Lab 06: Don't Test Me!

PSTAT 100, Summer Session A 2025 with Ethan P. Marzban

MEMBER 1 (NetID 1) MEMBER 2 (NetID 2)
MEMBER 3 (NetID 3)

July 15, 2025

Required Packages

```
library(ottr)        # for checking test cases (i.e. autograding)
library(pander)      # for nicer-looking formatting of dataframe outputs
library(tidyverse)   # for graphs, data wrangling, etc.
```

Logistical Details

Logistical Details

- This lab is due by **11:59pm on Wednesday, July 16, 2025**.
- Collaboration is allowed, and encouraged!
  - If you work in groups, list ALL of your group members' names and NetIDs (not Perm Numbers) in the appropriate spaces in the YAML header above.
  - Please delete any "MEMBER X" lines in the YAML header that are not needed.
  - No more than 3 people in a group, please.
- Ensure your Lab properly renders to a `.pdf`; non-`.pdf` submissions will not be graded and will receive a score of 0.
- Ensure all test cases pass (test cases that have passed will display a message stating `"All tests passed!"`)

Lab Overview and Objectives

Welcome to another PSTAT 100 Lab! In this lab, we will cover the following:

- Hypothesis Testing and Multiple testing
- ANOVA and the Kruskal-Wallis Test

Part 1: One-Sample *t*-Test

Recall the setup for testing a population mean: given an i.i.d. sample $X_1, \cdots, X_n$ from a normal distribution with unknown mean μ and unknown variance σ^2, we wish to test the null $H_0: \mu = \mu_0$ (for some specified μ_0) against various alternatives. We leverage the fact that

$$\text{TS} := \frac{\overline{X}_n - \mu_0}{S_X/\sqrt{n}} \overset{H_0}{\sim} t_{n-1}$$

Our ***p*-value** is the probability of observing something as or more extreme (in the direction of the alternative) than the observed value of the test statistic.

We call a test of this form a **one-sample *t*-test**. This is why the corresponding `R` function to carry out this test is called `t.test()`. As practice, let's write our own version of this test.

! Question 1

Write a function called `my_t_test()` that takes in three arguments:

- `x`: a data vector
- `mu_0`: the null value
- `alt`: one of `"lower"`, `"upper"`, or `"two"`, indicating the alternative

The output of your function should be the p-value of testing $H_0: \mu = \mu_0$ against the specified alternative.

Solution:

```
## replace this line with your code
my_t_test <- function(x, mu_0, alt) {
  n <- length(x)
  test_stat <- (mean(x) - mu_0) / (sd(x) / sqrt(n))

  if(alt == "lower") {
    p_val <- pt(test_stat, df = n - 1)
  } else if (alt == "upper") {
    p_val <- 1 - pt(test_stat, df = n - 1)
  } else if (alt == "two"){
    p_val <- 2 * pt(-1 * abs(test_stat), df = n - 1)
  }
  return(p_val)
}
```

Answer Check:

```
# DO NOT EDIT THIS LINE
invisible({check("tests/q1.R")})
```

```
All tests passed!
```

Part II: Multiple Testing

Recall that **Multiple Testing** refers to the phenomenon by which, as the number of tests we conduct on either the same data and/or the same hypotheses, the probability of falsely rejecting at least one null increases.

In this part of the lab, we will investigate this phenomenon empirically, using simulations.

! Question 2

Part (a)
Start by setting your seed to 100. Then, generate 1000 samples of size 100 from the $\mathcal{N}(10, 1)$ distribution. For each sample, use your `my_t_test()` function from Question 1 above to conduct a test of $H_0: \mu = 10$ against the two-sided alternative $H_0: \mu \neq 10$. Note crucially that our null value in this simulation is the true value of the parameter (which, again, in a real-world setting wouldn't be known to us)!

