STAT 403 HW7

Chongyi Xu

May 19, 2018

Question 1

In this question, we will focus on the iris dataset. In particular, we treat the variable `Petal.Width` as the response variable and the variables `Sepal.Length`, `Sepal.Width`, and `Petal.Length` as covariates. We will use the three bootstrap approaches to analyze the uncertainty of the linear regression. When using the bootstrap, please use at least $B = 10,000$ bootstrap samples.

(a) Fit a linear regression. What are the fitted coefficients?

```
dat <- iris
lm.model <- lm(data=dat, Petal.Width ~ Sepal.Length + Sepal.Width + Petal.Length)
lm.model$coefficients
```

```
##  (Intercept) Sepal.Length  Sepal.Width Petal.Length
##   -0.2403074   -0.2072661    0.2228285    0.5240831
```

(b) Apply the empirical, residual, and wild bootstrap to find the variance of the four fitted coefficients. Use a matrix to compare the variance of each fitted coefficients under the three bootstrap approaches.

```
B <- 10000
n <- nrow(dat)
empirical_BT_coef <- matrix(NA, nrow=B, ncol=4)
residual_BT_coef <- matrix(NA, nrow=B, ncol=4)
wild_BT_coef <- matrix(NA, nrow=B, ncol=4)
y_pred <- predict(lm.model)
set.seed(403)

for (i in 1:B) {
  w <- sample(n,n,replace=T)
  dat_BT <- dat[w,]

  # empirical BT
  em.model <- lm(data=dat_BT, Petal.Width ~
                   Sepal.Length + Sepal.Width + Petal.Length)
  empirical_BT_coef[i,] <- em.model$coefficients

  # residual BT
  res.y_BT <- y_pred + lm.model$residuals[w]
  res.dat_BT <- data.frame(Sepal.Length=dat$Sepal.Length,
                           Sepal.Width=dat$Sepal.Width,
                           Petal.Length=dat$Petal.Length,
                           Petal.Width=res.y_BT)
  res.model <- lm(data=res.dat_BT, Petal.Width ~
                   Sepal.Length + Sepal.Width + Petal.Length)
  residual_BT_coef[i,] <- res.model$coefficients

  # wild BT
  wild.y_BT <- y_pred + lm.model$residuals*rnorm(n)
  wild.dat_BT <- data.frame(Sepal.Length=dat$Sepal.Length,
```

```
                          Sepal.Width=dat$Sepal.Width,
                          Petal.Length=dat$Petal.Length,
                          Petal.Width=wild.y_BT)
  wild.model <- lm(data=wild.dat_BT, Petal.Width ~
                   Sepal.Length + Sepal.Width + Petal.Length)
  wild_BT_coef[i,] <- wild.model$coefficients
}
```

Then we would like to check the coefficients

```
coef_table <- matrix(NA, nrow=3, ncol=4)
colnames(coef_table) <- c('(Intercept)', 'Sepal.Length', 'Sepal.Width', 'Petal.Length')
rownames(coef_table) <- c('Empirical BT', 'Residual BT', 'Wild BT')
for (i in 1:4) {
  coef_table[1,i] <- var(empirical_BT_coef[,i])
  coef_table[2,i] <- var(residual_BT_coef[,i])
  coef_table[3,i] <- var(wild_BT_coef[,i])
}
coef_table
```

```
##              (Intercept) Sepal.Length Sepal.Width Petal.Length
## Empirical BT  0.03673946  0.002375429 0.002227567 0.0006049514
## Residual BT   0.03074004  0.002177923 0.002327318 0.0005782879
## Wild BT       0.03486566  0.002381922 0.002095413 0.0006110160
```

