A2

Jefferson Li, Arib Shaikh

15/02/2021

Set Seed

```r
set.seed(1005057368)
```

Q1

a)

Note that

$$\hat{\beta}_1 \sim N(\beta_1, \sigma^2/Sxx)$$

We know $\beta_1 = 4, \sigma^2 = 25$.

We can calculate Sxx from the following code

```r
Xs = c(4, 8, 12, 16, 20)
Sxx = 0
for (x in Xs) {
  Sxx = Sxx + (x - mean(Xs))^2
}
Sxx
```

```
## [1] 160
```

Therefore

$$\hat{\beta}_1 \sim N(4, 25/160)$$

We want to calculate,

$$\begin{aligned}
P(|\hat{\beta}_1 - \beta_1| > 1) &= P(|\hat{\beta}_1 - 4| > 1) && (1)\\
&= P(\hat{\beta}_1 - 4 > 1) \quad or \quad P(\hat{\beta}_1 - 4 < -1) && (2)\\
&= P(\hat{\beta}_1 > 5) + P(\hat{\beta}_1 < 3) \qquad \text{Since they are disjoint} && (3)
\end{aligned}$$

This probability can be calculated in r with the following code

```r
prob = pnorm(3, 4, sqrt(25/Sxx)) + (1-pnorm(5, 4, sqrt(25/Sxx)))
```

Therefore,

$$P(|\hat{\beta}_1 - \beta_1| > 1) = 0.011412$$

b)

```r
Xs = c(4, 8, 12, 16, 20)

errors = rnorm(5,0,5)
errors
```

```
## [1]  5.910658  3.719423 -4.960113 -1.684409  5.930103
```

```r
Ys = rep(0, 5)
for (i in 1:length(Ys)) {
    Ys[i] = 20 + 4*Xs[i] + errors[i]
}
Ys
```

```
## [1]  41.91066  55.71942  63.03989  82.31559 105.93010
```

```r
Sxy = 0
for (i in 1:length(Ys)) {
    Sxy = Sxy + (Xs[i] - mean(Xs))*(Ys[i] - mean(Ys))
}
Sxy
```

```
## [1] 618.5402
```

```r
Bhat1 = Sxy / Sxx
Bhat0 = mean(Ys) - Bhat1*mean(Xs)
Bhat1
```

```
## [1] 3.865876
```

```r
Bhat0
```

```
## [1] 23.39261
```

```r
X0 = 10
Yhat0 =  Bhat0 + Bhat1 * X0
Yhat0
```

```
## [1] 62.05138
```

```r
fit = lm(formula = Ys ~ Xs)
CI = predict(fit, data.frame(Xs=10), interval="confidence")
```

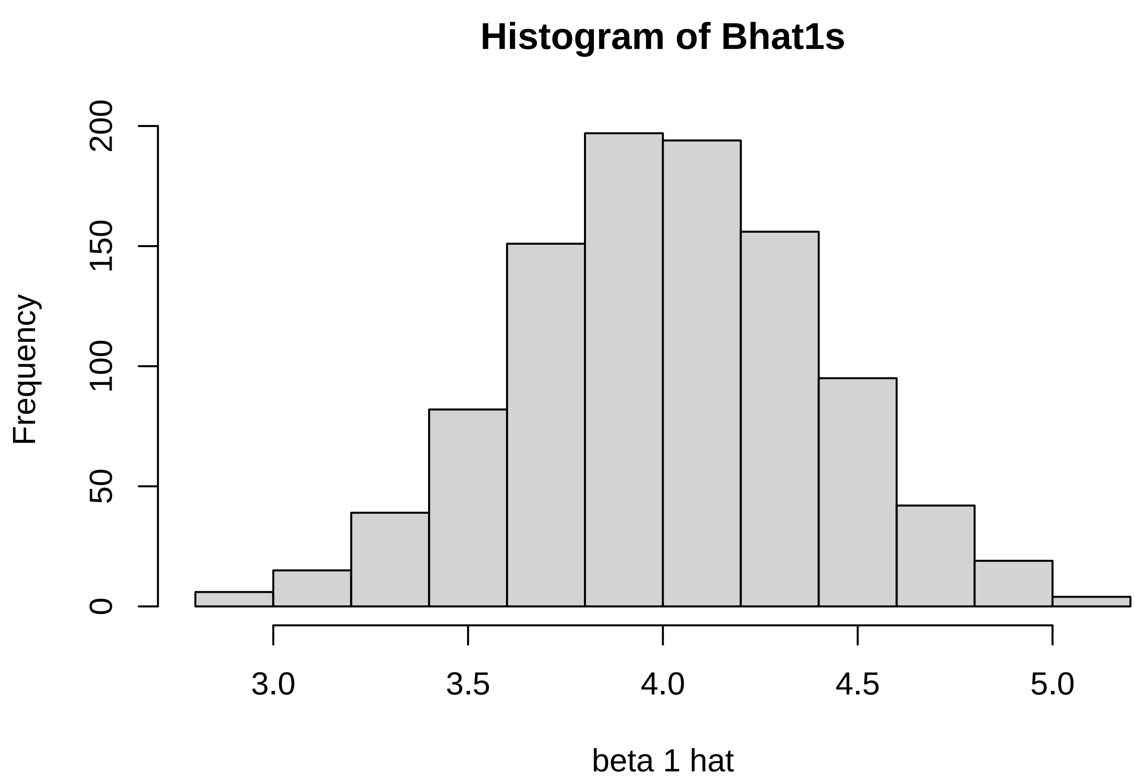
The 95% confidence interval for E(Y0) when X0 = 10 is given by

