# Resampling-based inference using the `mosaic` package

Daniel Kaplan* Macalester College St. Paul, MN

Nicholas J. Horton† Smith College Northampton, MA

Randall Pruim‡ Calvin College Grand Rapids, MI

March 1, 2012

## Contents



## 1 Introduction

This document is intended to describe relatively straightforward ways to undertake a variety of resampling-based inferences through use of the `mosaic` package within R.

One of the goals of the `mosaic` package is to provide elementary commands that can be easily strung together by novices without having to master the esoteric aspects of programming.

The `mosaic` operations allow students to implement each of the operations in what George Cobb calls the "3 Rs" of statistical inference: Randomization, Replication, and Rejection XX **?**. By putting the 3 Rs together in various ways, students learn to generalize and internalize the logic of inference, rather than just following formulaic methods.

*dtkaplan@gmail.com
†nhorton@smith.edu
‡rpruim@calvin.edu

There's an interesting discussion of the role of simulation in **?**, where he notes the changing role of simulation. It used to be:

> something that people did when they can't do the math. ... It now seems that we are heading into an era when all statistical analysis can be done by simulation.

Arguably, the most important operation in statistics is sampling: ideally, selecting a random subset from a population. Regrettably, sampling takes work and time, so instructors tend to de-emphasize the actual practice of sampling in favor of theoretical descriptions. What's more, the algebraic notation in which much of conventional textbook statistics is written does not offer an obvious notation for sampling.

With the computer, however, these efficiency and notation obstacles can be overcome. Sampling can be placed in its rightfully central place among the statistical concepts in our courses.

Resampling-based inference using permutation testing and bootstrapping are an increasingly important set of techniques for introductory statistics and beyond.

Bootstrapping and permutation testing are powerful and elegant approaches to estimation of sample statistics and testing, respectively that can be implemented even in many situations where asymptotic results are difficult to find or otherwise unsatisfactory **??**. Bootstrapping involves sampling *with* replacement from a population, repeatedly calculating a sample statistic of interest to empirically construct the sampling distribution. Permutation testing for a 2 sample comparison is doing by *permuting* the labels for the grouping variable, then calculating the sample statistic (e.g. difference between two groups using these new labels) to empirically construct the null distribution.

## 2 R and RStudio

R is an open-source statistical environment that has been used at a number of institutions to teach introductory statistics. Among other advantages, R makes it easy to demonstrate the concepts of statistical inference through randomization while providing a sensible path for beginners to progress to advanced and professional statistics. RStudio (http://www.rstudio.org) is an open-source integrated development environment for R which facilitates use of the system.

## 3 Setup

The mosaic package is available over the Internet and can be installed into R using the standard features of the system (this needs only be done once).

```
> install.packages("mosaic")
```

Once installed, the package must be loaded so that it is available (this must be done within each R session).

```
> require(mosaic)
> options(digits=3)
> numsim = 100
```

This command would typically be provided in a set-up file for students so that it's executed automatically each time an R session is started.

XX need to find a better home for these datasets. Can we add them to the package? XX

The instructions for the breakout session did not give any details about how to access the data. So that you may follow these examples from any session of R, here are the commands to read the files from a public web site.

```
> mustangs = read.csv("http://www.mosaic-web.org/MustangPrice.csv")
> sleep = read.csv("http://www.mosaic-web.org/SleepCaffeine.csv")
```

```
> mustangs = read.csv("MustangPrice.csv")
> sleep = read.csv("SleepCaffeine.csv")
```

# 4 Problem 1: Used Mustangs (bootstrapping a mean)

*A student collected data on the selling prices for a sample of used Mustang cars being offered for sale at an internet website. The price (in $1,000's), age (in years) and miles driven (in 1,000's) for the 25 cars in the sample are given in the [file* `MustangPrice.csv`*]. Use these data to construct a 90% confidence interval for the mean price (in $1,000's) for used Mustangs.*

The mean price can be calculated as

```
> with(mustangs, mean(Price))
```

```
[1] 16
```

