

# Lab 5

*Shuyuan Wang, sw3449*

*November 8, 2019*

## Instructions

Make sure that you upload the PDF (or HTML) output after you have knitted the file. The files you upload to the Canvas page should be updated with commands you provide to answer each of the questions below. You can edit this file directly to produce your final solutions.

## Goal

The goal of this lab is to investigate the empirical behavior of a common hypothesis testing procedure through simulation using R. We consider the traditional two-sample t-test.

## Two-Sample T-Test

Consider an experiment testing if a 35 year old male's heart rate statistically differs between a control group and a dosage group. Let  $X$  denote the control group and let  $Y$  denote the drug group. One common method used to solve this problem is the two-sample t-test. The null hypothesis for this study is:

$$H_0 : \mu_1 - \mu_2 = \Delta_0,$$

where  $\Delta_0$  is the hypothesized value. The assumptions of the two sample pooled t-test follow below:

### Assumptions

1.  $X_1, X_2, \dots, X_m$  is a random sample from a normal distribution with mean  $\mu_1$  and variance  $\sigma_1^2$ .
2.  $Y_1, Y_2, \dots, Y_n$  is a random sample from a normal distribution with mean  $\mu_2$  and variance  $\sigma_2^2$ .
3. The  $X$  and  $Y$  samples are independent of one another.

### Procedure

The test statistic is

$$t_{calc} = \frac{\bar{x} - \bar{y} - \Delta_0}{\sqrt{\frac{s_1^2}{m} + \frac{s_2^2}{n}}},$$

where  $\bar{x}, \bar{y}$  are the respective sample means and  $s_1^2, s_2^2$  are the respective sample standard deviations.

The approximate degrees of freedom is

$$df = \frac{\left(\frac{s_1^2}{m} + \frac{s_2^2}{n}\right)^2}{\frac{(s_1^2/m)^2}{m-1} + \frac{(s_2^2/n)^2}{n-1}}$$

Under the null hypothesis,  $t_{calc}$  has a student's t-distribution with  $df$  degrees of freedom.

### Rejection rules

Reject  $H_0$  when:

$$Pvalue \leq \alpha$$

Alternative Hypothesis	P-value calculation
$H_A : \mu_1 - \mu_2 > \Delta_0$ (upper-tailed)	$P(t_{calc} > T)$
$H_A : \mu_1 - \mu_2 < \Delta_0$ (lower-tailed)	$P(t_{calc} < T)$
$H_A : \mu_1 - \mu_2 \neq \Delta_0$ (two-tailed)	$2 * P( t_{calc}  > T)$

## Tasks

- 1) Using the **R** function **t.test**, run the two sample t-test on the following simulated dataset. Note that the **t.test** function defaults a two-tailed alternative. Also briefly interpret the output.

```
set.seed(5)
sigma=5
Control <- rnorm(30,mean=10,sd=sigma)
Dosage <- rnorm(35,mean=12,sd=sigma)
t.test(Control,Dosage,mu=0,conf.level = 0.95)

##
## Welch Two Sample t-test
##
## data: Control and Dosage
## t = -1.9684, df = 62.014, p-value = 0.05349
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -4.96460632 0.03821408
## sample estimates:
## mean of x mean of y
## 10.05649 12.51969
```

Interpretation: Since p-value is slightly larger than 0.05, we fail to reject  $H_0 : \mu_1 - \mu_2 = \Delta_0$  at 5% significance level. It means we tend to believe that a 35 year old male's heart rate does not statistically differ between a control group and a dosage group.

- 2) Write a function called **t.test.sim** that simulates **R** different samples of  $X$  for control and **R** different samples of  $Y$  for the drug group and computes the proportion of test statistics that fall in the rejection region. The function should include the following:

- Inputs:
  - **R** is the number of simulated data sets (simulated test statistics). Let **R** have default 10,000.
  - Parameters **mu1**, **mu2**, **sigma1** and **sigma2** which are the respective true means and true standard deviations of  $X$  &  $Y$ . Let the parameters have respective defaults **mu1=0**, **mu2=0**, **sigma1=1** and **sigma2=1**.
  - Sample sizes **n** and **m** defaulted at **m=n=30**.
  - **level** is the significance level as a decimal with default at  $\alpha = .05$ .
  - **value** is the hypothesized value defaulted at 0.
- The output should be a **list** with the following labeled elements:
  - **statistic.list** vector of simulated t-statistics (this should have length **R**).
  - **pvalue.list** vector of empirical p-values (this should have length **R**).
  - **empirical.size** is a single number that represents the proportion of simulated test statistics that fell in the rejection region.

