

SOC-GA 2332 Intro to Stats Lab 3

Risa Gelles-Watnick

9/19/2025

Logistics & Announcement

- **Problem Set 1** is due on Tue Sep. 26th, 11:59 pm.
 - Make sure to comment on your code. This will help me give you feedback and is also just good coding practice.
 - Any questions so far?
- **Lab on October 24th** will be at NYU Academic Resource Center, Room ARC_LL01 because of a fire alarm check.
- **Final Project**
 - Progress steps will be built into the problem sets
 - Will go over some steps for getting started in the next lab
 - Intro to [Overleaf](#) (+ RMarkdown template)

```
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)

# Import data
weight_df <- read.csv("data/weight.csv")
```

Part 1 Review: Population and Sample Exercise (10 minutes)

1. Write down the formula you use to calculate the following sample statistics (assume your sample size = n):
 - Sample mean:
 - Sample variance:
 - Sample standard deviation:
 - Standard error of sample mean:
 - 95% confidence interval of the population mean:
2. You have collected a sample of 25 on tree heights in Brooklyn. The sample mean is 23 inches and sample variance is 4 inches.
 - What is the point estimate of the population mean?
 - What is the 95% confidence interval of the population mean (round to 2 decimal places)?
 - What is the 95% confidence interval of the population mean if the sample size is 10,000 (round to 2 decimal places)?

Part 2: Hypothesis and Significance Test

First, let's review the standard steps for conducting a significance test:

2.1 The standard procedure of a significance test

1. Formulate our research question in the null and alternative hypotheses
2. Select a significance level (α) (in social science, usually $\alpha = 0.05$)
3. Select which test statistics to use (for small samples, we use the ***t* test statistics**)
4. If you are collecting first-hand data, select a sample size that provides you with sufficient statistical power
5. Derive the **sampling distribution of the test statistic** under the assumption that **the null hypothesis is true**
 - For the *t* test statistics, its sampling distribution is approximately the Student *t* distribution with $n - 1$ degrees of freedom
 - When n gets larger (usually $n > 30$), the *t* distribution is approximately a standard normal distribution (see graph below)
 - The *t* test statistic formula is: $t = \frac{\bar{y} - \mu_0}{se}$ (μ_0 is the population mean in the null hypothesis)

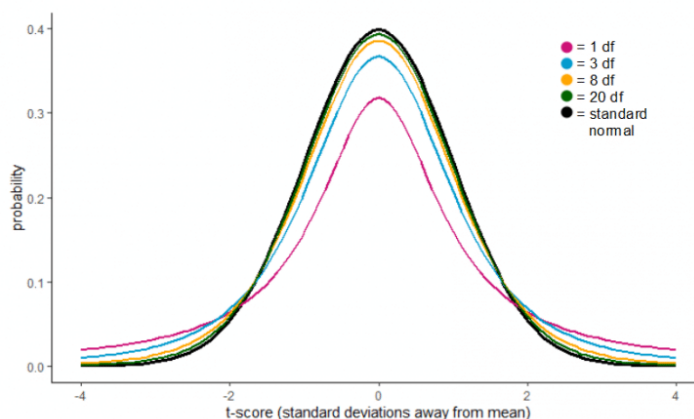


Figure 1: t distribution 1

6. (A) Derive the critical value of *t* and your rejection region according to the null hypothesis
 - The critical value of *t* (CV_t) is the value beyond which we will regard our observed *t* as “unusual”
 - The rejection region will be $(-\infty, -|CV_t|) \cup (|CV_t|, \infty)$.
 - For samples with a $df \geq 100$, the critical value of *t* is **1.96** for a significance level at 0.05. The rejection region is $(-\infty, -1.96) \cup (1.96, \infty)$
 - For samples with a $df \leq 100$, you can use the “t Distribution Critical Values” table in your textbook to find out the critical value and rejection region:
 - For a **two-tailed test** that have a significance level at 0.05, we find values from the $t_{.025}$ column
 - For a **one-tailed test** that have a significance level at 0.05, we find values from the $t_{.050}$ column
 - You can also use the `qt()` function in R to find out the critical value:
 - To find out critical value of *t* for a **two-tailed test**, use `qt(p = 0.5*your_alpha, df = your_degree_of_freedom)`

- To find out critical value of t for a **one-tailed test**, use `qt(p = your_alpha, df = your_degree_of_freedom)`
- *Note:* the `qt()` function is the quantile function for the Student t distribution in base R that gives the t value based on the percentile you input

