IS5 in R: Paired Samples and Blocks (Chapter 18)

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 18: Paired Samples and Blocks

```
library(mosaic)
library(readr)
library(janitor)
Dexterity <- read_csv("http://nhorton.people.amherst.edu/is5/data/Dexterity.csv") %>%
  clean_names()
```

```
## Warning: Duplicated column names deduplicated: 'Dominant' =>
## 'Dominant_1' [6]
```

```
## Parsed with column specification:
## cols(
##   `Age(months)` = col_integer(),
##   Gender = col_character(),
##   `Dominant Hand` = col_character(),
##   Dominant = col_double(),
##   `non-dominant` = col_double(),
##   Dominant_1 = col_double(),
##   `Non-dominant` = col_double()
## )
```

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.
Here we use the `clean_names()` function from the `janitor` package to sanitize the names of the columns (which would otherwise contain special characters or whitespace).

```
Dexterity %>%
  select(age_months, dominant_1, non_dominant_2, gender) %>%
  head(n = 7)
```

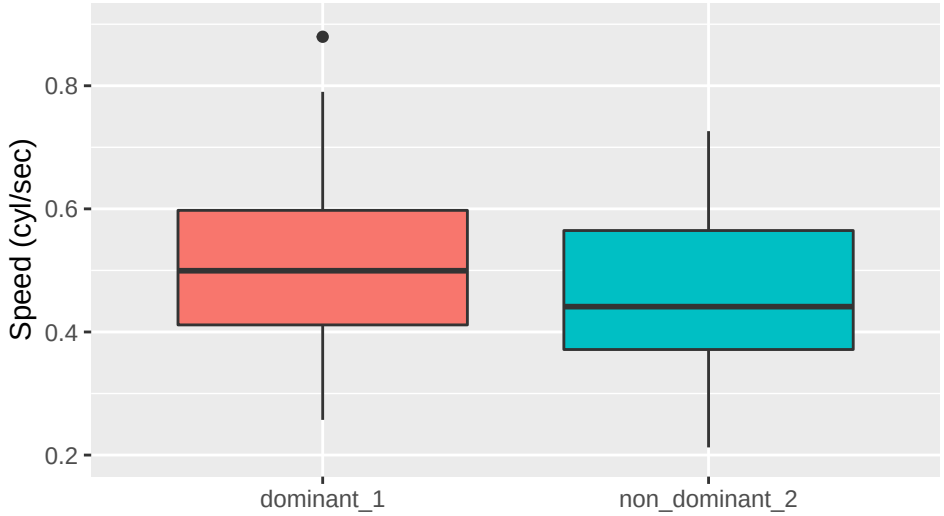
```
## # A tibble: 7 x 4
##   age_months dominant_1 non_dominant_2 gender
##        <int>      <dbl>          <dbl> <chr>
```

```
## 1         117       0.353           0.216 male
## 2         101       0.257           0.343 male
## 3         135       0.537           0.497 male
## 4         119       0.444           0.496 male
## 5         124       0.483           0.388 female
## 6         127       0.524           0.422 female
## 7         101       0.455           0.381 male
```

Section 18.1: Paired Data

```
library(tidyr) # for the gather() function
# Figure 18.1, page 586
Dexterity %>%
  select(dominant_1, non_dominant_2) %>%
  gather(key = hand_type, value = speed, dominant_1, non_dominant_2) %>%
  gf_boxplot(speed ~ hand_type, fill = ~ hand_type) %>%
  gf_labs(x = "", y = "Speed (cyl/sec)") +
  ylim(.2, .9) +
  guides(fill = FALSE)
```

