IS5 in R: Comparing Groups (Chapter 17)

Nicholas Horton (nhorton@amherst.edu)

2025-01-20

Table of contents

Introduction and background

This document is intended to help describe how to undertake analyses introduced as examples in the Fifth Edition of *Intro Stats* (2018) by De Veaux, Velleman, and Bock. This file as well as the associated Quarto reproducible analysis source file used to create it can be found at http://nhorton.people.amherst.edu/is5.

This work leverages initiatives undertaken by Project MOSAIC (http://www.mosaic-web.org), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the `mosaic` package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the mosaic package vignettes (https://cran.r-project.org/web/packages/mosaic). A paper describing the mosaic approach was published in the *R Journal*: https://journal.r-project.org/archive/2017/RJ-2017-024.

We begin by loading packages that will be required for our analyses.

```
library(mosaic)
library(tidyverse)
```

Section 17.1: A Confidence Interval for the Difference Between Two Proportions

```
# Creating a data frame for Seatbelts
Seatbelts <- bind_rows(
  do(2777) * data.frame(passenger = "F", belted = TRUE),
  do(4208 - 2777) * data.frame(passenger = "F", belted = FALSE),
  do(1363) * data.frame(passenger = "M", belted = TRUE),
  do(2763 - 1363) * data.frame(passenger = "M", belted = FALSE)
) |>
  select(passenger, belted)
```

The `mosaic::do()` function constructs the correct number of rows for the data frame from provided cell counts.

```
set.seed(234)
num_sim <- 10000

# What does do() do?
abs(diffmean(belted ~ passenger, data = resample(Seatbelts)))
```

```
 diffmean
0.1765479
```

```
# Difference of proportions from one random resample
abs(diffmean(belted ~ passenger, data = resample(Seatbelts)))
```

```
 diffmean
0.1637818
```

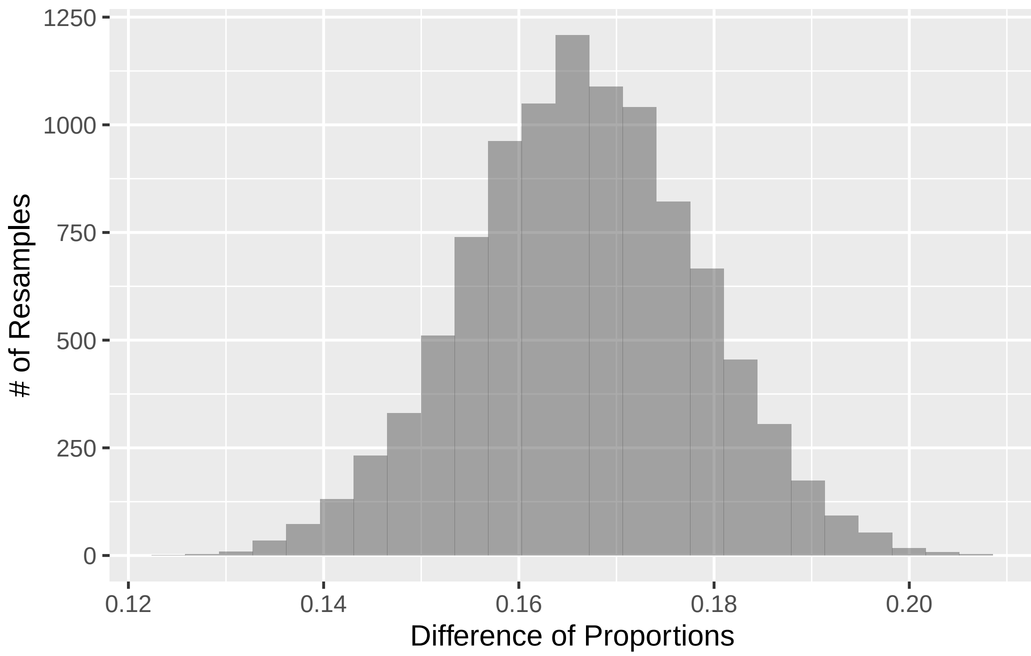
```
# Difference of proportions from another random resample

do(2) *             # Calculates two differences
  abs(diffmean(belted ~ passenger, data = resample(Seatbelts)))
```

```
   diffmean
1 0.1622740
2 0.1459405
```

```
# We want to calculate num_sim resampled differences of proportions
seatbeltresamples <-
  do(num_sim) * abs(diffmean(belted ~ passenger, data = resample(Seatbelts)))
```

