STAT INFERENCE

Research Methods in Psychology II ▪ Department of Psychology ▪ Colorado State University

BY THE END OF THIS UNIT YOU WILL:

1. Understand the ideas behind the central limit theorem and the sampling distribution.
2. Know how to calculate and interpret the standard error of the mean.
3. Know how to calculate and interpret a confidence interval around a mean.
4. Know how to conduct and interpret a one-sample t-test to determine if a mean is significantly different from 0.

What are Inferential Statistics?

When we pose research questions, we typically have some population in mind. Very rarely does a scientist study the entire population, rather, she draws a random sample of cases from the population. She studies this sample in order to understand the phenomena of interest in the population, without having to actually study every case in the population. In this way, inferential statistics are used 1.) to generalize from the sample to the larger population, 2.) to test hypotheses, and 3.) to make predictions (and understand the accuracy of those predictions).

Typically there is a parameter, or set of parameters, in the population (e.g., mean, correlation, regression coefficient, variance explained) that we want to estimate using our sample.
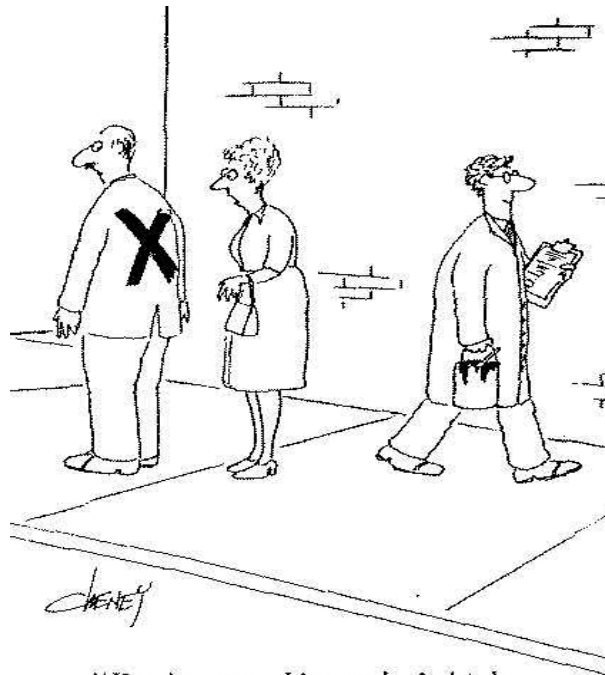
"Don't move, Howard...I think you've just been randomly sampled."

© Kimberly L. Henry, Ph.D. ▪ kim.henry@colostate.edu

AN EXAMPLE

Successful Weight Loss

A research team is interested in assessing the relationship between caloric deficit and pounds lost over the course of a 1 month weight loss program designed for men age 40 to 50 in CO. 100 obese men were randomly selected to participate. The team measured the following variables:

1. **lbslost** = pounds lost after one month on the weight loss program
2. **caldeff** = caloric deficit over the course of the program (expressed in 1000 calories). A caloric deficit is a state in which you are burning more calories than you eat.
3. **selfeff** = self efficacy for weight loss at the start of the program

Load Libraries & Import Data

To begin, we will simply examine the pounds lost (lbslost) by each participant in the study. The data are in the Unit 3 folder in dropbox, the file is called wtloss.csv. There is a notebook created for this Unit, it's called InferenceExploration_Notebook. Please copy the data and this notebook file into your MyClassActivities folder. Next, open up your MyClassActivities R project, and then the InferenceExploration_Notebook Rmd file.

To begin, execute the following two codechunks.

```
---
title: "R Notebook for Unit 3"
subtitle: "Exploration of the Principles of Statistical Inference"
output: html_notebook
---


# Load libraries
```{r, message = FALSE}

library(tidyverse)
library(descriptr)
library(mosaic)

```


Import data
```{r, message = FALSE}

wtloss <- read_csv(file = "wtloss.csv")

screener(wtloss)

```
```

```
--------------------------------------------------------------------
|  Column Name  |  Data Type  |  Levels  |  Missing  |  Missing (%)  |
--------------------------------------------------------------------
|    lbslost    |   numeric   |    NA    |     0     |      0        |
|    caldef     |   numeric   |    NA    |     0     |      0        |
|    selfeff    |   numeric   |    NA    |     0     |      0        |
|      id       |   integer   |    NA    |     0     |      0        |
--------------------------------------------------------------------

Overall Missing Values              0
Percentage of Missing Values        0 %
Rows with Missing Values            0
Columns With Missing Values         0
```

Get Some Descriptive Statistics

`summary_stats(wtloss$lbslost)`

```
                    Univariate Analysis

N                    100.00    Variance                4.40
Missing                0.00    Std Deviation           2.10
Mean                   3.06    Range                  10.76
Median                 3.14    Interquartile Range     3.07
Mode                  -3.19    Uncorrected SS       1374.33
Trimmed Mean           3.11    Corrected SS          435.83
Skewness              -0.33    Coeff Variation        68.49
Kurtosis              -0.22    Std Error Mean          0.21

                        Quantiles

         Quantile                        Value

         Max                              7.56
         99%                              7.30
         95%                              5.58
         90%                              6.04
         Q3                               4.70
         Median                           3.14
         Q1                               1.62
         10%                              0.30
         5%                              -0.07
         1%                              -1.58
         Min                             -3.19

                      Extreme Values

         Low                           High

  Obs           Value         Obs             Value
  93        -3.1904565         13          7.5647785
  66        -1.5595839         24          7.2924021
  80        -1.0083407         22          6.3813312
  71        -0.7865195         43          6.3052286
  41         -0.75858           4          6.2482989
```