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




```
Dexterity %>%
  select(dominant_1, non_dominant_2) %>%
  mutate(difference = dominant_1 - non_dominant_2) %>%
  head(n = 18)
```

```
## # A tibble: 18 x 3
##    dominant_1 non_dominant_2 difference
##         <dbl>          <dbl>      <dbl>
##  1      0.353          0.216    0.137
##  2      0.257          0.343   -0.0863
##  3      0.537          0.497    0.0392
##  4      0.444          0.496   -0.0524
##  5      0.483          0.388    0.0947
##  6      0.524          0.422    0.102
##  7      0.455          0.381    0.0742
```

```
##  8      0.394          0.403   -0.00904
##  9      0.451          0.328    0.124
## 10      0.527          0.271    0.256
## 11      0.565          0.415    0.149
## 12      0.653          0.298    0.355
## 13      0.421          0.337    0.0833
## 14      0.320          0.233    0.0872
## 15      0.344          0.241    0.102
## 16      0.428          0.612   -0.184
## 17      0.556          0.521    0.0357
## 18      0.465          0.411    0.0543
```

Section 18.2: The Paired *t*-Test

Step-By-Step Example: A Paired *t*-Test

XX NH can't find data for 2006 winter olympics speed skating data (page 589)

```
# This data doesn't seem to be correct!! (wrong year an doesn't have lane data)
SpeedySkatey <- read_csv("http://nhorton.people.amherst.edu/is5/data/Winter_Olympics_2010_speed_skating
```

```
## Parsed with column specification:
## cols(
##   Nation = col_character(),
##   Athlete = col_character(),
##   Result = col_double()
## )
```

```
SpeedySkatey %>%
  arrange(Result) %>%
  head()
```

```
## # A tibble: 6 x 3
##   Nation      Athlete       Result
##   <chr>       <chr>          <dbl>
## 1 Korea       Sang-Hwa Lee    38.2
## 2 Germany     Jenny Wolf      38.3
## 3 China       Beixing Wang    38.5
## 4 Netherlands Margot Boer     38.5
## 5 China       Shuang Zhang    38.5
## 6 Japan       Sayuri Yoshii   38.6
```

Section 18.3: Confidence Intervals for Matched Pairs

```
Couples <- read_csv("http://nhorton.people.amherst.edu/is5/data/Couples.csv") %>%
  filter(wAge != "*") %>%
  mutate(wAge = as.numeric(wAge))
```

```
## Parsed with column specification:
## cols(
##   Names = col_character(),
##   wAge = col_character(),
##   hAge = col_integer(),
##   wHeight = col_integer(),
```

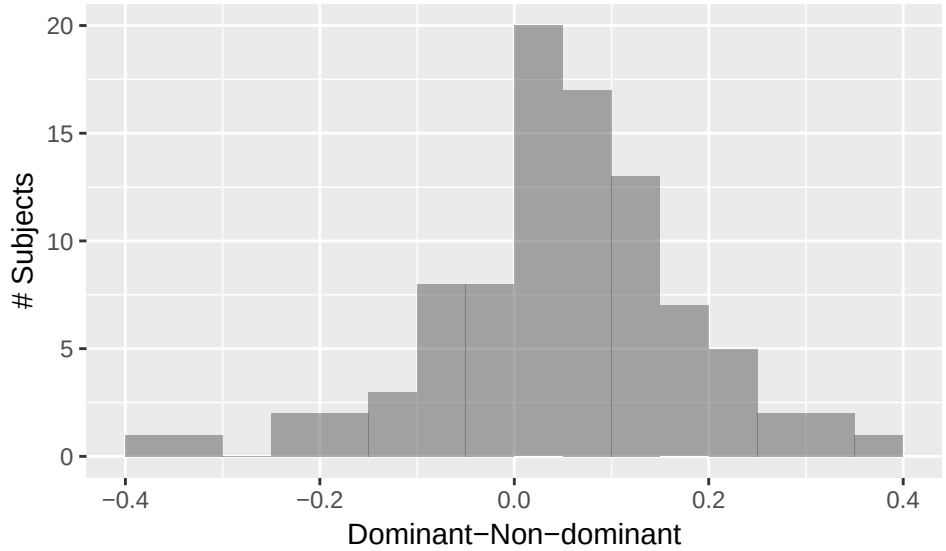
```
##   hHeight = col_integer()
## )
```

```
# table on page 592
Couples %>%
  select(wAge, hAge) %>%
  mutate(difference = hAge - wAge) %>%
  head(n = 7)
```

```
## # A tibble: 7 x 3
##    wAge  hAge difference
##   <dbl> <int>      <dbl>
## 1    43    49          6
## 2    28    25         -3
## 3    30    40         10
## 4    57    52         -5
## 5    52    58          6
## 6    27    32          5
## 7    52    43         -9
```

