IS5 in R: Comparing Groups (Chapter 17)

Margaret Chien and Nicholas Horton (nhorton@amherst.edu)

July 17, 2018

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. More information about the book can be found at http://wps.aw.com/aw_deveaux_stats_series. This file as well as the associated R Markdown 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 (http://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.

Chapter 17: Comparing Groups

```
library(mosaic)
library(readr)
library(janitor)
```

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

```
# Creating a data frame for Seatbelts
Seatbelts <- rbind(
  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 `do()` function constructs the correct number of rows for the data frame from provided cell counts.

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

# What does do() do?
resample(Seatbelts) %>%
  group_by(passenger) %>%
  summarise(proportion = sum(belted)/n()) %>%
  summarise(diffprop = abs(diff(proportion))) # Difference of proportions from one random resample
```

```
## # A tibble: 1 x 1
##   diffprop
##      <dbl>
```

```
## 1    0.170
```

```
resample(Seatbelts) %>%
  group_by(passenger) %>%
  summarise(proportion = sum(belted)/n()) %>%
  summarise(diffprop = abs(diff(proportion))) # Difference of proportions from another random resample
```

```
## # A tibble: 1 x 1
##   diffprop
##      <dbl>
## 1    0.178
```

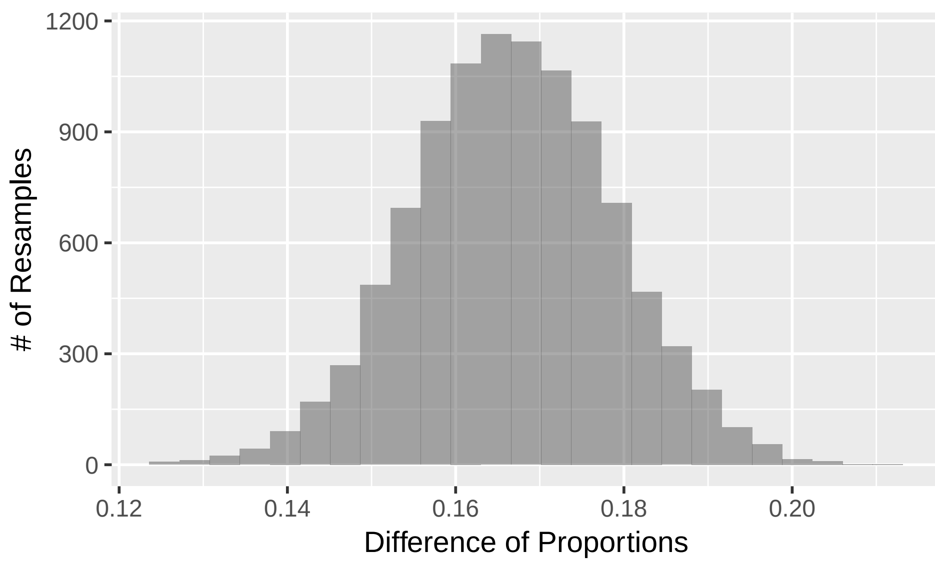
```
do(2) * resample(Seatbelts) %>%
  group_by(passenger) %>%
  summarise(proportion = sum(belted)/n()) %>%
  summarise(diffprop = abs(diff(proportion))) # Calculates two differences
```

```
##    diffprop
## 1 0.1573754
## 2 0.1548267
```

```
# We need numsim differences of proportions
seatbeltresamples <- do(numsim) * resample(Seatbelts) %>%
  group_by(passenger) %>%
  summarise(proportion = sum(belted)/n()) %>%
  summarise(diffprop = abs(diff(proportion)))
```

For more information about `resample()`, refer to the resample vignette in mosaic.

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



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

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

```r
  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
##   M   F
## 248 256
```

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

```
## [1] 248
```

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

```
## [1] 0.5685484
```

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

```
## [1] 0.03145024
```

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

```
## [1] 256
```

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

```
## [1] 0.6992188
```

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

```
## [1] 0.02866236
```

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

```
## [1] 0.04255171
```

Example 17.2: Finding a Two-Proportion z-Interval

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

```
## [1] 0.04727054 0.21407019
```

```r
# 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(179, 141) out of c(nF, 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

```
# Create the data set
SleepHabits <- rbind(
  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
```

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

```
## [1] 0.03535867
```

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

```
## [1] 0.1985926
```

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

```
## [1] 5.616518
```

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

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

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

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

```r
# 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
```

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

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

```r
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

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

```
## Parsed with column specification:
## cols(
##   Friend = col_integer(),
##   Stranger = col_integer()
```

```
## )
```