<u>Visualize Pounds Lost in Accordance with the Empirical Rule</u>

```
ggplot(wtloss, aes(x = lbslost)) +
  geom_histogram(binwidth = .5, color = "grey") +
  geom_vline(xintercept = mean(wtloss$lbslost), lwd = 2, color="red") +
  geom_vline(xintercept = mean(wtloss$lbslost) + 2*sd(wtloss$lbslost), lwd = 2, color="blue") +
  geom_vline(xintercept = mean(wtloss$lbslost) - 2*sd(wtloss$lbslost), lwd = 2, color="blue") +
  labs(title = "Distribution of pounds lost for the 100 men in the study",
      subtitle = "red line represents sample mean, blue lines represents +/- 2 standard deviations from the mean",
      x = "pounds lost")
```

This is just a run of the mill histogram, but it's been enhanced to include 3 vertical lines on the graph. First, we add a red vertical line at the mean of lbslost (3.06). Note that I could have simply typed 3.06, instead of mean(wtloss$lbslost), and this line of code would work in the same way. Second, two blue lines are added to denote 2 standard deviations above and below the mean. Note that again here, I could have computed the values corresponding to 2 standard deviations surrounding the mean (-1.13 for below mean, 7.26 for above mean), and simply listed these numbers after "xintercept =" for the corresponding geom_vline.

If the distribution of this variable is normal, then according to the empirical rule, 95% of the observations should fall within plus or minus two standard deviations of the mean.

Distribution of pounds lost for the 100 men in the study
red line represents sample mean, blue lines represents +/- 2 standard deviations from the mean

<u>Visualize Pounds Lost in Accordance with the Empirical Rule, 1.96 vs. 2</u>

The empirical rule states that ~ 95% of the scores from a standard normal distribution are within 2 standard deviations of the mean (so between a z-score of -2 and a z-score of +2). This value of 2 is just a close approximation, in fact, the exact value is 1.959964. How do we get that? We use the qnorm function in R — which is the quantile function for the cumulative distribution function. In this case, we want to know the precise z-score for the percentiles of 2.5 and 97.5 of a standard normal distribution, which leaves 5% in the tails (2.5% in the lower tail, 2.5% in the upper tail), and encompasses the middle 95% of the distribution. Using qnorm to find a quantile answers the question "What is the z-score associated with the a certain percentile score?" Here, I use the combine function (c) to indicate that I want two quantile scores at once. These are the precise values, and we will exchange 2 for this value in the plot.

```
qnorm(c(.025, .975))          [1] -1.959964  1.959964
```

```
ggplot(wtloss, aes(x = lbslost)) +
  geom_histogram(binwidth = .5, color = "grey") +
  geom_vline(xintercept = mean(wtloss$lbslost), lwd = 2, color="red") +
  geom_vline(xintercept = mean(wtloss$lbslost) + 1.959964*sd(wtloss$lbslost), lwd = 2, color="blue") +
  geom_vline(xintercept = mean(wtloss$lbslost) - 1.959964*sd(wtloss$lbslost), lwd = 2, color="blue") +
  labs(title = "Distribution of pounds lost for the 100 men in the study",
      subtitle = "red line represents sample mean, blue lines represents +/- 1.96 standard deviations from the mean",
      x = "pounds lost")
```


Distribution of pounds lost for the 100 men in the study
red line represents sample mean, blue lines represents +/- 1.96 standard deviations from the mean

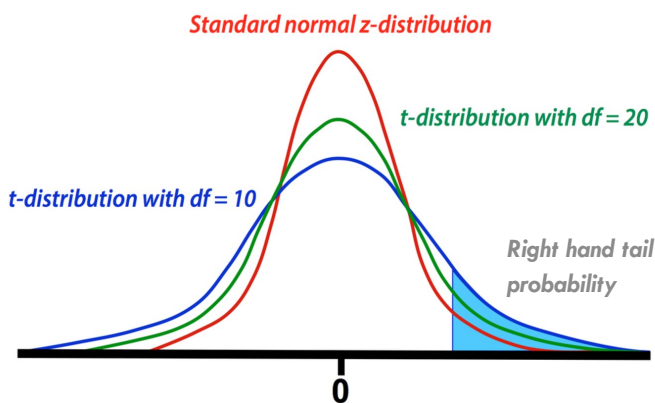# Visualize Pounds Lost, Now Use the Student's t-distribution

There is one more modification that we can apply. So far, we have been working with a standard normal distribution, also referred to as a z-distribution. Applying the principles of empirical rule and the cumulative distribution function for a standard normal distribution is appropriate if the sample is large or the population standard deviation is known. When these rules do not hold, we should use the Student's t-distribution. The Student's t-distribution is also symmetric and has a bell-shaped curve, but rather than a single curve, it's a family of curves based on the corresponding degrees of freedom (df) for the statistic under consideration. As the df increases, the Student's t-distribution approaches the standard normal distribution.

