Final Project: Implementation Bootstrap Regression

Namrata Ray

12/5/2019

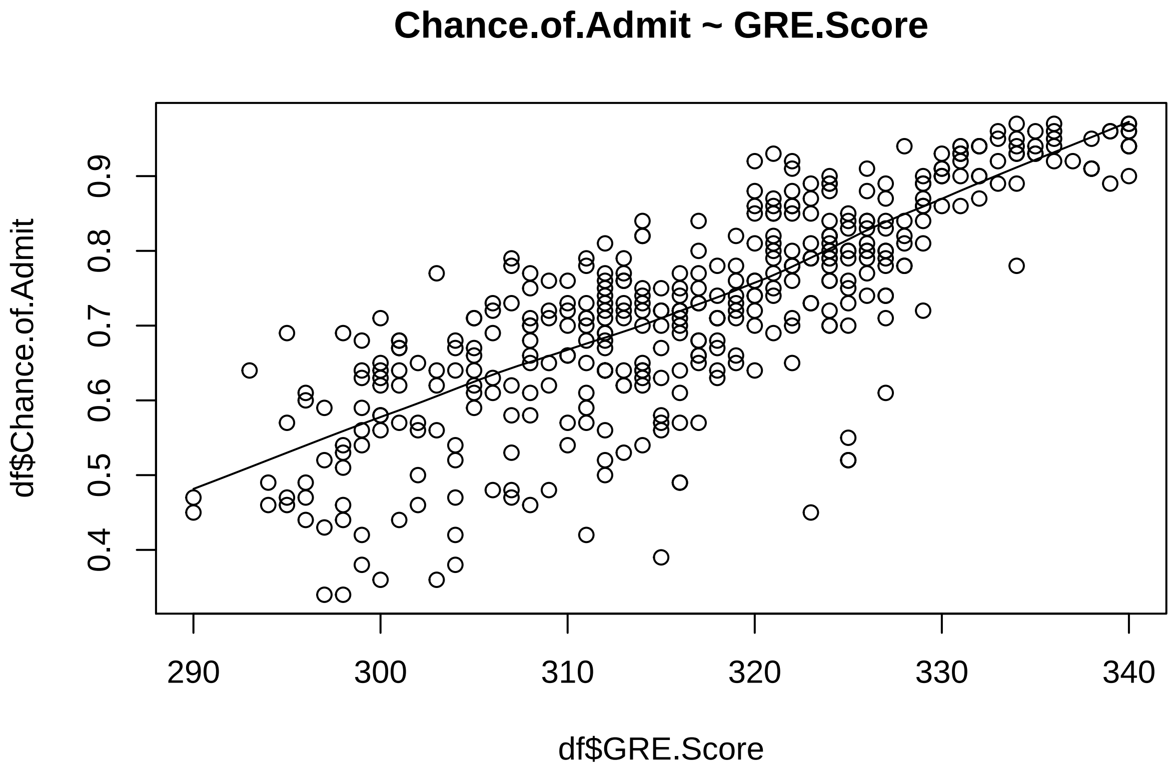
Bootstrap Regression

Data

```
setwd("C:/Users/rayna/Downloads")
df = read.csv("C:/Users/rayna/Downloads/Admission_Predict.csv")
head(df)
```

```
##   Serial.No. GRE.Score TOEFL.Score University.Rating SOP LOR CGPA Research
## 1          1       337         118                 4 4.5 4.5 9.65        1
## 2          2       324         107                 4 4.0 4.5 8.87        1
## 3          3       316         104                 3 3.0 3.5 8.00        1
## 4          4       322         110                 3 3.5 2.5 8.67        1
## 5          5       314         103                 2 2.0 3.0 8.21        0
## 6          6       330         115                 5 4.5 3.0 9.34        1
##   Chance.of.Admit
## 1            0.92
## 2            0.76
## 3            0.72
## 4            0.80
## 5            0.65
## 6            0.90
```

```
scatter.smooth(x=df$GRE.Score, y=df$Chance.of.Admit, main="Chance.of.Admit ~ GRE.Score")
```



