Lab 5

Shreya Rao sr3843

June 4, 2021

```
library(formatR)
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
```

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 Δ_0 is the hypothesized value. The assumptions of the two-sample 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} (or 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)	$2 * P(\lvert t_{calc} \rvert > T)$

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(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
```

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**).
  - **rejection.rate** 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 = 0.05, 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] <- testing.procedure$statistic
        pvalue.list[i] <- testing.procedure$p.value
    }
    rejection.rate <- mean(pvalue.list < level)
    return(list(statistic.list = statistic.list, pvalue.list = pvalue.list,
        rejection.rate = rejection.rate))
}
```

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,
    m = 30, n = 30, level = 0.05, value = 0, direction = "Two")
```

```
## $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 0.617452414
##  [7] 0.858106351 0.008684406 0.036512133 0.429195763
##
## $rejection.rate
## [1] 0.3
```

3) Assuming the null hypothesis

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

is true, compute the empirical size (or rejection rate) 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$rejection.rate**. 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**.

[H0 is true]

```r
sim <- t.test.sim(R = 10000, mu1 = 10, mu2 = 10, sigma1 = 5,
    sigma2 = 5, m = 30, n = 30, level = 0.05, value = 0, direction = "Two")

sim$rejection.rate
```

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

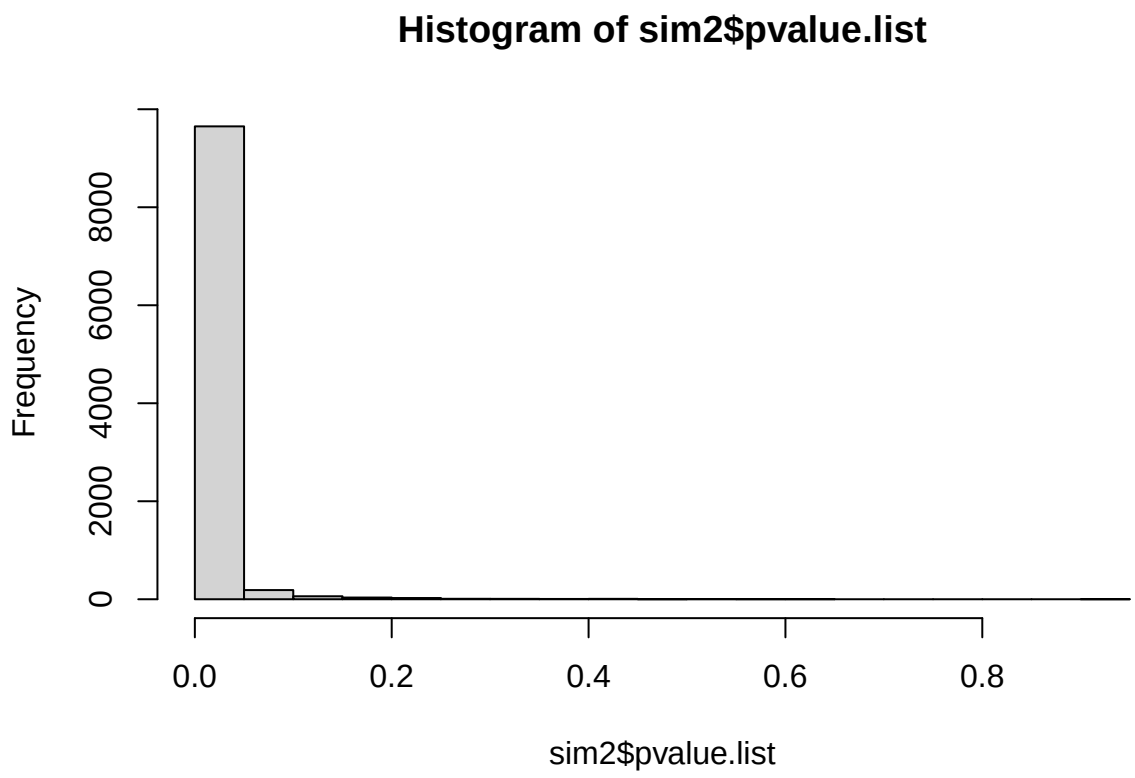
The rejection rate is extremely close to 0.5. This implies that a 5% of the 10000 samples created rejected the null hypothesis that mu1 = mu2, which is consistent with the 95% confidence interval. This happens because mu1 = mu2. The significance level alpha is the probability that we will make the mistake of rejecting the null hypothesis when in fact it is true. The p-value measures the probability of getting a more extreme value than the one we get from the experiment. [This value is close to 0.05 which is the significance level. This is accurate since our null hypothesis is true so the proportion rejected will be close to the alpha.]

NOTE: HA is true

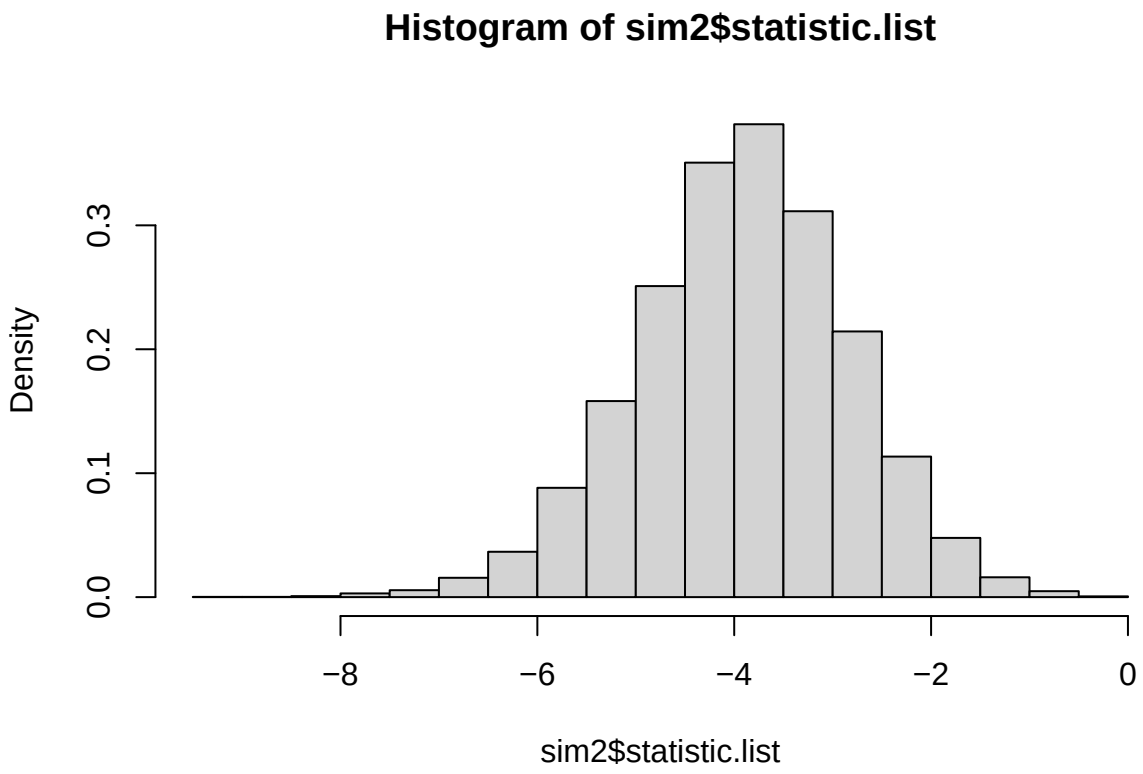
```r
sim2 <- t.test.sim(R = 10000, mu1 = 10, mu2 = 15, sigma1 = 5,
    sigma2 = 5, m = 30, n = 30, level = 0.05, value = 0, direction = "Two")


