Lab 5

Yuhao Wang, yw3204

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. The 95% confidence interval also leads to the some result.

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

1. $X_1, X_2, \ldots, X_m$ is a random sample from a normal distribution with mean μ_1 and variance σ_1^2.
2. $Y_1, Y_2, \ldots, Y_n$ is a random sample from a normal distribution with mean μ_2 and variance σ_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

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)	$2 * P(\|t_{calc}\| > T)$

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()
t.test(Control, Dosage, alternative = "t", mu = -5)
```

```
##
##  Welch Two Sample t-test
##
## data:  Control and Dosage
## t = 2.0272, df = 62.014, p-value = 0.04694
## alternative hypothesis: true difference in means is not equal to -5
## 95 percent confidence interval:
##  -4.96460632  0.03821408
## sample estimates:
## mean of x mean of y
##  10.05649  12.51969
```

```
t.test(Control, Dosage, alternative = "g", mu = -5)
```

```
##
##  Welch Two Sample t-test
##
## data:  Control and Dosage
## t = 2.0272, df = 62.014, p-value = 0.02347
## alternative hypothesis: true difference in means is greater than -5
## 95 percent confidence interval:
##  -4.552705       Inf
```

```
## sample estimates:
## mean of x mean of y
##  10.05649  12.51969
```

```
t.test(Control, Dosage, alternative = "l", mu = -5)
```

```
##
##  Welch Two Sample t-test
##
## data:  Control and Dosage
## t = 2.0272, df = 62.014, p-value = 0.9765
## alternative hypothesis: true difference in means is less than -5
## 95 percent confidence interval:
##        -Inf -0.3736869
## sample estimates:
## mean of x mean of y
##  10.05649  12.51969
```

Assume the null is $H_0 : \mu_1 - \mu_2 = -5$. The P-value for alternative $H_{A1} : \mu_1 - \mu_2 \neq -5$, $H_{A2} : \mu_1 - \mu_2 > -5$, and $H_{A3} : \mu_1 - \mu_2 < -5$ are seperately 0.047, 0.023 and 0.98, which are all insignificant when $\alpha = 0.01$. Therefore, we accept the null that the difference of mean is -5.

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:

- 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**, **mu1=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:

```
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 c(1:R)) {

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

    t <- mean(Control) - mean(Dosage) - value
    t <- t / sqrt(sigma1 **2 / m + sigma2 ** 2 / n)

    df <- (sigma1 **2 / m + sigma2 ** 2 / n) ** 2
    df <- df / (sigma1^4 / (m^2 * (m-1)) + sigma2^4 / (n^2 * (n-1)))

    # assume a two sided test
    p <- (1 - pt(abs(t), df)) * 2

    statistic.list[i] <- t
    pvalue.list[i] <- p
  }

  empirical.size <- sum(pvalue.list < level) / R

  return (list(statistic.list, pvalue.list, empirical.size))

  #for (i in 1:R) {

    # Sample realized data
    #Control <-
    #Dosage <-

    # Testing values
    #testing.procedure <-
    #statistic.list[i] <-
    #pvalue.list[i] <-
    #}
   #size.list <-
  #return()

}

emperical.size(R = 100, mu1 = 10, mu2 = 12, sigma1 = 5, sigma2 = 5)
```

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