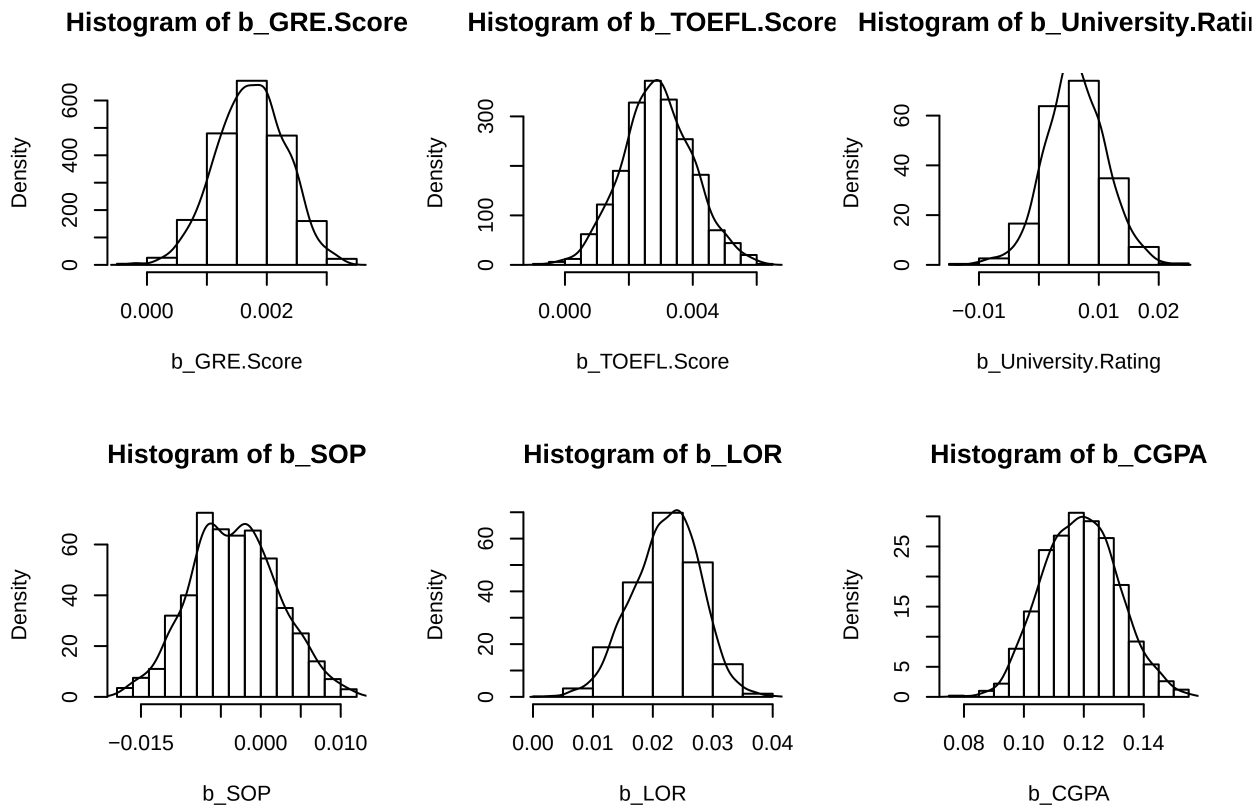
Sample regression

```
linearMod = lm(Chance.of.Admit ~ GRE.Score + TOEFL.Score + University.Rating
                + SOP + LOR + CGPA + Research, data=df)
betas_reg = linearMod$coefficients
se = sqrt(diag(vcov(linearMod)))
summary(linearMod)
```

```
##
## Call:
## lm(formula = Chance.of.Admit ~ GRE.Score + TOEFL.Score + University.Rating +
##     SOP + LOR + CGPA + Research, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.26259 -0.02103  0.01005  0.03628  0.15928
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -1.2594325  0.1247307 -10.097  < 2e-16 ***
## GRE.Score          0.0017374  0.0005979   2.906  0.00387 **
## TOEFL.Score        0.0029196  0.0010895   2.680  0.00768 **
## University.Rating  0.0057167  0.0047704   1.198  0.23150
## SOP               -0.0033052  0.0055616  -0.594  0.55267
## LOR                0.0223531  0.0055415   4.034  6.6e-05 ***
## CGPA               0.1189395  0.0122194   9.734  < 2e-16 ***
## Research           0.0245251  0.0079598   3.081  0.00221 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06378 on 392 degrees of freedom
## Multiple R-squared:  0.8035, Adjusted R-squared:    0.8
## F-statistic: 228.9 on 7 and 392 DF,  p-value: < 2.2e-16
```