Even though a single trial is of little use, it's a nice idea to have students do the calculation to show that they are getting a different (usually!) result than without resampling. One resampling trial can be carried out with

```
> with(resample(mustangs), mean(Price))
```

```
[1] 16.1
```

Another trial can be carried out with the command:

```
> with(resample(mustangs), mean(Price))
```

```
[1] 15.4
```

Let's generate five more:

```
> trials = do(5) * with(resample(mustangs), mean(Price))
> trials
```

```
  result
1   14.2
2   14.3
3   13.2
4   18.0
5   16.3
```

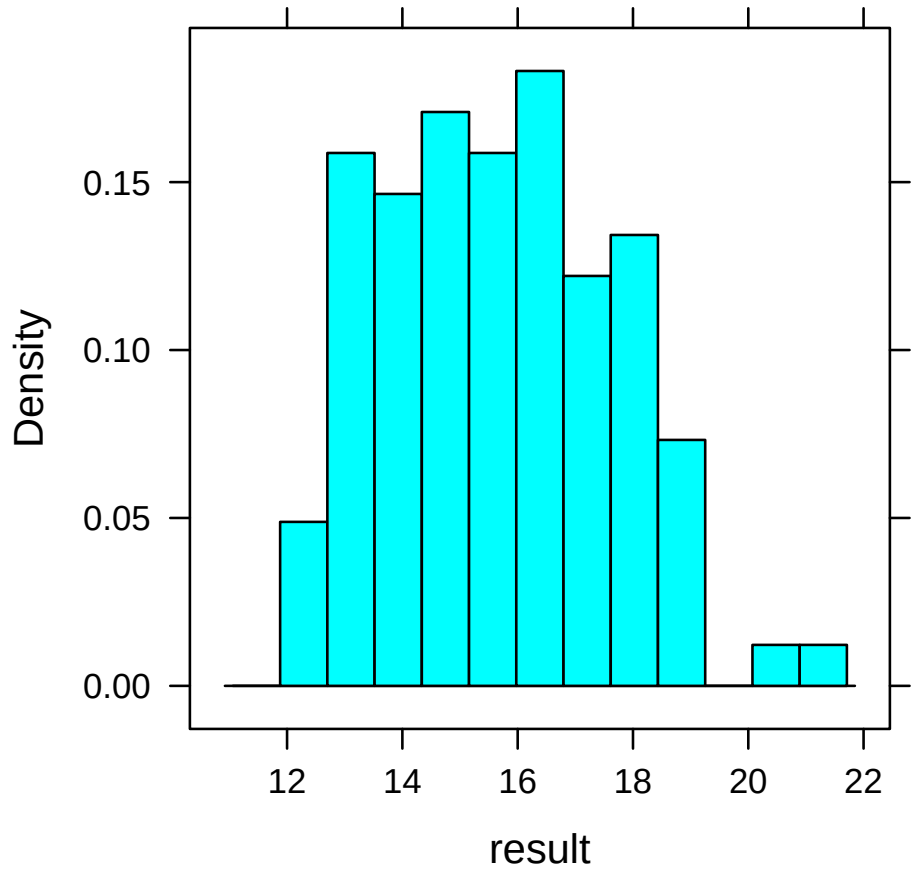
Now conduct 100 resampling trials, replacing the results in an object called `trials`:

```
> numsim
[1] 100
> trials = do(numsim) * with(resample(mustangs), mean(Price))
```

This creates a new set of data with the result from each of the $numsim = 100$ trials.