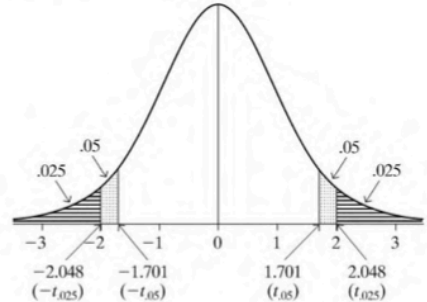


FIGURE 5.5: t Distribution with $df = 28$

Figure 2: t distribution 2

- (B) Alternatively, you can calculate the p -value of your observed t statistic
 - p -value is the probability that the test statistic equals to (or is more extreme than) what we observed
 - To find out the **two-tail** p -value, use `2*pt(q = observed_t, df = your_degree_of_freedom, lower.tail = FALSE)`
 - To find out the **one-tail** p -value, use `pt(q = observed_t, df = your_degree_of_freedom, lower.tail = FALSE)`

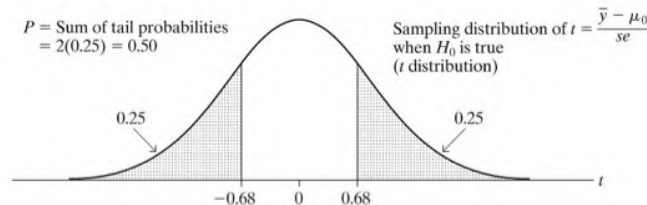


FIGURE 6.3: Calculation of P -Value when $t = 0.68$, for Testing $H_0: \mu = \mu_0$ against $H_a: \mu \neq \mu_0$. The P -value is the two-tail probability of a more extreme result than the observed one.

Figure 3: t distribution 3

- Make a conclusion about whether to reject the null hypothesis
 - You can use this [online tool](#) to visualize a t -test

Exercise (10 minutes)

With $\mu_0 = 0$, $\bar{y} = 1.54$, sample $n = 27$, $s = 3.25$, derive:

(1) The t test statistic

code here

(2) The critical value of t given H_0 is true

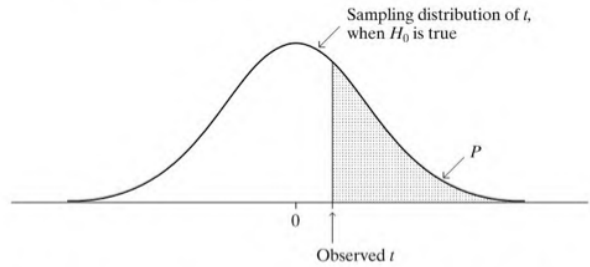


FIGURE 6.4: Calculation of P -value in Testing $H_0: \mu = \mu_0$ against $H_a: \mu > \mu_0$. The P -value is the probability of values to the right of the observed test statistic.

Figure 4: t distribution 4

code here

(3) Your rejection region

formulahere

formulahere

(4) p -value

code here

(5) Your conclusion of the significance test

2.2 One-sample t-test using R

- **Question:** When do you use one-sample t-test?
- R provides a simple function `t.test()` to perform hypothesis testing using the t test statistics
- For example, the data object `weight_df` we just imported records the weight change of rats who researchers put on different diets, and we want to know which diet increases the rats' weight most.

```
## check data
head(weight_df)
```

```
##   subj diet before after change
## 1    1   b  80.5  82.2    1.7
## 2    2   b  84.9  85.6    0.7
## 3    3   b  81.5  81.4   -0.1
## 4    4   b  82.6  81.9   -0.7
## 5    5   b  79.9  76.4   -3.5
## 6    6   b  88.7 103.6   14.9
```

- Before performing any statistical test, it will be useful to (1) check the descriptive statistics and (2) plot the variables of interest.

```
## check descriptive statistics of all variables
summary(weight_df)
```

```
##      subj      diet      before      after
## Min.   : 1.00 Length:72 Min.   :70.00 Min.   : 71.30
```