Distribution of Beta Coefficient based on Std. Error

```r
par(mfrow=c(2,3))

b_GRE.Score = rnorm(1000,  betas_reg[2], se[2])
hist(b_GRE.Score, freq=FALSE); lines(density(b_GRE.Score))

b_TOEFL.Score = rnorm(1000,  betas_reg[3], se[3])
hist(b_TOEFL.Score, freq=FALSE); lines(density(b_TOEFL.Score))

b_University.Rating = rnorm(1000,  betas_reg[4], se[4])
hist(b_University.Rating, freq=FALSE); lines(density(b_University.Rating))

b_SOP = rnorm(1000,  betas_reg[5], se[5])
hist(b_SOP, freq=FALSE); lines(density(b_SOP))

b_LOR = rnorm(1000,  betas_reg[6], se[6])
hist(b_LOR, freq=FALSE); lines(density(b_LOR))

b_CGPA = rnorm(1000,  betas_reg[7], se[7])
hist(b_CGPA , freq=FALSE); lines(density(b_CGPA))
```



Bootstrap Regression

```r
n = length(df$Serial.No.)
b_GRE.Score = c()
b_TOEFL.Score = c()
b_University.Rating = c()
b_SOP = c()
b_LOR = c()
b_CGPA = c()


for(i in 1:1000){
  df_temp = df[sample(nrow(df), n, replace=TRUE), ]

  linearMod1 = lm(Chance.of.Admit ~ GRE.Score + TOEFL.Score + University.Rating
                  + SOP + LOR + CGPA + Research, data=df_temp)
  betas = linearMod1$coefficients
  b_GRE.Score = c(b_GRE.Score, unname(betas[2]))
  b_TOEFL.Score = c(b_TOEFL.Score, unname(betas[3]))
  b_University.Rating = c(b_University.Rating, unname(betas[4]))
  b_SOP = c(b_SOP, unname(betas[5]))
  b_LOR = c(b_LOR, unname(betas[6]))
  b_CGPA = c(b_CGPA, unname(betas[7]))
}
```

```
par(mfrow=c(2,3))
hist(b_GRE.Score, freq=FALSE); lines(density(b_GRE.Score))
hist(b_TOEFL.Score, freq=FALSE); lines(density(b_TOEFL.Score))
hist(b_University.Rating, freq=FALSE); lines(density(b_University.Rating))
hist(b_SOP, freq=FALSE); lines(density(b_SOP))
hist(b_LOR, freq=FALSE); lines(density(b_LOR))
hist(b_CGPA , freq=FALSE); lines(density(b_CGPA))
```