I started the function below:

```
t.test.sim <- function(R=10000,
                      mu1=0,mu2=0,
                      sigma1=1,sigma2=1,
                      m=30,n=30,
                      level=.05,
                      value=0,
                      direction="two.sided") {

  #Define empty lists
  statistic.list <- rep(0,R)
  pvalue.list <- rep(0,R)

  for (i in 1:R) {

    # Sample realized data
    Control <- rnorm(m,mean=mu1,sd=sigma1)
    Dosage <- rnorm(n,mean=mu2,sd=sigma2)

    # Testing values
    testing.procedure <- t.test(Control,Dosage,mu=value,
                                conf.level = 1-level,alternative = direction)
    statistic.list[i] <- testing.procedure$statistic
    pvalue.list[i] <- testing.procedure$p.value
  }
  empirical.size <- mean(pvalue.list<level)
  return(list(statistic.list=statistic.list,pvalue.list=pvalue.list,
             empirical.size=empirical.size))
}
```

Evaluate your function with the following inputs **R=10,mu1=10,mu2=12,sigma1=5** and **sigma2=5**.

```
t.test.sim(R=10,mu1=10,mu2=12,sigma1 = 5,sigma2=5)

## $statistic.list
## [1] -1.5594821 -1.6265940 -0.3916181 -3.0267377 -0.6315979  0.5023321
## [7]  0.1796087 -2.7168991 -2.1448224 -0.7961500
##
## $pvalue.list
## [1] 0.124574455 0.109418121 0.697191662 0.003707140 0.530363425
## [6] 0.617452414 0.858106351 0.008684406 0.036512133 0.429195763
##
## $empirical.size
## [1] 0.3
```

3) Assuming the null hypothesis

$$H_0 : \mu_1 - \mu_2 = 0$$

is true, compute the empirical size using 10,000 simulated data sets. Use the function **t.test.sim** to accomplish this task and store the object as **sim**. Output the empirical size quantity **sim\$empirical.size**. Comment on this value. What is it close to?

**Note:** use **mu1=mu2=10** (i.e., the null is true). Also set **sigma1=5,sigma2=5** and **n=m=30**.

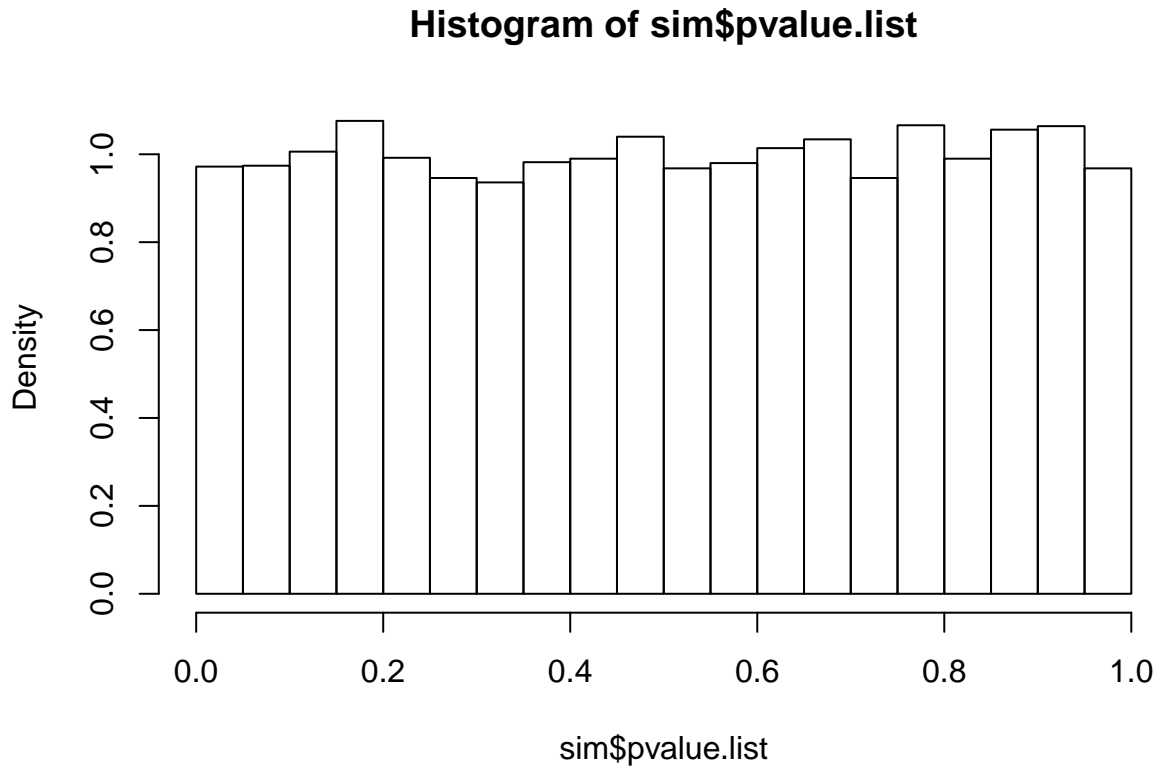
```
sim <- t.test.sim(R=10000,mu1=10,mu2=10,sigma1=5,sigma2=5,n=30,m=30)
sim$empirical.size
```

```
## [1] 0.0486
```