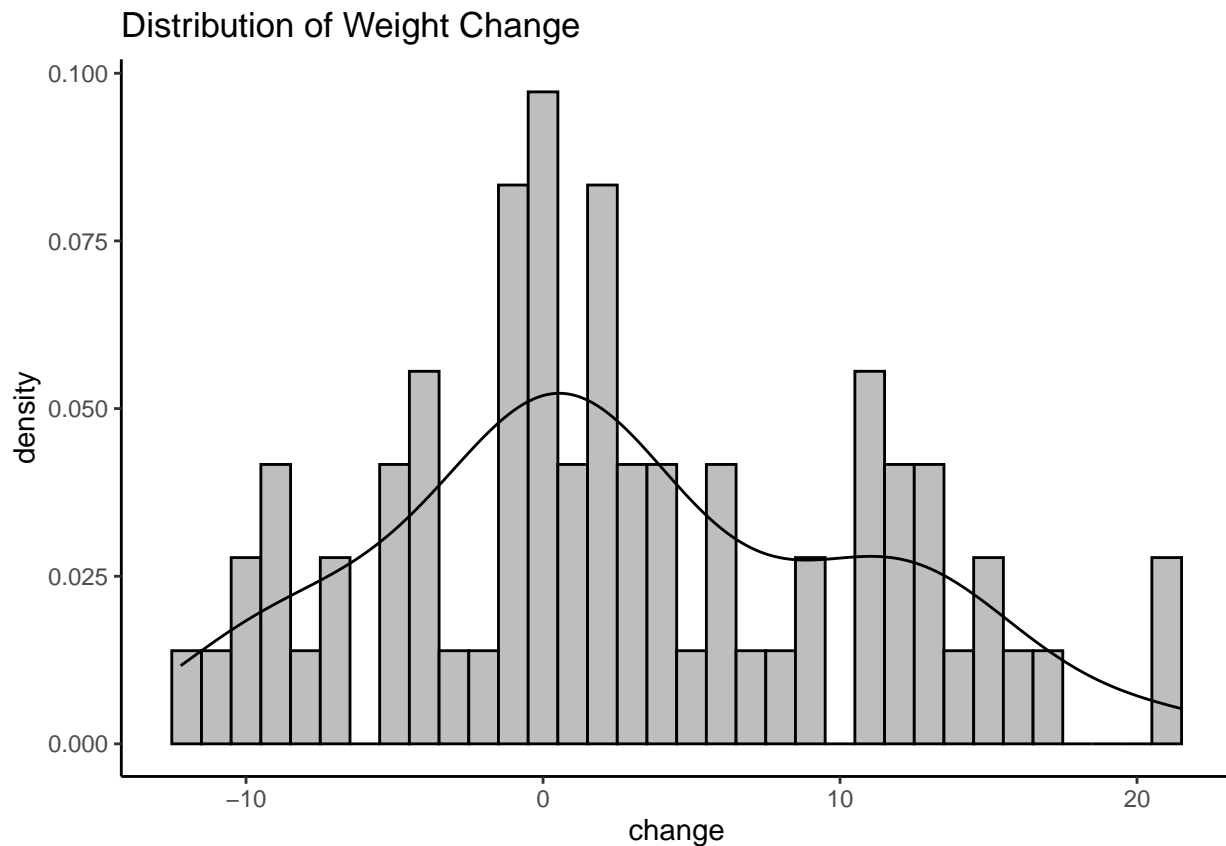
```
## 1st Qu.:18.75   Class :character   1st Qu.:79.60   1st Qu.: 79.33
## Median :36.50   Mode  :character   Median :82.30   Median : 84.05
## Mean   :36.50                   Mean  :82.41   Mean  : 85.17
## 3rd Qu.:54.25                   3rd Qu.:86.00   3rd Qu.: 91.55
## Max.   :72.00                   Max.   :94.90   Max.   :103.60
##      change
## Min.    :-12.200
## 1st Qu. : -2.225
## Median  :  1.650
## Mean    :  2.764
## 3rd Qu. :  9.100
## Max.    : 21.500
```

```
## plot histogram and density curve
```

```
weight_df %>%
```

```
  ggplot(aes(x = change, y = ..density..)) +
    geom_histogram(binwidth = 1, fill = "grey", color = "black") +
    geom_density() +
    labs(title = "Distribution of Weight Change") +
    theme_classic()
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



- We test:

$$H_0 : \mu_{\text{change}} = 0$$

- the mean weight change is 0 against:

$$H_{a1} : \mu_{\text{change}} \neq 0$$

- the mean weight change is not 0, a two-tailed test:

$$H_{a2} : \mu_{\text{change}} > 0$$

- the mean weight change is larger than 0, a one-tailed test, using the following code:

```
## mean of weight change
mean(weight_df$change)

## [1] 2.763889

# ---- one sample two-tail t-test ---- #
two_tail_t <- t.test(
  weight_df$change,          # the sample value vector that you want to test
  mu = 0,                   # mean given by your null hypothesis
  alternative = "two.sided", # direction of alternative hypothesis
  conf.level = 0.95         # significance level
)

