Introduction to the Practice of Statistics using R: Chapter 16

Ben Baumer Nicholas J. Horton*

March 29, 2013

Contents

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:

```
> 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:

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

```
> 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)
```

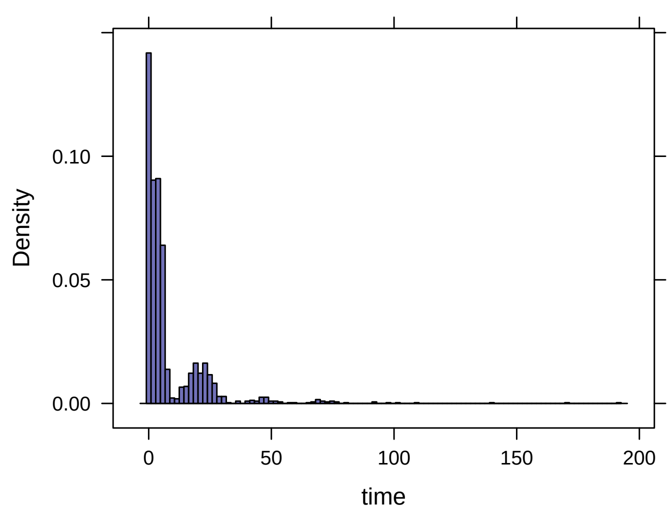
The specific goal of this document is to demonstrate how to replicate the analysis described in Chapter 16: Bootstrap Methods and Permutation Tests.

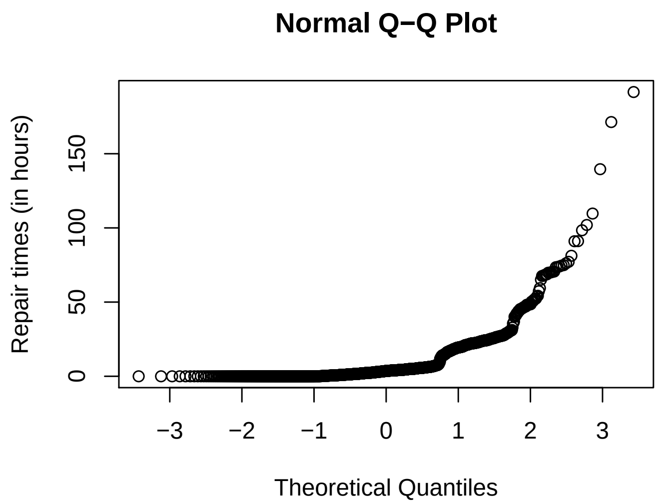
1 The Bootstrap Idea

The bootstrap is a fundamental concept in statistical computing, and the requisite calculations are very easy to perform in R.

The repair time data from Verizon shown in Figure 16.1 (page 16-4) can be plotted thusly:

```
> verizon = read.csv("http://www.math.smith.edu/ips6eR/ch16/eg16_001.csv")
> xhistogram(~time, data=verizon, nint=100)
> with(verizon, qqnorm(time, ylab="Repair times (in hours)"))
```





A command to facilitate resampling within the `mosaic` package is `resample()`. We get our first example on page 16-5, which considers a subset of size $n = 6$ from the Verizon dataset.

```
> data = c(3.12, 0, 1.57, 19.67, 0.22, 2.2)
> mean(data)

[1] 4.46

> s1 = resample(data)
> s1

[1] 0.00 0.22 1.57 2.20 2.20 3.12

> mean(s1)

[1] 1.55

> s2 = resample(data)
> s2

[1] 19.67  2.20 19.67  1.57  2.20  1.57

> mean(s2)

[1] 7.81

> s3 = resample(data)
> s3

[1]  0.22 19.67  3.12  2.20  0.00  3.12

> mean(s3)