So, what are degrees of freedom. Imagine we have a variable measured for 5 people randomly selected from the population — for example, the weight loss of 5 men. Before estimating anything, each man's weight is free to take on any value. Now, imagine, we want to use these five men to estimate the mean weight loss in the population. You now have a constraint—the estimation of the mean. Once the mean is estimated, 4 of the 5 values are free to vary, but one is not. That is, if we know a mean, and the value of 4 of the cases, then the value of the 5th case is also known and not free to vary. With each test that we conduct this year, there will be a formula for calculating the degrees of freedom. When estimating the population mean, the formula for the degrees of freedom is simply $n - 1$, where n is the sample size. In our example, n = 100, so if we desire to use our sample to estimate the mean in the population, then we have 99 df left.

The R function qt gives the quantile function for the Student's t-distribution It answers the question "What is the t-score associated with the a certain percentile score?" Here, I use the combine function (c) to indicate that I want two quantile scores at once.

qt(c(.025, .975), df = 99)   `[1] -1.984217  1.984217`

Distribution of pounds lost for the 100 men in the study
red line represents sample mean, blue lines represents +/- 1.98 standard deviations from the mean

```
ggplot(wtloss, aes(x = lbslost)) +
  geom_histogram(binwidth = .5, color = "grey") +
  geom_vline(xintercept = mean(wtloss$lbslost), lwd = 2, color="red") +
  geom_vline(xintercept = mean(wtloss$lbslost) + 1.984217*sd(wtloss$lbslost), lwd = 2, color="blue") +
  geom_vline(xintercept = mean(wtloss$lbslost) - 1.984217*sd(wtloss$lbslost), lwd = 2, color="blue") +
  labs(title = "Distribution of pounds lost for the 100 men in the study",
       subtitle = "red line represents sample mean, blue lines represents +/- 1.98 standard deviations from the mean",
       x = "pounds lost")
```

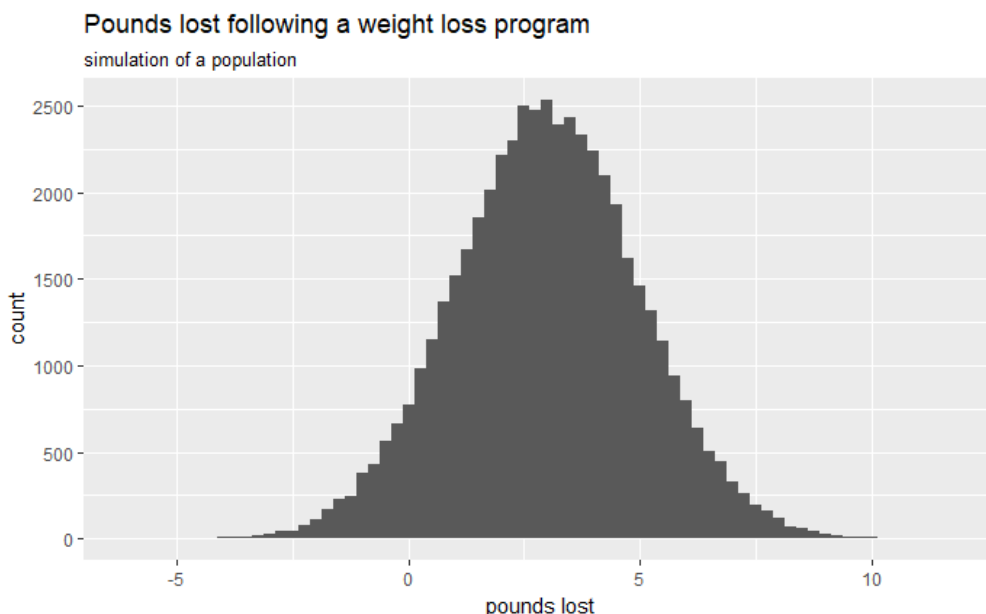## Our Sample is Just One Possible Sample

As diligent scientists, we carefully define our population, then draw a random sample from that population. In doing so, we must realize that the sample we draw is just one of many, many random samples that could have been drawn. When estimating a statistic, for example the mean pound's lost, we must realize that the statistic will undoubtedly vary from sample to sample, this is referred to as sampling variability. We must take this variability, and the associated uncertainty about the true population mean, into account. Let's simulate this process to get a feel for the process and issues.

Let's pretend that we had the resources to deliver the weight loss program to every single obese man age 40 to 50 in Northern Colorado. In this hypothetical scenario, we have data on every person in the population. In this population we will assert that the average pounds lost is 3.00, with a standard deviation of 2.00.