`sim$empirical.size` is close to significance level (probability of type I error, i.e. the probability of rejecting  $H_0$  when  $H_0$  is true).

- 4) Plot a histogram of the simulated P-values, i.e., `hist(sim$pvalue.list)`. What is the probability distribution shown from this histogram? Does this surprise you?

```
hist(sim$pvalue.list,freq = F)
```

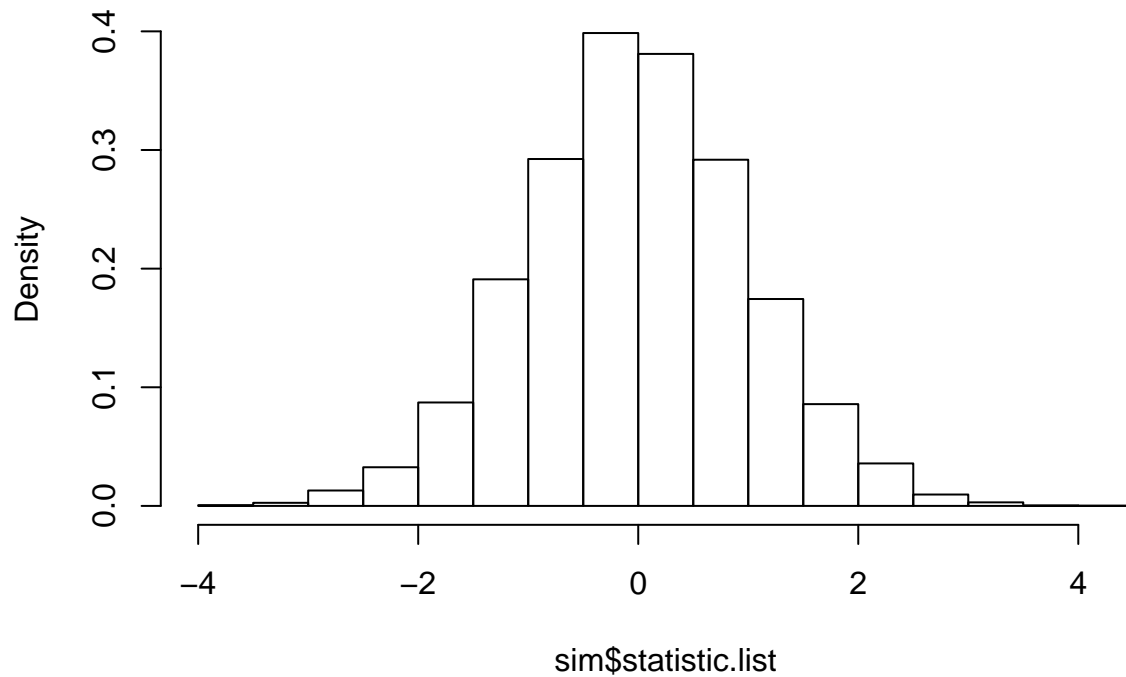


It seems like a Uniform distribution. Yes, it does surprise me.

