ISLR | Chapter 4 Exercises

Marshall McQuillen

6/29/2018

Conceptual

1

$$f(\alpha) = Var(\alpha X + (1 - \alpha)Y)$$

Using the statistical property that $Var(X + Y) = Var(X) + Var(Y) + 2Cov(X, Y)$, the above equation can be rewritten as:

$$f(\alpha) = Var(\alpha X) + Var((1 - \alpha)Y) + 2Cov(\alpha X, (1 - \alpha)Y)$$

Then, using the statistical property that $Var(cX) = c^2Var(X)$ and $Cov(aX, bY) = abCov(X, Y)$, the equation can once again be rewritten as:

$$f(\alpha) = \alpha^2Var(X) + (1 - \alpha)^2Var(Y) + 2\alpha(1 - \alpha)Cov(X, Y)$$

Multiplying the $\alpha(1 - \alpha)$ comes out to:

$$f(\alpha) = \alpha^2Var(X) + (1 - \alpha)^2Var(Y) + 2(\alpha - \alpha^2)Cov(X, Y)$$

By then taking the partial derivative of $f(\alpha)$ with respect to α, the slope of the function at a given alpha can be obtained:

$$\frac{\partial f(\alpha)}{\partial \alpha} = 2\alpha\sigma_X^2 + 2(1 - \alpha)(-1)\sigma_Y^2 + 2(1 - 2\alpha)\sigma_{XY}$$

Divide by 2:

$$\frac{\partial f(\alpha)}{\partial \alpha} = \alpha\sigma_X^2 + (-1 + \alpha)\sigma_Y^2 + (1 - 2\alpha)\sigma_{XY}$$

Expand the second and third terms in the equation:

$$\frac{\partial f(\alpha)}{\partial \alpha} = \alpha\sigma_X^2 + -\sigma_Y^2 + \alpha\sigma_Y^2 + \sigma_{XY} - 2\alpha\sigma_{XY}$$

Factor α out of all possible terms:

$$\frac{\partial f(\alpha)}{\partial \alpha} = \alpha(\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}) - \sigma_Y^2 + \sigma_{XY}$$

Divide each term by $(\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY})$:

$$\frac{\partial f(\alpha)}{\partial \alpha} = \alpha - \frac{\sigma_Y^2 + \sigma_{XY}}{(\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY})}$$

Since the goal is to minimize the equation, setting the partial derivative to zero will return an equation that is a minimum.

$$0 = \alpha - \frac{\sigma_Y^2 + \sigma_{XY}}{(\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY})}$$

Subtract α

$$-\alpha = - \frac{\sigma_Y^2 + \sigma_{XY}}{(\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY})}$$

Multiply by -1:

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}}$$

2

- **A**. Since a bootstrapped sample contains N observations of the original sample of the population, each sample being chosen at random with replacement, the probability that the first observation in a bootstrapped sample is *not* the jth observation is $\frac{n - 1}{n}$.
- **B**. The probability that the second bootstrap observation is *not* the jth observation is $\left(\frac{n - 1}{n}\right)^2$.
- **C**. Since a boostrapped sample contains N observations, the probability that the jth observation (x_j) is *not* in the bootstapped sample (S_b) is:

  $$P(x_j \ not \ in \ S_b) = \left(\frac{n - 1}{n}\right)^n$$

  Which can be simplified to:

  $$P(x_j \ not \ in \ S_b) = \left(1 - \frac{1}{n}\right)^n$$

- **D**. Since the probability that the jth observation is *not* in the boostrap sample is $\left(1 - \frac{1}{n}\right)^n$, the probability that the jth observation *is* in the bootstrap sample would be the complement, $1 - \left(1 - \frac{1}{n}\right)^n$. When $n = 5$, this comes out to $1 - \left(1 - \frac{1}{5}\right)^5 = 0.67232 = 67.23\%$
- **E**.