We will generate a population of size 50,000 using the rnorm function — which generates a set of random variables from a standard normal distribution with a given mean and standard deviation.

```
# generate some data
set.seed(8642)
p_lbslost <- rnorm(n=50000, m=3.00, sd=2.00)
my_pop <- data.frame(p_lbslost)

# plot the normal distribution
ggplot(my_pop, aes(x = p_lbslost)) +
  geom_histogram(binwidth = .25) +
  labs(title = "Pounds lost following a weight loss program", subtitle = "simulation of a population", x = "pounds lost")
```


This is a histogram of the 50,000 simulated scores for pounds lost for our hypothetical population.

Randomly Select 3 Samples from this Population

To begin let's draw three random samples from our hypothetical population to see how pounds lost varies.

```
set.seed(19657)
s1 <- sample_n(my_pop, 100, replace = TRUE)
s2 <- sample_n(my_pop, 100, replace = TRUE)
s3 <- sample_n(my_pop, 100, replace = TRUE)

my_samples <- bind_rows(s1, s2, s3, .id='sample')

my_samples %>%
  group_by(sample) %>%
  summarize(mean_in_sample = mean(p_lbslost))

ggplot(my_samples, aes(x = p_lbslost, group = sample, fill = sample)) +
  geom_density(alpha = .5) +
  labs(title = "Pounds lost following a weight loss program in the three samples", x = "pounds lost")
```
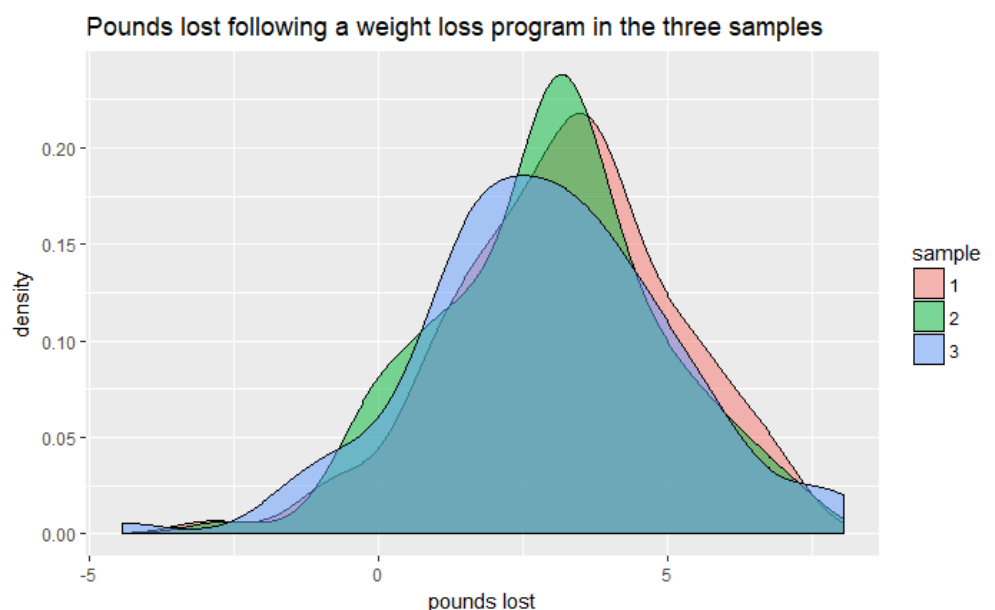
sample <chr>	mean_in_sample <dbl>
1	3.212966
2	2.952674
3	2.876155

The distribution of pounds lost in each of the samples is similar, but clearly different. We also see that each of the means is around 3, but also vary substantially.

Pounds lost following a weight loss program in the three samples

Randomly Select 10,000 Samples from the Population

Rather than just 3 samples, lets now select many, many random samples — 10,000 samples of size 100 is a good start.

```
set.seed(47256)
sample_means <- my_pop %>%
  rep_sample_n(size = 100, reps = 10000, replace = TRUE) %>%
  summarize(mean_lbslost = mean(p_lbslost), sd_lbslost = sd(p_lbslost))
```

	replicate	mean_lbslost	sd_lbslost
1	1	3.094337	1.956708
2	2	2.920634	2.165699
3	3	3.003978	2.165654
4	4	3.134027	1.774678
5	5	2.944227	1.947747
6	6	3.190237	1.951895
7	7	3.475803	2.098721
8	8	3.032388	1.863397
9	9	3.010078	1.776071
10	10	3.268840	1.981819
11	11	3.119842	2.176504
12	12	2.798634	2.121862
13	13	2.728742	1.779156
14	14	2.901259	2.263805
15	15	2.799985	1.813757
16	16	2.583812	2.115549
17	17	3.054015	2.027214
18	18	2.712764	2.255358
19	19	2.854545	2.089046
20	20	2.625643	1.756653
21	21	3.096244	1.990946
22	22	3.006533	1.894356
23	23	2.853870	2.034015
24	24	3.018812	1.694358
25	25	3.038068	2.121856