For more information about `resample()`, refer to the resampling vignette: https://cran.r-project.org/web/packages/mosaic/vignettes/Resampling.html

```
# Figure 17.1, page 542
gf_histogram(~ diffmean, data = seatbeltresamples) |>
  gf_labs(x = "Difference of Proportions", y = "# of Resamples")
```



Example 17.1: Finding the Standard Error of a Difference in Proportions

We begin with some wrangling to create the dataset.

```
# Creating the data set for online profiles
OnlineProf <- bind_rows(
  do(141) * data.frame(gender = "M", profile = TRUE), # 248 * .57 rounds to 141
```

```
  do(107) * data.frame(gender = "M", profile = FALSE), # 248 - 141
  do(179) * data.frame(gender = "F", profile = TRUE),
  do(77) * data.frame(gender = "F", profile = FALSE)
)
tally(~ gender, data = OnlineProf)
```

```
gender
  F   M
256 248
```

```
OnlineProfM <- OnlineProf |>
  filter(gender == "M") # Make a data set for male observations
nM <- nrow(OnlineProfM)
nM # n for males
```

```
[1] 248
```

```
propMyes <- mean(~profile, data = OnlineProfM)
propMyes # p for males
```

```
[1] 0.5685484
```

```
sepboys <- ((propMyes * (1 - propMyes)) / nM)^.5
sepboys # SE for males
```

```
[1] 0.03145024
```

```
OnlineProfF <- OnlineProf |>
  filter(gender == "F") # Make a data set for female observations
nF <- nrow(OnlineProfF)
nF # n for females
```

```
[1] 256
```

```
propFyes <- mean(~profile, data = OnlineProfF)
propFyes # p for females
```

```
[1] 0.6992188
```

```
sepgirls <- ((propFyes * (1 - propFyes)) / nF)^.5
sepgirls # SE for females
```

```
[1] 0.02866236
```

```
sep <- (sepboys^2 + sepgirls^2)^.5
sep # overall SE
```

```
[1] 0.04255171
```

Example 17.2: Finding a Two-Proportion *z*-Interval

We can calculate the desired Z interval.

```
zstats <- qnorm(p = c(.025, .975))
(propFyes - propMyes) + zstats * sep
```

```
[1] 0.04727054 0.21407019
```

```
# Or, you can use:
prop.test(x = c(179, 141), n = c(nF, nM), correct = FALSE)
```

```

    2-sample test for equality of proportions without continuity correction

data:  c out of c179 out of nF141 out of nM
X-squared = 9.2792, df = 1, p-value = 0.002318
alternative hypothesis: two.sided
95 percent confidence interval:
 0.04727054 0.21407019
sample estimates:
   prop 1    prop 2 
0.6992188 0.5685484 
```

The `prop.test()` function can be used to find confidence intervals and p-values of both one or two proportion z-tests.

Section 17.2: Assumptions and Conditions for Comparing Proportions

Section 17.3: The Two-Sample *z*-Test: Testing for the Difference Between Proportions

Step-By-Step Example: A Two-Proportion *z*-Test

Again, we need to create the data table of counts.

```
# Create the data set
SleepHabits <- bind_rows(
  do(205) * data.frame(gen = "GenY", internet = TRUE),
  do(293 - 205) * data.frame(gen = "GenY", internet = FALSE),
  do(235) * data.frame(gen = "GenX", internet = TRUE),
  do(469 - 235) * data.frame(gen = "GenX", internet = FALSE)
)
```

```
# Mechanics
ngeny <- nrow(filter(SleepHabits, gen == "GenY"))
ngeny # n for GenY
```

```
[1] 293
```

```
ygeny <- nrow(filter(SleepHabits, gen == "GenY" & internet == TRUE))
ygeny # y for GenY
```

```
[1] 205
```

```
pgeny <- mean(~ internet, data = filter(SleepHabits, gen == "GenY"))
pgeny # proportion for GenY
```

```
[1] 0.6996587
```

```
ngenx <- nrow(filter(SleepHabits, gen == "GenX"))
ngenx # n for GenX
```

```
[1] 469
```

```
ygenx <- nrow(filter(SleepHabits, gen == "GenX" & internet == TRUE))
ygenx # y for GenX
```

```
[1] 235
```

```
pgenx <- mean(~ internet, data = filter(SleepHabits, gen == "GenX"))
pgenx # proportion for GenX
```

```
[1] 0.5010661
```

```
sepgen <- ((pgeny * (1 - pgeny)) / ngeny + (pgenx * (1 - pgenx)) / ngenx)^.5
sepgen # overall SE
```

```
[1] 0.03535867
```

```
pdiff <- pgeny - pgenx
pdiff # difference between proportions
```

```
[1] 0.1985926
```

```
z <- (pdiff - 0) / sepgen
z
```

```
[1] 5.616518
```

```
2 * pnorm(q = z, lower.tail = FALSE)
```

```
[1] 1.948444e-08
```

Section 17.4: A Confidence Interval for the Difference Between Two Means

The `t.test()` function can be used to generate a confidence interval for the difference between two means. The `conf.level` option can be used to create different intervals.

