Assignment 7

EMATM0061: Statistical Computing and Empirical Methods, TB1, 2024

Introduction

This is the 7th assignment for Statistical Computing and Empirical Methods. This assignment is mainly based on Lectures 18, 19, and 20 (see the Blackboard).

You can *optionally* submit this assignment by 13:00 Monday 11th November. This will help us understand your work but will *not* count towards your final grade. The submission point can be found under the assignment tab at Blackboards (click the title “Assignment 07” and upload a pdf file).

Load packages

Some of the questions in this assignment require the tidyverse package. If it hasn’t been installed on your computer, please use “install.packages()” to install them first.

To load the tidyverse package:

```r
library(tidyverse)
```

1. One sample hypothesis testing

1.1 One sample t-test on penguins data

In this question, we will apply one sample t-test to the population mean of the Adelie penguin’s bill lengths.

(Q1)

Begin by loading the “Palmer penguins” library. Next extract a vector called ““bill_adelie”” consisting of the bill lengths of the Adelie penguins belonging to the Adelie species.

Carry out a statistical hypothesis test to test the hypothesis that the population mean of the Adelie penguin’s bill lengths is 40 mm. Use a significance level of 0.01. You can use the “t.test()” function. What assumptions are required for this hypothesis test?

1.2 Implementing a one-sample t-test

(Q1)

Implement a function that carries out a two-sided one-sample t-test. Your sample should take in two arguments

1) a vector “x” corresponding to a sample $X_1, \cdots, X_n \in \mathcal{N}(\mu, \sigma^2)$ and
2) the value μ_0 corresponding to a null hypothesis of $\mu = \mu_0$.

The output of your function should be the corresponding p-value of the test.

You can test your implementation by confirming your function gives the same p-value as the “t.test()” function for the example in question 1.1(Q1) of the assignment.

2. paired t-test and effect size

The Barley data set gives the yields of two types of barley - Glabron and Velvet across twelve different fields. The data is paired as yields are given for both types of barley across each of the twelve fields. Install the package called “PairedData” first (if you haven’t done so) and then load the data:

```
library(PairedData) # you might need to install the package first
data("Barley")
detach('package:PairedData', unload=TRUE)
detach('package:MASS', unload=TRUE)
# unload package because it contains another select() function

head(Barley, 4)

##   Farm Glabron Velvet
## 1  F01      49     42
## 2  F02      47     47
## 3  F03      39     38
## 4  F04      37     32
```

(Q1)

Carry out a paired t-test to determine whether there is a difference in average yield between the two types of barley. Use a significance level of 0.01. You can use the t.test() function.

(Q2)

Compute the effect size using Cohen’s d statistic.

(Q3)

What assumptions are required for the paired t-test? Are these assumptions justified in this case?

3. Implementing unpaired t-test

In this question the goal is to create a function called “t_test_function()” which implements an unpaired Student’s t-test, in order to test for a difference of population means between two unpaired samples from two distributions. Your function will play a similar role to the following standard R function:

```
t.test(body_mass_g~species, data=peng_AC,var.equal = TRUE)
```

Begin by creating a data frame called ““peng_AC”” which is a subset of the Palmer penguins data set consisting of all those penguins which belong to either the ““Adelie”” or the ““Chinstrap”” species of penguins.

```
library(palmerpenguins)
peng_AC<-penguins %>%
  drop_na(species,body_mass_g) %>%
  filter(species !="Gentoo")
head(peng_AC %>% select(species, flipper_length_mm, body_mass_g), 5)

## # A tibble: 5 × 3
##   species flipper_length_mm body_mass_g
##   <fct>               <int>       <int>
## 1 Adelie                181        3750
## 2 Adelie                186        3800
## 3 Adelie                195        3250
## 4 Adelie                193        3450
## 5 Adelie                190        3650
```

(Q1)

First, try to understand what the following piece of code does.

```
val_col <- "body_mass_g"
group_col <- "species"
data <- peng_AC

data_new <- data %>%
  # rename the columns; note that you can not drop the "!!" (why?)
  rename(group=(!!group_col),val=(!!val_col))%>%
  group_by(group) %>%
  drop_na(val) %>%
  summarise(mn=mean(val))

data_new
```

```
## # A tibble: 2 × 2
##   group        mn
##   <fct>     <dbl>
## 1 Adelie    3701.
## 2 Chinstrap 3733.
```

```
data_new$mn[2]
```

```
## [1] 3733.088
```

Now, let’s create the function “t_test_function()”. Your function should take in the following arguments:

1. ““data”” - A data frame argument,
2. ““val_col””- A string argument. This argument is for the column name for a continuous variable (e.g., the body mass column “body_mass_g”),
3. ““group_col”” - A string argument. This argument is for the column name for a binary variable (e.g., the species column “species”).

