Lab 5

Hanao Li hl3202

Nov 12, 2018

Instructions

Make sure that you upload an RMarkdown file to the canvas page (this should have a .Rmd extension) as well as the PDF output after you have knitted the file (this will have a .pdf extension). 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. The lab is due 11:59pm on Saturday, November 9th.

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 Δ_0 is the hypothesized value. The assumptions of the two sample pooled t-test follow below:

Assumptions

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:

$$Pvalue \leq \alpha$$

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)
```

```
##
##  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
```

```
# From the output, we could see that the 95 percent confidence interval is from -4.96 to 0.038 w
hich includes 0. Since p-value is larger than 0.05, we could say that we do not reject the null
 hypothesis, and we can conclude Control and Dosage has the same mean.
```

2. Write a function called **emperical.size** 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:

I started the function below:

```
emperical.size <- function(R=10000,
                           mu1=0,mu2=0,
                           sigma1=1,sigma2=1,
                           m=30,n=30,
                           level=.05,
                           value=0,
                           direction="Two") {

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

  for (i in 1:R) {

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

    # Testing values
    testing.procedure <- t.test(Control, Dosage)
    statistic.list[i] <- as.numeric(testing.procedure[1])
    pvalue.list[i] <- as.numeric(testing.procedure[3])

  }

  esize = sum((pvalue.list <= level)) / R

   size.list <- list(statistic.list, pvalue.list, esize)
   names(size.list) <- c("statistic.list", "pvalue.list", "emperical.size")

  return(size.list)

}
```

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

```
emperical.size(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
##
## $emperical.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 **emperical.size** to accomplish this task and store the object as **sim**. Output the empirical size quantity **sim$size**. Comment on this value. What is it close to?

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

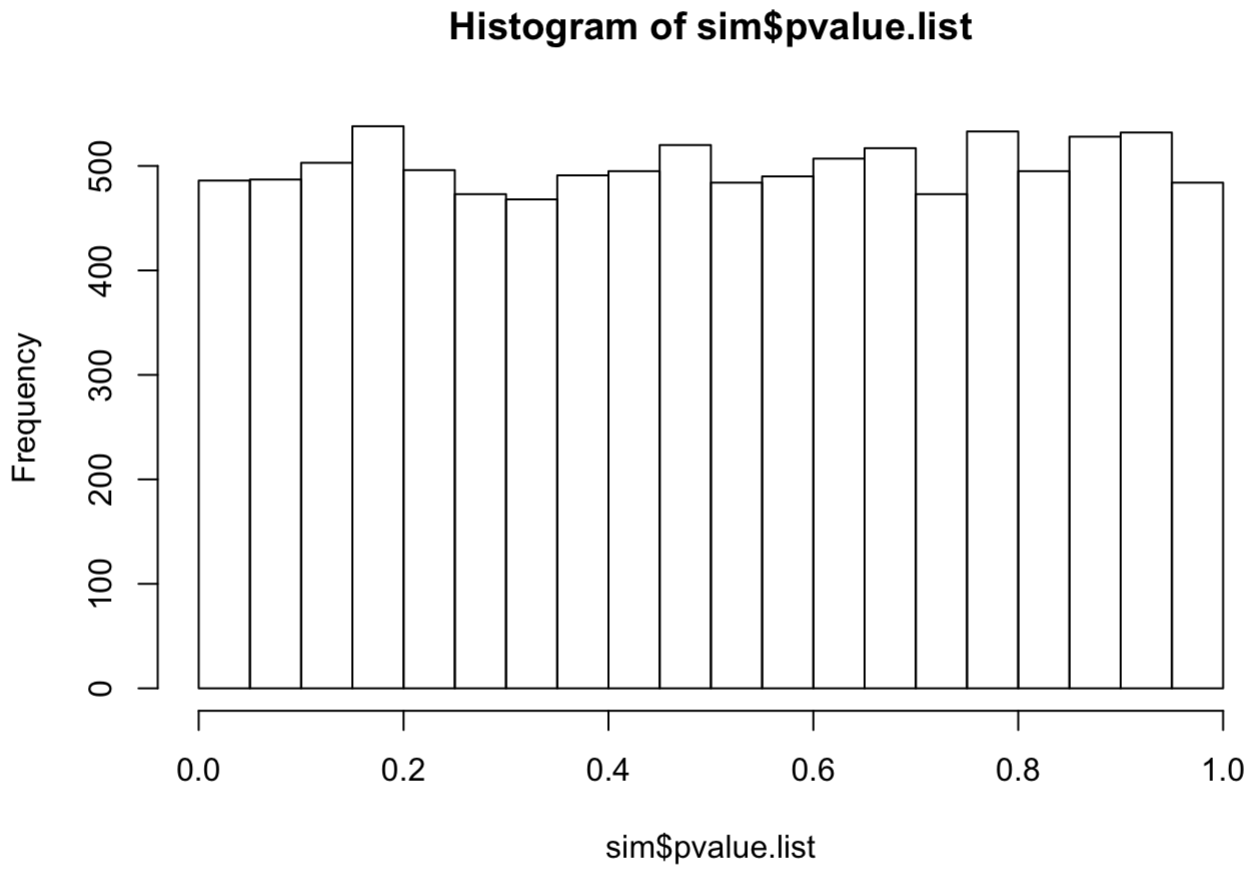
```
sim <- emperical.size(R = 10000, mu1 = 10, mu2 = 10, sigma1 = 5, sigma2 = 5, m = 30, n = 30)
sim$emperical.size
```