Example 17.7: Finding a Confidence Interval for the Difference in Sample Means

We can calculate the confidence interval using summary statistics.

```
# page 555
nord <- 27 # n for ordinary bowls
nref <- 27 # n for refilling bowls
yord <- 8.5 # y for ordinary bowls
yref <- 14.7 # y for refilling bowls
sord <- 6.1 # standard deviation for ordinary bowls
sref <- 8.4 # standard deviation for refilling bowls

seys <- 2.0 # overall SE
diffy <- yref - yord # difference between y's is 6.2
tstats <- qt(p = c(.025, .975), df = 47.46)
tstats
```

```
[1] -2.011226  2.011226
```

```
me <- tstats * seys
me # margin of error
```

```
[1] -4.022452  4.022452
```

```
diffy + me # confidence interval
```

```
[1]  2.177548 10.222452
```

Section 17.5: The Two-Sample *t*-Test: Testing for the Difference Between Two Means

Step-By-Step Example: A Two-Sample *t*-Test for the Difference Between the Two Means

We begin by reading the data.

```
# page 556
BuyingCam <- read_csv("http://nhorton.people.amherst.edu/is5/data/Buy_from_a_friend.csv")
```

```
head(BuyingCam) # before reshaping using `pivot_longer()`
```

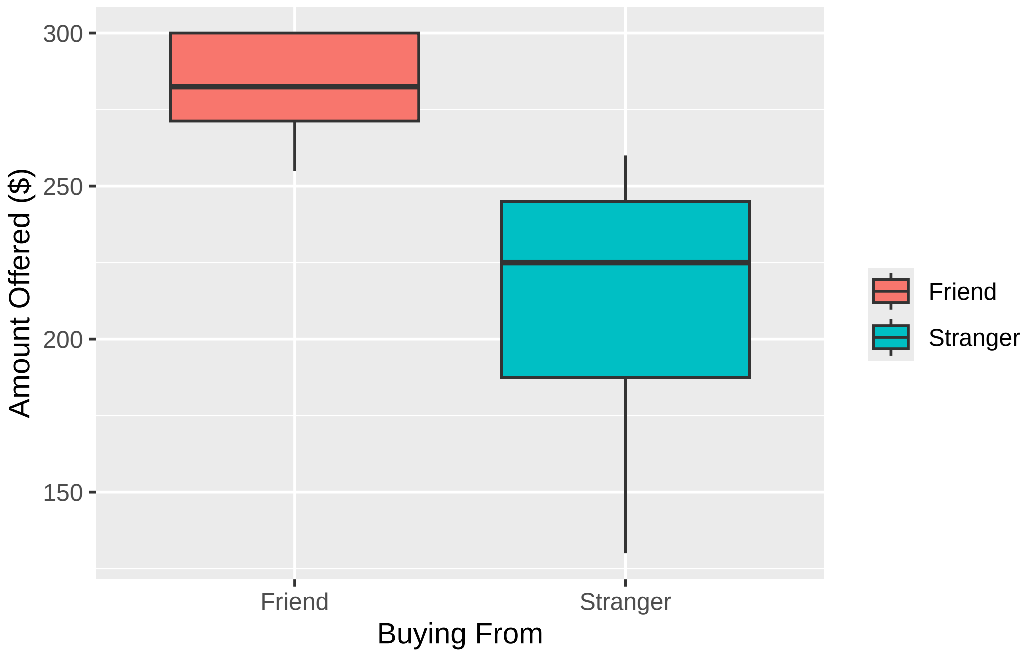
```
# A tibble: 6 x 2
  Friend Stranger
   <dbl>    <dbl>
```

```
1    275      260
2    300      250
3    260      175
4    300      130
5    255      200
6    275      225
```