I calculated the mean and standard deviation of lbslost in each of the 10,000 random samples. Here I display these descriptive statistics for the first 25 samples.
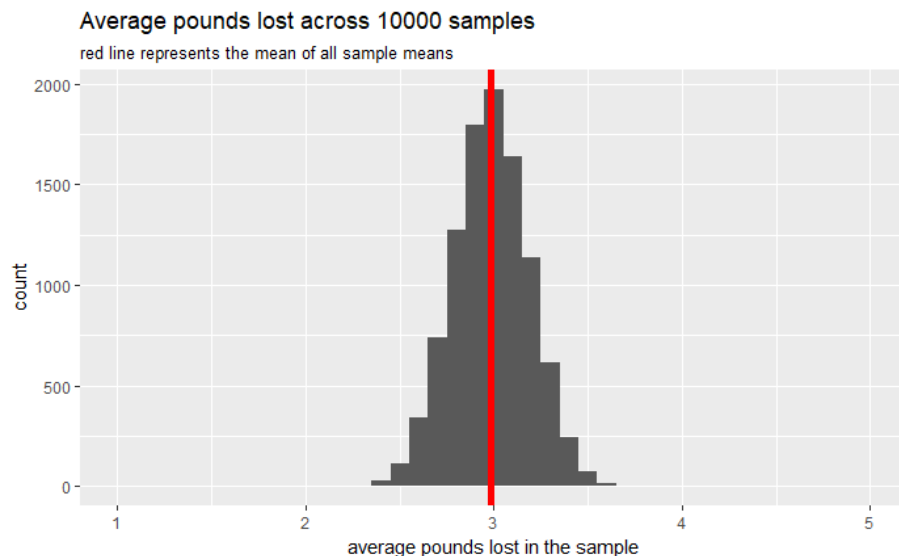
Describe the 10,000 Samples

```
# print mean and standard deviation of average pounds lost across the 10,000 samples
mean(sample_means$mean_lbslost)
sd(sample_means$mean_lbslost)

ggplot(sample_means, aes(x = mean_lbslost)) +
 geom_histogram(binwidth = .1) +
 geom_vline(xintercept = mean(sample_means$mean_lbslost), lwd = 2, color="red") +
 labs(title = "Average pounds lost across 10000 samples",
     subtitle = "red line represents the mean of all sample means", x = "average pounds lost in the sample") +
 scale_x_continuous(limits = c(1, 5))
```

[1] 2.985371
[1] 0.2002739

Using the summarized dataframe that represents the mean pounds lost for all 10,000 random samples, I calculated the mean and sd of the 10,000 means—so the mean of means.

Average pounds lost across 10000 samples
red line represents the mean of all sample means

This is a plot of the 10,000 means obtained by calculating the average pounds lost in each of the 10,000 random samples that we drew. This plot represents the sampling distribution for mean pounds lost. The sampling distribution is the distribution of a statistic of interest across all possible random samples. It describes how the statistic differs across random samples. You can see in the chart that the mean differs quite a bit across the random samples that we generated — from a low of 2.27 to a high of 3.68.   The sampling distribution is centered around the true population value (the average pounds lost in the population), and is normally distributed. The standard deviation of the sampling distribution is called the standard error of the statistic of interest, in this case the standard error of the mean. The degree of spread across the random samples speaks to the precision of our estimated statistic. A larger standard error relative to the mean indicates less precision (more variability across samples), a small standard error relative to the mean indicates more precision (less variability across samples).

The Central Limit Theorem (CLT)

The CLT dictates that provided the sample size is sufficiently large, the sampling distribution of a statistic is normally distributed, regardless of the distribution of the underlying variable (e.g., continuous, Poisson, binomial, even skewed, etc.). The definition of a sufficiently large sample size depends on the skewness of the distribution from which the random sample is drawn. ==If the distribution is symmetric, unimodal or continuous, then only a very small sample size is needed (e.g., 5 cases), if the distribution is skewed, then a larger sample size is needed (~30 or more cases).== Given this theorem, we can then use what we know about the empirical rule to calculate a confidence interval around the estimated statistic (i.e., the mean pounds lost). For example, we can multiply the standard error by 1.98 (so, 1.98 · .20) = .40, then add and subtract that quantity from the mean to arrive at an interval within which 95% of the means will fall. This range is 2.59, 3.38.

```
me <- 1.984217*0.2002739 # margin of error
lower <- 2.985371 - (1.984217*0.2002739) # lower bound of confidence limit
upper <- 2.985371 + (1.984217*0.2002739) # upper bound of confidence limit


sample_means <- sample_means %>%
  mutate(out_range = ifelse(mean_lbslost < lower | mean_lbslost > upper, 1, 0))


mean(sample_means$out_range)


# add 95% confidence band to our histogram
ggplot(sample_means, aes(x = mean_lbslost)) +
  geom_histogram(binwidth = .1) +
  geom_vline(xintercept = mean(sample_means$mean_lbslost), lwd = 2, color="red") +
  geom_vline(xintercept = lower, lwd = 2, color="blue") +
  geom_vline(xintercept = upper, lwd = 2, color="blue") +
  labs(title = "Average pounds lost across 10000 samples",
       subtitle = "red line represents the mean of all sample means, blue line represents +/- 1.98 SDs from this mean",
       x = "average pounds lost in the sample") +
  scale_x_continuous(limits = c(1, 5))
```
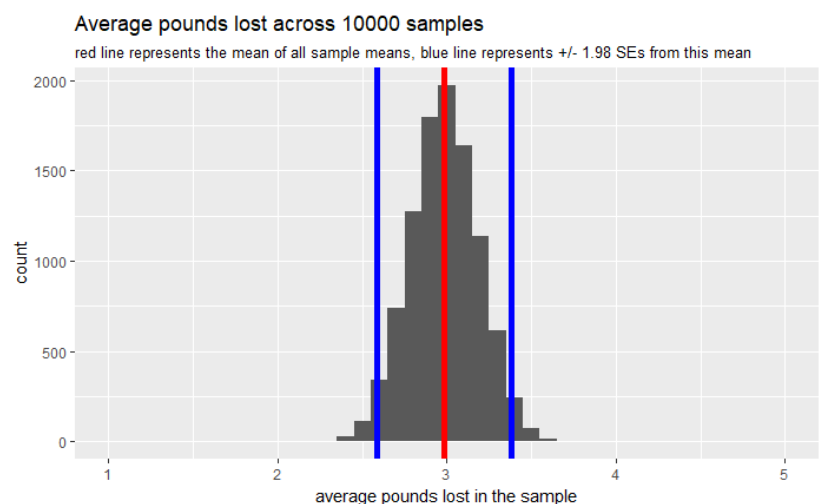
```
[1] 0.0475
```
Proportion of samples outside of +/- 1.98 SEs