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

```
# This value is close to 0.05 which is the significance level. This is accurate since our null h
ypothesis is true so the proportion rejected will be close to the alpha.
```

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?

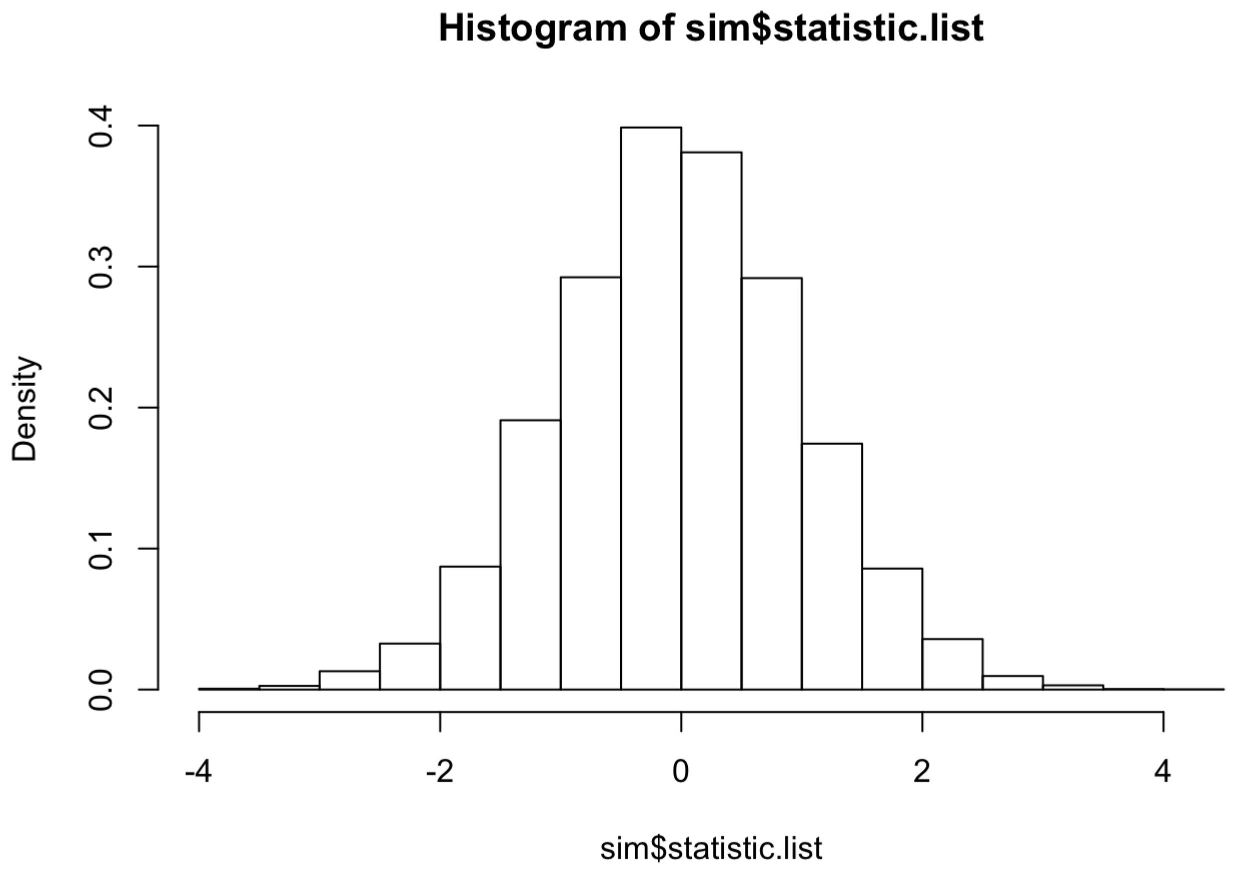
```
hist(sim$pvalue.list)
```



```
# This is the uniform distribution. because the distribution of an invertible CDF of a random va
riable is unifrom from 0 to 1.
```