The function should carry out an unpaired Student’s t test based on the value of the continuous variable in the column whose name is stored in “val_col”:

1. The function should begin by partitioning the sample into two groups based on the value of the binary variable named ““group_col””. For example, suppose that the column ““group_col”” contains entries “Adelie” and “Chinstrap”, then you should partition the sample into two groups corresponding to “Adelie” and “Chinstrap” respectively. You can use the “group_by()” function.
2. Your function should then compute the sample mean, sample variance and sample size for each of these two groups, based upon the variable within the column whose name is stored in ““val_col””.
3. Your function should compute a test statistic for the Student’s unpaired t-test (Lecture 20). In addition, the function should compute the corresponding p-value. Finally, your function should compute an estimate for the effect size.
4. Your function should return a data frame containing the test statistic, p-value and effect size.

Your function should have the following output:

```
t_test_function(data=peng_AC,val_col="body_mass_g",group_col="species")
```

```
##       t_stat effect_size     p_val
## 1 -0.5080869 -0.07420226 0.6119085
```

(Q2)

As an additional challenge you can modify your function so that it takes a fourth argument called ““var_equal”” which takes a Boolean value. If the input of ““var_equal”” is the Boolean “TRUE” your function should compute the test statistic and p-value for an unpaired Student’s t-test. If, on the other hand, the input of ““var_equal”” is the Boolean “FALSE” your function should compute the test statistic and p-value for Welch’s t-test. Your function should have the following output:

```
t_test_function(data=peng_AC,val_col="body_mass_g",group_col="species",
 val_equal=FALSE)
```

You can compare the output of your function with R’s inbuilt “t.test()” function.

4. Useful concepts in statistical hypothesis testing

This question is about the basic concepts in statistical hypothesis testing.

(Q1)

Explain the following concepts. The aim is to get familiar with these definitions. Try to do this yourselves but you may want to go back to the lectures 18 and 19 to find the definitions if needed.

1. Null hypothesis
2. Alternative hypothesis
3. Test statistic
4. Type 1 error
5. Type 2 error
6. The size of a test
7. The power of a test
8. The significance level
9. The p-value
10. Effect size

(Q2)

(1). Is the p-value the probability that the null hypothesis is true?

(2). If I conduct a statistical hypothesis test, and my p-value exceeds the significance level, do I have good evidence that the null hypothesis is true?

5. Investigating test size for an unpaired Student’s t-test

In this question, we shall investigate the performance of an unpaired Student’s t-test from the perspective of test size. Recall a Type 1 error occurs when we reject the null hypothesis when the null hypothesis is true. The size of a test is the probability of a Type 1 error. A key property of valid statistical hypothesis tests with a given significance level is that the size of the test does not exceed the significance level.

Note that we can apply unpaired Student's t-test with significance level alpha on a samples “sample_0”, “sample_1”:

```
t.test(sample_0,sample_1,var.equal = TRUE, conf.level = 1-alpha)
```

We can apply an unpaired Student's t-test and extract the p-value as follows:

```
t.test(sample_0,sample_1,var.equal = TRUE)$p.value
```

Notice that the significance level wasn't supplied as an argument. Is this a problem?

The following code checks the size of an unpaired Student's t-test with a significance level of $\alpha = 0.05$.

```
num_trials<-10000
sample_size<-30
mu_0<-1
mu_1<-1
sigma_0<-3
sigma_1<-3
alpha<-0.05
set.seed(0) # set random seed for reproducibility

single_alpha_test_size_simulation_df <- data.frame(trial=seq(num_trials
)) %>%
  # generate random Gaussian samples
  mutate(sample_0=map(.x=trial,.f=~rnorm(n=sample_size,mean=mu_0,sd=sig
ma_0)),
         sample_1=map(.x=trial,.f=~rnorm(n=sample_size,mean=mu_1,sd=sig
ma_1))) %>%
  # generate p values
  mutate(p_value=pmap(.l=list(trial,sample_0,sample_1),
                      .f=~t.test(..2,..3,var.equal = TRUE)$p.value))%>%
  # type I error
  mutate(type_1_error=p_value<alpha)

single_alpha_test_size_simulation_df %>%
  pull(type_1_error) %>%
  mean() # estimate of coverage probability

## [1] 0.0502
```

Check that you understand the above code.