```
##   [1] -1.5861090 -1.7323453 -0.3708647 -2.8195378 -0.6699651  0.5150688
##   [7]  0.1798320 -2.8038017 -2.0920774 -0.8695075 -2.2345974 -1.5726986
##  [13] -3.4947412 -0.8110779 -2.2711868 -1.7650207 -2.8832824  0.2491932
##  [19] -2.1918651 -1.0245902 -0.1064628 -3.5248851 -1.3916341 -2.8751203
##  [25] -0.9515481 -1.8009965 -0.9498246 -1.7703880 -1.6162308 -1.3601658
##  [31] -3.8492429 -0.6354300 -1.6580171 -0.1990177 -0.7069224 -1.5624353
##  [37] -1.6909669 -1.7736820 -3.1153419 -2.1873317 -1.8635657 -0.9598042
##  [43] -2.7151990  1.3071357 -1.8003936 -1.9518850  0.2324198 -2.9091141
##  [49] -0.5095471 -0.6901077 -0.9630197 -1.5815257 -0.9536170 -1.4156164
##  [55] -2.8153275 -1.0099221 -2.2102093 -1.4708188 -1.9429152 -3.6013784
##  [61] -1.6838108 -2.3754702 -2.9645544 -0.9822021 -0.7341464 -2.3771721
##  [67] -1.7219494 -1.6642979 -2.3322694 -4.7986772 -2.5569528 -2.3198962
##  [73] -2.5083586 -2.1424678 -1.0826765 -1.6995505 -3.3615842 -1.0711714
##  [79] -1.9091514 -2.6606972 -0.6517892 -1.5601054 -1.5369666 -1.4045908
##  [85] -2.3312826 -1.8489132 -2.0316815 -1.3823133 -1.2846733 -2.1055652
##  [91] -1.5318899 -0.2985285 -2.8199845 -1.2927492 -0.4772538 -1.7022132
##  [97] -1.8042835 -2.4857706 -1.9721002 -0.8932217
##
## [[2]]
##   [1] 1.181521e-01 8.852729e-02 7.120894e-01 6.570122e-03 5.055387e-01
##   [6] 6.084624e-01 8.579113e-01 6.858479e-03 4.081895e-02 3.881538e-01
##  [11] 2.931129e-02 1.212283e-01 9.161396e-04 4.206369e-01 2.686368e-02
##  [16] 8.282472e-02 5.512812e-03 8.040917e-01 3.241729e-02 3.098110e-01
##  [21] 9.155825e-01 8.344961e-04 1.693476e-01 5.638799e-03 3.452748e-01
##  [26] 7.690285e-02 3.461426e-01 8.191787e-02 1.114714e-01 1.790409e-01
##  [31] 2.978273e-04 5.276472e-01 1.027135e-01 8.429449e-01 4.824453e-01
##  [36] 1.236258e-01 9.621070e-02 8.136541e-02 2.855045e-03 3.276320e-02
##  [41] 6.744769e-02 3.411380e-01 8.711392e-03 1.963252e-01 7.699911e-02
##  [46] 5.578564e-02 8.170301e-01 5.131058e-03 6.123022e-01 4.928788e-01
##  [51] 3.395356e-01 1.191963e-01 3.442351e-01 1.622354e-01 6.646157e-03
##  [56] 3.167274e-01 3.105014e-02 1.467449e-01 5.688567e-02 6.572547e-04
##  [61] 9.759350e-02 2.085227e-02 4.393049e-03 3.300798e-01 4.658171e-01
##  [66] 2.076503e-02 9.040829e-02 1.014470e-01 2.317922e-02 1.158713e-05
##  [71] 1.319812e-02 2.388700e-02 1.494793e-02 3.636500e-02 2.834327e-01
##  [76] 9.457333e-02 1.376497e-03 2.885296e-01 6.119383e-02 1.006863e-02
##  [81] 5.171118e-01 1.241753e-01 1.297394e-01 1.654759e-01 2.323497e-02
##  [86] 6.956913e-02 4.677573e-02 1.721757e-01 2.040137e-01 3.958253e-02
##  [91] 1.309863e-01 7.663667e-01 6.562103e-03 2.012240e-01 6.349749e-01
##  [96] 9.407009e-02 7.637990e-02 1.583057e-02 5.337301e-02 3.754298e-01
##
## [[3]]
## [1] 0.34
```

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

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?

