

Introduction to the Practice of Statistics using R: Chapter 7

Nicholas J. Horton* Ben Baumer

March 29, 2013

Contents

| | | |
|----------|---|----------|
| 1 | Inference for the mean of a population | 2 |
| 1.1 | Transformations | 4 |
| 1.2 | Sign test | 6 |
| 2 | Comparing two means | 6 |
| 2.1 | Confidence interval | 7 |
| 2.2 | Pooled two-sample t procedure | 7 |
| 3 | Optional topics | 9 |
| 3.1 | F test for equality of spread | 9 |
| 3.2 | Power | 9 |

Introduction

This document is intended to help describe how to undertake analyses introduced as examples in the Sixth Edition of *Introduction to the Practice of Statistics* (2002) by David Moore, George McCabe and Bruce Craig. More information about the book can be found at <http://bcs.whfreeman.com/ips6e/>. This file as well as the associated `knitr` reproducible analysis source file can be found at <http://www.math.smith.edu/~nhorton/ips6e>.

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 vignette (<http://cran.r-project.org/web/packages/mosaic/vignettes/MinimalR.pdf>).

To use a package within R, it must be installed (one time), and loaded (each session). The package can be installed using the following command:

*Department of Mathematics and Statistics, Smith College, nhorton@smith.edu

```
> install.packages('mosaic') # note the quotation marks
```

The # character is a comment in R, and all text after that on the current line is ignored. Once the package is installed (one time only), it can be loaded by running the command:

```
> require(mosaic)
```

This needs to be done once per session. We also set some options to improve legibility of graphs and output.

```
> trellis.par.set(theme=col.mosaic()) # get a better color scheme for lattice
> options(digits=3)
```

The specific goal of this document is to demonstrate how to replicate the analysis described in Chapter 7: Inference for Distributions.

1 Inference for the mean of a population

It is straightforward to undertake inference for a single population using R. For example, the results from Example 7.1 (page 421) can be reproduced using the following commands

```
> x = c(5,6, 0, 4, 11, 9, 2, 3)
> favstats(x)

min   Q1 median   Q3 max mean   sd n missing
  0 2.75   4.5 6.75  11   5 3.63 8      0

> length(x)

[1] 8

> tstar = qt(.975, df=length(x) - 1)
> tstar

[1] 2.36

> moe = sd(x) / sqrt(length(x))
> moe

[1] 1.28

> mean(x) + c(-tstar, tstar) * moe

[1] 1.97 8.03
```

As the authors note (page 421), we are 95% confident that the US population's average time spent listening to full-track music on a cell phone is between 2.0 and 8.0 hours per month. Since

this interval does not contain the null value of 8.3 hours, these data suggest that on average, a US subscriber listens to less full-track music.

Example 7.2 (pages 422–423) continues this example using a one-sample t-test, specifically assessing whether the mean in the US is different than 8.3 hours.

```
> t = (mean(x) - 8.3) / (sd(x)/sqrt(length(x)))
> t

[1] -2.57

> pt(t, df=length(x) - 1) # one tail

[1] 0.0184

> 1 - pt(abs(t), df=length(x) - 1) # right tail

[1] 0.0184

> 2* pt(t, df=length(x) - 1) # two sided test (since our statistic was negative)

[1] 0.0368

> 2*(1 - pt(abs(t), df=length(x) - 1)) # two sided test (this always works)

[1] 0.0368
```

Example 7.4 (pages 424–425) considers a one-sample test of stock portfolio diversification.

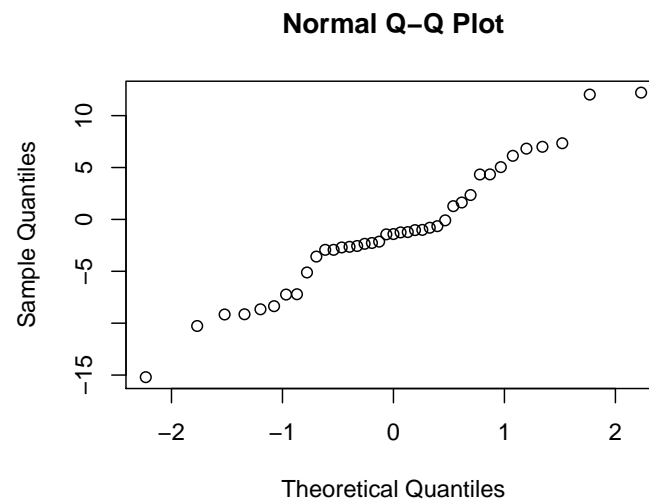
```
> ds = read.csv("http://www.math.smith.edu/ips6eR/ch07/ta07_001.csv")
> favstats(~ return, data=ds)

  min    Q1 median    Q3  max mean   sd  n missing
-15.2 -3.25  -1.41  1.99 12.2 -1.1 5.99 39         0

> with(ds, qqnorm(return))
> with(ds, t.test(return-0.95))
```

One Sample t-test

```
data:  return - 0.95
t = -2.14, df = 38, p-value = 0.03914
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -3.989 -0.107
sample estimates:
mean of x
 -2.05
```