Step-By-Step Example: A Paired t-Interval

```
DexData <- Dexterity %>%
  select(dominant_1, non_dominant_2) %>%
  mutate(difference = dominant_1 - non_dominant_2) %>%
  filter(dominant_1 < 1) # For some reason, the book has removed one observation where dominant_1 = 1,
gf_histogram(~ difference, data = DexData, binwidth = .05, center = .025) %>%
  gf_labs(x = "Dominant-Non-dominant", y = "# Subjects")
```




```
# Mechanics
ndex <- nrow(DexData) + 1 # the book kept n at 93 for some reason
ndex # number of pairs (children)
```

```
## [1] 93
```

```
ddex <- mean(~ difference, data = DexData)
ddex # mean difference
```

```
## [1] 0.05148209
```

```
sdex <- sd(~ difference, data = DexData)
sdex # standard deviation of the differences
```

```
## [1] 0.1298746
```

```
sedex <- sdex/(ndex^.5)
sedex # standard error of the differences
```

```
## [1] 0.01346736
```

```
df <- ndex - 1
df
```

```
## [1] 92
```

```
tstats <- qt(p = c(.025, .975), df = df)
tstats
```

```
## [1] -1.986086  1.986086
```

```
medex <- tstats  * sedex
medex # margin of error of the differences
```

```
## [1] -0.02674735  0.02674735
```

```
ddex + medex
```

```
## [1] 0.02473474 0.07822943
```

```
# Or, if you don't want to go through all those calculations:
t.test(~ difference, data = DexData, df = df)
```

```
##
##  One Sample t-test
##
## data:  difference
## t = 3.8021, df = 91, p-value = 0.0002592
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.02458583 0.07837834
## sample estimates:
##  mean of x
## 0.05148209
```

Effect Size

XX NH skating data (page 594), workweek data (page 595)

Section 18.4: Blocking

What's Independent?

Random Matters: A Bootstrapped Paired Data Confidence Interval and Hypothesis Test

```
set.seed(2345)
numsim <- 5000
```

```
# What does do() do?
mean(~ difference, data = resample(DexData)) # One mean of a random resample
```

```
## [1] 0.04257654
```

```
mean(~ difference, data = resample(DexData)) # Another mean of a random resample
```

```
## [1] 0.04985414
```

```
do(2) * mean(~ difference, data = resample(DexData)) # Calculates two means
```

```
##         mean
## 1 0.03828900
## 2 0.02499735
```

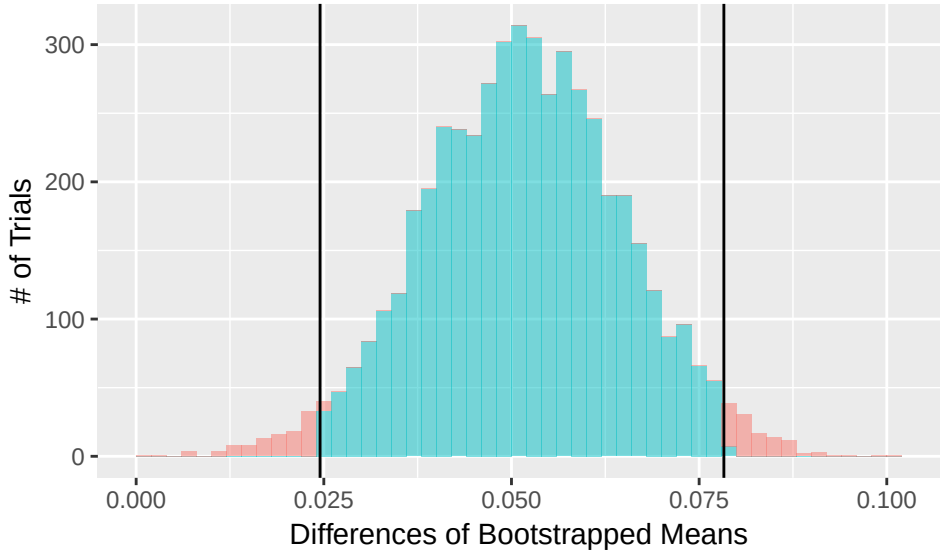
```
# We need numsim means
DexBoots <- do(numsim) * mean(~ difference, data = resample(DexData))

qdata(~ mean, p = c(.025, .975), data = DexBoots)
```