[1] 4.72
```

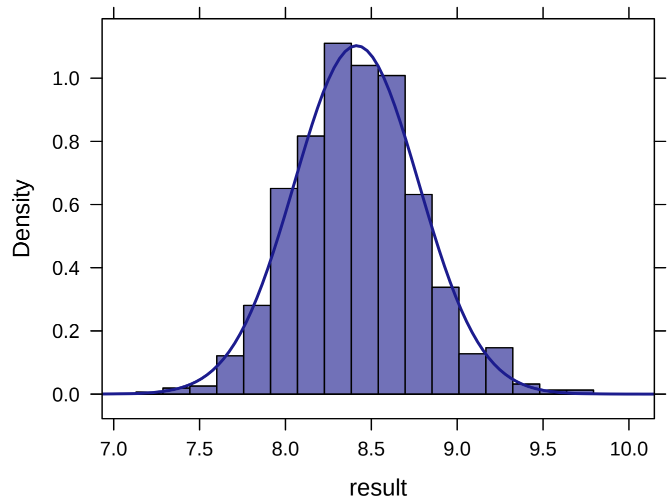
Note that the results shown here do not match the book, due to the random nature of resampling.

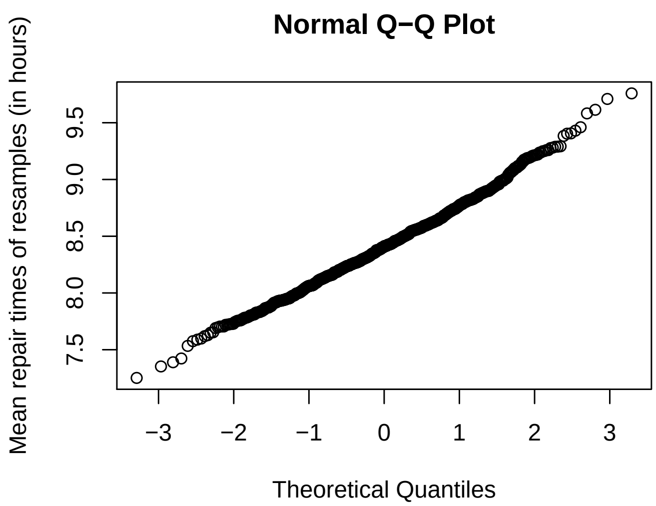
In Figure 16.3 (page 16-6) we visualize a bootstrap distribution. To construct such a thing, we use the `do()` command, which simply repeats some operation many times, and collects the results in a data frame.

```
> mean(~time, data=verizon)

[1] 8.41

> mean(~time, data=resample(verizon))

[1] 8.26

> mean(~time, data=resample(verizon))

[1] 8.94

> mean(~time, data=resample(verizon))

[1] 8.66

> bootstrap = do(1000) * mean(time, data=resample(verizon))
> favstats(~result, data=bootstrap)

  min   Q1 median   Q3  max mean    sd    n missing
 7.25 8.17   8.41 8.63 9.76 8.41 0.362 1000       0

> # Theoretical standard error
> 14.69 / sqrt(1664)

[1] 0.36
```

Note how the theoretical standard error (i.e. standard deviation of the sampling distribution of the mean) compares to the standard deviation from the bootstrap sample.