Solution:

```
## replace this line with your code
set.seed(100)
p_vals <- c()
for(b in 1:1000) {
  X <- rnorm(100, 10)
  p_vals <- c(p_vals, my_t_test(X, 10, "two"))
}
```

Answer Check:

```
# DO NOT EDIT THIS LINE
invisible({check("tests/q2a.R")})
```

```
All tests passed!
```

Part (b)
Using a 5% level of significance, how many of the 1000 tests you conducted in part (a) lead to rejection of the null? Assign this to a variable called `num_false_rej`. Note that these are all "false" rejections, in the sense that our null value is equal to the true value of the parameter - again, we wouldn't know that these are "false" rejections in a real-world setting, since we wouldn't know the true value of μ!

Solution:

```
## replace this line with your code
(num_false_rej <- sum(p_vals < 0.05))
```

```
[1] 41
```

Answer Check:

```
# DO NOT EDIT THIS LINE
invisible({check("tests/q2b.R")})
```

```
All tests passed!
```

Now, suppose we generate N independent samples of size n from a population with mean μ'. Further suppose we use each of these N samples to test the null $H_0: \mu = \mu'$ at an α level of significance; i.e., unbeknownst to us, we are using the true population mean as our null value. As was mentioned in lecture, the number X of these N tests that (falsely) reject the null satisfies $X \sim \text{Bin}(N, \alpha)$.

! Question 3

Carry out the following:

1) Take 100 samples, each of size 100, from the $\mathcal{N}(10, 1)$ distribution, and record the number of samples for which we would reject the null $H_0: \mu = \mu_0$ in favor of the two-sided alternative $H_A: \mu \neq \mu_0$ at a 5% level of significance.

2) Repeat this 1000 times, to obtain a vector of 1000 numbers: each number representing the number of false rejections that took place out of 100 samples of size 100. Call this vector `num_rej`.

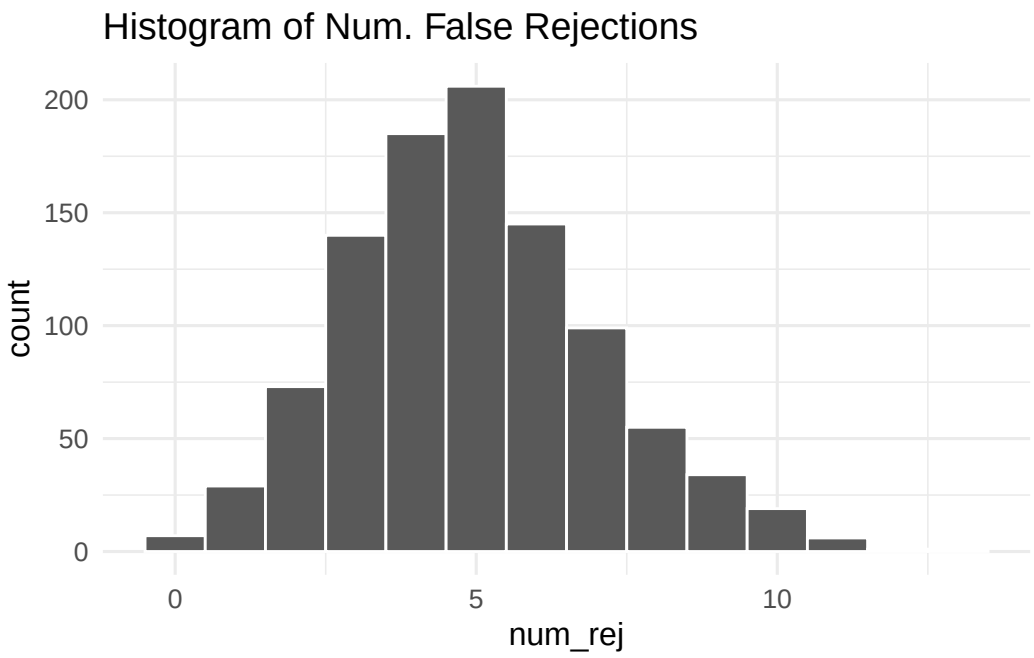
The distribution of the values in `num_rej` should be a Bin(100, 0.05) distribution. Plot a histogram of the `num_rej`, and also plot a QQ-plot comparing the `num_rej` values to the Bin(100, 0.05) distribution.