Plots of distributions are straightforward, e.g.:

```
> xhistogram(~ result, data=trials)
```




Calculation of the 90% confidence interval can be done directly.

```
> qdata(c(.05, .95), trials$result)
  5%  95%
12.9 18.7
```

Alternatively, the standard error from this distribution can be used to estimate the 90% margin of error. First calculate the t (or z) multiplier for the appropriate degrees of freedom:

```
> tstar = qt(.95, df=24)
> zstar = qnorm(.95)
```

The resulting margin of error will be

```
> tstar * sd(trials$result)

[1] 3.37

> zstar * sd(trials$result)

[1] 3.24
```

# 5 Problem 2: NFL Overtimes (single test of proportion)

*The National Football League (NFL) uses an overtime period to determine a winner for games that are tied at the end of regulation time. The first team to score in the overtime wins the game and a coin flip is used to determine which team gets the ball first. Is there an advantage to winning the coin flip? Data from the 1974 through 2009 seasons show that the coin flip winner won 240 of the 428 games where a winner was determined in overtime. Treat these as a sample of NFL games to test whether there is sufficient evidence to show that the proportion of overtime games won by the coin flip winner is more than one half.*

If the coin-flip result were unrelated to the outcome of the game, the observed 240 game wins out of 428 events would itself be a plausible outcome of a coin flip.

**Style 1** Using the built-in binomial distribution operators.

Generate a simulation where each trial is a random sample of 428 games from a world in which the null hypothesis holds true.

```
> proptable(rbinom(100000, prob=0.5, size=428) >= 240)

  FALSE    TRUE
0.99336 0.00664
```

It's very unlikely, if the null were true, that the coin-flip winner would win 240 or more times.

Of course, such a calculation can be done directly, but that raises issues such as which tail `pbinom` is calculating (R always does the left tail) and adjusting the cut-off appropriately

```
> pbinom(239, prob=0.5, size=428)

[1] 0.993
```

**Style 2** Explicitly simulating a coin flip.

Recognizing that coin flips are a staple of statistics courses, the `mosaic` package offers a random flip operator that does the tabulation for you. Here is one trial involving flipping 428 coins:

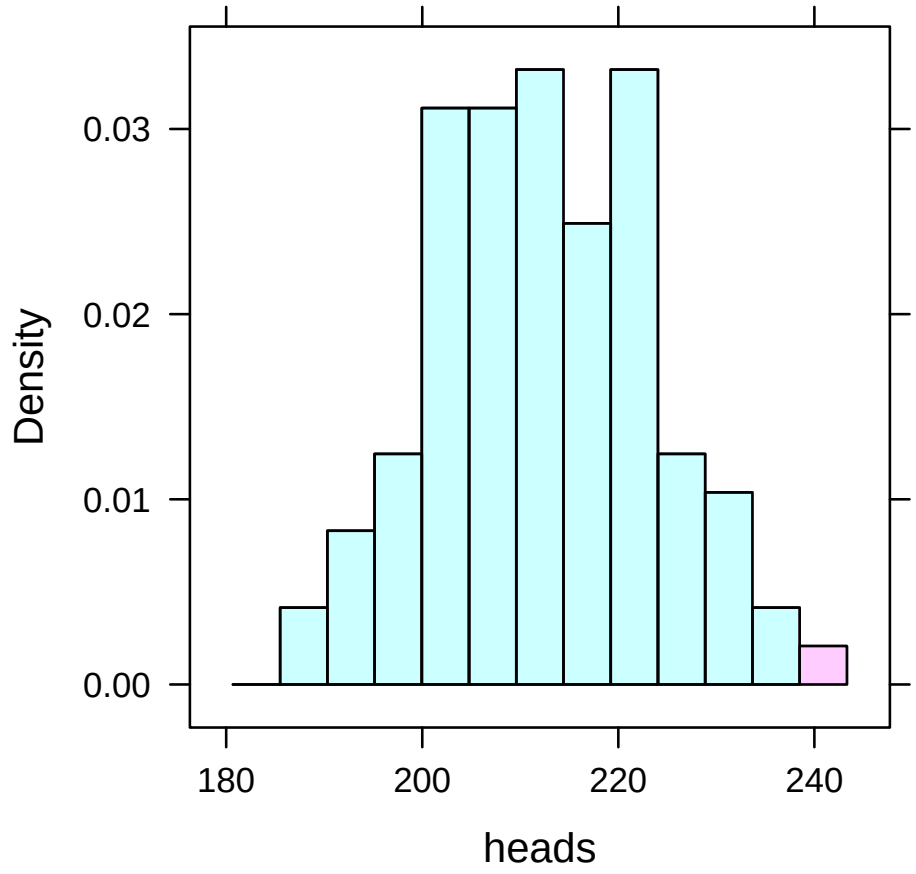
```
> do(1) * rflip(428)
```

```
    n heads tails
1 428   231   197
```

