

IPS9 in R: Inference for means (Chapter 7)

Bonnie Lin and Nicholas Horton (nhorton@amherst.edu)

July 22, 2018

Introduction and background

These documents are intended to help describe how to undertake analyses introduced as examples in the Ninth Edition of *Introduction to the Practice of Statistics* (2017) by Moore, McCabe, and Craig.

More information about the book can be found [here](#). The data used in these documents can be found under Data Sets in the Student Site. This file as well as the associated R Markdown reproducible analysis source file used to create it can be found at <https://nhorton.people.amherst.edu/ips9/>.

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 7: Inference for means

This file replicates the analyses from Chapter 7: Inference for means.

First, load the packages that will be needed for this document:

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

Section 7.1: Inference for the mean of a population

First, we need to clean up the data of average time spent watching TV and draw a simple random sample (SRS) of size 8 for this problem. We use the following functions to find the mean, standard deviation, and 95% confidence interval as shown on page 411-412. We also check the assumptions and conditions for a Student's t-test by looking at the qq plot.

```
TVTime <- read_csv("https://nhorton.people.amherst.edu/ips9/data/chapter07/EG07-01TVTIME.csv")
```

```
## Warning: Missing column names filled in: 'X2' [2], 'X3' [3], 'X4' [4],
## 'X5' [5], 'X6' [6]
```

```
TVTime <- TVTime %>% select(Time) %>% head(., 8)
favstats(~ Time, data = TVTime)
```

```
##   min    Q1 median    Q3   max mean      sd n missing
##    0 4.875    8.5 20.75 40.5 14.5 14.85405 8      0
```

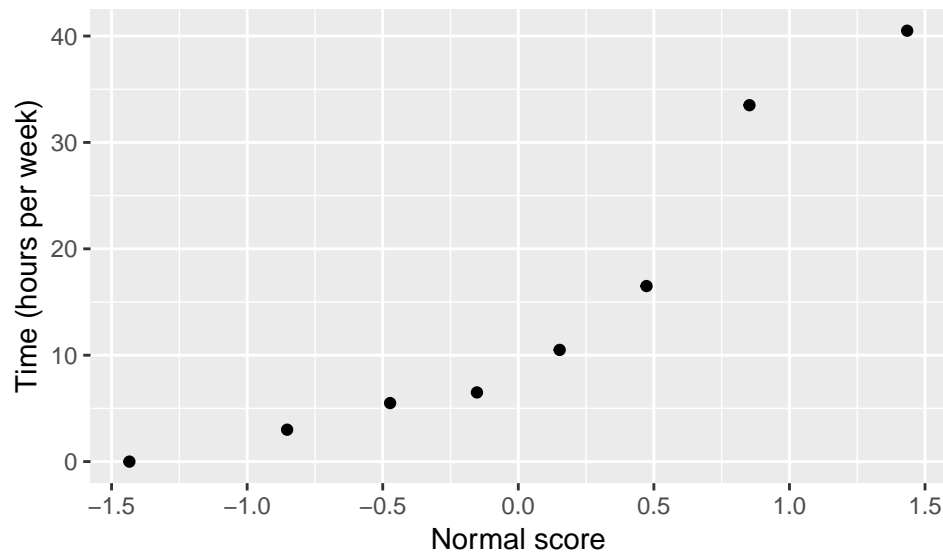
```
t.test(~ Time, data = TVTime)
```

```
##
## One Sample t-test
##
## data:  Time
```

```
## t = 2.761, df = 7, p-value = 0.02806
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 2.081702 26.918298
## sample estimates:
## mean of x
## 14.5
```

Figure 7.2

```
gf_qq(~ Time, data = TVTime) %>%
  gf_labs(x = "Normal score", y = "Time (hours per week)")
```



