repetitions
  combined_sample <- c(group1, group2) #combine the data of both groups into
one group

  for (i in 1:B) {
    indices <- floor(runif(18, min=1, max=19)) #generate 18 random integers
representing the indices of the observations to be selected
    bootstrap_sample <- combined_sample[indices] #sample the observations
based on the randomly generated indices
    tgroup1 <- bootstrap_sample[1:N] #assign the first N sampled units to the
first group
    tgroup2 <- bootstrap_sample[(N+1):(N+M)] #assign the remaining sampled
units to the second group
```

```
    differences[i] <- mean(tgroup1) - mean(tgroup2) #calculate the mean
difference between the groups and assign it to the differences vector
  }

  emp_p <- mean(abs(differences) > abs(sample_difference)) #calculate the
proportion of absolute mean differences that is greater than the sample mean
differences. We use the absolute value since we want to calculate a two-sided
p-value.
  hist(differences); abline(v = sample_difference, lwd = 2, col = "red")
#create a histogram of the distribution of the mean differences. Draw a red
vertical line at the position of the observed mean difference
  list("Sample difference" = sample_difference, "Distribution mean" =
mean(differences), "Distribution sd"= sd(differences), "P-value" = emp_p)
#return a list with the observed mean difference, the mean of the bootstrap
distribution, the standard deviation of the bootstrap distribution and the
empirical p-value
}

bootstrapTestnosample(csfi, tfi, 100000)
```




```
## $`Sample difference`
## [1] 1.9
##
## $`Distribution mean`
## [1] 0.0015955
##
## $`Distribution sd`
```

```
## [1] 1.160572
##
## $`P-value`
## [1] 0.09936
```

The results from the approach that did not use the `sample()` function are very close to the results from the approach that did use the `sample()` function. The slight difference in the two sets of results are possibly due to a difference in the distributions the `sample()` and `runif()` functions sample from. The `runif()` function samples from a uniform distribution, whereas the `sample()` function samples from the underlying distribution of the data at hand. This difference becomes negligible as the number of iterations tends to infinity.

### 3. Which resampling technique(s) would you use, if you would be interested in estimation instead of hypothesis testing? Make an R function that performs resampling for estimation purposes using the mean difference between two groups from the t-test function. Your function should produce all relevant aspects of estimation, and you should discuss these aspects and put them in perspective. Again use the Flight instruction data to show the results of your function.

When talking about resampling methods for estimation, there are three options to choose from: bootstrapping, the jackknife approach, and the jackknife-after-bootstrap approach. Bootstrapping generally works well but can become computationally heavy. Jackknife produces lower bias, but is not useful for non-smooth statistics. According to the book by Rizzo, jackknife-after-bootstrap produces lowest standard error.

Generally, bootstrap methods are often used when the sample data is the only available data. This is the case for this question. Additionally, bootstrap methods can be used to calculate the bias in the estimation and to provide an estimate of the standard error of the bootstrap distribution. The jackknife-after-bootstrap can only provide an estimate of the standard error of it's distribution. Since we believe bias is a relevant aspect of estimation, we decided to use bootstrapping as a resampling technique for estimation. In addition, we will provide a 95% confidence interval of the bootstrap distribution.

```
set.seed(123)
bootstrapEstimation <- function(csfi, tfi, B) {
  sample_difference <- mean(csfi) - mean(tfi) #calculate the observed mean
difference between the groups
  differences <- numeric(B) #create vector of zeros equal in length to the
number of replications

  for (i in 1:B) {
    csfi_sample <- sample(csfi, replace = TRUE) #sample with replacement from
the experimental group
    tfi_sample <- sample(tfi, replace = TRUE) #sample with replacement from
the control group
    differences[i] <- mean(csfi_sample) - mean(tfi_sample) #calculate the
```

```
mean difference between the groups and store it inside the differences vector
  }

  bias <- mean(differences - sample_difference) #calculate the bias of the
estimate
  se <- sd(differences) #calculate the standard error of the bootstrap
distribution
  list("Estimated mean difference" = mean(differences), "Bias" = bias,
"Standard Error" = se) #return a list that contains the mean of the bootstrap
distribution, the bias of the estimated mean difference, and the standard
error of the bootstrap distribution
}

estimates <- bootstrapEstimation(csfi, tfi, 100000)
estimates
```

```
## $`Estimated mean difference`
## [1] 1.895916
##
## $Bias
## [1] -0.0040845
##
## $`Standard Error`
## [1] 1.041127
```

```
mean(csfi) - mean(tfi) #the observed mean difference between the control
group and the experimental group, which is 1.9
```

```
## [1] 1.9
```

```
ci_lb <- estimates[[1]] - 1.96 * estimates[[3]] #calculate the lower bound of
the 95% confidence interval
ci_ub <- estimates[[1]] + 1.96 * estimates[[3]] #calculate the upper bound of
the 95% confidence interval

confidence_interval <- c(ci_lb, ci_ub) #create the 95% confidence interval of
the mean difference between the computer-simulated flight instructions group
and the traditional flight instructions group
confidence_interval
```

```
## [1] -0.1446927  3.9365237
```

Our bootstrap estimation produced a bias of -0.004. This means that there is a difference of 0.004 between our estimate and the true mean difference between the two groups. Since this value is very close to zero, it means that our approach almost provided a unbiased estimate.

The standard error is a measure of the dispersion of sample means around the population mean. Our approach yielded a standard error of 1.04, which is a reasonable value. It gives us an indication that if we would sample all possible samples from our data and calculate

the mean difference for all those samples, their distribution would likely behave like a standard normal distribution.

Finally, we provided a 95% confidence interval of our bootstrap distribution. The 95% confidence interval ranges from -0.14 to 3.95. This means that if we sampled all possible samples from our data, we would expect that 95% of the true differences falls between these values. However, the 95% confidence interval contains 0, indicating that the true difference between the means could be 0. This leads us to conclude once more that there is no statistical evidence supporting the alternative hypothesis that the means of these two groups differ from each other significantly.

All in all, both methods would lead us to draw the same conclusion about the absence of evidence to support the alternative hypothesis.