95% percent of the samples produce an estimate of the mean that is within 1.98 standard errors of the true mean, 5% do not. We never know if the one random sample that we draw is one of the lucky ones inside this **confidence interval.**

Average pounds lost across 10000 samples
red line represents the mean of all sample means, blue line represents +/- 1.98 SEs from this mean

Estimation of the Standard Error in Our Sample

Of course, we do not draw many, many random samples and conduct our study within each — therefore, we need a method for estimating the standard error another way. We use this formula to calculate the standard error (SE) using our sample statistics.

$$SE = \frac{SD}{\sqrt{n}} = \frac{2.10}{\sqrt{100}} = .21$$

```
# calculate and store relevant quantities
mean <- mean(wtloss$lbslost) # mean in the sample
sd <- sd(wtloss$lbslost) # standard deviation in the sample
n <- length(wtloss$lbslost) # n of the sample


se <- sd/sqrt(n) # standard error
se
```

```
Univariate Analysis

N                100.00    Variance                  4.40
Missing            0.00    Std Deviation             2.10
Mean               3.06    Range                    10.76
Median             3.14    Interquartile Range       3.07
Mode              -3.19    Uncorrected SS         1374.33
Trimmed Mean       3.11    Corrected SS            435.83
Skewness          -0.33    Coeff Variation          68.49
Kurtosis          -0.22    Std Error Mean            0.21
```

```
[1] 0.2098183
```

Once the SE is calculated we can construct a confidence interval (CI). Typically the t-score from the Student's t distribution is used. Recall that we calculated the t-scores for the 2.5 and 97.5 percentiles (corresponding to a 95% CI) of the cumulative distribution function of the t-distribution to be -1.98 and 1.98 respectively. Therefore, given our sample size, the 95% CI is:

$3.06 \pm 1.98 \cdot .21 = 2.65, 3.48$

```
me <- 1.984217*se # 95% confidence interval
me


lower <- mean - me # calculate lower bound for confidence interval
upper <- mean + me # calculate upper bound for confidence interval
lower
upper
```

```
[1] 0.4163251
[1] 2.647166
[1] 3.479816
```

Confidence Intervals in All 10,000 Samples

Let's calculate the 95% CI in all of the samples and take a look.

```
sample_means <- sample_means %>%

  select(replicate, mean_lbslost, sd_lbslost) %>%

  mutate(se = sd_lbslost/sqrt(99),

      me = 1.984217*se,

      lower = mean_lbslost - me,

      upper = mean_lbslost + me,

      not_in_ci = ifelse(2.985371 < lower | 2.985371 > upper, 1, 0))
```

```
[1] 0.0469
```

Across all of the samples, about 4.7% do not include the population mean in the calculated 95% CI. This is the essence of the 95% CI — if we were to randomly select many, many random samples, and in each construct a 95% CI using the sample's mean and SE, in 95% of the samples the true population mean will be within the constructed 95% CI of the sample.