## extract test statistic
two_tail_t$statistic

##          t
## 2.93757

## extract p-value
two_tail_t$p.value

## [1] 0.004457718

## extract the confidence interval of the mean
two_tail_t$conf.int

## [1] 0.8878354 4.6399424
## attr(,"conf.level")
## [1] 0.95

## display full result
two_tail_t

##
## One Sample t-test
##
## data:  weight_df$change
## t = 2.9376, df = 71, p-value = 0.004458
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.8878354 4.6399424
## sample estimates:
## mean of x
## 2.763889
```

```
# ---- one sample one-tail t-test ---- #
t.test(weight_df$change,
       mu = 0,
       alternative = "greater",
       conf.level = 0.95)

##
## One Sample t-test
##
## data: weight_df$change
## t = 2.9376, df = 71, p-value = 0.002229
## alternative hypothesis: true mean is greater than 0
## 95 percent confidence interval:
##  1.195825      Inf
## sample estimates:
## mean of x
##  2.763889
```

Exercise (5 minutes)

The research lab claims that these diets will lead to a weight increase of 4 lbs. Use the `weight_df` data and with $\alpha = 0.05$, perform both a two-tailed and a one-tailed test:

$$H_0 : \mu_{\text{change}} = 4$$

against

$$H_{a1} : \mu_{\text{change}} \neq 4 \text{ and } H_{a2} : \mu_{\text{change}} < 4$$

- Report your hypothesis testing result.
- *Hint:* Make sure you put correct arguments for your `t.test()` function! (Are you testing for “two.sided”, “less”, or “greater”? What’s your mu?)

```
# code here
```

Part 3: Comparing the Mean of Two Groups (Two-sample t-test)

3.1 Two independent samples

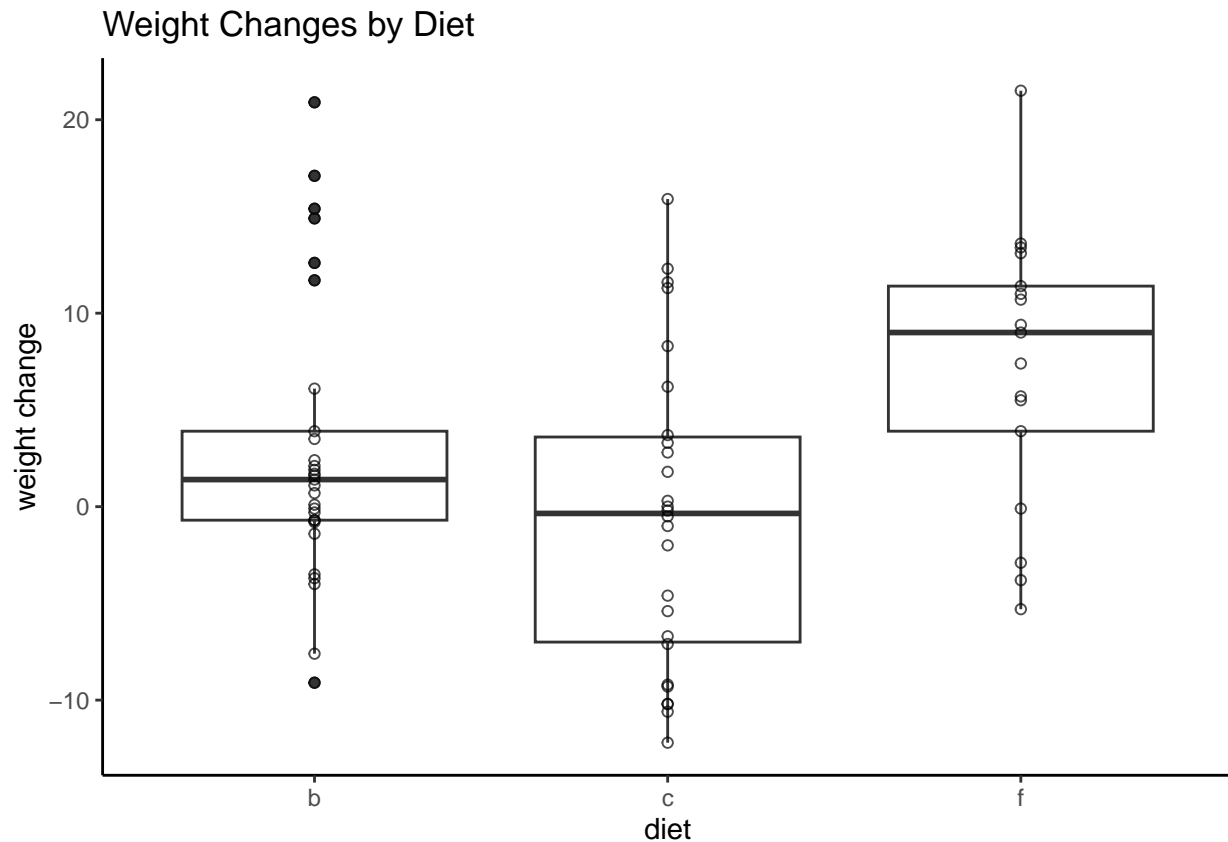
In the case of comparing the mean of two independent samples, we follow the same procedures as the one sample *t*-test, except the statistics for finding the *t* test statistics change. We will not review all the formulas here. Please review lecture slides and the textbook.

Using R, we can perform a two-sample *t*-test by using the same `t.test()` function but adding a second sample mean vector.

For example, in studying the rats, three different diets are used. We can plot a boxplot to visualize how weight changes differ across these diets