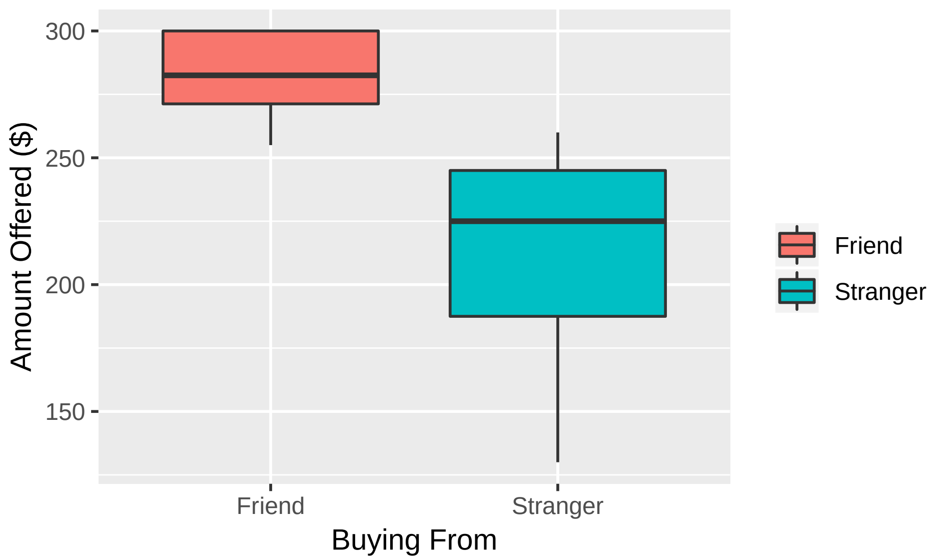
By default, `read_csv()` prints the variable names. These messages can be suppressed using the `message=FALSE` code chunk option to save space and improve readability.

```r
library(tidyr) # for gather() function
```

```
##
## Attaching package: 'tidyr'
## The following object is masked from 'package:Matrix':
##
##     expand
```

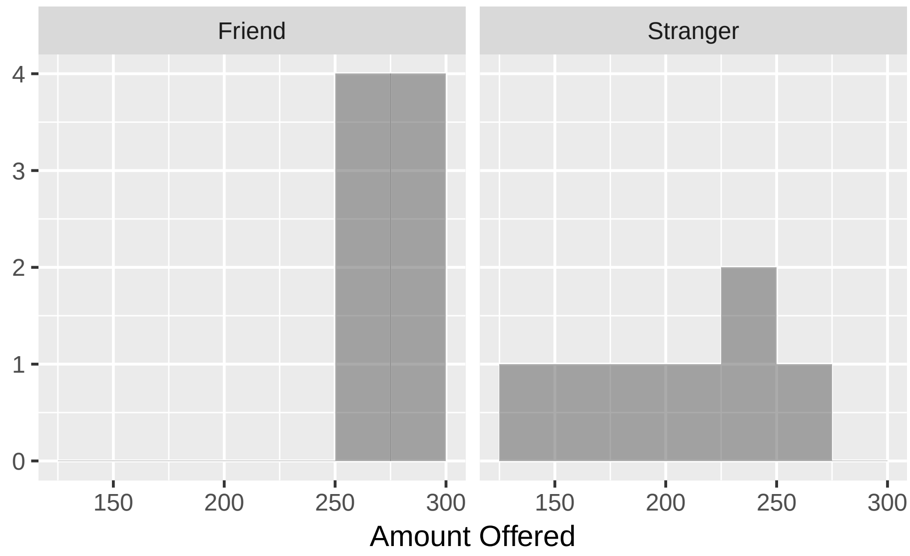
```r
BuyingCam <- BuyingCam %>%
  gather(key = buying_type, value = amount_offered, Friend, Stranger)
# Model
gf_boxplot(amount_offered ~ buying_type, fill = ~ buying_type, data = BuyingCam) %>%
  gf_labs(x = "Buying From", y = "Amount Offered ($)", fill = "")
```

```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```




```r
gf_histogram(~ amount_offered, binwidth = 25, center = 12.5, data = BuyingCam) %>%
  gf_facet_wrap(buying_type ~ .) %>%
  gf_labs(x = "Amount Offered", y = "")
```

```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```




```
# Mechanics
favstats(~ amount_offered | buying_type, data = BuyingCam)
```

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

Section 17.6: Randomization Tests and Confidence Intervals for Two Means

```
Cars <- read_csv("http://nhorton.people.amherst.edu/is5/data/Car_speeds.csv")
```