sim2$rejection.rate
```

```
## [1] 0.9651
```

```r
# Most p values will be close to 0
hist(sim2$pvalue.list)
```



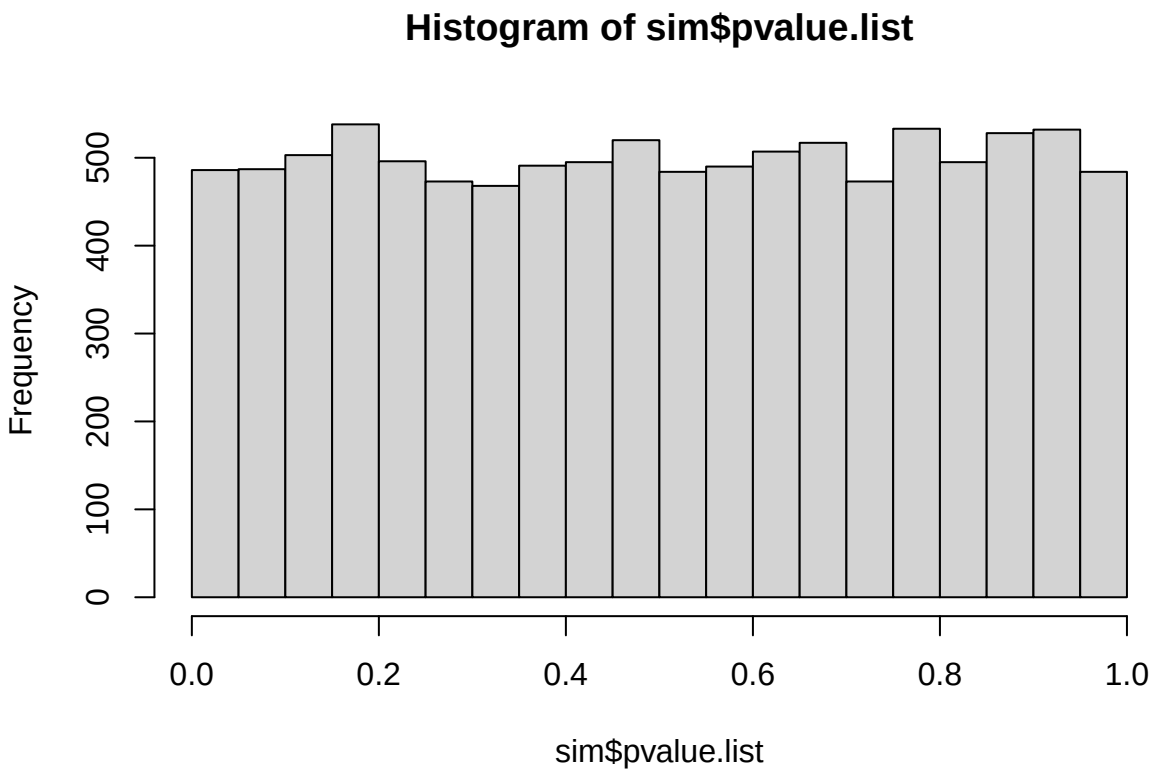
```r
# t-statistic will follow a normal dustribution. Under the
# null hypothesis, the t distribution will be centered away
# from 0.