Solution:

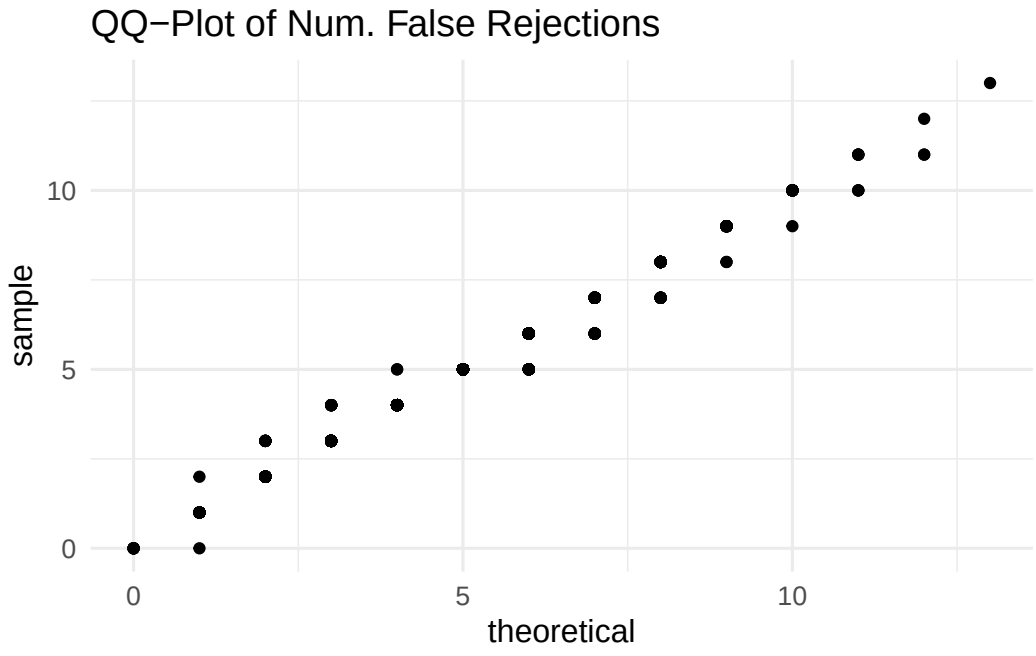
```
## replace this line with your code
set.seed(100)
num_rej <- c()

for(j in 1:1000){
  p_vals <- c()
    for(i in 1:100) {
      X <- rnorm(100, 10)
      p_vals <- c(p_vals, my_t_test(X, 10, "two"))
    }
  num_rej <- c(num_rej, sum(p_vals < 0.05))
}

data.frame(num_rej) %>% ggplot(aes(x = num_rej)) +
  geom_histogram(bins = 14, col = "white") +
  theme_minimal(base_size = 12) +
  ggtitle("Histogram of Num. False Rejections")
```




```
data.frame(num_rej) %>% ggplot(aes(sample = num_rej)) +
  geom_qq(distribution = stats::qbinom,
          dparams = list(100, 0.05)) +
  theme_minimal(base_size = 12) +
  xlab("theoretical") + ylab("sample") +
  ggtitle("QQ-Plot of Num. False Rejections")
```



Answer Check:
There is no autograder for this question; your TA will manually check that your answers are correct.

Part III: ANOVA

A natural question that arises is: how can we compare means across *several* (i.e. more than 3 groups). For example, suppose we want to determine whether the average (mean) air pollution levels are the same across three different cities.