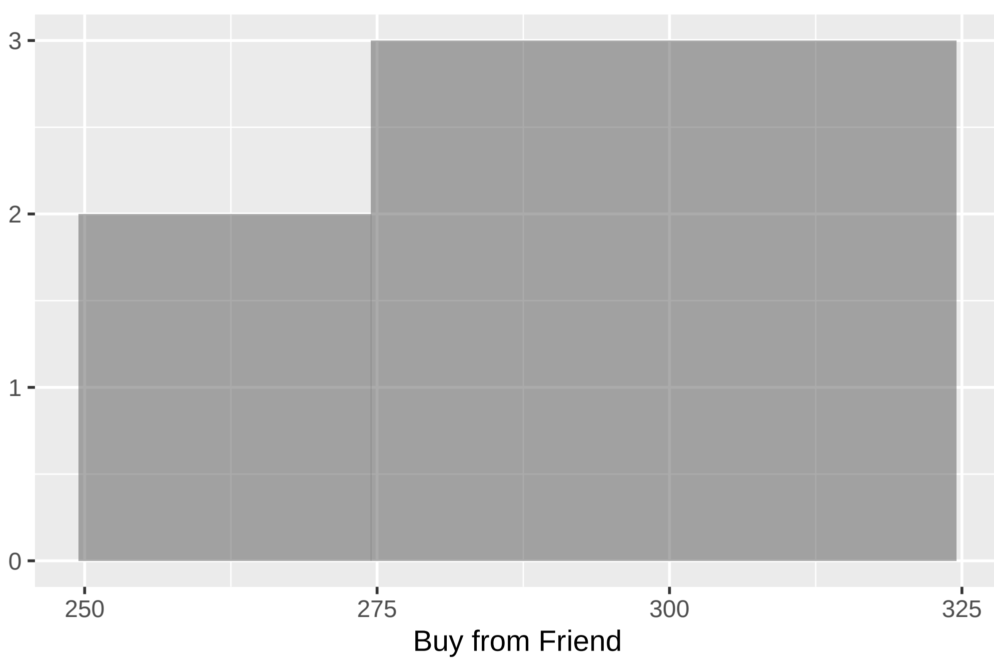
```
BuyingCam <- BuyingCam |>
  pivot_longer(
    Friend:Stranger,
    names_to = "buying_type",
    values_to = "amount_offered"
  )
head(BuyingCam) # after reshaping
```

```
# A tibble: 6 x 2
  buying_type amount_offered
  <chr>                <dbl>
1 Friend                 275
2 Stranger               260
3 Friend                 300
4 Stranger               250
5 Friend                 260
6 Stranger               175
```

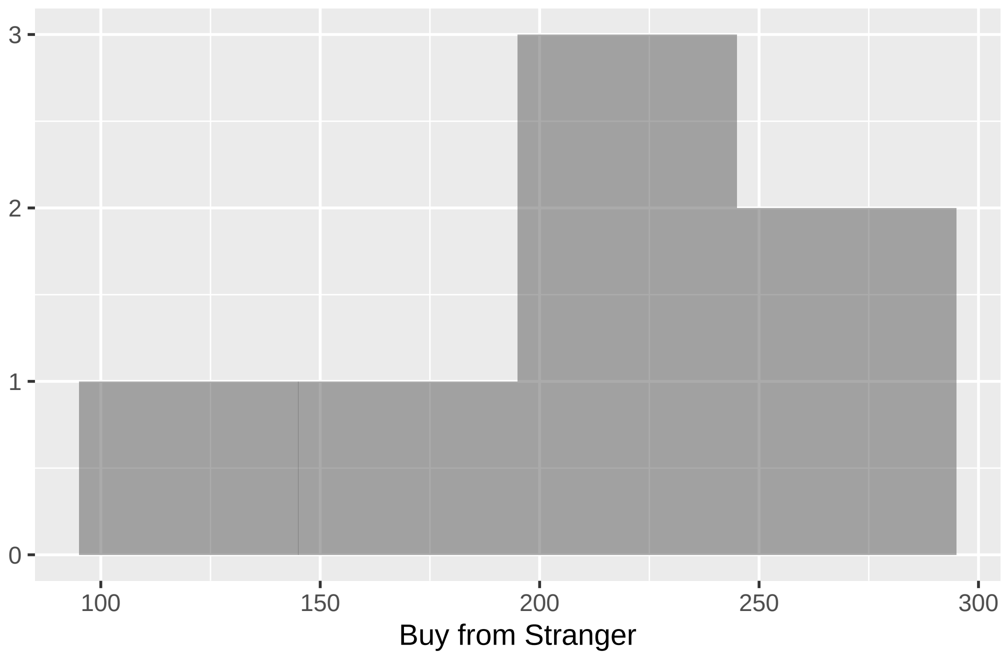
```
# Model
gf_boxplot(amount_offered ~ buying_type, fill = ~ buying_type, data = BuyingCam) |>
  gf_labs(x = "Buying From", y = "Amount Offered ($)", fill = "")
```




```
BuyingCam |>
  filter(buying_type == "Friend") |>
  gf_histogram(~ amount_offered, binwidth = 25, center = 12) |>
  gf_labs(x = "Buy from Friend", y = "")
```



```
BuyingCam |>
  filter(buying_type == "Stranger") |>
  gf_histogram(~ amount_offered, binwidth = 50, center = 20) |>
  gf_labs(x = "Buy from Stranger", y = "")
```