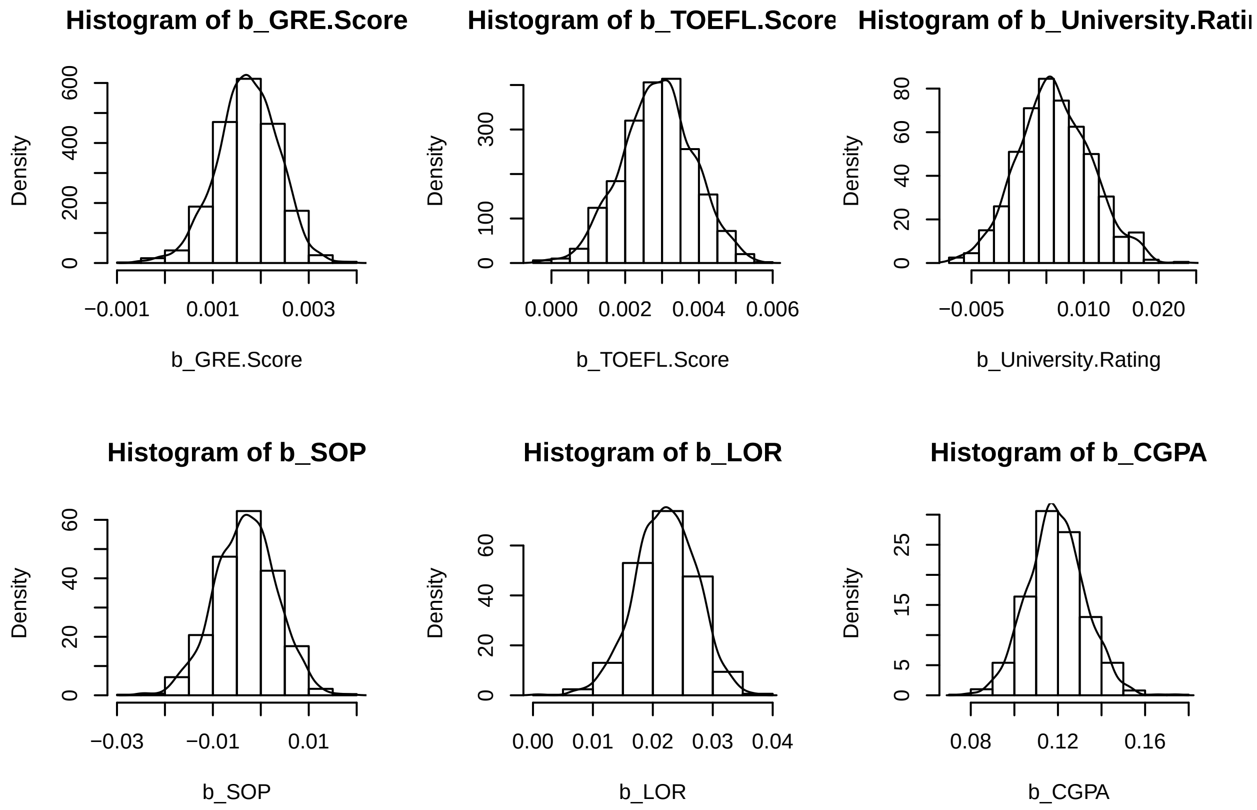


Table for Compariosn

```
library(knitr)
betas_bs_mean = c(mean(b_GRE.Score), mean(b_TOEFL.Score), mean(b_University.Rating),
                  mean(b_SOP), mean(b_LOR), mean(b_CGPA))

betas_bs_se = c(sqrt(var(b_GRE.Score)), sqrt(var(b_TOEFL.Score)), sqrt(var(b_University.Rating)),
                sqrt(var(b_SOP)), sqrt(var(b_LOR)), sqrt(var(b_CGPA)))

result = data.frame(variable_name=colnames(df)[2:7],
                    sample_b=round(unname(betas_reg[2:7]), 4),
                    sample_se=round(unname(se[2:7]), 7),
                    bootstrap_b_mean= round(betas_bs_mean, 4),
                    bootstrap_b_se= round(betas_bs_se, 7) )
kable(result)
```

variable_name	sample_b	sample_se	bootstrap_b_mean	bootstrap_b_se
GRE.Score	0.0017	0.0005979	0.0017	0.0006337
TOEFL.Score	0.0029	0.0010895	0.0029	0.0009647
University.Rating	0.0057	0.0047704	0.0061	0.0048448
SOP	-0.0033	0.0055616	-0.0032	0.0063356
LOR	0.0224	0.0055415	0.0221	0.0050532
CGPA	0.1189	0.0122194	0.1192	0.0129639

```
print('   ')
```

```
## [1] "   "
```