Then, we can conduct a significance test on the null hypothesis that the sample mean would be equal to the overall U.S. average as demonstrated on page 414:

```
t.test(~ Time, data = TVTime, alternative = "less")
```

```
##
## One Sample t-test
##
## data: Time
## t = 2.761, df = 7, p-value = 0.986
## alternative hypothesis: true mean is less than 0
## 95 percent confidence interval:
## -Inf 24.44976
## sample estimates:
## mean of x
## 14.5
```

Section 7.2: Comparing two means

By performing a significance test between the S&P 500 return and an investor's stock portfolio (page 415-418), we can assess the quality of a broker's management of this portfolio.

```
STOCK <- read_csv("https://nhorton.people.amherst.edu/ips9/data/chapter07/EG07-04STOCK.csv")
favstats(~ Return, data = STOCK)
```

```
##      min      Q1 median      Q3      max      mean      sd  n missing
```

```
## -15.25 -3.255 -1.41 1.99 12.22 -1.099744 5.990888 39 0
sigtest_STOCK <- t.test(~ Return, data = STOCK, alternative = "two.sided")
confint_STOCK <- with(sigtest_STOCK, conf.int)
confint_STOCK - 0.95
```

```
## [1] -3.9917650 -0.1077222
## attr(,"conf.level")
## [1] 0.95
```

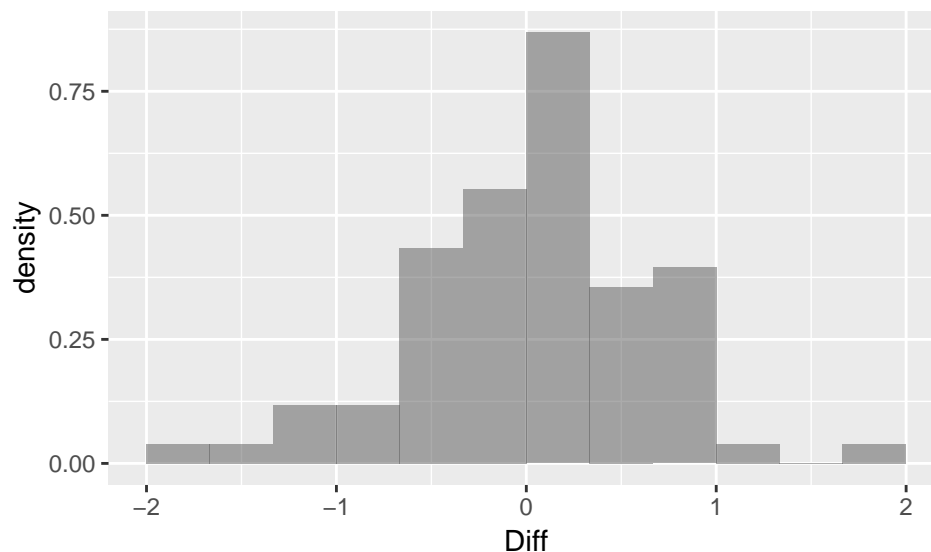
We can use the `with()` function to extract the confidence interval from the `t.test` output. To obtain the corresponding interval for the underperformance, we can estimate the confidence interval of the amount that the investor should be compensated with.

```
GEPARTS <- read_csv("https://nhorton.people.amherst.edu/ips9/data/chapter07/EG07-07GEPARTS.csv")
```

```
## Warning: Missing column names filled in: 'X5' [5]
```

```
gf_dhistogram(~ Diff, data = GEPARTS, binwidth = 1/3, center = 1/6)
```

```
## Warning: Removed 58 rows containing non-finite values (stat_bin).
```



```
with(GEPARTS, t.test(OptionOn, OptionOff, var.equal = TRUE, conf.level = 0.90))
```