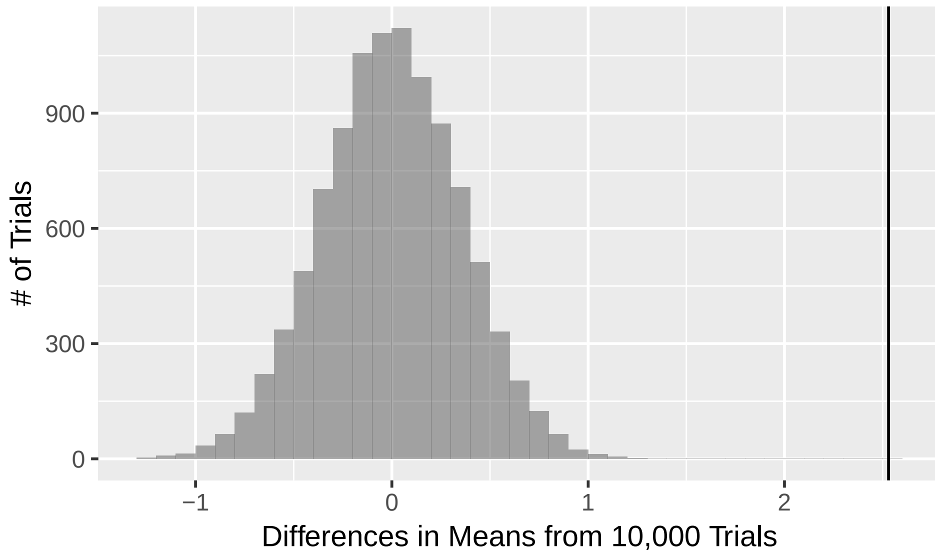
```
## Parsed with column specification:
## cols(
##   direction = col_character(),
##   speed = col_double()
## )
```

```
# Figure 17.2 (page 560) is the same as Figure 4.4 (page 102)
favstats(~ speed | direction, data = Cars)
```

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

```
set.seed(23456)
numsim <- 10000

CarSims <- do(numsim) * Cars %>%
  mutate(direction = shuffle(direction)) %>%
  group_by(direction) %>%
  summarise(means = mean(~ speed)) %>%
  summarise(diffmeans = diff(means))
```

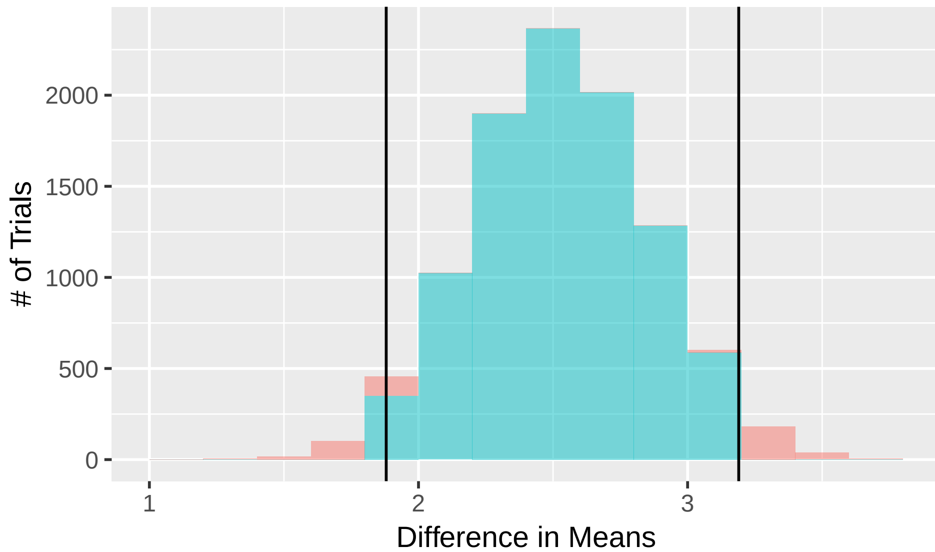
```r
# Figure 17.3, page 560
gf_histogram(~ diffmeans, data = CarSims, binwidth = .1, center = .05) %>%
  gf_vline(xintercept = 2.53) %>%
  gf_labs(x = "Differences in Means from 10,000 Trials", y = "# of Trials")
```




```r
set.seed(32453)
numsim <- 10000

CarBoots <- do(numsim) * Cars %>%
  resample() %>%
  group_by(direction) %>%
  summarise(means = mean(~ speed)) %>%
  summarise(diffmeans = diff(means))

qdata(~ diffmeans, p = c(.025, .975), data = CarBoots)
```

```
##       quantile     p
## 2.5%  1.887197 0.025
## 97.5% 3.185255 0.975
```

```r
CarBoots <- CarBoots %>%
  mutate(interval = ifelse(diffmeans > 1.88 & diffmeans < 3.19, "Within 95% Confidence",
                           "Outside 95% Confidence"))

# Figure 17.4
gf_histogram(~ diffmeans, fill = ~ interval, data = CarBoots, binwidth = .2,
             center = .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

Section 17.8: The Standard Deviation of a Difference