```
sim <- emperical.size(R = 100000, mu1 = 10, mu2 = 10, sigma1 = 5, sigma2 = 5)
sim[[3]]
```

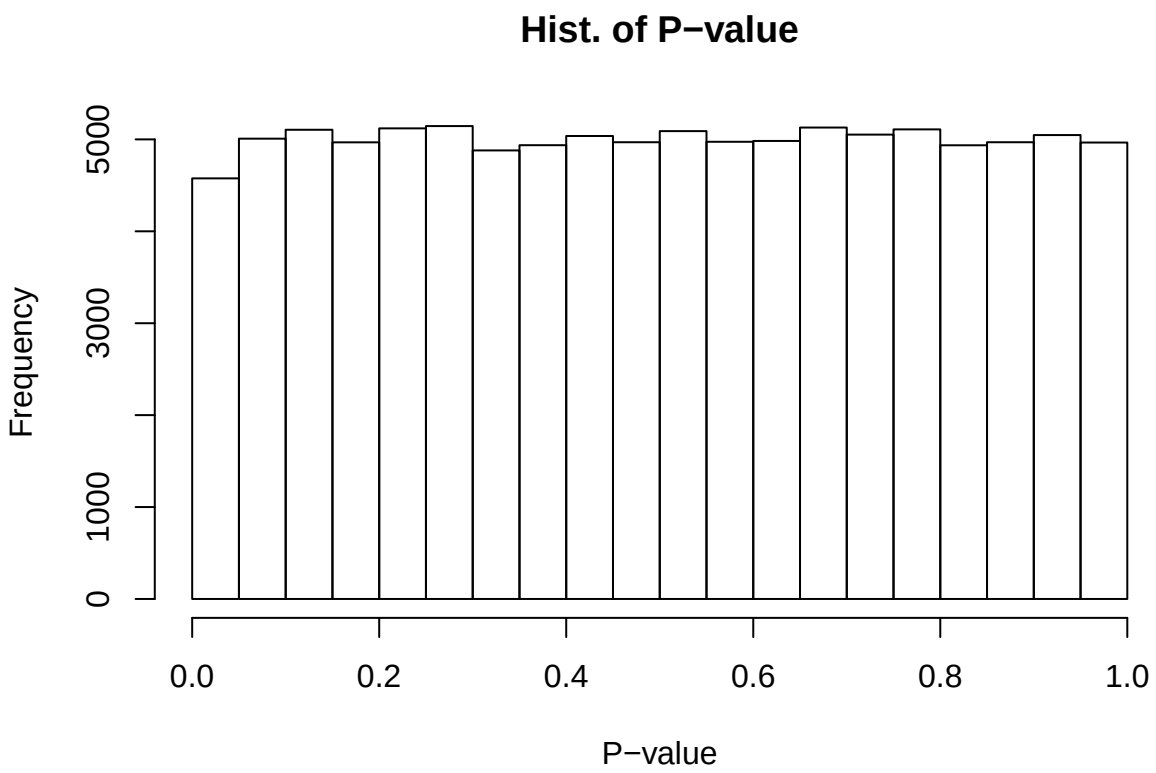
```
## [1] 0.04576
```

It is close to 0.04.

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

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?

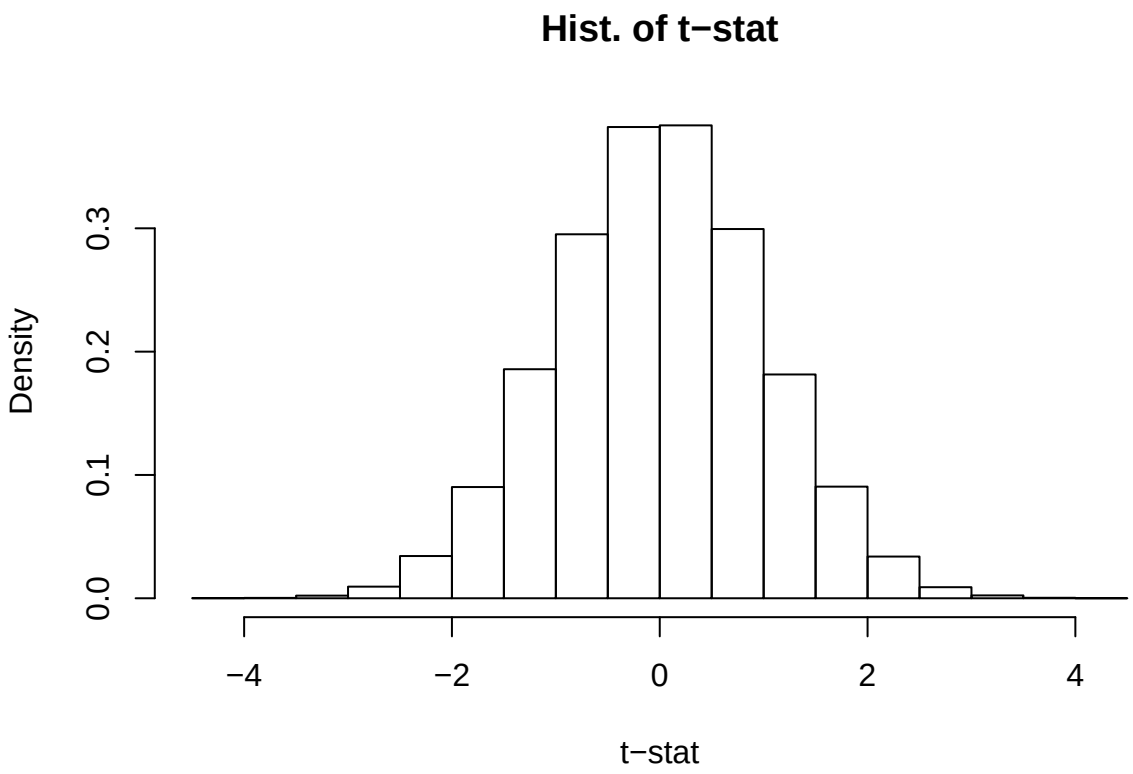
```
hist(sim[[2]], xlab = "P-value", main = "Hist. of P-value")
```



The distribution is close to the uniform distribution on the interavl $(0, 1)$. Actually, it is not dificult to prove under the null hypothesis its distribution is exactly uniform $U(0, 1)$.

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[[1]], probability = TRUE, xlab = "t-stat", main = "Hist. of t-stat")
```