Example 7.7 (pages 428–429) considers whether there is a statistically significant difference in the aggressive behaviors of dementia patients on moon days vs. other days.

```
> ds = read.csv("http://www.math.smith.edu/ips6eR/ch07/ta07_002.csv")
> favstats(~ aggdiff, data=ds)
```

```
   min   Q1 median   Q3  max mean   sd  n missing
-0.02 1.84   2.68 3.35 4.41 2.43 1.46 15      0
```

```
> with(ds, stem(aggdiff))
```

The decimal point is at the |

```
-0 | 0
 0 | 11
 1 | 6
 2 | 1347
 3 | 11167
 4 | 44
```

```
> with(ds, t.test(aggdiff))
```

One Sample t-test

```
data:  aggdiff
t = 6.45, df = 14, p-value = 1.518e-05
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
```

```
1.62 3.24
sample estimates:
mean of x
      2.43
```

These results are consistent with those from the text.

1.1 Transformations

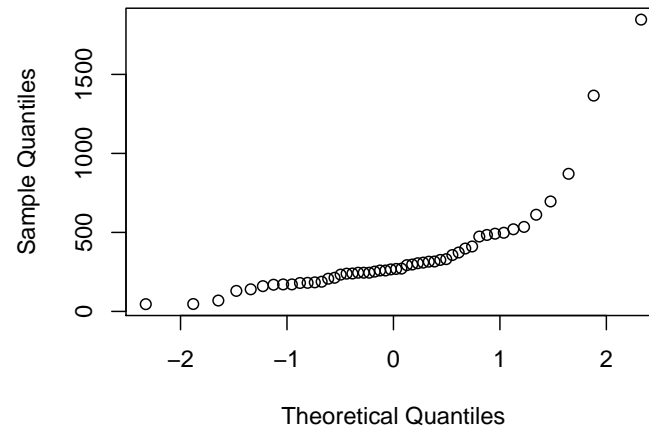
Example 7.11 (pages 436–437) considers the length of audio files on an iPod (which are dramatically right skewed). A log transformation is indicated (as seen below).

```
> ds = read.csv("http://www.math.smith.edu/ips6eR/ch07/ta07_003.csv")
> names(ds)

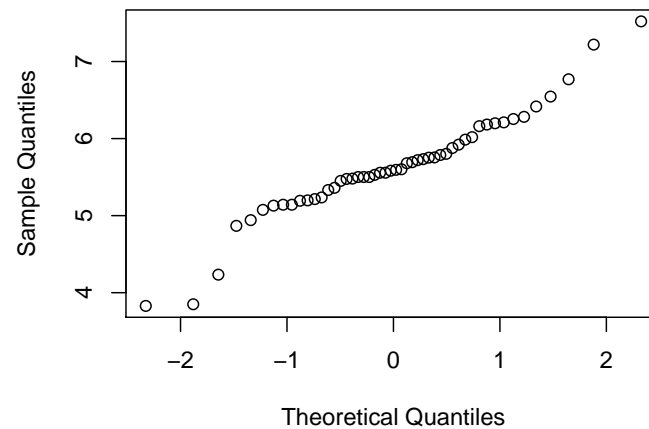
[1] "min"      "sec"      "total_secs"

> ds = transform(ds, logtotal = log(total_secs))
> with(ds, qqnorm(total_secs))
> with(ds, qqnorm(logtotal))
```

Normal Q-Q Plot



Normal Q-Q Plot



```
> with(ds, t.test(logtotal))
```

One Sample t-test

data: logtotal

t = 58.2, df = 49, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

5.44 5.83

sample estimates:

mean of x

5.63

1.2 Sign test

The sign test can be undertaken using the `pbinom()` command (as described in Example 7.12 on pages 438–439).

```
> 1 - pbinom(13, size=15, prob=0.5)
[1] 0.000488
```

2 Comparing two means

Example 7.14 (pages 450–453) compares the DRP scores from two samples of third-graders randomly assigned to a treatment group and control group. We can replicate the parts and pieces of this comparison as well as undertake a two sample (equal variance) t-test.

```
> ds = read.csv("http://www.math.smith.edu/ips6eR/ch07/ta07_004.csv")
> mean(drp ~ group, data=ds)

Control   Treat
   41.5    51.5

> diff(mean(drp ~ group, data=ds))

Treat
 9.95

> sd(drp ~ group, data=ds)

Control   Treat
   17.1    11.0
```

We wouldn't undertake a one-sided test, but that's the approach suggested by the authors (page 453):

```
> t.test(drp ~ group, alternative="less", data=ds)

Welch Two Sample t-test

data:  drp by group
t = -2.31, df = 37.9, p-value = 0.01319
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
 -Inf -2.69
sample estimates:
mean in group Control   mean in group Treat
           41.5           51.5
```

2.1 Confidence interval

Example 7.15 (page 454) calculates a 95% confidence interval for the mean improvement in the entire population of third-graders.

```
> t.test(drp ~ group, data=ds) # to match bottom of page 452

Welch Two Sample t-test

data: drp by group
t = -2.31, df = 37.9, p-value = 0.02638
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -18.68 -1.23
sample estimates:
mean in group Control    mean in group Treat
                41.5                51.5
```

This interval matches the results at the top of page 455.

