Measuring the Uncertainty of a Prediction Rule

Ken Wood

7/26/2024

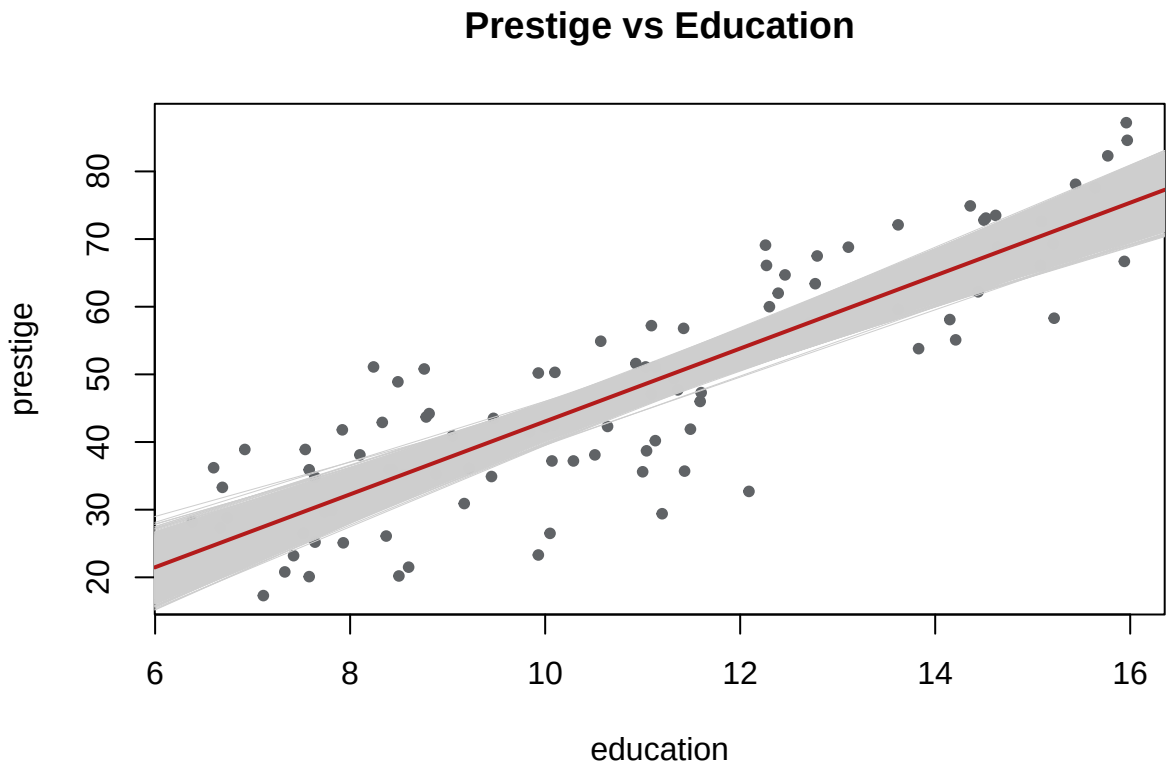
```
# The Prestige data set is available in the cars library
library(carData)
# Load the Prestige data set
data(Prestige)
# Exclude any observations that do not have an entry in the type column
Prestige = Prestige[!is.na(Prestige$type),]

# eCornell Hex Codes:
crimson = '#b31b1b'   #crimson
lightGray = '#cecece' #lightGray
darkGray = '#606366'  #darkGray
skyBlue = '#92b2c4'   #skyblue
gold = '#fbb040'      #gold
ecBlack = '#393f47'   #ecBlack
```

Step 1: Load the data and define colors.

Step 2: Generate many boot-strapped datasets. In the code below, you'll generate a bootstrapped data set 10,000 times and store the values of correlations, intercepts, and slopes for each data set.

