TOOL

Measuring the Uncertainty Around a Regression Line

You can fit a regression line to your data to build a linear prediction rule. However, when you use linear regression, you should assess the uncertainty around the prediction rule by assessing the uncertainties around its correlation coefficient and slope. This accomplishes two key things: First, it allows you to test the hypothesis that the response value changes with the predictor variable; and second, if the response value does change with the predictor variable, analyzing the uncertainty around the slope will help you determine how much you should trust the predictions you make using the rule overall, as well as at individual points within the prediction rule.

Use this tool as a guide to measuring uncertainty around these values with bootstrap methods in R. The first step in this method is to use bootstrap methods to calculate the correlation coefficient, intercept, and slope of regression lines fitted to many different bootstrapped data sets. Then, you can use these values to calculate the bootstrap confidence intervals around each value and to create a confidence band around the prediction rule. This tool demonstrates how to do this using an example from the `Prestige` data set.

Using R With This Tool

The portions of this tool with a grey background are code text you can use to do the examples included in this tool. You can also modify them to use with your own data. In these examples:

- Commands are the lines of code that don't begin with a pound sign (#). Type these lines into R to carry out the command.
- Commented text begins with one pound sign and explains what the code does.
- Code output begins with two pound signs.

Data Set Information

The **Prestige** data set contains information about different job types in Canada in 1971 that you examined throughout this course. It is part of the **carData** package in R. In this data set, **education** refers to the average number of years of education the employees that hold that job have had, and **prestige** refers to the Pineo-Porter prestige value which measures the pride an employee has in their job.

You can load and view the data set with the following code:

```
install.packages("carData")                 # Install the carData package
library(carData)                            # Load the carData package
data(Prestige)                              # Load the Prestige data set
Prestige = Prestige[!is.na(Prestige$type),]  # Remove rows that do not
                              # contain information on the profession type
head(Prestige)                # View the first 6 rows of the data set

##                     education income women prestige census type
## gov.administrators      13.11  12351 11.16     68.8   1113 prof
## general.managers        12.26  25879  4.02     69.1   1130 prof
## accountants             12.77   9271 15.70     63.4   1171 prof
## purchasing.officers     11.42   8865  9.11     56.8   1175 prof
## chemists                14.62   8403 11.68     73.5   2111 prof
## physicists              15.64  11030  5.13     77.6   2113 prof
```

Measuring the Uncertainty Around the Correlation Coefficient and Slope

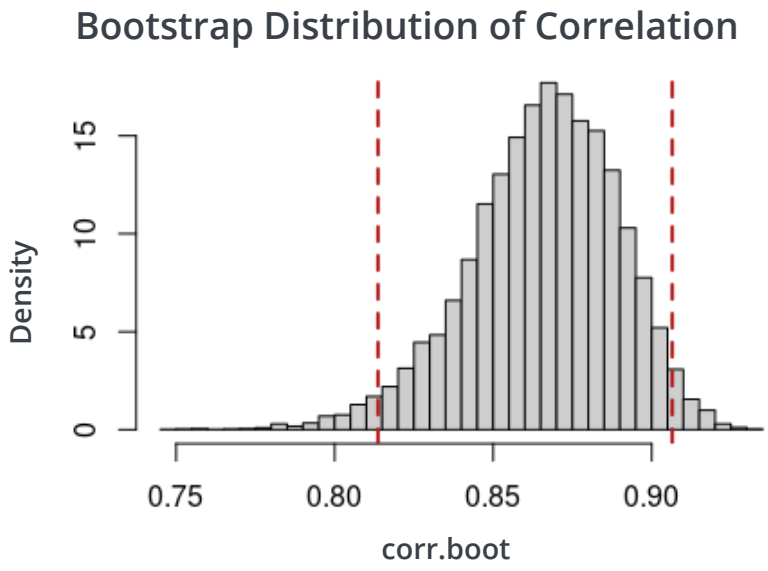
You can use bootstrap methods to assess uncertainty around the Pearson correlation and regression coefficients calculated on your sample. To do this:

1. Build the bootstrap distribution by generating a large number of samples (`n = 10000`), fit a linear prediction rule to each sample, and store the correlation coefficient, intercept, and slope from each sample in three vectors of length, `n`.

```
set.seed(1)            # Set seed for reproducibility

n = 10000              # Number of bootstrap samples to draw
corr.boot = rep(0, n)  # Vector to store n correlation coefficients
a.boot = rep(0,n)      # Vector to store n intercept values
b.boot = rep(0, n)     # Vector to store n slope values

for (b in 1:n){        # For loop to draw n bootstrap samples

    boot.id = sample(98, replace = TRUE)  # Random sample of 98 values
    Prestige.boot = Prestige[boot.id,]    # Create bootstrap sample from the
                                          # Prestige data
# Calculate and store correlation coefficient:
corr.boot[b] = cor(Prestige.boot$education, Prestige.boot$prestige)

# linear regression of prestige and education:
fit.boot <- lm(prestige ~ education, data = Prestige.boot)

# store intercept and slope:
a.boot[b] = fit.boot$coefficients[1]
b.boot[b] = fit.boot$coefficients[2]
}
```