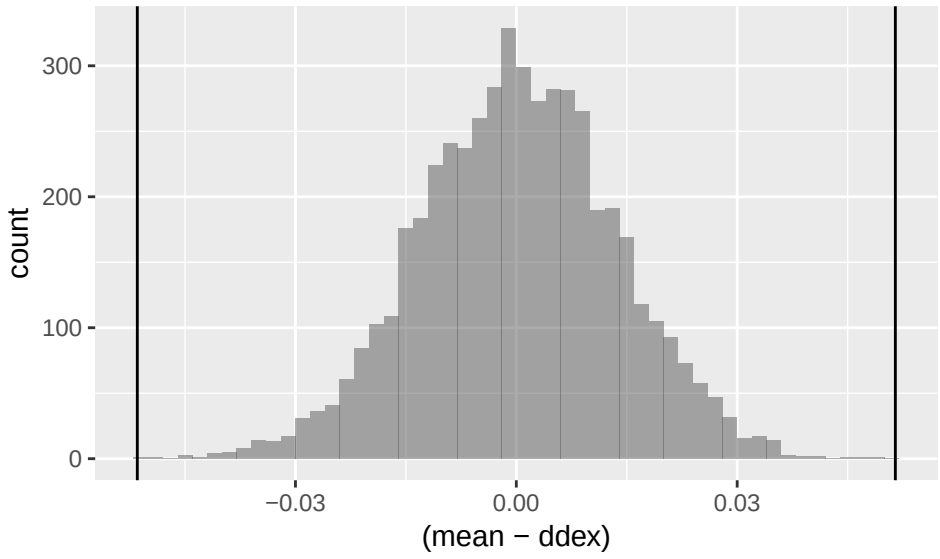
```
##         quantile     p
## 2.5%  0.02515483 0.025
## 97.5% 0.07794129 0.975
```

```
DexBoots <- DexBoots %>%
  mutate(interval = ifelse(mean > .0245 & mean < .0783, "Within 95% Confidence",
                           "Outside 95% Confidence"))

# Figure 18.4, page 597
gf_histogram(~ mean, fill = ~ interval, data = DexBoots, binwidth = .002, center = .001) %>%
  gf_vline(xintercept = .0245) %>%
  gf_vline(xintercept = .0783) %>%
  gf_labs(x = "Differences of Bootstrapped Means", y = "# of Trials") +
  guides(fill = FALSE)
```




```
# Figure 18.5
gf_histogram(~ (mean - ddex), data = DexBoots, binwidth = .002, center = .001) %>%
  gf_vline(xintercept = ddex) %>%
  gf_vline(xintercept = -ddex)
```




```
favstats(~ (mean - ddex), data = DexBoots)
```

```
##          min           Q1      median          Q3         max         mean
##  -0.05087679 -0.008911368 0.000258236 0.009129865 0.04873482 0.0002299513
##           sd    n missing
##  0.01336809 5000       0
```

With `favstats()`, we can see that our minimum is within the interval, but our maximum isn't.

```
DexBoots %>%
  filter((mean - ddex) > ddex)
```

```
## [1] mean     interval
## <0 rows> (or 0-length row.names)
```

Like the book, there is one instance (out of 5,000), so we estimate the P-value as 1/5,000 (the book says 50,000, which is incorrect), or .0002.