$$[53.6645619, 70.4381973]$$

c)

```r
Bhat0s = rep(0,1000)
Bhat1s = rep(0,1000)
lowers = rep(0,1000)
uppers = rep(0,1000)

Xs = c(4, 8, 12, 16, 20)

for (i in 1:length(Bhat0s)) {
  errors = rnorm(5,0,5)

  Ys = rep(0, 5)
  for (j in 1:length(Ys)) {
      Ys[j] = 20 + 4*Xs[j] + errors[j]
  }
  fit = lm(formula = Ys ~ Xs)

  Bhat0s[i] = fit$coefficients[1]
  Bhat1s[i] = fit$coefficients[2]
  CI = predict(fit, data.frame(Xs=10), interval="confidence")
  lowers[i] = CI[2]
  uppers[i] = CI[3]
 }
```

```r
hist(Bhat1s, xlab = "beta 1 hat")
```



```r
mean(Bhat1s)
```

```
## [1] 4.01303
```

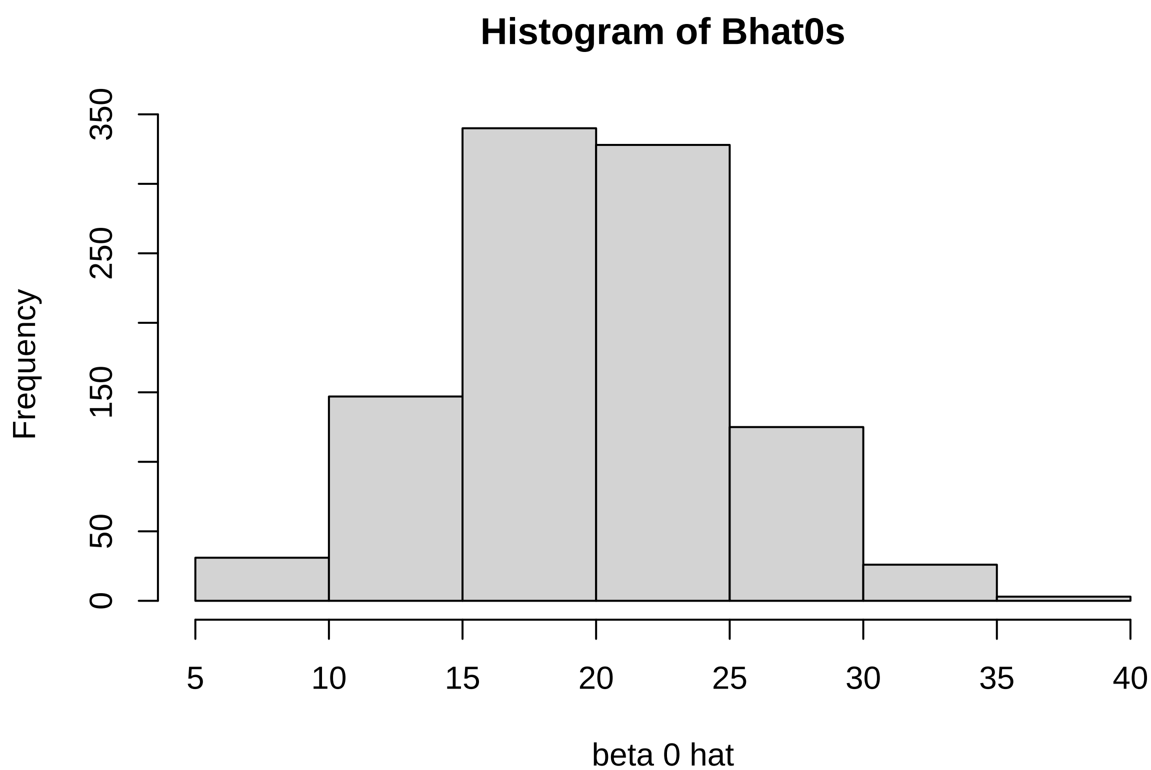
```r
sd(Bhat1s)
```

```
## [1] 0.389916
```

The sample mean of 4.0130298 is consistent with the theoretical mean of 4.

The sample standard deviation of 0.389916 is consistent with the theoretical standard deviation of $\sqrt{\sigma^2/Sxx} = \sqrt{25/160} = 0.3952$.

```r
hist(Bhat0s, xlab = "beta 0 hat")
```




```r
mean(Bhat0s)
```

```
## [1] 19.83268
```

```r
sd(Bhat0s)
```

```
## [1] 5.184647
```

The sample mean of 19.8326773 is consistent with the theoretical mean of 20.

The sample standard deviation of 5.1846469 is consistent with the theoretical standard deviation of $\sqrt{\frac{1}{n} + \frac{\bar{x}^2}{Sxx}}\sigma = \sqrt{\frac{1}{5} + \frac{12^2}{160}}5 = 5.244$.

d)

```r
numsWithinCI = 0

for (i in 1:length(lowers)) {
  Ey = 20 + 4*X0
  if (Ey > lowers[i] && Ey < uppers[i]){
    numsWithinCI = numsWithinCI + 1
  }
}
numsWithinCI
```

```
## [1] 951
```

This result of 951/1000 is consistent with the theoretical proportion of 950/1000

Q4

a)

```r
Xs = c(3,5,4,6,7)
Ys = c(4,6.5,5,7,7.5)
fit = lm(formula = Ys ~ Xs,x=TRUE)
model.matrix(fit)
```

1)

```
##   (Intercept) Xs
## 1           1  3
## 2           1  5
## 3           1  4
## 4           1  6
## 5           1  7
## attr(,"assign")
## [1] 0 1
```