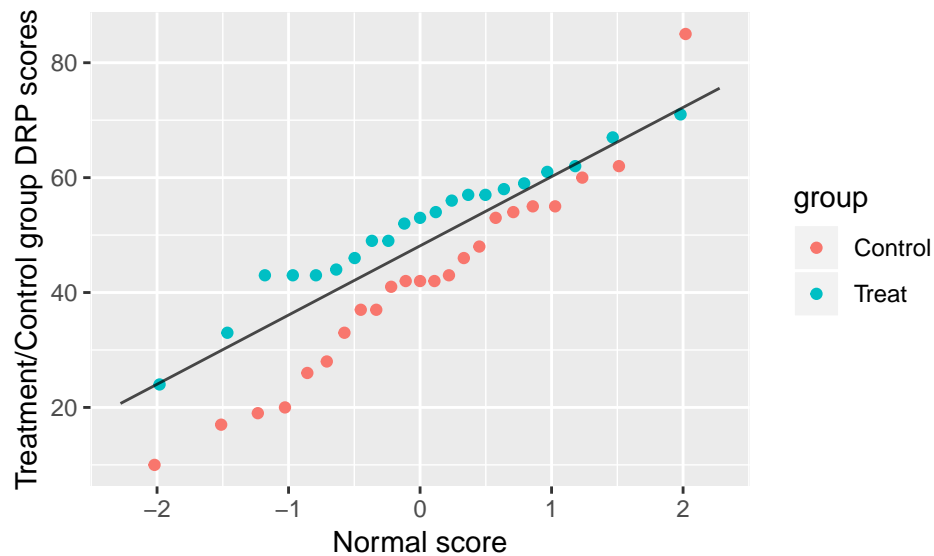
```
##
## Two Sample t-test
##
## data: OptionOn and OptionOff
## t = -0.20815, df = 150, p-value = 0.8354
## alternative hypothesis: true difference in means is not equal to 0
## 90 percent confidence interval:
## -0.2438016 0.1893279
## sample estimates:
## mean of x mean of y
## 118.7345 118.7617
```

```
DRP <- read_csv("https://nhorton.people.amherst.edu/ips9/data/chapter07/EG07-11DRP.csv")
```

```
# Figure 7.12, page 438
```

```
gf_qq(~ drp, color = ~ group, data = DRP) %>%
  gf_qqline(color = "black", linetype = "solid") %>%
```

```
gf_labs(x = "Normal score", y = "Treatment/Control group DRP scores")
```



```
# Summary statistics, page 439
```

```
favstats(drp ~ group, data = DRP)
```

```
##      group min   Q1 median   Q3 max    mean      sd  n missing
## 1 Control  10 30.5    42 53.5  85 41.52174 17.14873 23      0
## 2  Treat  24 44.0    53 58.0  71 51.47619 11.00736 21      0
```

```
# 95% confidence interval for difference between treatment and control groups
```

```
t.test(drp ~ group, data = DRP)
```

```
##
## Welch Two Sample t-test
##
## data: drp by group
## t = -2.3109, df = 37.855, p-value = 0.02638
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -18.67588 -1.23302
## sample estimates:
## mean in group Control    mean in group Treat
##           41.52174           51.47619
```

Note that textbook reports the difference as the mean of treatment minus the mean of the control, while the `t.test()` function here reports the difference in the opposite order.

```
t.test(drp ~ group, data = DRP, alternative = "greater")
```

```
##
## Welch Two Sample t-test
##
## data: drp by group
## t = -2.3109, df = 37.855, p-value = 0.9868
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
## -17.21761      Inf
## sample estimates:
```

```
## mean in group Control    mean in group Treat
##           41.52174           51.47619
```