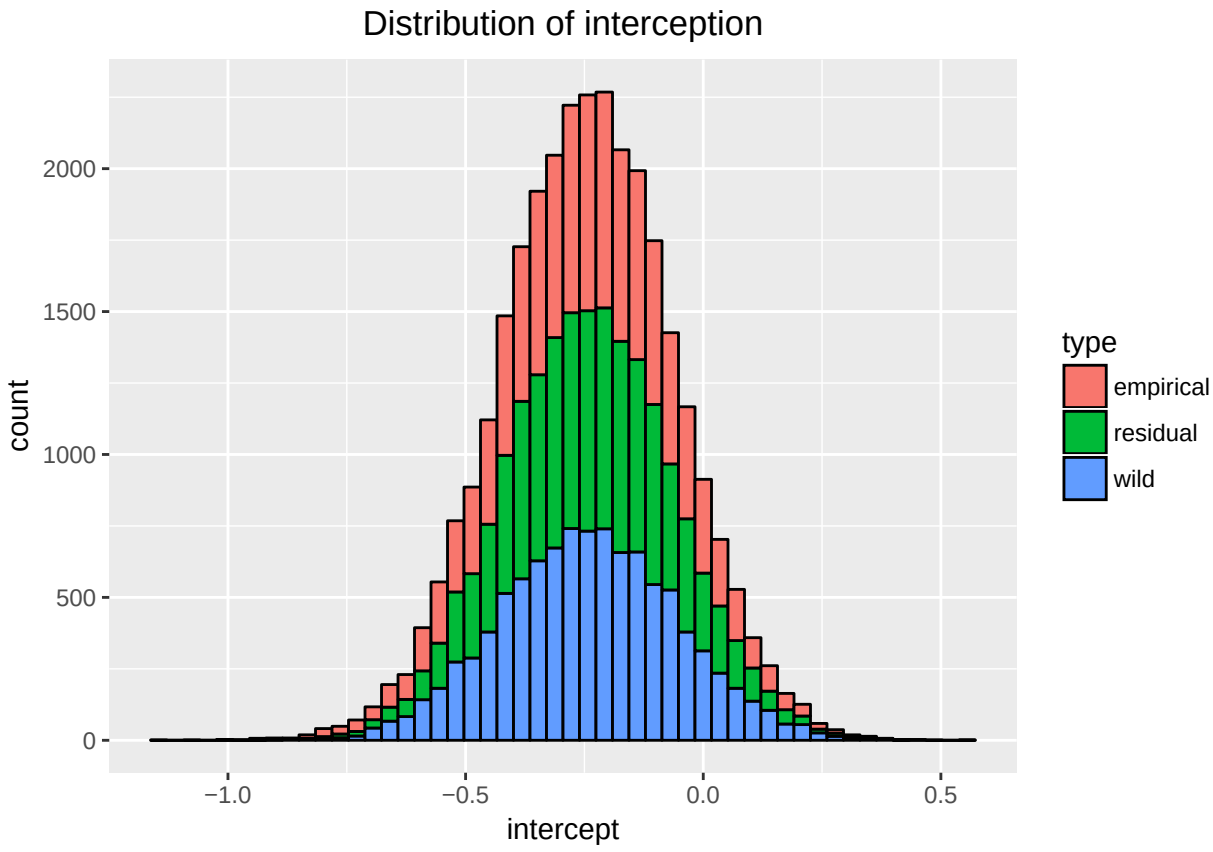
(c) For the intercept, use a single plot to compare its distribution fro mthe three bootstrap approaches.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

```
empirical <- data.frame(intercept=empirical_BT_coef[,1])
residual <- data.frame(intercept=residual_BT_coef[,1])
wild <- data.frame(intercept=wild_BT_coef[,1])
empirical$type <- 'empirical'
residual$type <- 'residual'
wild$type <- 'wild'
intercept_coef <- rbind(empirical, residual, wild)

ggplot(intercept_coef, aes(intercept, fill=type)) +
  geom_histogram(bins=50, color="black") +
  ggtitle('Distribution of interception') +
  theme(plot.title = element_text(hjust = 0.5))
```



Question 2

In this question, we will use the dataset we used in Lab 7. In particular, we will fit a logistic regression model with response Y=`admit` and two covariates `gre` and `gpa`. When using the bootstrap, please use at least $B = 10000$ bootstrap samples.