The distribution shown above is quite like a normal distribution with mean close to 0.

6) Run the following four lines of code:

```
emperical.size(R=1000,mu1=10,mu2=10,sigma1=5,sigma2=5)[[3]]
```

```
## [1] 0.053
```

```
emperical.size(R=1000,mu1=10,mu2=12,sigma1=5,sigma2=5)[[3]]
```

```
## [1] 0.348
```

```
emperical.size(R=1000,mu1=10,mu2=14,sigma1=5,sigma2=5)[[3]]
```

```
## [1] 0.865
```

```
emperical.size(R=1000,mu1=10,mu2=16,sigma1=5,sigma2=5)[[3]]
```

```
## [1] 0.997
```

```
  **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.
```

The proportion of rejecting the null is increasing as mu2 increases and the increasing speed is not linear.

7) Run the following four lines of code:

```
emperical.size(R=10000,mu1=10,mu2=12,sigma1=10,sigma2=10,m=10,n=10)[[3]]
```

```
## [1] 0.0547
```

```
emperical.size(R=10000,mu1=10,mu2=12,sigma1=10,sigma2=10,m=30,n=30)[[3]]
```

```
## [1] 0.1149
```

```
emperical.size(R=10000,mu1=10,mu2=12,sigma1=10,sigma2=10,m=50,n=50)[[3]]
```

```
## [1] 0.1609
```

```
emperical.size(R=10000,mu1=10,mu2=12,sigma1=10,sigma2=10,m=100,n=100)[[3]]
```

```
## [1] 0.2916
```

```
  **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.
```

The proportion of rejection is also increasing as the sample size of the two group increases simultaneously.