5. Plot a histogram illustrating the empirical sampling 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)
```



```
# This is a normal distribution.
```

6. Run the following four lines of code:

   emperical.size(R=1000,mu1=10,mu1=10,sigma1=5,sigma2=5)$emperical.size

   emperical.size(R=1000,mu1=10,mu1=12,sigma1=5,sigma2=5)$emperical.size

   emperical.size(R=1000,mu1=10,mu1=14,sigma1=5,sigma2=5)$emperical.size

   emperical.size(R=1000,mu1=10,mu1=16,sigma1=5,sigma2=5)$emperical.size

   Comment on the results.

```
emperical.size(R=1000,mu1=10,mu2=10,sigma1=5,sigma2=5)$emperical.size
```

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

```
emperical.size(R=1000,mu1=10,mu2=12,sigma1=5,sigma2=5)$emperical.size
```

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

```
emperical.size(R=1000,mu1=10,mu2=14,sigma1=5,sigma2=5)$emperical.size
```

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

```
emperical.size(R=1000,mu1=10,mu2=16,sigma1=5,sigma2=5)$emperical.size
```

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

```
# The emperical size is becoming larger because the difference between the true mean is becoming more and more significant which means that more simulated datasets will be rejected.
```

7. Run the following four lines of code:

   emperical.size(R=10000,mu1=10,mu2=12,sigma1=10,sigma2=10,m=10,n=10)$emperical.size

   emperical.size(R=10000,mu1=10,mu2=12,sigma1=10,sigma2=10,m=30,n=30)$emperical.size

   emperical.size(R=10000,mu1=10,mu2=12,sigma1=10,sigma2=10,m=50,n=50)$emperical.size

   emperical.size(R=10000,mu1=10,mu2=12,sigma1=10,sigma2=10,m=100,n=100)$emperical.size

   Comment on the results.

```
emperical.size(R=10000,mu1=10,mu2=12,sigma1=10,sigma2=10,m=10,n=10)$emperical.size
```

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

```
emperical.size(R=10000,mu1=10,mu2=12,sigma1=10,sigma2=10,m=30,n=30)$emperical.size
```

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

```
emperical.size(R=10000,mu1=10,mu2=12,sigma1=10,sigma2=10,m=50,n=50)$emperical.size
```

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

```
emperical.size(R=10000,mu1=10,mu2=12,sigma1=10,sigma2=10,m=100,n=100)$emperical.size
```

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

```
# The emperical size is becoming larger because since the true means are different, as the samples increases, the results will be more accurate which means that more simulated datasets will be rejected.
```