We can replicate the analyses on pages 557-558.

```
df_stats(amount_offered ~ buying_type, data = BuyingCam)
```

```
        response buying_type min     Q1 median  Q3 max     mean       sd n
1 amount_offered      Friend 255 271.25  282.5 300 300 281.8750 18.31032 8
2 amount_offered    Stranger 130 187.50  225.0 245 260 211.4286 46.43223 7
  missing
1       0
2       1
```

```
t.test(amount_offered ~ buying_type, data = BuyingCam)
```

```

    Welch Two Sample t-test

data:  amount_offered by buying_type
```

```
t = 3.766, df = 7.6229, p-value = 0.006003
alternative hypothesis: true difference in means between group Friend and group Stranger is n
95 percent confidence interval:
  26.93688 113.95597
sample estimates:
  mean in group Friend mean in group Stranger
              281.8750               211.4286
```

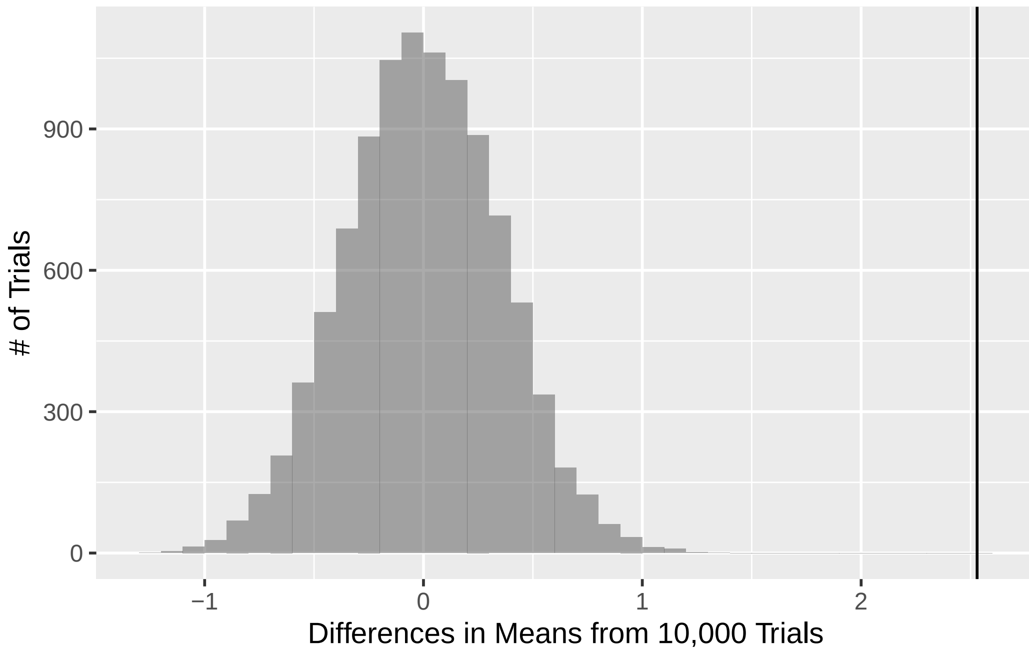
Section 17.6: Randomization Tests and Confidence Intervals for Two Means