(Q1)

Modify the above code to explore how the size of the test varies as a function of the significance level α. You might want to use visualization.

6. The statistical power of an unpaired t-test

In this question, we shall investigate the performance of an unpaired Student's t-test from the perspective of statistical power. Recall that the statistical power of a test is the probability of correctly rejecting the null hypothesis when an alternative hypothesis holds.

Consider a setting in which we have two samples i.i.d with Gaussian distribution. The first sample consists of n_0 observations with a population mean μ_0 and population variance σ_0^2. The second sample consists of n_1 observations with a population mean μ_1 and population variance σ_1^2.

The following code checks the statistical power of an unpaired Student's t-test in sample sizes $n_0 = n_1 = 30$, $\mu_0 = 3$, $\mu_1 = 4$, $\sigma_0 = \sigma_1 = 1$ and with a significance level of $\alpha = 0.05$.

```
num_trials<-10000

n_0<-30
n_1<-30
mu_0<-3
mu_1<-4
sigma_0<-2
sigma_1<-2

alpha<-0.05
set.seed(0) # set random seed for reproducibility

data.frame(trial=seq(num_trials)) %>%
  # generate random Gaussian samples
  mutate(sample_0 = map(.x=trial,.f =~ rnorm(n=n_0,mean=mu_0,sd=sigma_0
)),
         sample_1 = map(.x=trial,.f =~ rnorm(n=n_1,mean=mu_1,sd=sigma_1
))) %>%
  # for each sample, generate p value; check examples of pmap() with ?m
ap
  mutate(p_value=pmap(.l = list(trial,sample_0,sample_1),
                      .f =~ t.test(..2, ..3, var.equal = TRUE)$p.value)
) %>%
  # estimate of coverage probability
  mutate(reject_null = p_value<alpha ) %>%
  # extract a column
  pull(reject_null) %>%
  # compute probability
  mean()
```

```
## [1] 0.4862
```

(Q1)

Conduct a simulation study to explore how the statistical power varies as a function of the significance level.

(Q2)

Conduct a simulation study to explore how the statistical power varies as a function of the difference in means $\mu_1 - \mu_0$.

(Q3)

Conduct a simulation study to explore how the statistical power varies as a function of the population standard deviation $\sigma = \sigma_0 = \sigma_1$.

(Q4)

Conduct a simulation study to explore how the statistical power varies as a function of the sample size $n = n_0 = n_1$

7. (*Optional) Comparing the paired and unpaired t-tests on paired data

The aim of this question is to explore the benefits of using a paired test when a natural pairing is available. Consider a situation in which we have two i.i.d. samples $X_1, \cdots, X_n$ and $Y_1, \cdots, Y_n$.

Suppose that $X_1, \cdots, X_n \sim \mathcal{N}(\mu_X, \sigma_X^2)$ and for each $i = 1, \cdots, n$, we have $Y_i = X_i + Z_i$ where $Z_1, \cdots, Z_n \sim \mathcal{N}(\mu_Z, \sigma_Z^2)$ are independent and identically distributed random variables. It follows that $Y_1, \cdots, Y_n \sim \mathcal{N}(\mu_Y, \sigma_Y^2)$ are independent and identically distributed with $\mu_Y = \mu_X + \mu_Z$ and $\sigma_Y^2 = \sigma_X^2 + \sigma_Z^2$.

In this situation we only observe the two samples $X_1, \cdots, X_n$ and $Y_1, \cdots, Y_n$. We are interested in performing a statistical hypothesis test to see if $\mu_X \neq \mu_Y$. We have two options here. We could either (1) use the pairing and apply a paired test or (2) ignore the pairing and use an unpaired test. In the console run “?t.test()” to see how to carry out an unpaired and a pared test within R.

(Q1)

Conduct a simulation study to explore the statistical power of these two approaches. You may want to consider a setting in which $n = 30, \mu_X = 10, \sigma_X = 5, \mu_Z = 1$ and $\sigma_Z = 1$. Consider a range of different significance levels α