Again, note that the negated t value can be attributed to the same reason as above.

```
# Example 7.16, page 444
EATER <- read_csv("https://nhorton.people.amherst.edu/ips9/data/chapter07/EG07-16EATER.csv") %>%
  na.omit()
favstats(WTLOSS ~ Group, data = EATER)

##   Group min  Q1 median   Q3 max  mean      sd n missing
## 1 Early 6.3 9.4   10.2 15.1 16.8 11.56 4.306158 5      0
## 2 Late 0.2 1.5    4.6  7.8 11.5  5.12 4.622445 5      0

diffmean(WTLOSS ~ Group, data = EATER)

## diffmean
##      -6.44

# Note that R calculates the difference of the early-eater mean from the later-eater mean

# 95% confidence intervals, page 476
t.test(WTLOSS ~ Group, data = EATER, var.equal = TRUE)

##
## Two Sample t-test
##
## data:  WTLOSS by Group
## t = 2.2794, df = 8, p-value = 0.05212
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.07502285 12.95502285
## sample estimates:
## mean in group Early  mean in group Late
##           11.56           5.12

#Equal variances assumed
t.test(WTLOSS ~ Group, data = EATER)

##
## Welch Two Sample t-test
##
## data:  WTLOSS by Group
## t = 2.2794, df = 7.9601, p-value = 0.05228
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.08070514 12.96070514
## sample estimates:
## mean in group Early  mean in group Late
##           11.56           5.12

#Equal variances not assumed
#var.equal is FALSE by default
```

Since the last row of the dataset had missing values, we piped the data into the `na.omit()` to remove the N/A's from our analysis.

Another way to think about the `var.equal` argument in the `t.test()` function above is in terms of pooled variances. If we want to use the pooled two-sample t procedure, we have to specify `var.equal` to be `TRUE`.

We will demonstrate that in the following example:

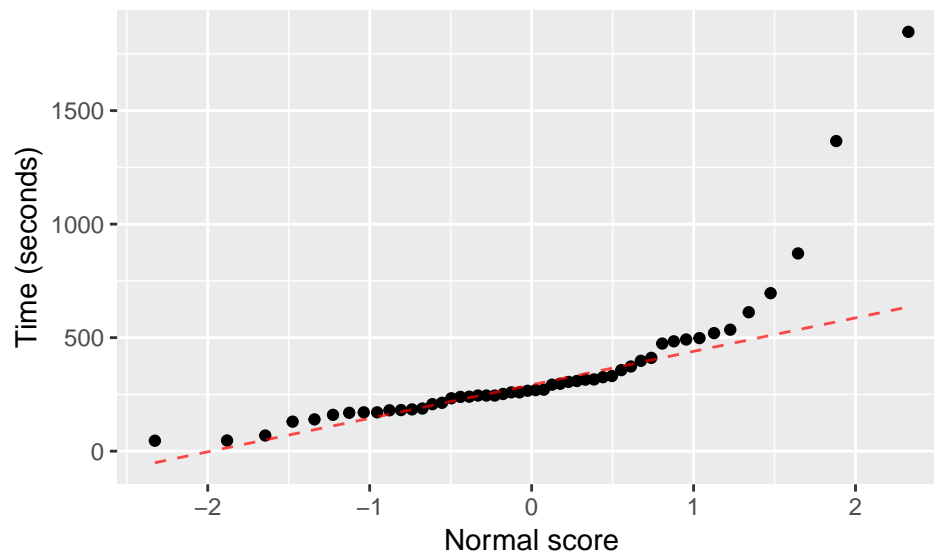
```
# Example 7.19, page 451-452
BP_CA <- read_csv("https://nhorton.people.amherst.edu/ips9/data/chapter07/EG07-18BP_CA.csv")
## XX possibly wrong datapoint?
favstats(dec ~ group, data = BP_CA)

##      group min    Q1 median    Q3 max      mean      sd n missing
## 1 Calcium  -5 -2.75     4 10.75 18  5.0000000 8.743251 10      0
## 2 Placebo -11 -3.00    -1  1.00 12 -0.6363636 5.869799 11      0

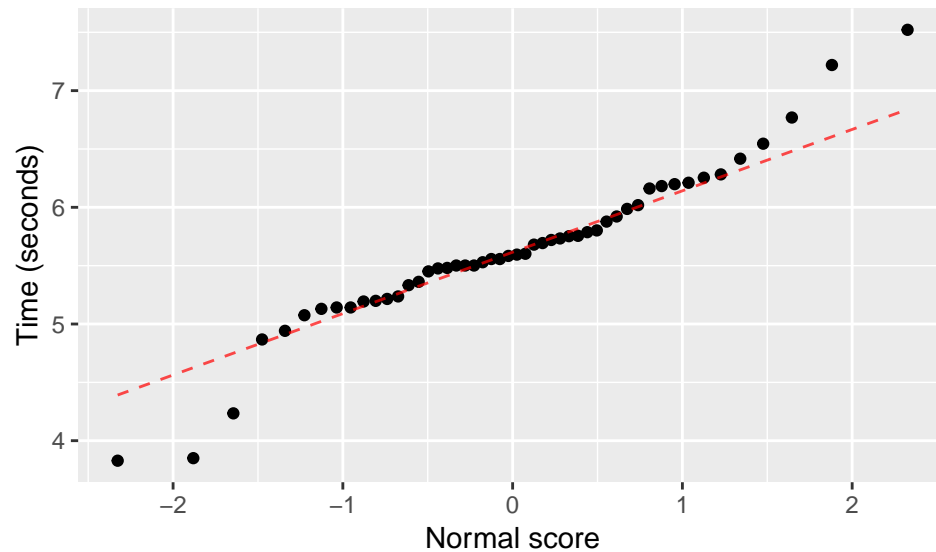
t.test(dec ~ group, data = BP_CA, var.equal = TRUE, conf.level = 0.90)

##
## Two Sample t-test
##
## data:  dec by group
## t = 1.7499, df = 19, p-value = 0.09627
## alternative hypothesis: true difference in means is not equal to 0
## 90 percent confidence interval:
##  0.06682212 11.20590516
## sample estimates:
## mean in group Calcium mean in group Placebo
##      5.0000000      -0.6363636

### Example 7.25, page 470
SONGS <- read_csv("https://nhorton.people.amherst.edu/ips9/data/chapter07/EG07-25SONGS.csv")
## Checking the Normality condition
gf_qq(~ total_secs, data = SONGS) %>%
  gf_qqline(linetype = "dashed", color = "red") %>%
  gf_labs(x = "Normal score", y = "Time (seconds)")
```



```
## Check the condition after *transforming* the variable
gf_qq(~ log(total_secs), data = SONGS) %>%
  gf_qqline(linetype = "dashed", color = "red") %>%
  gf_labs(x = "Normal score", y = "Time (seconds)")
```



```
log_total_secs_SONGS <- SONGS %>% mutate(log_total_secs = log(total_secs))
```

```
## Comparing the 95% confidence intervals
```

```
### With transformation
```

```
t.test(~ log_total_secs, data = log_total_secs_SONGS)
```

```
##
```

```
## One Sample t-test
```

```
##
```

```
## data: log_total_secs
```

```
## t = 58.217, df = 49, p-value < 2.2e-16
```

```
## alternative hypothesis: true mean is not equal to 0
```

```
## 95 percent confidence interval:
```

```
## 5.437069 5.825853
```

```
## sample estimates:
```

```
## mean of x
```

```
## 5.631461
```

```
### Without transformation
```

```
t.test(~ total_secs, data = SONGS)
```

```
##
```

```
## One Sample t-test
```

```
##
```

```
## data: total_secs
```

```
## t = 8.1308, df = 49, p-value = 1.206e-10
```

```
## alternative hypothesis: true mean is not equal to 0
```

```
## 95 percent confidence interval:
```

```
## 266.5823 441.6177
```

```
## sample estimates:
```

```
## mean of x
```

```
## 354.1
```

Since the logarithmic transformation made the Normal quantile plot distribution appear approximately Normal, we created a dataset called `log_total_secs` with the transformed variable. `###` Section 7.3: Additional topics on inference