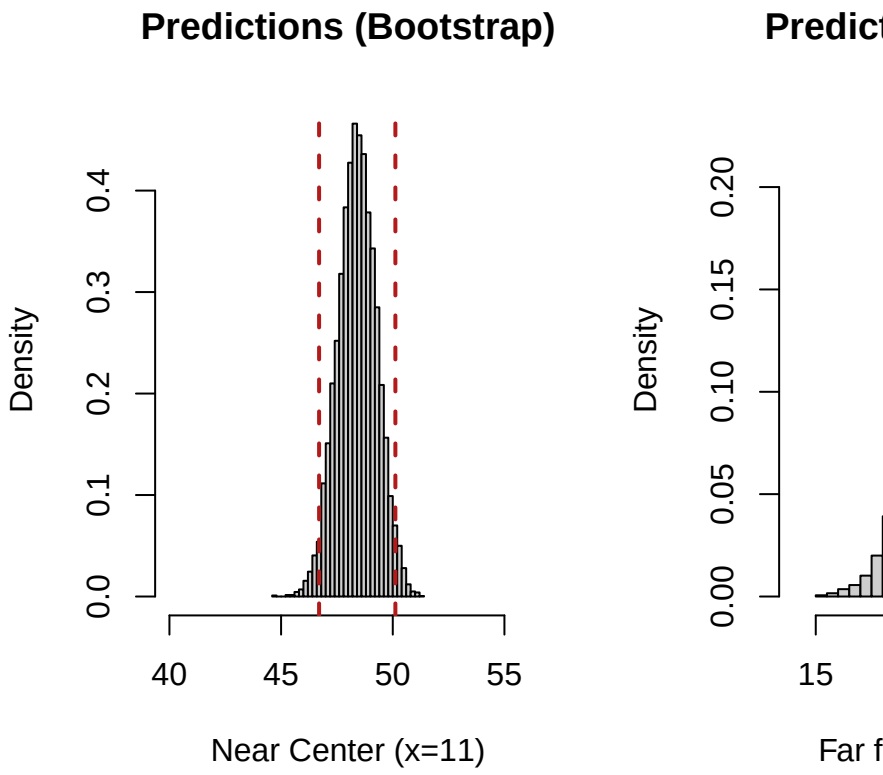
```
set.seed(1) # Set the seed for reproducibility
B = 10000   # Number of bootstrapped data sets
corr.boot = rep(0, B) # Vector to store correlation coefficients
a.boot = rep(0,B)     # Vector to store intercept values
b.boot = rep(0, B)    # Vector to store slope values

# Create plot of observed data:
plot(Prestige$education, Prestige$prestige,
     pch = 20, col = darkGray,
     xlab = 'education', ylab = 'prestige',
     main = 'Prestige vs Education')

# Use a for loop to generate B bootstrapped data sets:
for (b in 1:B){
  boot.id = sample(98, replace = TRUE)
  Prestige.boot = Prestige[boot.id,]

# Store coefficients on each bootstrapped sample:
  corr.boot[b] = cor(Prestige.boot$education,
                     Prestige.boot$prestige)
  fit.boot <- lm(prestige ~ education, data = Prestige.boot)
  a.boot[b] = fit.boot$coefficients[1]
```

```
  b.boot[b] = fit.boot$coefficients[2]

# Visualize each bootstrapped regression
# line on the plot in gray:
  abline(fit.boot, lwd = 0.5, col = lightGray)
}
# Add the regression line for the observed data in red
fit <- lm(prestige ~ education, data = Prestige)
abline(fit, col = crimson, lwd = 2)
```




```
par(mfrow = c(1,2))

# Fitted values near the center of data set
prestige.pred.11 = a.boot + 11*b.boot

hist(prestige.pred.11, breaks = 30, freq = FALSE,
     col = lightGray,
     xlim = c(40,56),
     main = 'Predictions (Bootstrap)',
     xlab = 'Near Center (x=11)')
abline(v = quantile(prestige.pred.11,
       probs = c(0.025, 0.975)),
       lty = 2, col = crimson, lwd = 2)

# Fitted values near the left end of data set
prestige.pred.6 = a.boot + 6*b.boot

hist(prestige.pred.6, breaks = 30, freq = FALSE,
```

```
     col = lightGray,
     xlim = c(14, 30),
     main = 'Predictions (Bootstrap)',
     xlab = 'Far from Center (x=6)')
abline(v = quantile(prestige.pred.6,
       probs = c(0.025, 0.975)),
       lty = 2, col = crimson, lwd = 2)
```



Step 3: Examine the bootstrapped values.

```
par(mfrow = c(1,1))
```

Step 4: Generate the 95% confidence band. Repeatedly calculating such 95% intervals for every value of X, we get a 95% confidence band around the regression line.

```
# Create a vector of points across the range of X values
x.all = seq(6, 18, by = 0.01)
# Create a vector to store the upper confidence intervals
ci.upper = rep(0, length(x.all))
# Create a vector to store the lower confidence intervals
ci.lower = rep(0, length(x.all))

# 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 add the regression line:
```

```r
plot(prestige ~ education, data = Prestige, pch = 19,
     col = ecBlack, main = "Prestige vs. Education")
fit.prestige <- lm(prestige ~ education, data = Prestige)
abline(fit.prestige, col = crimson, lwd = 3)

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