2.2 Pooled two-sample t procedure

While not generally recommended, the pooled two sample procedures can be fit within R. By default the *unequal* variance test is calculated. The `var.equal=` option can be set to change this. (But first note the error in the dataset, where the eighth placebo subject is miscoded (should be 114, 112, decrease=2, not -2).

```
> ds = read.csv("http://www.math.smith.edu/ips6eR/ch07/ta07_005.csv")
> ds[18,]

   id  group g beg end dec
18 18 Placebo 1 112 114 -2

> ds[18,"dec"] = 2 # note error from table 7.5 page 463
> ds[18,]

   id  group g beg end dec
18 18 Placebo 1 112 114  2

> t.test(dec ~ group, alternative="greater", var.equal=TRUE, data=ds)

Two Sample t-test

data: dec by group
t = 1.63, df = 19, p-value = 0.05935
```



```

alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
-0.307      Inf
sample estimates:
mean in group Calcium mean in group Placebo
           5.000           -0.273

```

Again: the one-sided test is used (though we might question this).

To get a two-sided confidence interval, we fit the `t.test` without the `alternative=` option:

```

> t.test(dec ~ group, var.equal=TRUE, data=ds)

Two Sample t-test

data:  dec by group
t = 1.63, df = 19, p-value = 0.1187
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-1.48 12.03
sample estimates:
mean in group Calcium mean in group Placebo
           5.000           -0.273

```

We would probably still suggest reporting the unequal variance (unpooled) results:

```

> t.test(dec ~ group, var.equal=FALSE, data=ds)

Welch Two Sample t-test

data:  dec by group
t = 1.6, df = 15.6, p-value = 0.1288
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-1.71 12.26
sample estimates:
mean in group Calcium mean in group Placebo
           5.000           -0.273

```

These are slighter wide (but require less unverifiable assumptions).

3 Optional topics

3.1 F test for equality of spread

While we don't recommend using the test for equality for spread, this is straightforward to undertake in R. The values displayed on page 475 can be generated with the command:

```
> qf(c(0.90, 0.95, 0.0975, 0.99, 0.999), df1=9, df2=10)
[1] 2.35 3.02 0.41 4.94 8.96
```

We can carry out the test of equal variances using `var.test()`:

```
> var.test(dec ~ group, data=ds)

F test to compare two variances

data:  dec by group
F = 2.2, num df = 9, denom df = 10, p-value = 0.2365
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.581 8.703
sample estimates:
ratio of variances
                2.2
```

3.2 Power

Power calculations, such as the one described in Example 7.23 (page 478) can be undertaken using the `power.t.test()` function. Suppose that we wanted to plan a new study to provide convincing evidence at the $\alpha = 0.01$ level, with 45 subjects in each of our two groups. We believe that the true different in means is 5 points, and we assume that the population standard deviation is 7.4 for both groups (this corresponds to an effect size of $5/7.4 = 0.676$).

```
> power.t.test(delta=5, n=45, sd=7.4, alternative="one.sided", sig.level=0.01)

Two-sample t test power calculation

      n = 45
  delta = 5
     sd = 7.4
sig.level = 0.01
  power = 0.797
alternative = one.sided

NOTE: n is number in *each* group
```

We'd still do the two-sided test (which will have slightly less power):

```
> power.t.test(delta=5, n=45, sd=7.4, sig.level=0.01)
```

```
Two-sample t test power calculation
```

```

      n = 45
    delta = 5
      sd = 7.4
sig.level = 0.01
  power = 0.715
alternative = two.sided
```

```
NOTE: n is number in *each* group
```

An even more flexible approach would be to simulate data to calculate the power. This can be extended to other settings for which the existing power functions do not handle.

We first need to write a function which samples the two groups under the assumptions of the power calculation.

```
> gendata = function() {
  n = 45
  sd = 7.4
  diff = 5
  y1 = rnorm(n, mean=0, sd=sd)
  y2 = rnorm(n, mean=diff, sd=sd)
  y = c(y1, y2)
  x = c(rep("Control", n), rep("Treatment", n))
  return(data.frame(y, x))
}
```

We can repeatedly call this function, carry out our *t*-test, and save the *p*-value (using the `pval()` function in the `mosaic` package. Using 5000 simulations is sufficient to estimate the proportion of times with fair accuracy (if the power was 0.50, then the standard error of the proportion would be $\sqrt{0.5^2/5000} = 0.007$).

Finally, we tally how many of these were less than our desired alpha level (0.01).

```
> powersim = do(5000) * pval(t.test(y ~ x, data=gendata()))
> head(powersim)
```

```

  p.value
1 0.015684
2 0.062431
3 0.013455
4 0.000212
5 0.008053
6 0.010064
```

```
> options(digits=5)
> tally(~ p.value <= 0.01, format="percent", data=powersim)
```

| TRUE | FALSE | Total |
|-------|-------|--------|
| 71.96 | 28.04 | 100.00 |

The results are quite consistent with the analytic power calculation.