More concretely, consider k populations $\mathcal{P}_1, \cdots, \mathcal{P}_3$ with means $\mu_1, \cdots, \mu_k$ and variances $\sigma_1^2, \cdots, \sigma_k^2$. Additionally, consider testing

$$\begin{bmatrix} H_0: & \mu_1 = \mu_2 = \cdots = \mu_k \\ H_A: & \text{At least one of the means are different} \end{bmatrix}$$

Such a set of hypotheses can be tested using what is known as an **Analysis of Variance** (or **ANOVA**, for short). Here's the main idea of how ANOVA works. Suppose we have samples (potentially of different sizes) from each population. Even if all populations have the same means, we wouldn't be surprised in our samples had slightly different observed sample means. This is because there will be some baseline variability due to chance. What ANOVA seeks to do is compare the variances within and across samples and see whether or not the overall variability exceeds what we would expect due to chance alone (which would lead credence *away* from the null, that the populations all have the same mean).

In the interest of time, I'll bypass the theoretical derivations of ANOVA and jump straight to how we can perform an ANOVA in `R`. There are actually a couple of functions which can be used for conducting an ANOVA - we'll use the function `aov()` [and we'll return to ANOVA in a future lecture].

One thing I want to make very clear is the fact that the alternative hypothesis in ANOVA is **NOT** $H_A: \mu_1 \neq \mu_2 \neq \cdots \neq \mu_k$. Rather, the alternative is simply that *at least one* of the group means differs from the rest.

As a concrete example, consider the following (fictional) situation. Suppose we want to determine whether or not the average scores of students on a particular exam differ significantly based on class standing (i.e. freshmen, sophomore, junior, senior). Additionally, suppose we collect the following (fictional) data:

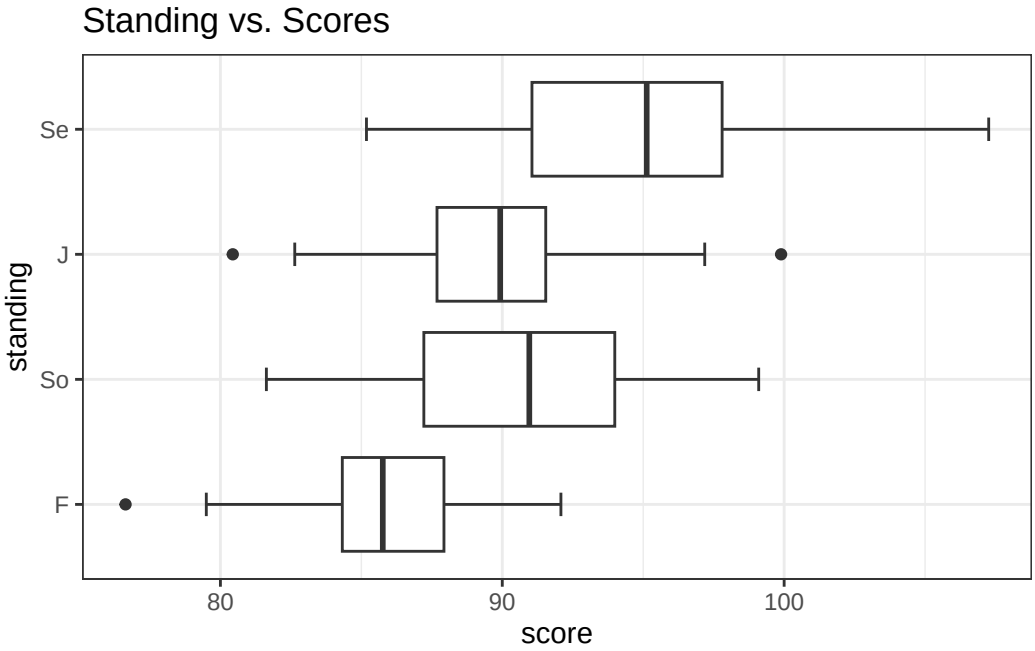
```
freshmen <- rnorm(50, 85, 3)
sophomores <- rnorm(60, 90, 5)
juniors <- rnorm(70, 90, 4)
seniors <- rnorm(40, 95, 5)

scores <- data.frame(
  score = c(freshmen, sophomores, juniors, seniors),
  standing = factor(
    c(rep("F", length(freshmen)),
                rep("So", length(sophomores)),
                rep("J", length(juniors)),
                rep("Se", length(seniors))
```

```
                ),
    ordered = T,
    levels = c("F", "So", "J", "Se")
  )
)
```