We'll do 100 trials, and count in what fraction of the trials the winner (say, "heads") wins 240 or more

```
> trials = do(numsim) * rflip(428)
> xhistogram(~ heads, groups = heads >= 240, data=trials)
> proptable(trials$heads >= 240)
```

```
FALSE  TRUE
 0.99  0.01
```




The observed pattern of 240 wins is not a likely outcome under the null hypothesis.

## 6 Problem 3: Sleep vs. Caffeine (2 sample permutation test of means)

*In an experiment on memory, students were given lists of 24 words to memorize. After hearing the words they were assigned at random to different groups. One group of 12 students took a nap for*

*1.5 hours while a second group of 12 students stayed awake and was given a caffeine pill. The table below shows the number of words each participant was able to recall after the break. Test whether the data indicate a difference in mean number of words recalled between the two treatments.*

The Sleep group seems to have remember somewhat more words:

```
> mean(Words ~ Group, data=sleep)

     Group    S  N Missing
1 Caffeine 12.2 12       0
2    Sleep 15.2 12       0
```

```
> obs = compareMean(Words ~ Group, data=sleep); obs

[1] 3
```

To implement the null hypothesis, scramble the Group with respect to the outcome, Words:

```
> compareMean(Words ~ shuffle(Group), data=sleep)

[1] -0.167
```

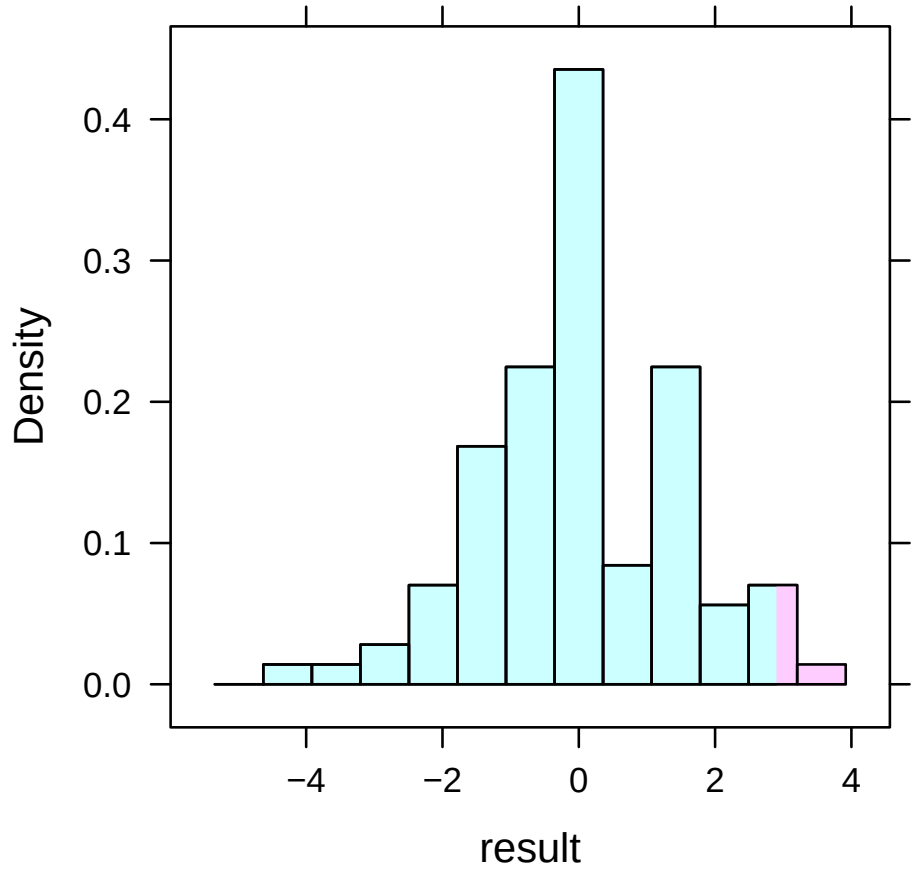
That's just one trial. Let's try again:

```
> compareMean(Words ~ shuffle(Group), data=sleep)

[1] -0.667
```