hist(sim2$statistic.list, probability = TRUE)
```



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?

[H0 is true]

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



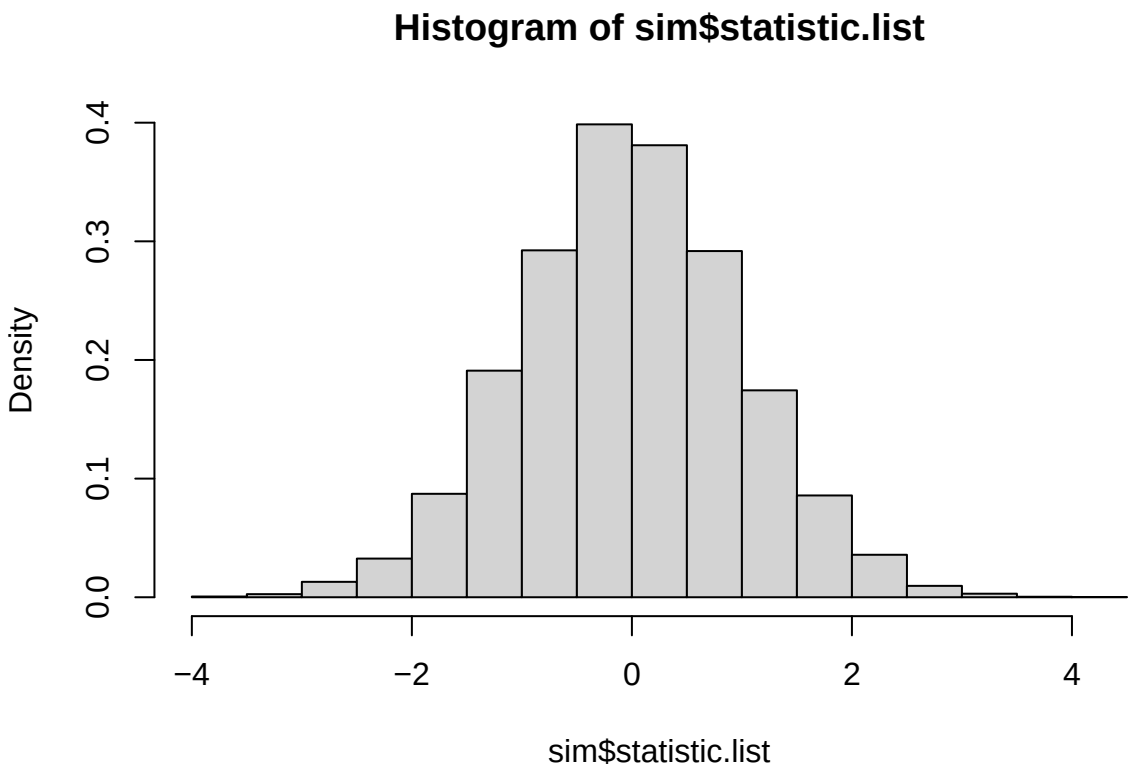