```
> xhistogram(~result, data=bootstrap, fit="normal")
> with(bootstrap, qqnorm(result, ylab="Mean repair times of resamples (in hours)"))
```

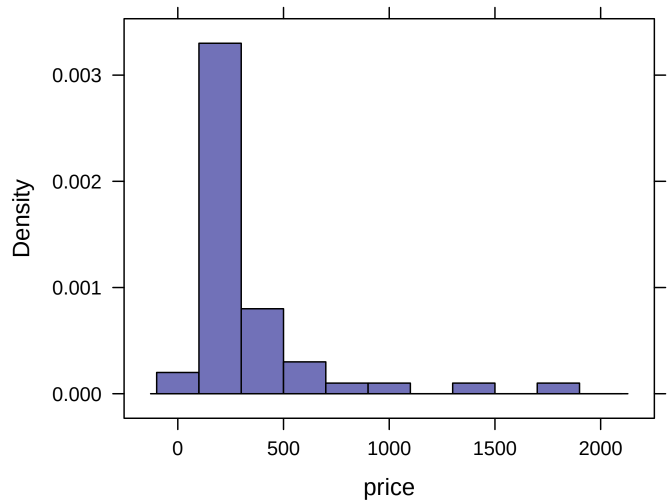


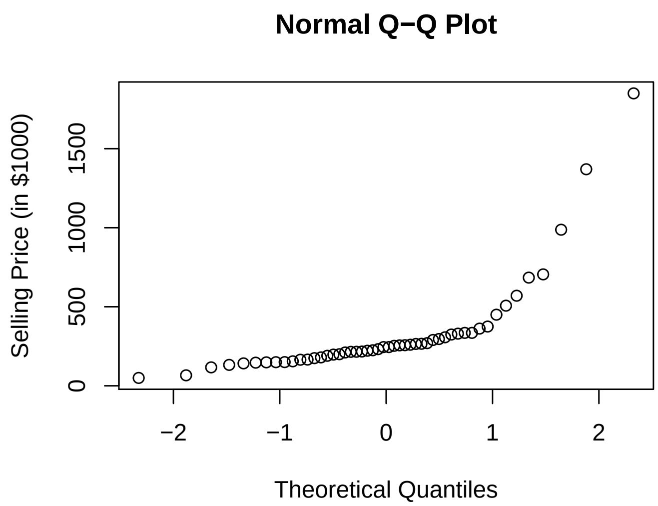


2 First Steps in Using the Bootstrap

Table 16.1 and Figure 16.6 (page 16-14) display residential and commercial real estate prices in Seattle.

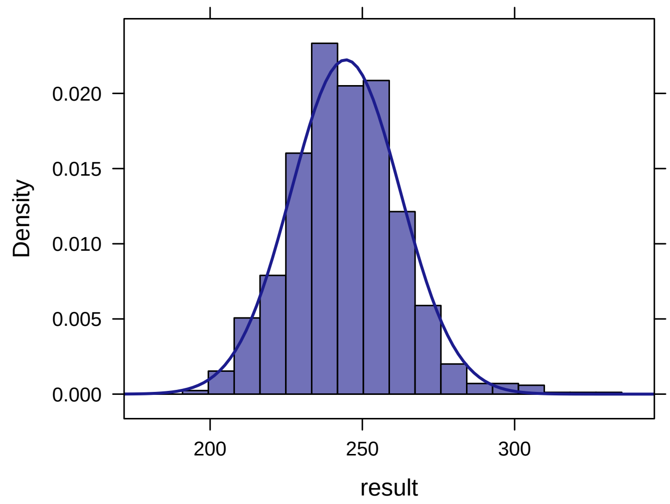
```
> seattle = read.csv("http://www.math.smith.edu/ips6eR/ch16/ta16_001.csv")
> names(seattle) = c("price")
> xhistogram(~price, data=seattle)
> with(seattle, qqnorm(price, ylab="Selling Price (in $1000)"))
```

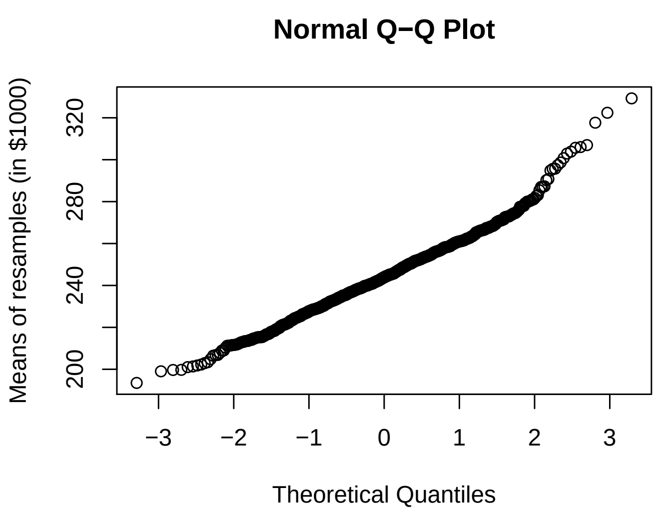




In this example we are working with the 25% trimmed mean. To find the 25% trimmed mean, we grab only the middle 50% of the data, and compute the mean on this subset. This can be achieved using the `trim` argument to `mean()`.

```
> mean(~price, trim=0.25, data=seattle)

[1] 244

> bootstrap = do(1000) * mean(~price, trim=0.25, data=resample(seattle))
> favstats(~result, data=bootstrap)

 min  Q1 median  Q3 max mean sd    n missing
 194 233    244 256 329  245 18 1000       0

> xhistogram(~result, data=bootstrap, fit="normal")
> with(bootstrap, qqnorm(result, ylab="Means of resamples (in $1000)"))
```





We compute the bias as the difference between the average of the bootstrapped means and the trimmed mean from the original sample.