We begin by reading in the Cars dataset.

```
Cars <- readr::read_csv("http://nhorton.people.amherst.edu/is5/data/Car_speeds.csv")
# Figure 17.2 (page 560) is the same as Figure 4.4 (page 102)
df_stats(speed ~ direction, data = Cars)
```

```
  response direction   min      Q1 median      Q3   max     mean       sd   n
1    speed      Down 10.27 20.4675 22.885 25.3525 32.95 22.71708 3.622006 250
2    speed        Up 15.08 22.4975 25.155 28.1600 34.97 25.25172 3.856331 250
  missing
1       0
2       0
```

```
set.seed(23456)
num_sim <- 10000
CarSims <- do(num_sim) * diffmean(speed ~ shuffle(direction), data = Cars)
# Figure 17.3, page 560
gf_histogram(~ diffmean, data = CarSims, binwidth = 0.1, center = 0.05) |>
  gf_vline(xintercept = 2.53) |>
  gf_labs(x = "Differences in Means from 10,000 Trials", y = "# of Trials")
```



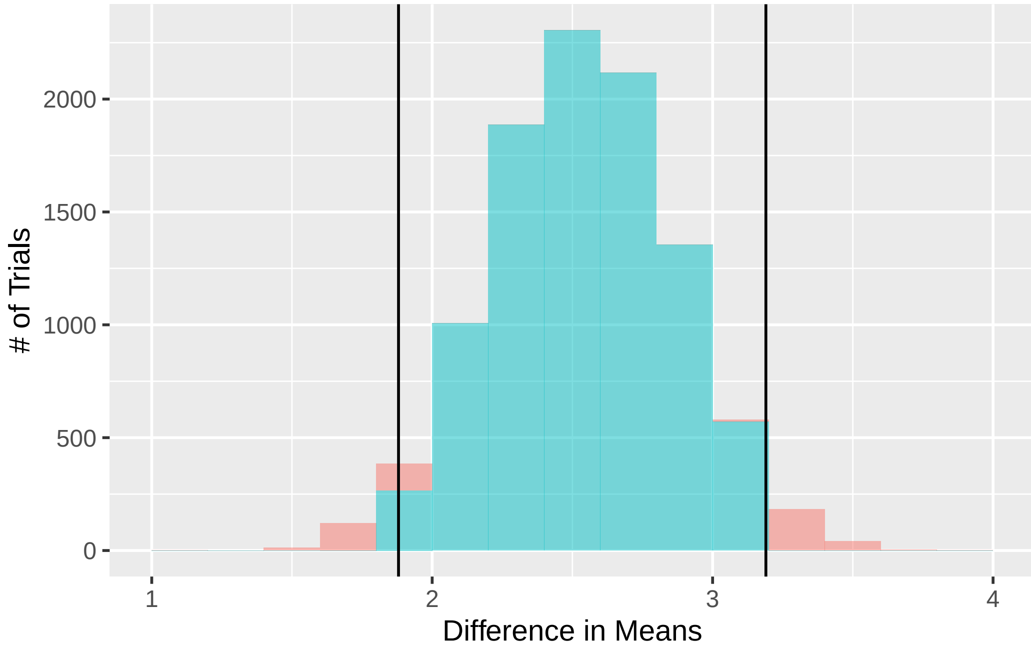

```
set.seed(32453)
num_sim <- 10000

CarBoots <- do(num_sim) * diffmean(speed ~ direction, data = resample(Cars))
qdata(~ diffmean, p = c(.025, .975), data = CarBoots)
```

```
    2.5%    97.5%
1.877961 3.183384
```

```
CarBoots <- CarBoots |>
  mutate(interval = ifelse(
    diffmean > 1.88 & diffmean < 3.19,
    "Within 95% Confidence", # if TRUE
    "Outside 95% Confidence" # if FALSE
  ))
```

```
# Figure 17.4
gf_histogram(
  ~ diffmean,
  fill = ~ interval, data = CarBoots,
  binwidth = 0.2, center = 0.1
) |>
  gf_vline(xintercept = 1.88) |>
```

```
  gf_vline(xintercept = 3.19) |>
  gf_labs(x = "Difference in Means", y = "# of Trials") +
  guides(fill = FALSE) # to remove the legend
```



Section 17.7: Pooling

The pooled variance t.test can be generated by using the option `var.equal = TRUE`.

```
t.test(amount_offered ~ buying_type, var.equal = TRUE, data = BuyingCam)
```

```

	Two Sample t-test

data:  amount_offered by buying_type
t = 3.9699, df = 13, p-value = 0.0016
alternative hypothesis: true difference in means between group Friend and group Stranger is n
95 percent confidence interval:
  32.11047 108.78238
sample estimates:
  mean in group Friend mean in group Stranger
              281.8750               211.4286
```

Section 17.8: The Standard Deviation of a Difference