- 5) Plot a histogram illustrating the empirical sampling of the t-statistic, i.e., `hist(sim$statistic.list,probability=TRUE)`. What is the probability distribution shown from this histogram?

```
hist(sim$statistic.list,probability=TRUE)
```

## Histogram of sim\$statistic.list



It's a bell shaped distribution, likely a t-distribution.

6) Run the following four lines of code:

```
t.test.sim(R=1000,mu1=10,mu2=10,sigma1=5,sigma2=5)$empirical.size
```

```
## [1] 0.055
```

```
t.test.sim(R=1000,mu1=10,mu2=12,sigma1=5,sigma2=5)$empirical.size
```

```
## [1] 0.315
```

```
t.test.sim(R=1000,mu1=10,mu2=14,sigma1=5,sigma2=5)$empirical.size
```

```
## [1] 0.847
```

```
t.test.sim(R=1000,mu1=10,mu2=16,sigma1=5,sigma2=5)$empirical.size
```

```
## [1] 0.998
```

Comment on the results.

Holding other parameters still, as the true difference between the two groups enlarges, the proportion of rejection increases. That is to say, the larger the true difference, the easier to detect the difference and the easier to reject  $H_0$  (i.e. the larger power of the test).

7) Run the following four lines of code:

```
t.test.sim(R=10000,mu1=10,mu2=12,sigma1=10,sigma2=10,m=10,n=10)$empirical.size
```

```
## [1] 0.0658
```

```
t.test.sim(R=10000,mu1=10,mu2=12,sigma1=10,sigma2=10,m=30,n=30)$empirical.size
```

```
## [1] 0.1242
```

```
t.test.sim(R=10000,mu1=10,mu2=12,sigma1=10,sigma2=10,m=50,n=50)$empirical.size
```

```
## [1] 0.1702
```

```
t.test.sim(R=10000,mu1=10,mu2=12,sigma1=10,sigma2=10,m=100,n=100)$empirical.size
```

```
## [1] 0.2896
```

Comment on the results.

Holding other parameters still and under  $H_1$ , as the sample sizes of the two groups both become larger, the proportion of rejection increases. That is to say, the larger the sample sizes, the easier to detect the difference and the easier to reject  $H_0$  (i.e. the larger power of the test).

- 8) **Extra credit:** Modify the `t.test.sim()` function to investigate how the power and size behave in the presence of heavy tailed data. I.e., test how **robust** the t-test is in the presence of violations from normality. **Hint:** the Cauchy distribution very heavy tailed.

```
t.test.sim2 <- function(R=10000,
                        location1=0,location2=0,
                        scale1=1,scale2=1,
                        m=30,n=30,
                        level=.05,
                        value=0,
                        direction="two.sided") {

  #Define empty lists
  statistic.list <- rep(0,R)
  pvalue.list <- rep(0,R)

  for (i in 1:R) {

    # Sample realized data
    Control <- rcauchy(m,location=location1,scale=scale1)
    Dosage <- rcauchy(n,location=location2,scale=scale2)

    # Testing values
    testing.procedure <- t.test(Control,Dosage,mu=value,
                                conf.level = 1-level,alternative = direction)
    statistic.list[i] <- testing.procedure$statistic
    pvalue.list[i] <- testing.procedure$p.value
  }
  empirical.size <- mean(pvalue.list<level)
  return(list(statistic.list=statistic.list,pvalue.list=pvalue.list,
              empirical.size=empirical.size))
}

# how power and size behave in the presence of heavy tailed data
t.test.sim2(R=10000,location1=0,location2=2,scale1=1,scale2=1,m=10,n=10)$empirical.size

## [1] 0.1784

t.test.sim2(R=10000,location1=0,location2=2,scale1=1,scale2=1,m=30,n=30)$empirical.size

## [1] 0.1941
```

```
t.test.sim2(R=10000,location1=0,location2=2,scale1=1,scale2=1,m=50,n=50)$empirical.size
```

```
## [1] 0.2095
```

```
t.test.sim2(R=10000,location1=0,location2=2,scale1=1,scale2=1,m=100,n=100)$empirical.size
```

```
## [1] 0.2139
```

In the presence of violations from normality, t-test behaves in the same manner that holding others still as the size increases, the power increases as well.