As a first pass, we can generate a side-by-side boxplot:

```
scores %>%
  ggplot(aes(y = standing,
             x = score)) +
  geom_boxplot(staplewidth = 0.25) +
  theme_bw() +
  ggtitle("Standing vs. Scores")
```



Based on the boxplot alone, it looks like there are some clear differences in the scores across the different class standings. To formally test this using an ANOVA, we use:

```
aov(score ~ standing, data = scores) %>% summary()
```

```
             Df Sum Sq Mean Sq F value Pr(>F)
standing      3   1820   606.7    39.2 <2e-16 ***
Residuals   216   3343    15.5
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We'll discuss the different components of the output in a few lectures (after we discuss regression). For now, focus on the `Pr(>F)` column - this is (essentially) a p-value of the hypotheses posited at the start of this section.

One thing to note is that ANOVA is very prone to the effects of Multiple Testing. Indeed, when we test the null $H_0 : \mu_1 = \cdots = \mu_k$, we are actually simultaneously testing $\binom{k}{2}$ hypotheses! Hence, a simple way to mitigate against the effects of multiple testing in an ANOVA is to divide the overall level of significance by $\binom{k}{2}$ - we call this the **Bonferroni Correction.**

! Question 4

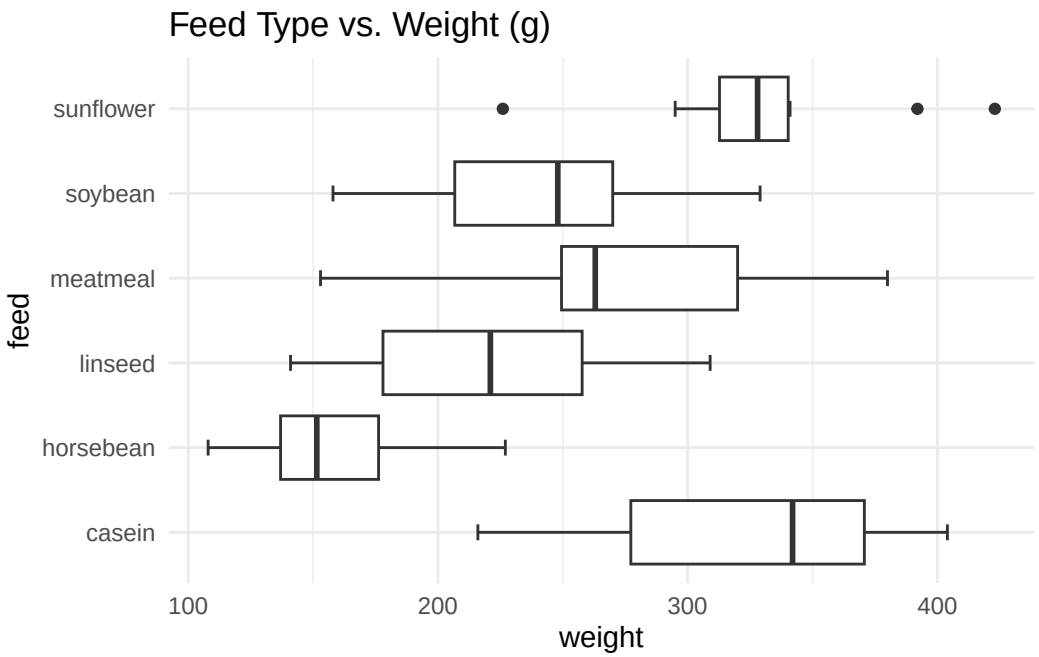
One of the datasets built into `R` is called `chickwts`, and contains the weights of various chickens placed on one of six different feed supplements.

a) Generate a side-by-side boxplot of the weight (in grams) vs. feed type. Based on the graph, does there appear to be a difference in average (mean) weight across the different feed types?

b) Conduct an ANOVA to test whether there is a statistically significant difference in average chick weights across the different feed types. Be sure to use a Bonferroni correction.

Solution:

```
## replace this line with your code;
## feel free to add more code chunks as you see fit.
chickwts %>%
  ggplot(aes(x = weight, y = feed)) +
  geom_boxplot(staplewidth = 0.25) +
  theme_minimal() +
  ggtitle("Feed Type vs. Weight (g)")
```



Based on this boxplot, it appears as though chick weights do differ across the different feed types. To formally test this with an ANOVA, we use

```
aov(weight ~ feed, chickwts) %>% summary()
```

```
            Df Sum Sq Mean Sq F value   Pr(>F)
feed         5 231129   46226   15.37 5.94e-10 ***
Residuals   65 195556    3009
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Sure enough, the p-value is `5.94e-10` (i.e. 5.94×10^{-10}), which is much smaller than a 0.05 level of significance meaning:

> At a 5% level of significance, there is evidence to suggest that there exists a difference in average (mean) chick weight across the different feed types.

Answer Check:

There is no autograder for this question.

Crucially, one of the assumptions of ANOVA is that each population $\mathcal{P}_i$ is normally distributed. If this assumption is violated, ANOVA can produce incorrect results. An easy way to check for normality is - you guessed it - QQ-plots!

If the normality assumption is violated, it may be prudent to apply a **nonparametric test**. We'll discuss nonparametris in more detail in a future lab; for now, I'll introduce the **Kruskal-Wallis Test** which is essentially an extension of ANOVA to non-normal populations. In `R`, we carry out the Kruskal-Wallis test using the function `kruskal.test()`.

! Question 5

Redo the analysis in Question 4(b) above, this time using a Kruskal-Wallis test. Comment on any differences between the result of this test and the result of the ANOVA from before.

Solution:

```
## replace this line with your code;
## feel free to add more code chunks as you see fit.
kruskal.test(weight ~ feed, chickwts)
```

```
    Kruskal-Wallis rank sum test

data:  weight by feed
Kruskal-Wallis chi-squared = 37.343, df = 5, p-value = 5.113e-07
```

Once again, the p-value is very small, leading us to reject the null in favor of the alternative. That is, at a 5% level of significance, there is sufficient evidence to suggest that there exists a difference in average (mean) chick weight across the different feed types.

Answer Check:

There is no autograder for this question.

Part IV: Two-Sample *t*-Test

Finally, let us return to the notion of a **two-sample *t*-test.** In certain cases, it may be desired to test whether or not two populations have the same mean. For example, we might ask ourselves: is the true average (mean) commute time of all Los Angelites the same as the true average (mean) commute time of New Yorkers?

More mathematically, consider two populations $\mathcal{P}_1$ (with mean μ_1 and standard deviation σ_1) and $\mathcal{P}_2$ (with mean μ_2 and standard deviation σ_2). The null hypothesis we wish to test can be formulated as

$$H_0: \ \mu_1 = \mu_2$$

and some possible alternatives are:

- $H_A: \ \mu_1 < \mu_2$ (what `R` calls **`less`**)
- $H_A: \ \mu_1 > \mu_2$ (what `R` calls **`greater`**)
- $H_A: \ \mu_1 \neq \mu_2$ (**two-sided**)

It's customary to reparametrize the null and alternative hypotheses to be in terms of parameter *differences*:

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

and

- $H_A: \ \mu_1 - \mu_2 < 0$ (what `R` calls **`less`**)
- $H_A: \ \mu_1 - \mu_2 > 0$ (what `R` calls **`greater`**)
- $H_A: \ \mu_1 - \mu_2 < 0$ (**two-sided**)

The reason we do so is, if we view $\delta := \mu_1 - \mu_2$ as its own parameter, our test can be rephrased as a test solely on δ - that is, we can effectively treat the problem as a one-sample problem (which we are now very familiar with).