replicate	mean_lbslost	sd_lbslost	se	me	lower	upper	not_in_ci
1	3.094337	1.956708	0.1966566	0.3902093	2.704128	3.484547	0
2	2.920634	2.165699	0.2176609	0.4318865	2.488747	3.352520	0
3	3.003978	2.165654	0.2176564	0.4318775	2.572101	3.435856	0
4	3.134027	1.774678	0.1783619	0.3539086	2.780119	3.487936	0
5	2.944227	1.947747	0.1957560	0.3884223	2.555804	3.332649	0
6	3.190237	1.951895	0.1961728	0.3892495	2.800988	3.579487	0
7	3.475803	2.098721	0.2109294	0.4185298	3.057273	3.894333	1
8	3.032388	1.863397	0.1872784	0.3716010	2.660787	3.403989	0
9	3.010078	1.776071	0.1785018	0.3541863	2.655891	3.364264	0
10	3.268840	1.981819	0.1991803	0.3952169	2.873623	3.664057	0
11	3.119842	2.176504	0.2187469	0.4340413	2.685800	3.553883	0
12	2.798634	2.121862	0.2132552	0.4231446	2.375490	3.221779	0
13	2.728742	1.779156	0.1788119	0.3548017	2.373940	3.083543	0
14	2.901259	2.263805	0.2275210	0.4514510	2.449807	3.352710	0
15	2.799985	1.813757	0.1822895	0.3617018	2.438283	3.161687	0
16	2.583812	2.115549	0.2126206	0.4218855	2.161926	3.005697	0
17	3.054015	2.027214	0.2037426	0.4042696	2.649745	3.458284	0
18	2.712764	2.255358	0.2266720	0.4497664	2.262998	3.162531	0
19	2.854545	2.089046	0.2099570	0.4166003	2.437945	3.271146	0
20	2.625643	1.756653	0.1765503	0.3503141	2.275328	2.975957	1

Alpha and Confidence Intervals

A 95% confidence interval is associated with an alpha (α) of .05. In terms of constructing confidence intervals, this alpha of .05 is the probability that the following will happen: we will estimate the statistic of interest in our random sample, calculate the 95% CI, and that the true parameter in the population (the value that we are trying to estimate) will not be included in our constructed CI. You will need to decide how comfortable you are with this probability. If that feels to risky, you can construct a wider confidence interval (e.g., 99%).

Calculate the critical value of t given alpha (2 sided) and df

```
# t-scores needed for a 95% CI
# two-tailed with alpha = .05, df = 99 (equivalent ways to write)
qt(1-.05/2, df = 99)
qt(c(.025, .975), df = 99)


# t-scores needed for a 99% CI
# two-tailed with alpha = .01, df = 99 (equivalent ways to write)
qt(1-.01/2, df = 99)
qt(c(.005, .995), df = 99)
```

```
[1] 1.984217
[1] -1.984217   1.984217
[1] 2.626405
[1] -2.626405   2.626405
```

Calculate the 99% CI to compare to our previous 95% CI

```
me <- 2.626405*se # 99% confidence interval
lower <- mean - me # calculate lower bound for confidence interval
upper <- mean + me # calculate upper bound for confidence interval
```
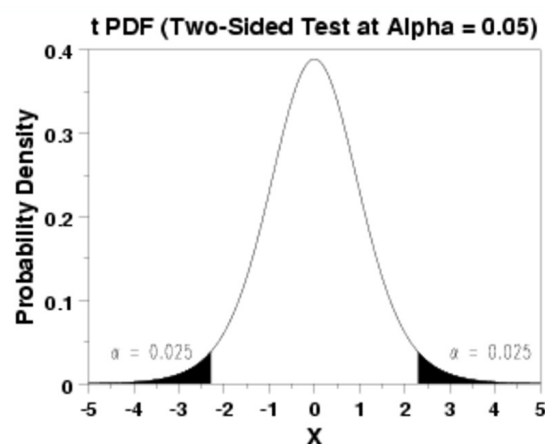
```
[1] 0.5510679
[1] 2.512423
[1] 3.614559
```

95% CI: 3.06 ± 1.98 · .21 = 2.65, 3.48

99% CI: 3.06 ± 2.63 · .21 = 2.51, 3.61

t PDF (Two-Sided Test at Alpha = 0.05)

The Sampling Distibution for Hypothesis Testing

We can use the principles of the sampling distribution to conduct hypothesis tests. When we conduct a hypothesis test, we begin by stating the null hypothesis and our alternative hypothesis. For example, let's determine if the weight loss program resulted in significant weight loss — specifically we can determine if the average weight loss is significantly different than 0 (no weight change after the first month on the program).
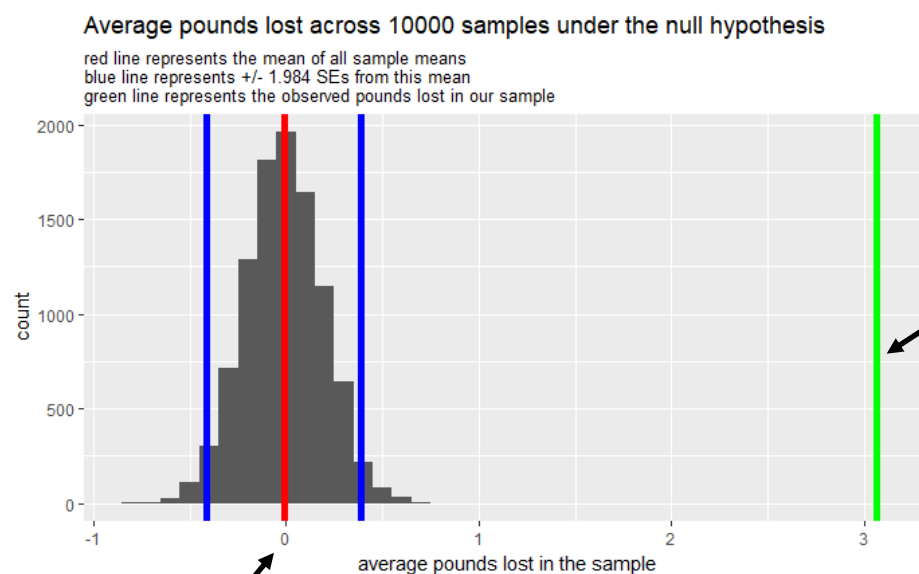
First let's define the population parameter of interest. We are interested in the average weight loss in the population, a population mean is written as μ (mu), and we want to determine if this average weight loss is significantly different from 0.