```
> # bias
> mean(~result, data=bootstrap) - mean(~price, trim=0.25, data=seattle)

[1] 0.539
```

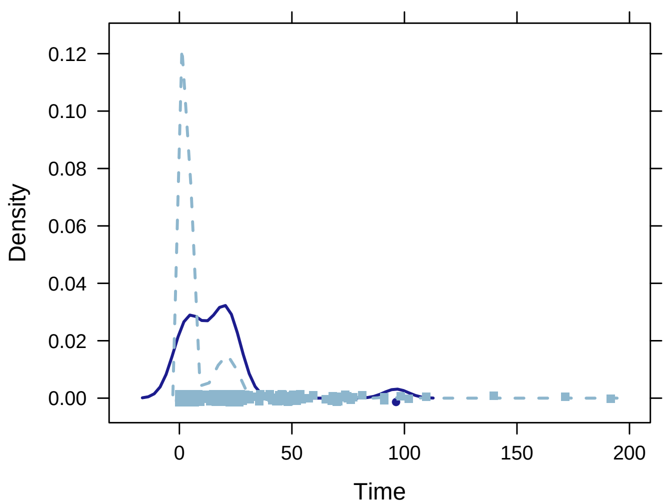
The computation of the confidence interval in Example 16.5 (page 16-16) makes use of the t-distribution.

```
> se.boot = sd(~result, bootstrap)
> t.star = qt(0.975, df=49)
> t.star
```

```
[1] 2.01

> moe = t.star * se.boot
> mean(~price, trim=0.25, data=seattle) + c(-moe, moe)

[1] 208 280
```

In Example 16.6, we compare the means of two groups of service providers.

```
> CLEC = read.csv("http://www.math.smith.edu/ips6eR/ch16/eg16_006.csv")
> mean(Time ~ Group, data=CLEC)

 CLEC  ILEC
16.51  8.41

> densityplot(~Time, groups=Group, data=CLEC)
```



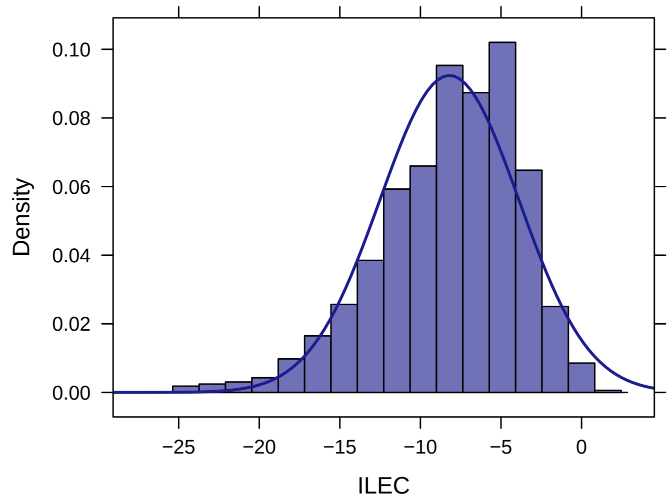
We then construct a bootstrap distribution for the difference in means among the two groups.

```
> bstrap = do(1000) * diff(mean(Time ~ Group, data=resample(CLEC)))
> favstats(~ILEC, data=bstrap)

   min    Q1 median Q3   max mean   sd    n missing
 -25.3 -10.8  -7.64 -5 0.917 -8.2 4.32 1000       0
```

Note that the resulting distribution is not quite so normal. Thus, we can use the quantile method to produce a bootstrap percentile confidence interval for the mean.

```
> xhistogram(~ILEC, fit="normal", data=bstrap)
> qdata(c(0.025, 0.975), vals=ILEC, data=bstrap)
```

```
 2.5% 97.5%
-18.0  -1.4
```



3 How Accurate is a Bootstrap Distribution?

4 Bootstrap Confidence Intervals

We return to the construction of a confidence interval for the mean price of real estate in Seattle explored in Example 16-5. To the t-based confidence interval we constructed previously, we can add the percentile-based confidence interval

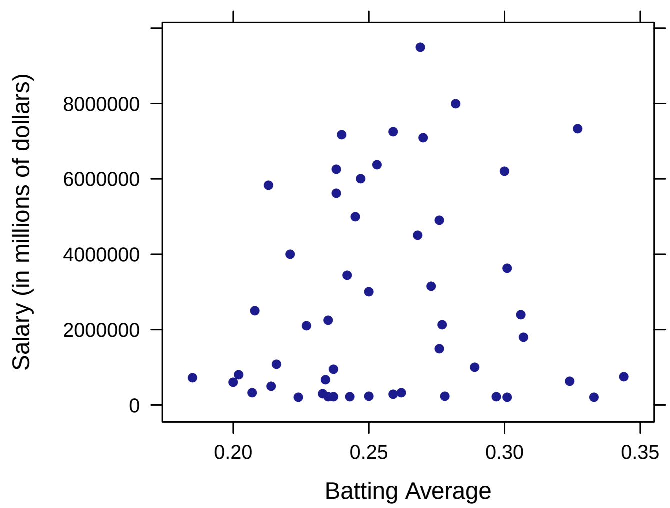
```
> mean(~price, trim=0.25, data=seattle) + c(-moe, moe)

