2. Sampling Residuals and Kable

Cheng Peng

Lab Note: STA321 Topics of Advanced Statistics

Contents

1 Introduction

In this note, we first introduce the bootstrap regression model through sampling residuals. The resulting bootstrap confidence intervals of regression coefficients will be included in the inference table of the output of **lm()**.

We will use some R functions to create nice-looking tables to be used in any professional reports.

2 Concept of Bootstrap Residuals

Bootstrapping residuals is another way to generate bootstrap random samples that are supposed to have the same distribution as that Y in the original random sample. The following flow chart explains the process of how to generate bootstrap random samples.
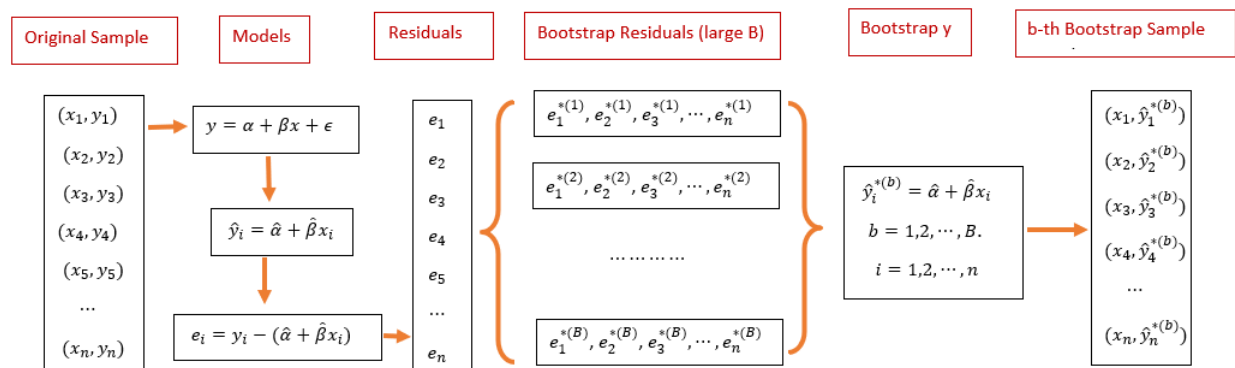
Next, we use a simple to demonstrate the steps to generate bootstrap samples based on sampling bootstrap residuals. The data set at https://raw.githubusercontent.com/pengdsci/sta321/main/ww03/handheight.txt has three variables: sex, height, and hand span. We will this data set to assess the linear correlation between

height and hand span. We first do some exploratory analysis to visualize the potential association between height and hand size.

```r
handheight = read.table("https://online.stat.psu.edu/stat501/sites/stat501/files/data/handheight.txt", I
MID = which(handheight$Sex=="Male")
MaleData = handheight[MID,]
FealeData = handheight[-MID,]
plot(handheight$Height, handheight$HandSpan, pch=16, col="white",
                        xlab = "Hand Span",
                        ylab = "Height",
                        main = "Hand Span vs Height",
                        col.main = "navy",
                        cex.main = 0.8,
                        bty="n")
points(handheight$Height[MID], handheight$HandSpan[MID], pch=16, col=alpha("darkred", 0.5))
points(handheight$Height[-MID], handheight$HandSpan[-MID], pch=19, col=alpha("blue", 0.5))
```
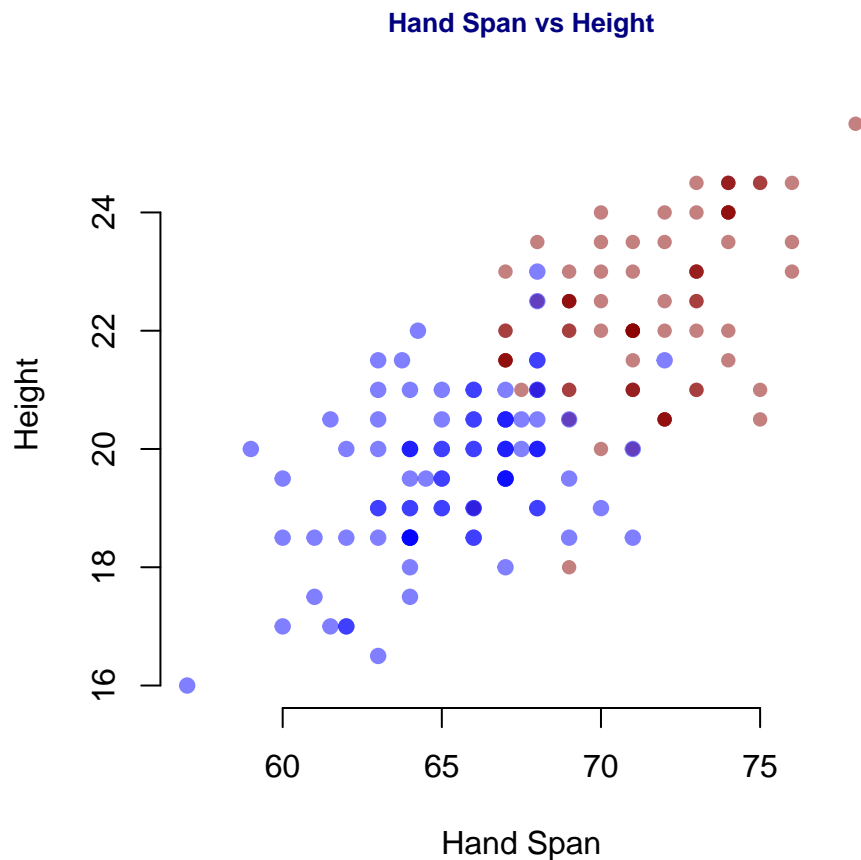

3 Implementing Bootstrapping Residuals

The following code reflects the steps in the above flow-chart

```r
height = handheight$Height
handspan = handheight$HandSpan
```

```
m0 = lm(height~handspan)
E = resid(m0)                    # Original residuals
a.hat = coef(m0)[1]
b.hat = coef(m0)[2]
##
B = 1000                         # generating 1000 bootstrap samples
bt.alpha = rep(0, B)
bt.beta = bt.alpha
for(i in 1:B){
  bt.e = sample(E, replace = TRUE)        # bootstrap residuals
  y.hat = a.hat + b.hat*handspan + bt.e   # bootstrap heights
  ## bootstrap SLR
  bt.m = lm(y.hat ~ handspan)
  bt.alpha[i] = coef(bt.m)[1]
  bt.beta[i] = coef(bt.m)[2]
 }
alpha.CI = quantile(bt.alpha, c(0.025, 0.975))
beta.CI = quantile(bt.beta, c(0.025, 0.975))
##
per.025 = c(alpha.CI[1],beta.CI[1])     # lower CI for alpha and beta
per.975 = c(alpha.CI[2],beta.CI[2])     # upper CI for alpha and beta
```

Next, we add the confidence limits to the output inferential table from the SLR based on the original sample.

```
lm.inference = as.data.frame((summary(m0))$coef)
lm.inference$per.025 = per.025
lm.inference$per.975 = per.975
kable(as.matrix(lm.inference))
```

	Estimate	Std. Error	t value	Pr(>|t|)	per.025	per.975
(Intercept)	35.52504	2.3159512	15.33929	0	30.968086	39.934765
handspan	1.56008	0.1105437	14.11278	0	1.353349	1.775964