(a) Use the parametric boostrap to construct a 90% CI for the slope of `gpa`.

```
dat <- read.csv("binary.csv")
glm.model <- glm(data=dat, admit ~ gpa + gre, family='binomial')
B <- 10000
n <- nrow(dat)

parametric_BT_coef <- matrix(NA, nrow=B, ncol=3)
empirical_BT_coef <- matrix(NA, nrow=B, ncol=3)
set.seed(403)
for (i in 1:B) {
  # parametric BT
  y_BT <- rbinom(n, size=1,
                 prob=predict(glm.model, type='response'))
  para.dat_BT <- data.frame(admit=y_BT, gpa=dat$gpa, gre=dat$gre)
  parametric.model <- glm(data=para.dat_BT, admit ~ gpa + gre, family='binomial')
  parametric_BT_coef[i,] <- parametric.model$coefficients

  # empirical BT
  w <- sample(n,n,replace=T)
```

```
  em.dat_BT <- dat[w,]
  em.model <- glm(data=em.dat_BT, admit ~ gpa + gre, family='binomial')
  empirical_BT_coef[i,] <- em.model$coefficients
}
```

First we would like to know the 90% CI

```
gpa_coef <- parametric_BT_coef[,2]
CI <- quantile(gpa_coef, probs=c(0.05,0.95))
print(paste('The 90% CI is [', CI[1], ',', CI[2], ']'))
```

```
## [1] "The 90% CI is [ 0.236187652410462 , 1.28947556652807 ]"
```

(b) Apply both the parametrix and bootstrap to estimate the standard error of the intercept, slope of **gre**, and slope of **gpa**. Use a single matrix to compare the standard errors of the three parameters obtained using the two bootstrap methods and the value from `summary()` function.

```
sd_table <- matrix(NA, nrow=3, ncol=3)
colnames(sd_table) <- c('(Intercept)', 'gpa', 'gre')
rownames(sd_table) <- c('Parametric', 'Empirical', 'summary()')

for (i in 1:3) {
  sd_table[1,i] <- sd(parametric_BT_coef[,i])
  sd_table[2,i] <- sd(empirical_BT_coef[,i])
}

sd_table[3,] <- summary(glm.model)$coefficients[, 'Std. Error']

sd_table
```

```
##             (Intercept)       gpa         gre
## Parametric    1.088288 0.3206930 0.001069349
## Empirical     1.103602 0.3418348 0.001091002
## summary()     1.075093 0.3195856 0.001057491
```

(c) Assume that we are interested in the following quantity:

$$\lambda = P(\mathbf{admit} = 1 | \mathbf{gre} = 500, \mathbf{gpa} = 3.7)$$

Use a bootstrao to compute a 90% confidence interval of λ

```
B <- 10000
lambda <- rep(NA, B)

dat1 <- data.frame(gpa=3.7, gre=500)

for (i in 1:B) {
  w <- sample(n,n,replace=T)
  em.dat_BT <- dat[w,]
  em.model <- glm(data=em.dat_BT, admit ~ gpa + gre, family='binomial')
  lambda[i] <- predict(em.model, newdata=dat1, type='response')
}

CI <- quantile(lambda, probs=c(0.05,0.95))
print(paste('The 90% CI is [', CI[1], ',', CI[2], ']'))
```

```
## [1] "The 90% CI is [ 0.240658973211319 , 0.37780369222498 ]"
```

(d) Test the null hypothesis

$$H_0 : P(\textbf{admit} = 1|\textbf{gre} = 670, \textbf{gpa} = 3.9) = P(\textbf{admit} = 1|\textbf{gre} = 700, \textbf{gpa} = 2.3)$$

Let the significance level $\alpha = 0.1$ Use a bootstrap approach to see if we can reject the null hypothesis.

```
B <- 10000
difference <- rep(NA, B)
john <- data.frame(gpa=2.3, gre=700)
sam <- data.frame(gpa=3.9, gre=670)

for (i in 1:B) {
  w <- sample(n,n,replace=T)
  dat_BT <- dat[w,]
  model <- glm(data=dat_BT, admit ~ gpa + gre, family='binomial')
  john.pred <- predict(model, newdata=john, type='response')
  sam.pred <- predict(model, newdata=sam, type='response')
  difference[i] <- john.pred - sam.pred
}

CI <- quantile(difference, probs=c(0.05,0.95))
print(paste('The 90% CI of the difference between John and Sam is [', CI[1], ',', CI[2], ']'))
```

```
## [1] "The 90% CI of the difference between John and Sam is [ -0.392658336292249 , -0.0541368009995502
```

With $\alpha = 0.1$, we can see that there is a difference between John and Sam (the CI does not contain 0). So we could reject our null hypothesis.