To get the distribution under the null hypothesis, do many trials.

```
> trials = do(numsim) * compareMean(Words ~ shuffle(Group), data=sleep)
> xhistogram(~ result, groups = result > obs, data=trials)
```




## 7 Bonus Problem: test of equality of proportions

We can undertake a test of difference in two proportions, in this case, the proportion homeless by gender in the HELP randomized clinical trial.

```
> with(HELPrct, table(homeless, sex))
```

```
          sex
homeless   female male
  homeless     40  169
  housed       67  177
```

```
> mean(homeless=="housed" ~ sex, data=HELPrct)
```

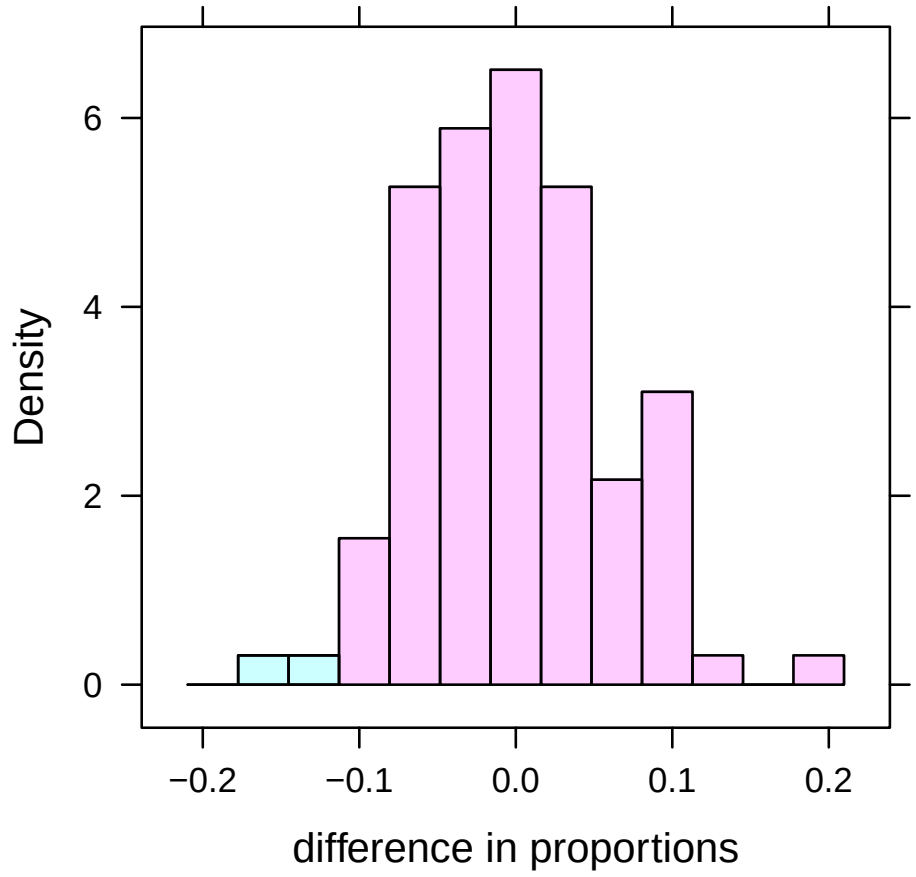
```
     sex     S   N Missing
1 female 0.626 107       0
2   male 0.512 346       0
```

```
> obs = compareProportion(homeless=="housed"~ sex, data=HELPrct); obs  # observed value
```

```
[1] -0.115
```