[1] 208 280

> qdata(c(0.025, 0.975), vals=result, data=bootstrap)

 2.5% 97.5%
  212   281
```

Note that the bootstrapped confidence interval is not quite symmetric with respect to the sample mean of 244.

4.1 Confidence intervals for the correlation

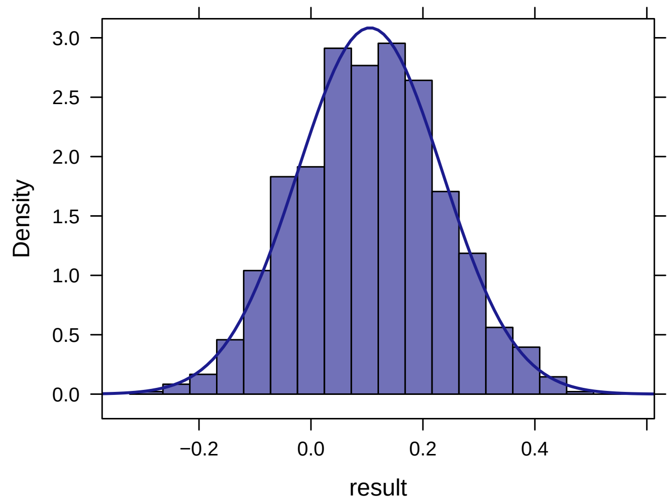
In Example 16.10 (page 16-35), we explore the correlation between batting average and player salary in Major League Baseball. The value of the correlation coefficient among the 50 players in Table 16.2 (page 16-36) is relatively small.

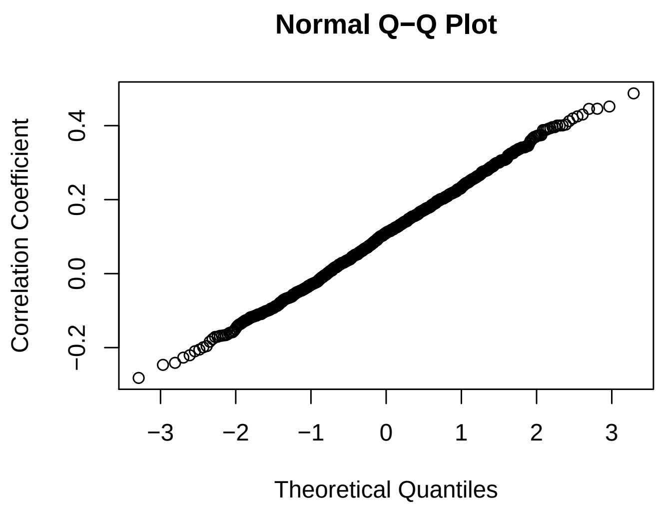
```
> MLB = read.csv("http://www.math.smith.edu/ips6eR/ch16/ta16_002.csv")
> names(MLB)[2] = "Salary"
> xyplot(Salary ~ Average, data=MLB, xlab="Batting Average"
        , ylab="Salary (in millions of dollars)")
> with(MLB, cor(Salary, Average))

[1] 0.107
```



To construct a bootstrap distribution for the correlation between batting average and salary, we resample the players and compute the correlation coefficient.

```
> cor.boot = do(1000) * with(resample(MLB), cor(Salary, Average))
> xhistogram(~result, data=cor.boot, fit="normal")
> with(cor.boot, qqnorm(result, ylab="Correlation Coefficient"))
```





In this case, the t-based confidence interval for the correlation coefficient

```
> se.boot = sd(~result, cor.boot)
> t.star = qt(0.975, df=(nrow(MLB) - 1))
> t.star

[1] 2.01

> moe = t.star * se.boot
> with(MLB, cor(Salary, Average)) + c(-moe, moe)

[1] -0.153  0.367
```

is in reasonable agreement with the percentile-based method.

```
> qdata(c(0.025, 0.975), vals=result, data=cor.boot)

  2.5%  97.5%
-0.137  0.366
```