Our null hypothesis is: $H_0: \mu = 0$,

Our alternative hypothesis: $H_a: \mu \neq 0$

Notice that our null hypothesis is non-directional, we want to determine if the population mean is different from 0, whether it's above (weight loss) or below (weight gain) 0. This non-directional test is what's typically used in behavioral sciences.

Once the hypothesis is defined, the next step is to set alpha for our test, **alpha is the probability of making a type I error, that is rejecting the null hypothesis, when in actuality it is true**. In this case, a type I error would mean that we conclude that weight changed over the course of the program, when in actuality it did not. When conducting a null hypothesis test, we need to set alpha *a priori*. Let's set it at .05, which corresponds to a 95% CI.

Average pounds lost across 10000 samples under the null hypothesis

red line represents the mean of all sample means
blue line represents +/- 1.984 SEs from this mean
green line represents the observed pounds lost in our sample

In our sample, the mean pounds lost is 3.06. How many standard errors away from 0 is this value? It's 3.06/.21 ≈ 14.6 standard errors. Given the expectation that 95% of samples will produce a mean that is between the two blue lines (1.98 standard errors), it's highly unusual that we would have randomly selected this sample if the null hypothesis is true. We can use the probability distribution function to determine jut how unusual.

Our null hypothesis asserts that the population mean is 0. Under this null hypothesis, the center of the sampling distribution is 0. If the null hypothesis is true, and the population mean is 0, then we could expect that 95% of the samples that we would select from the population will produce an estimate of the mean pounds lost within 1.98 standard errors from this mean, so $0 \pm 1.98 \cdot .21 = -.42, .42$.

2*(pt(14.6, 99, lower.tail = FALSE))

```
[1] 1.977673e-26
```

The probability of obtaining a mean that is this many standard errors away from 0 (the value of our null hypothesis) if the null hypothesis is true is virtually 0. This would suggest that the null hypothesis should be rejected.

One-Sample t-test

The steps we just took to determine if the population mean if significantly different from a specified value (0 in our case) is called a one-sample t-test.

Steps to conduct a null hypothesis significance test — one-sample t-test

1. State null and alternative hypothesis. $H_0: \mu = 0$, $H_a: \mu \neq 0$

2. Set alpha.   .05

3. Calculate the critical value of t (t_{crit}) given your alpha and degrees of freedom (n - 1 for a one sample t-test), this is the corresponding t-score for the cumulative distribution function. -1.98, 1.98

4. Calculate the t-test (t*), in this case: $t^* = \dfrac{\bar{x} - 0}{SE} = \dfrac{3.06}{.21} = 14.60$

5. Compare t* to t_{crit}. For a 2-tailed test, if the absolute value of t* > the absolute value of t_{crit}, then reject the null hypothesis. $|14.60| > |1.98|$

6. Calculate the p-value — the probability we would have gotten the sample result we did (or a more extreme result) if the null hypothesis were really true. <.0001

7. Compute the confidence interval. $3.06 \pm 1.98 \cdot .21 = 2.65, 3.48$

Our One-Sample t-test

```
# Calculate the critical value of t given alpha and df

qt(1-.05/2, df = 99)

qt(c(.025, .975), df = 99)


# Conduct t-test

t.test (wtloss$lbslost, mu=0, conf.level = .95)


# Calculate p-value

2*(1-pt(14.601, df = 99))
```

Critical value of t (t_{crit})

```
[1] 1.984217
[1] -1.984217  1.984217

          One Sample t-test

t*
     data:  wtloss$lbslost
     t = 14.601, df = 99, p-value < 2.2e-16
     alternative hypothesis: true mean is not equal to 0
     95 percent confidence interval:
      2.647166 3.479816
     sample estimates:
     mean of x
      3.063491


 [1] 1.968496e-26
```

p-value

Write Up the Results

A sample of 100 men were randomly selected from the population of all obese men in Colorado between the ages of 40 and 50. All men participated in a 1 month weight loss program. We hypothesized that the weight loss program would result in substantial weight loss over the course of the program. We conducted a 2-sided null hypothesis test to determine if the average weight loss is significantly different from 0. A one-sample t-test was conducted in R, Version 3.4.1. On average the men lost about 3 pounds of weight (M = 3.06, SD = 2.10), reflecting a weight loss that is significantly greater than 0, $t(99)$ = 14.60, p<.001 (95% CI: 2.65, 3.48). Our study provides some preliminary evidence that the weight loss program may result in significant weight loss at the end of one month on the program. A randomized control trial is needed to determine if the program causes weight loss.