Exploring the Bootstrap Method with R Coding

Introduction

The bootstrap method is an interesting procedure used in data science to get more accurate results from the same amount of data. It is by far my favorite concept in data analysis and computing because it effectively creates new data out of old data. I also felt that the bootstrap method also was not well covered in our stat 133 class, especially since you need a computer to do it and its relevance in the data science world. In this post, I will explain what the bootstrap is and how it works with the real data collected by Michealson and Morley in 1882 on the speed of light in air.

Data

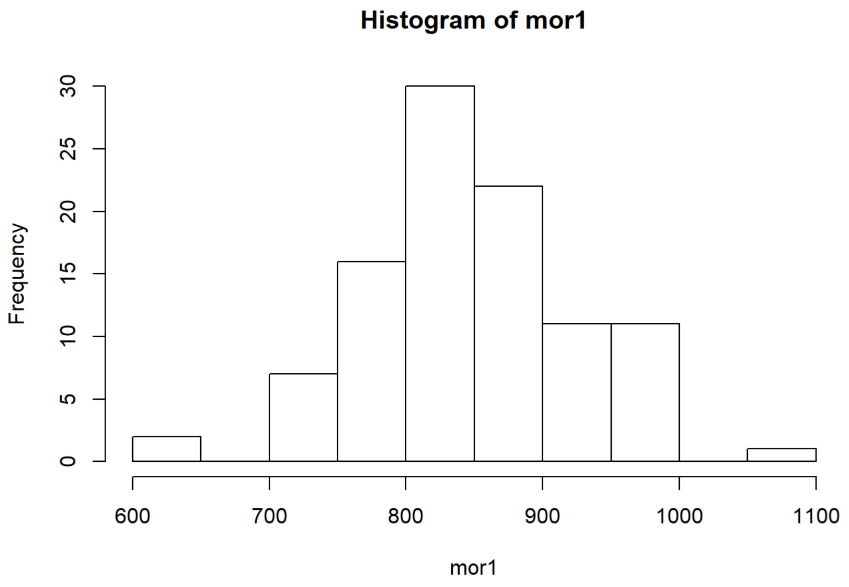
The data I am going to use is pre-loaded in R studio under the name ‘morley’. Since I am only interested in the speeds gathered, and not the experiment labels, I am going to define mor1 and the last column of the morley dataframe.

```
mor1 = morley[,3]
```

Visualize current data with histogram

To better grasp the data we are playing with, here is the 99.7% confidence interval, derived from the data captured by the 3 standard deviation calculation for normal distributions, for the data.

```
# histogram of the data
hist(mor1)
```



```
# standard deviation of the data
sdmor1 = sd(mor1)

# lower bound for 99.7% confidence interval
lowb = mean(mor1) - (sdmor1*3)

# upper bound
upb = mean(mor1) + (sdmor1*3)

# 99.7% confidence interval (add 29900 because thats the true speed in km/s according to the documentation of this
data)
paste('The true speed of light in air, with 99.7% confidence, is between', (lowb+299000), 'km/s', 'and', (upb+2990
00), 'km/s')
```

```
## [1] "The true speed of light in air, with 99.7% confidence, is between 299615.368356543 km/s and 300089.4316434
57 km/s"
```

```
# or make a value with plus or minus for errors
error = mean(mor1)-lowb
paste('The true speed of light in air is, with 99.7% confidence,', mean(mor1)+299000, '±', error, 'km/s')
```

```
## [1] "The true speed of light in air is, with 99.7% confidence, 299852.4 ± 237.031643457155 km/s"
```

Clearly this is not very accurate, since the error is over 200 km/s, so we will use the bootstrap method to get a more precise result. (not more accurate)