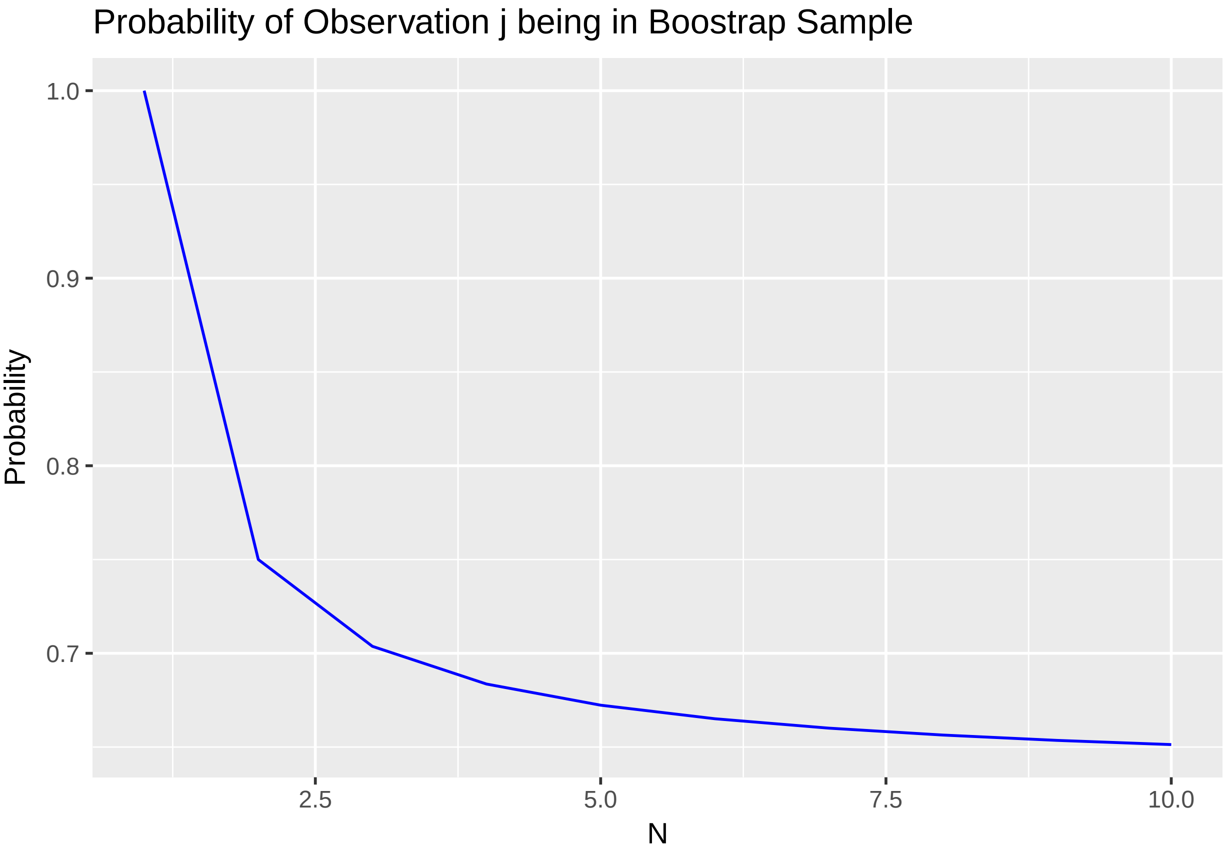
  $$1 - \left(1 - \frac{1}{100}\right)^{100} = 0.6339677 = 63.40\%$$

- **F**.

  $$1 - \left(1 - \frac{1}{100}\right)^{100} = 0.632139 = 63.21\%$$

- **G**. It is clear that as N increases the probability that the jth observation is in the bootstrap sample asymptotically approaches 0.632. The below plot illustrates this phenomenon (only displaying 1 to 10 for illustration purposes)

```
library(ggplot2)
x <- 1:100000
y <- 1 - (1 -(1/x))^x
df <- data.frame(x, y)
display_df <- df[1:10,]
ggplot(display_df, aes(x = x, y = y)) +
    geom_line(color = 'blue') +
    labs(x = "N", y = "Probability",
         title = "Probability of Observation j being in Boostrap Sample")
```



- **H**. The below code is showing mathematically what the plot above shows; that the limit of the function $1 - \left(1 - \frac{1}{x}\right)^x$ as x approaches infinity is 0.632.

```
store <- rep(NA, 10000)
for (i in 1:10000) {
    store[i] <- sum(sample(1:100, replace = TRUE)==4) > 0
}
mean(store)
```