We will use the same general approach to empirically calculate the null distribution by permuting the labels for the grouping variable.

```
> # compute permutation distribution
> permute = do(numsim)*compareProportion(homeless=="housed" ~ shuffle(sex), data=HELPrct)
> xhistogram(~ result, groups=result>=obs, permute, xlab="difference in proportions")
> ladd(panel.abline(v=obs, lwd=2))
```




## 8 Bonus Problem (bootstrapping a correlation)

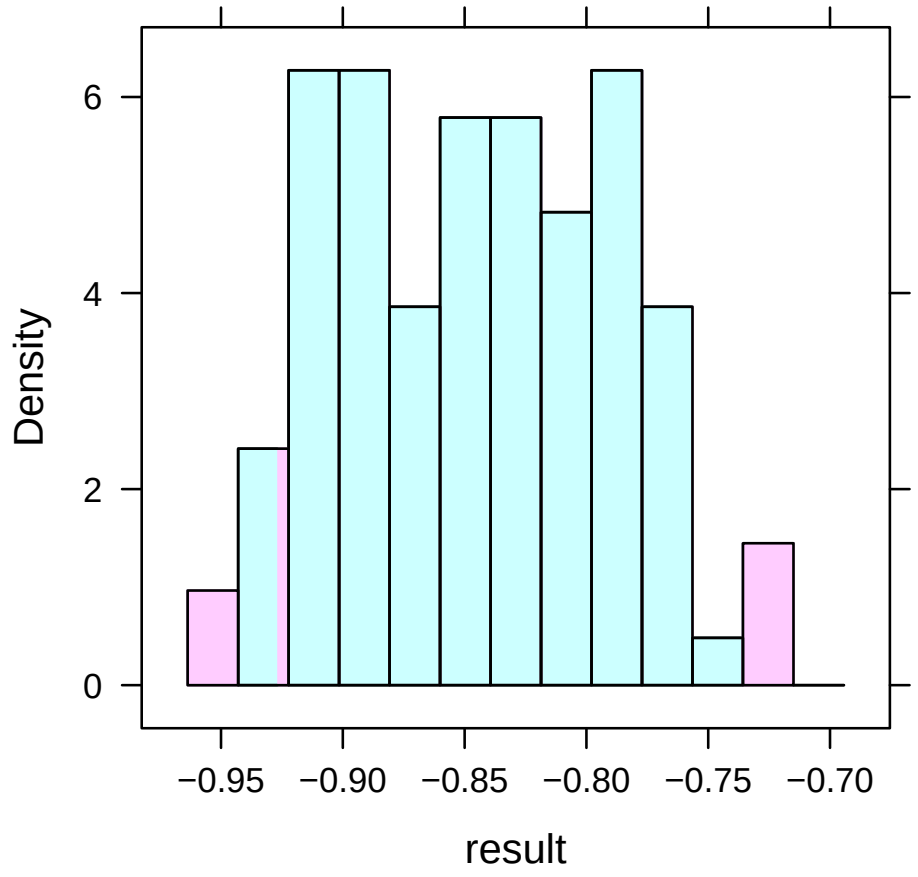
*The data on Mustang prices in Problem #1 also contains the number of miles each car had been driven (in thousands). Find a 95% confidence interval for the correlation between price and mileage.*

```
> with(mustangs, cor(Price, Miles))
[1] -0.825
```

```
> trials = do(numsim) * with(resample(mustangs), cor(Price, Miles))
> quantiles = qdata(c(.025, .975), trials$result); quantiles

  2.5%  97.5%
-0.940 -0.737

> xhistogram(~ result,
      groups = ((result <= quantiles[1]) | (result >= quantiles[2])),
      nbin=30, data=trials)
```