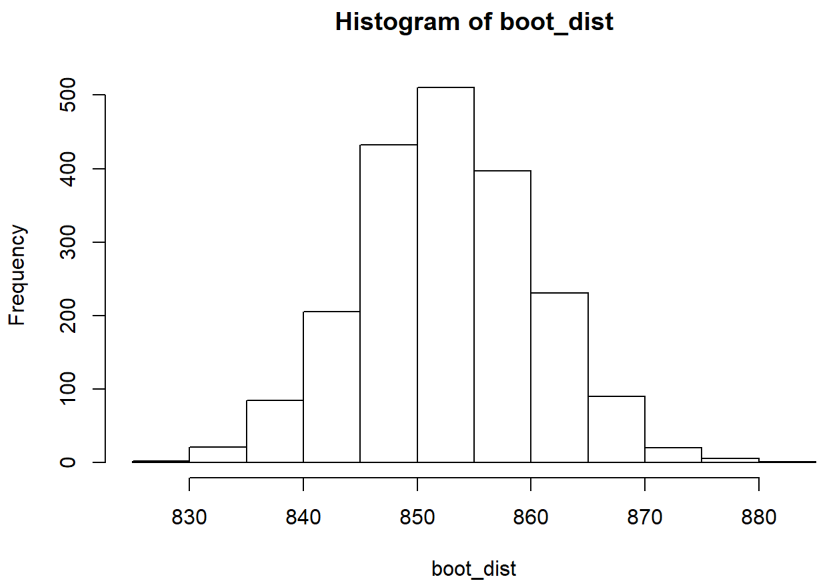
Calculate new data from old

The bootstrap method works through taking a sample of data, in this case mor1, and randomly taking values from the sample with replacement to form a new sample. After getting this new sample, you can find its mean, which is almost certainly different from the original sample. Running this same process over and over again provides a new distribution that will be more precise around the true value that I am looking for. In the next block of code, I define functions that do the described method and provide its respective histogram. The first function, resample_mean() resamples a given vector and returns the mean of the new resampled vector of equal length. The second function returns a distribution of 2000 resampled means.

```r
resample_mean = function(vec) {
    newval = 0
    resample = c()
    for (i in 1:100) # 100 because the size of the resample must equal the size of the sample
        newval = sample(vec, replace = TRUE)
        resample = append(resample, newval)
    return(mean(resample))
}


dist_means = function(vec) {
    new_mean = 0
    means_dist = c()
    for (i in 1:2000) { # I choose to run only 2000 repetitions to avoid waiting too long for the computation
        new_mean = resample_mean(vec)
        means_dist = append(means_dist, new_mean)
    }
    return(means_dist)
}


set.seed(7) # I use the set.seed() function here so that my results are reproducible
boot_dist = dist_means(mor1)
hist(boot_dist)
```



Histogram of boot_dist

Details of First Bootstrapped data

```r
# standard deviation of the new data
sdboot_dist = sd(boot_dist)

# lower bound for 99.7% confidence interval
lowb = mean(boot_dist) - (3*sdboot_dist)

# upper bound for 99.7% confidence interval
upb = mean(boot_dist) + (3*sdboot_dist)

# 99.7% confidence interval
paste('The true speed of light in air, with 99.7% confidence, is between', (lowb+299000), 'km/s', 'and', (upb+2990
00), 'km/s')
```

```
## [1] "The true speed of light in air, with 99.7% confidence, is between 299829.399021673 km/s and 299875.9938783
27 km/s"
```

```
# or make a value with plus or minus for errors
error = mean(boot_dist)-lowb
paste('The true speed of light in air is, with 99.7% confidence,', mean(boot_dist)+299000, '±', error, 'km/s')
```

```
## [1] "The true speed of light in air is, with 99.7% confidence, 299852.69645 ± 23.2974283271959 km/s"
```

The Cool Part

Something interestin about this data is that it is more accurate than one might expect. Michealson, one of the main scientists who collected this data, calculated the speed of light in air to be 299853 plus or minus 60 km/s. This can be found here.

His error was 60, whereas mine is only slightly over 20 when using 2000 repetitions. Perhaps if Michealson had had the computers of today or used the bootstrap method through extremely tedious calculations, he too would have gotten a more precise answer to the true speed of light in air. Admittedly, all of this data is still based on the vaues that Michealson discovered and the assumption that the data he collected was normally distributed.