```
## box plot
weight_df %>%
  ggplot(aes(x = diet, y = change)) +
  geom_boxplot() +
  geom_point(shape = 1, alpha = 0.7) +
  labs(title = "Weight Changes by Diet",
```

```
y = "weight change") +  
theme_classic()
```



It looks like diet f tends to result in a higher weight increase compared to other diet. Let's use a two-sample t-test to see if the mean weight change in diet f is statistically different from that in diet c:

$$H_0 : \mu_f - \mu_c = 0$$

against

$$H_a : \mu_f - \mu_c \neq 0$$

```
## filter data for each diet  
weight_f <- weight_df %>% filter(diet == "f")  
weight_c <- weight_df %>% filter(diet == "c")  
  
# ---- two-group independent two-tailed t-test ---- #  
t.test(  
  x = weight_f$change,      # mean value vector from the first sample  
  y = weight_c$change,      # mean value vector from the second sample  
  mu = 0,                  # mean difference given by your null hypothesis  
  alternative = "two.sided" # direction of alternative hypothesis  
)
```

```
##  
## Welch Two Sample t-test
```



```
##
## data: weight_f$change and weight_c$change
## t = 3.2992, df = 36.979, p-value = 0.002152
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## 2.976597 12.452815
## sample estimates:
## mean of x mean of y
## 7.264706 -0.450000
```

Note: The degrees of freedom of the t -distribution will be $n_0 + n_1 - 2$ **if the population variance of the two groups is equal**. This is often not a very realistic scenario. Because of this, Welch's approximation (which we will not define here, but can be found [here](#) if you are curious) is often used for the degrees of freedom of the t distribution. This is, in fact, the default option in the `t.test()` function that we use in R.

3.2 Two dependent samples

In fact, our example in the one-sample t -test in Part 2 is a two dependent sample t -test. For two dependent sample t -test, you can always create a new variable equal to the difference between the two dependent samples, like what we did in Part 2; or you can use the `t.test()` function and set the argument `paired = TRUE`.

For example, in the `weight_df` data, if we want to test whether the mean weight before the treatment is different from the mean weight after the treatment:

$$H_0 : \mu_{\text{before}} - \mu_{\text{after}} = 0$$

against

$$H_a : \mu_{\text{before}} - \mu_{\text{after}} \neq 0$$

```
# ---- two-group dependent two-tailed t-test ---- #
t.test(
  x = weight_df$before,      # mean value vector from the first sample
  y = weight_df$after,      # mean value vector from the second sample
  mu = 0,                   # mean difference given by your null hypothesis
  paired = TRUE,            # dependent samples
  alternative = "two.sided", # direction of alternative hypothesis
  conf.level = 0.95         # significance level
)
```

```
##
## Paired t-test
##
## data: weight_df$before and weight_df$after
## t = -2.9376, df = 71, p-value = 0.004458
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
## -4.6399424 -0.8878354
## sample estimates:
## mean difference
## -2.763889
```

Part 3 Exercise (5 minutes)

Perform a two-sample two-tailed t -test and interpret the results for the difference between diet b and c:

$$H_0 : \mu_b - \mu_c = 0$$

against

$$H_a : \mu_b - \mu_c \neq 0$$

```
# code here
```