```
# The distribution of an invertible CDF of a random variable
# is uniform from 0 to 1.
```

If means are the same then the curves will totally overlap. So all the p values are uniform.

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

[Under H0 is true]

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



This follows a normal distribution. When sample size is large, t-statistic values will be normally distributed. Under the null hypothesis, the t distribution will be centered around 0.

6) Run the following four lines of code:

```
# The effect size (diff between means) the the only thing
# that's increasing

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

```
## [1] 0.052
```

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

```
## [1] 0.346
```

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

```
## [1] 0.868
```

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

```
## [1] 0.996
```

As the effect size or departure from the null increases, with a fixed sample size, the power of the test increases. Power is the rejection rate if the null is false (complement of type 2 error, prob of rejecting the null when it should be rejected)

Comment on the results.

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)$rejection.rate
```

```
## [1] 0.0713
```

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

```
## [1] 0.1172
```

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

```
## [1] 0.1678
```

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

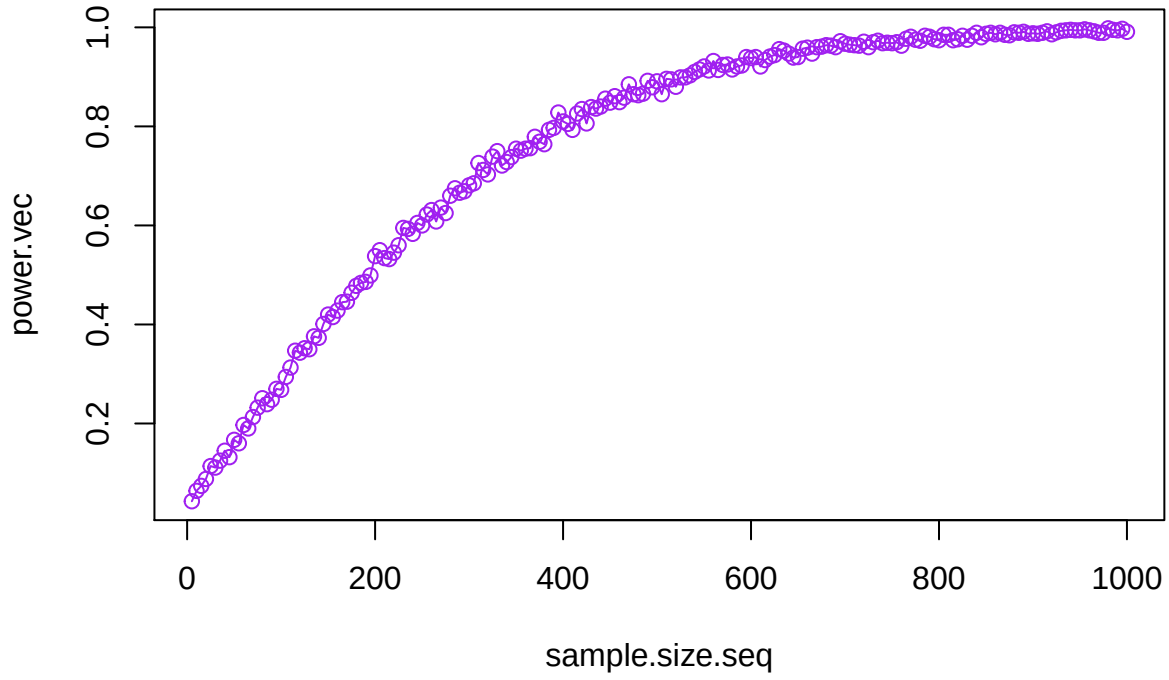
```
## [1] 0.291
```

Increasing the sample sizes here. As we increase the sample size, the probability of rejecting null increases when null is not true.

7b)

```
power.vec <- NULL
sample.size.seq <- seq(5, 1000, by = 5)
counter <- 1

for (i in sample.size.seq) {
    power.vec[counter] <- t.test.sim(R = 1000, mu1 = 10, mu2 = 12,
        sigma1 = 10, sigma2 = 10, m = i, n = i)$rejection.rate
    counter <- counter + 1
}
```

```
plot(sample.size.seq, power.vec, col = "purple", type = "o")
```



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., investigate how **robust** the t-test is in the presence of violations from normality.

Hint: The Cauchy distribution and the students' t-distribution with low df are both heavy tailed.

```
t.test.not.normal <- function(R = 10000, df1 = 0, df2 = 0, m = 30,
    n = 30, level = 0.05, value = 0, direction = "Two") {

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

    for (i in 1:R) {

        # Sample realized data
        Control <- rt(m, df = df1)
        Dosage <- rt(n, df = df2)

        # Testing values
        testing.procedure <- t.test(Control, Dosage)
        pvalue.list[i] <- testing.procedure$p.value
    }

    rejection.rate <- mean(pvalue.list < level)
    return(list(pvalue.list = pvalue.list, rejection.rate = rejection.rate))
}
```

```
power.vec <- NULL
sample.size.seq <- seq(5, 1000, by = 5)
counter <- 1

for (i in sample.size.seq) {
    power.vec[counter] <- t.test.not.normal(R = 1000, df1 = 10,
        df2 = 9, m = i, n = i)$rejection.rate
    counter <- counter + 1
}
```

```
plot(sample.size.seq, power.vec, col = "purple", type = "o")
```