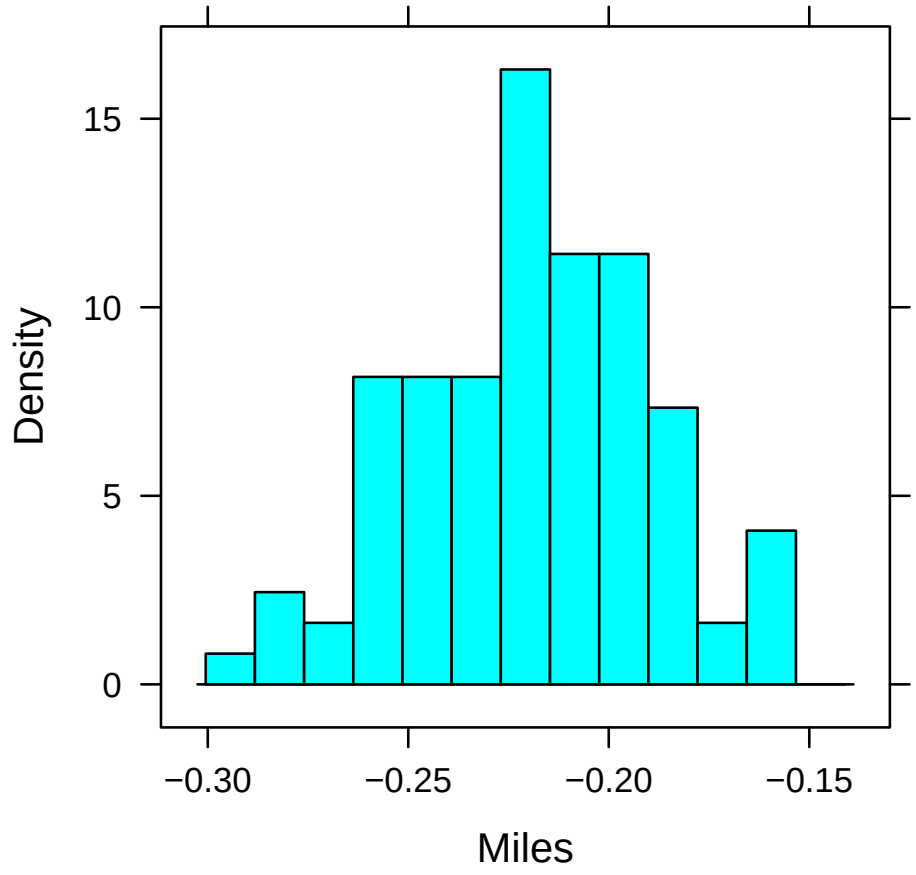

But there's no reason to restrict oneself to the correlation: we can also fit the linear model and consider the coefficients themselves:

```
> do(1) * lm(Price ~ Miles, data=mustangs)

  Intercept  Miles sigma r-squared
1      30.5 -0.219  6.42      0.68

> trials = do(numsim) * lm(Price ~ Miles, data=resample(mustangs))
> xhistogram(~ Miles, data=trials)
> sd(trials) #standard errors
```

```
Intercept      Miles      sigma r-squared
   3.0879     0.0303     1.6193     0.0911
```




The predicted average price goes down by $22 \pm 6$ cents per mile driven.

**Using Simulations in Other Ways** The basic technology of resampling and shuffling can be used to demonstrate many other concepts in statistics than the generation of confidence intervals and p-values. For example, it is very useful for showing the origins of distributions such as t and F. Similarly, it can be helpful to show students the distribution of p-values under the null hypothesis — students are surprised to see that it's uniform. Seeing this helps them to understand the sense in which the "significance level" refers to a false rejection of the null in a world in which the null is true.

For additional examples of the use of simulations in introductory statistics using R, see

- R. Pruim, N. Horton, & D. Kaplan, *Teaching Statistics with R*, http://mosaic-web.org/uscots2011/WorkshopNotes-001.pdf
- D. Kaplan, *Statistical Modeling: A Fresh Approach*, http://www.macalester.edu/~kaplan/ISM

Project MOSAIC, www.mosaic-web.org