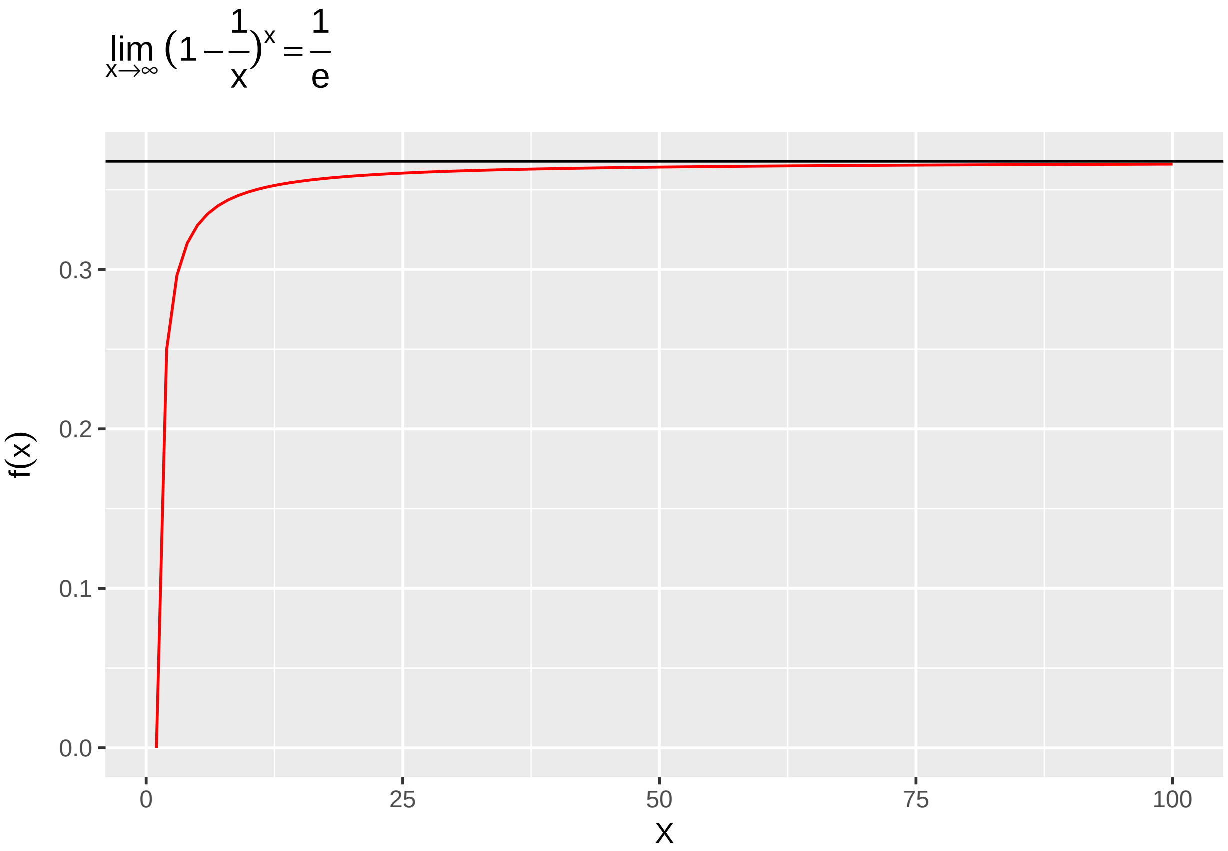
```
## [1] 0.6301
```

This can be written as:

$$\lim_{x\to\infty} \left(1 - \left(1 - \frac{1}{x}\right)^x\right) = 0.632$$

However, the inner part of that equation, $\lim_{x\to\infty} \left(1 - \frac{1}{x}\right)^x$, simplifies to $\frac{1}{\epsilon}$, proven by plot below:

```r
x <- 1:100
y <- (1 - (1/x))^x
asymptote <- rep(1/exp(1), 100)
df <- data.frame(x, y, asymptote)
ggplot(df, aes(x = x, y = y)) +
    geom_line(color = 'red') +
    geom_hline(aes(yintercept = asymptote)) +
    labs(x = "X", y = expression(f(x))) +
    ggtitle(expression(lim((1 - over(1, "x"))^"x", x %->% infinity) == frac(1, e)))
```



Therefore:

$$\lim_{x\to\infty}\left(1-\left(1-\frac{1}{x}\right)^x\right) = 0.632 = 1 - \frac{1}{\epsilon}$$

3

- **A**. K-Fold cross validation is the process of randomly dividing the entire data set into K separate subsets. A statistical model can then be trained on $K-1$ of those subsets, and the final Kth subset is used to test the model on unseen data, returning an estimate of the test error. This is performed K times, each time using a different subset as to estimate the test error. This results in K separate estimates of the testing error, which can be averaged to get the cross validated error estimate.
- **B**.

  i. There are a couple advantages of K-Fold CV over the validation set approach. First, K-Fold CV will return more than one estimate of the testing error, allowing insight into the variance of the testing error. In addition to this, since the number of observations in the training data set using the validation set approach is less than the number of observations used in the training data set in K-Fold CV, the validation set approach will typically overestimate the testing error. This is due to the fact that a model has a better chance of modeling the true relationship within the data the more observations too which it has access.

ii. LOOCV also has a couple downsides relative to K-Fold CV. First and foremost, since a total of N models are fit to the data, there is a large increase in computation time over K-Fold CV when K is equal to the usual 5 or 10. In addition to this, since there are N total models and each of the N models consists of $N - 1$ observations, *each of the N models will be trained on nearly identical data.* This leads to the CV error estimates being highly correlated, which corresponds to high variance and low bias.