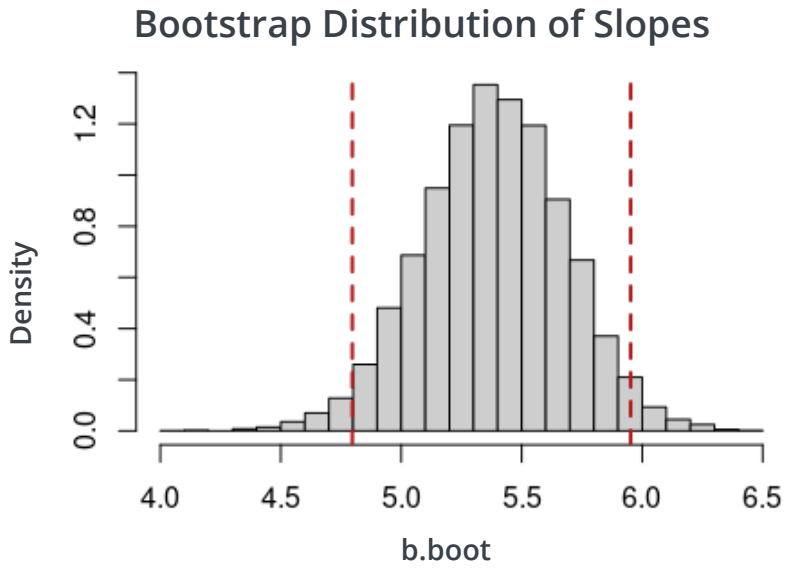
2. Use the bootstrap distribution of correlation coefficients to construct a 95% confidence interval (CI) to test if the correlation coefficient is 0. In the example below, 0 is not within the 95% CI, so you can be 95% confident that the correlation coefficient is not equal to 0. This means you can reject the null hypothesis that **education** has no association with **prestige**.

```
hist(corr.boot, breaks = 30,
   freq = FALSE, col = "lightgrey",
   main = "Bootstrap Distribution of
   Correlation")

abline(v = quantile(corr.boot,
   probs = c(0.025, 0.975)),
   lty = 2, col = "red", lwd = 2)
```



3. Use the bootstrap distribution of slopes to construct a 95% CI and test if the slope is 0. In the example below, 0 is not within the 95% CI, so you can reject the null hypothesis that **education** has no association with **prestige**.

```
hist(b.boot, breaks = 30,
   freq = FALSE, col = "lightgrey",
   main = "Bootstrap Distribution of
   Slopes")

abline(v = quantile(b.boot,
   probs = c(0.025, 0.975)),
   lty = 2, col = "red", lwd = 2)
```



You can similarly use the bootstrap distribution of the intercepts a.boot to construct a 95% CI for the baseline, i.e., the average value of the response variable when the predictor variable is 0. This is usually not of primary interest to data scientists, but is used to assess uncertainty around the overall prediction rule as described below.

4. Use bootstrap distributions of slope and intercept to quantify uncertainty around the prediction rule for different values of X. This is called a confidence band, and it can help you determine if there are any values in your data set that are too uncertain to predict effectively. Predictions of X values towards the center of the scatterplot are most precise, while predictions for X values around the left or right extreme of the scatterplot come with higher uncertainty.

In the prestige example below, you can see that the CI in the middle of the x-axis is narrower than the CI at either end of the x-axis, so you can be most certain about predictions you make about `education` values between ten and twelve.

```
x.all = seq(6, 18, by = 0.01)    # Create a vector of points across
                                 # the range of X values

ci.upper = rep(0, length(x.all)) # Create a vector to store the upper
                                 # confidence intervals

ci.lower = rep(0, length(x.all)) # Create a vector to store the lower
                                 # confidence intervals

# Use a for loop to calculate the 95% CI at each point in x.all:

for (i in 1:length(x.all)){
    x = x.all[i]
    prestige.pred.x = a.boot + x*b.boot
    ci.upper[i] = quantile(prestige.pred.x, probs = 0.975)
    ci.lower[i] = quantile(prestige.pred.x, probs = 0.025)
}

# Plot prestige vs education and the
# regression line:
plot(prestige ~ education, data = Prestige,
     pch = 19, col = "black",
     main = "Prestige vs. Education")
fit.prestige <- lm(prestige ~ education,
     data = Prestige)
abline(fit.prestige, col = "red", lwd = 3)

# Add the 95% CI intervals to the plot:
lines(x.all, ci.upper, lty = 2,
    col = "red", lwd = 2)
lines(x.all, ci.lower, lty = 2,
    col = "red", lwd = 2)
```

Prestige vs. Education