Now, consider a sample $X \sim \mathcal{P}_1$ of size n_1 and $Y \sim \mathcal{P}_2$ of size n_2 (note that we are allowing our two samples to be of different sizes!). An unbiased estimator for δ is $\Delta := \overline{X}_{n_1} - \overline{Y}_{n_2}$, and hence it makes sense to formulate a test statistic to be in terms of this difference. Note that, if we assume independence both within our samples and across our samples,

$$\begin{aligned}\text{Var}(\Delta) &:= \text{Var}\left(\overline{X}_{n_1} - \overline{Y}_{n_2}\right) \\ &= \text{Var}\left(\overline{X}_{n_1}\right) + \text{Var}\left(\overline{Y}_{n_2}\right) = \frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}\end{aligned}$$

Hence, a natural test statistic (assuming *unknown* population standard deviations) is

$$\text{TS} := \frac{\overline{X}_{n_1} - \overline{Y}_{n_2}}{\sqrt{\frac{S_X^2}{n_1} + \frac{S_Y^2}{n_1}}} \tag{1}$$

where S_X^2 and S_Y^2 denote the sample variances of our samples X and Y, respectively. It turns out that the exact distribution of TS is unknown, but very well-approximated by a $t-$distribution with degrees of freedom given by the **Satterthwaite Approximation**:

$$\text{df} = \text{round}\left\{ \frac{\left[\left(\frac{s_X^2}{n_1}\right) + \left(\frac{s_Y^2}{n_2}\right)\right]^2}{\frac{\left(\frac{s_X^2}{n_1}\right)^2}{n_1 - 1} + \frac{\left(\frac{s_Y^2}{n_2}\right)^2}{n_2 - 1}} \right\}$$

where s_X^2 and s_Y^2 denote the observed instances of S_X^2 and S_Y^2, respectively, and round$(\cdot)$ denotes the rounding function [e.g. round$(4.2) = 4$]

! Question 6

Consider the following vectors `x` and `y`, and interpret them as two samples from two different populations

```
x <- c(1, 2, 3, 4, 5)
y <- c(2, 2, 3, 3, 4, 5)
```

First conduct a two-sided t-test **by hand** (i.e. using only basic `R` functions and **NOT** using `t.test()`). Then, reconduct your test using `t.test()`, and compare results.

Solution:

```
## replace this line with your code
xbar <- mean(x)
ybar <- mean(y)

sx2 <- var(x)
sy2 <- var(y)
n1 <- length(x)
n2 <- length(y)

ts <- (xbar - ybar) / sqrt((sx2 / n1) + (sy2 / n2))

numerator <- ((sx2 / n1) + (sy2 / n2))^2
denom <- ((sx2 / n1)^2 / (n1 - 1)) + ((sy2 / n2)^2 / (n2 - 1))
```

The observed value of the test statistic is -0.1953662, and the degrees of freedom are 7.2679146. Compare this with the output of `t.test()`:

```
t.test(x, y, alternative = "two.sided")
```

```
    Welch Two Sample t-test

data:  x and y
t = -0.19537, df = 7.2679, p-value = 0.8505
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2.168948  1.835615
sample estimates:
mean of x mean of y
 3.000000  3.166667
```

Answer Check:

There is no autograder for this question.

Submission Details

Congrats on finishing this PSTAT 100 lab! Please carry out the following steps:

Submission Details

1) Check that all of your tables, plots, and code outputs are rendering correctly in your final `.pdf`.
2) Check that you passed all of the test cases (on questions that have autograders). You'll know that you passed all tests for a particular problem when you get the message "All tests passed!".
3) Submit **ONLY** your `.pdf` to Gradescope. Make sure to **match ALL pages to the ONE question on Gradescope**; failure to do so will incur a penalty of 0.1 points.