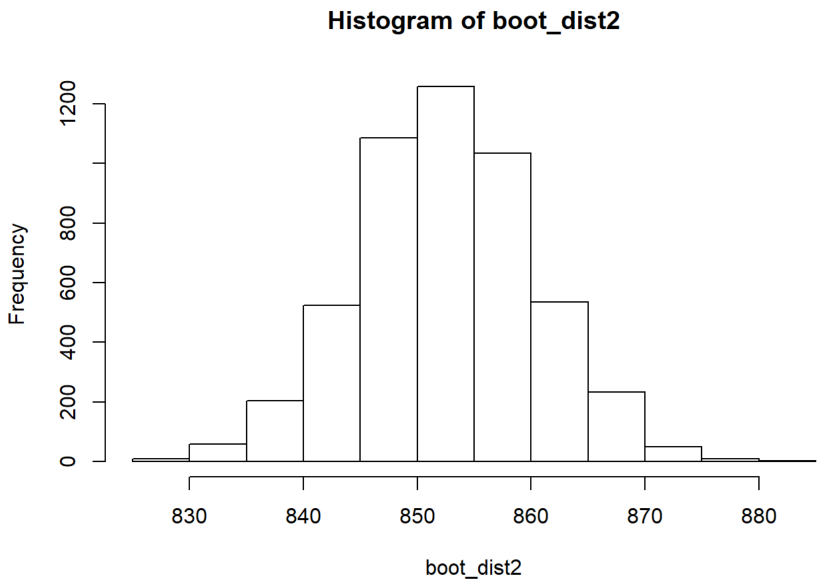
CAUTION: The bootstrap method finds the distribution of the average speed of light for many experiements. For this test I am making the assumption that the average of each simulated experiement will be part of a normal distribution centered on the true speed of light in air.

So, the results from the 2000 repetitions are clearly more precise than the normal data, but lets see what happens when we increase the number of repetitions

```
dist_means = function(vec) {
    new_mean = 0
    means_dist = c()
    for (i in 1:5000) { # the code is the same except I changed the repetitions to 5000
        new_mean = resample_mean(vec)
        means_dist = append(means_dist, new_mean)
    }
    return(means_dist)
}

set.seed(7) # I use the set.seed() function here so that my results are reproducible
boot_dist2 = dist_means(mor1)

# histogram of the second bootstrapped data
hist(boot_dist2)
```



```
# standard deviation of the new data
sdboot_dist2 = sd(boot_dist2)

# lower bound for 99.7% confidence interval
lowb = mean(boot_dist2) - (3*sdboot_dist2)

# upper bound for 99.7% confidence interval
upb = mean(boot_dist2) + (3*sdboot_dist2)

# 99.7% confidence interval
paste('The true speed of light in air, with 99.7% confidence, is between', (lowb+299000), 'km/s', 'and', (upb+2990
00), 'km/s')
```

```
## [1] "The true speed of light in air, with 99.7% confidence, is between 299829.310914805 km/s and 299875.8397651
95 km/s"
```

```
# or make a value with plus or minus for errors
error = mean(boot_dist2)-lowb
paste('The true speed of light in air is, with 99.7% confidence,', mean(boot_dist2)+299000, '±', error, 'km/s')
```

```
## [1] "The true speed of light in air is, with 99.7% confidence, 299852.57534 ± 23.2644251950383 km/s"
```

This shows the limitations of the bootstrap method. The error is about the same, despite more than doubling the number of repetitions. This means that my calulated value for the average speed of light in air can never be much better than plus or minus 20 km/s just by increasing the number of repetitions of the bootstrap. The only way that I could increase the precision and accuracy of the average velocity is if I had more data in the original sample.

Conclusion

I hope that this post helps explain the bootstrap method and the hidden power it has in the world of data science. Although it is essentaily only possible with the help of computers, it is a invluable tool able to make the precision of a prediction much better than with a normal set of data.

Refereneces

> https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/morley.html

> https://en.wikipedia.org/wiki/Albert_A._Michelson

> http://rmarkdown.rstudio.com/authoring_basics.html

> https://www.inferentialthinking.com/chapters/11/2/bootstrap.html

> https://www.statmethods.net/management/userfunctions.html

> http://rfunction.com/archives/62

> https://en.wikipedia.org/wiki/Bootstrapping_(statistics)