Regression and Bootstrapping

Soumik Ghosh Moulic

Minerva Schools at KGI

Question 1

A)

```
library(ggplot2) #ggplot2 helps us to plot 2 regression lines in the
same plot
set.seed(301) #this makes sure that the same results can be
reproduced on any computer
x <- c(1:99) #generating the initial set of n random numbers. Here n
= 99
y <- 0.15*x+5 + rnorm(99,0,1) #generating the y variables based on
the x random numbers
data <- data.frame(x, y) #creating the data set of x and y
```

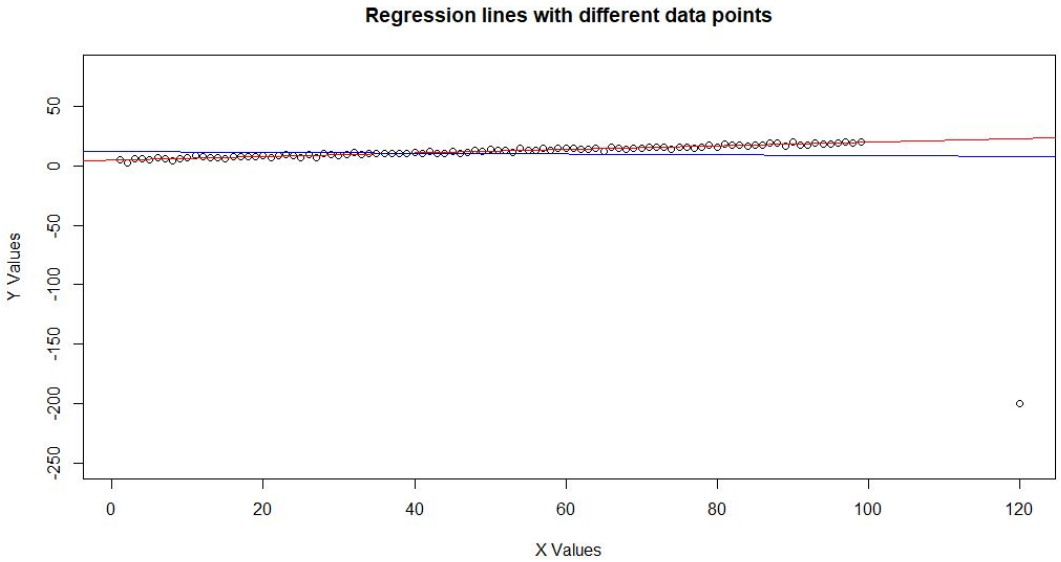
B)

```
Call:
lm(formula = y ~ x, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-2.87860 -0.47183  0.07577  0.59730  2.00287

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.02523    0.18773   26.77   <2e-16 ***
x            0.14809    0.00326   45.43   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9269 on 97 degrees of freedom
Multiple R-squared:  0.9551,     Adjusted R-squared:  0.9546
F-statistic:  2064 on 1 and 97 DF,  p-value: < 2.2e-16
```

C)

```
Call:
lm(formula = y ~ x, data = updatedData)

Residuals:
     Min       1Q   Median       3Q      Max
-208.083   -2.589    2.274    5.972   11.719

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 11.93130    4.35356   2.741  0.00729 **
x           -0.03207    0.07436  -0.431  0.66721
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 21.77 on 98 degrees of freedom
Multiple R-squared:  0.001894,   Adjusted R-squared:  -0.00829
F-statistic: 0.186 on 1 and 98 DF,  p-value: 0.6672
```

D)



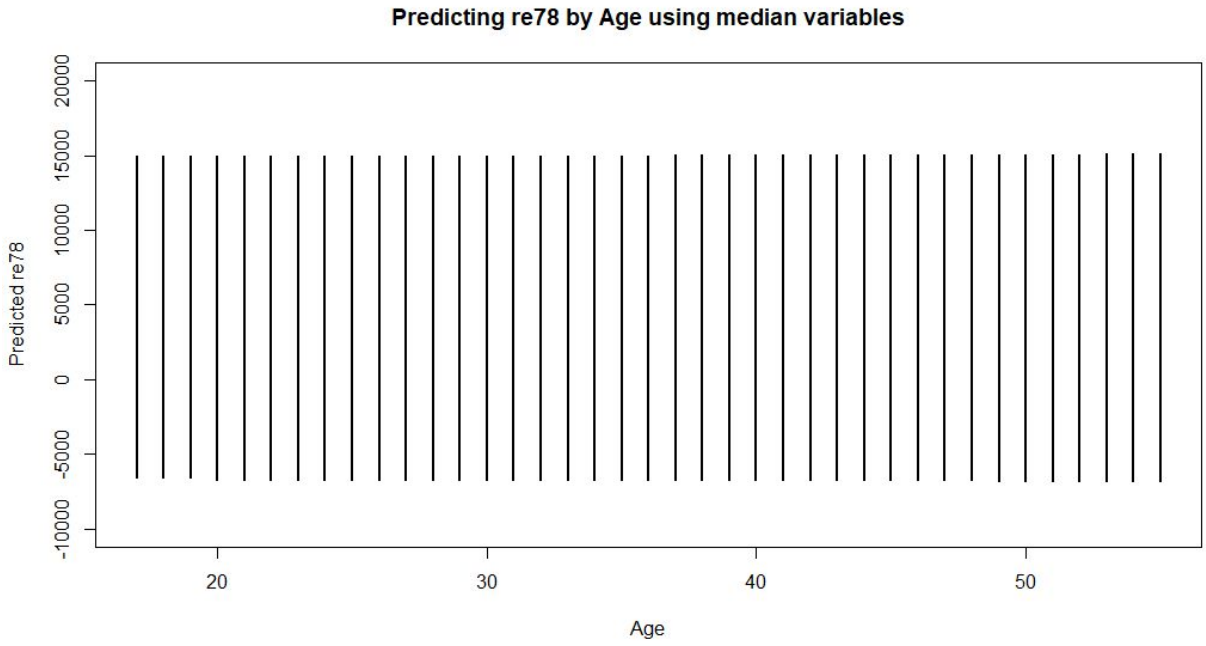
E) As we can see from the figure D), having an outlier that is not close to your original data invalidates the essence of the model in predicting the behaviour of a certain outcome that it was designed for. It is similar to the fact that trying to accommodate every behaviour of your partner makes you not attractive to them anymore because you lose yourself trying to live for them.

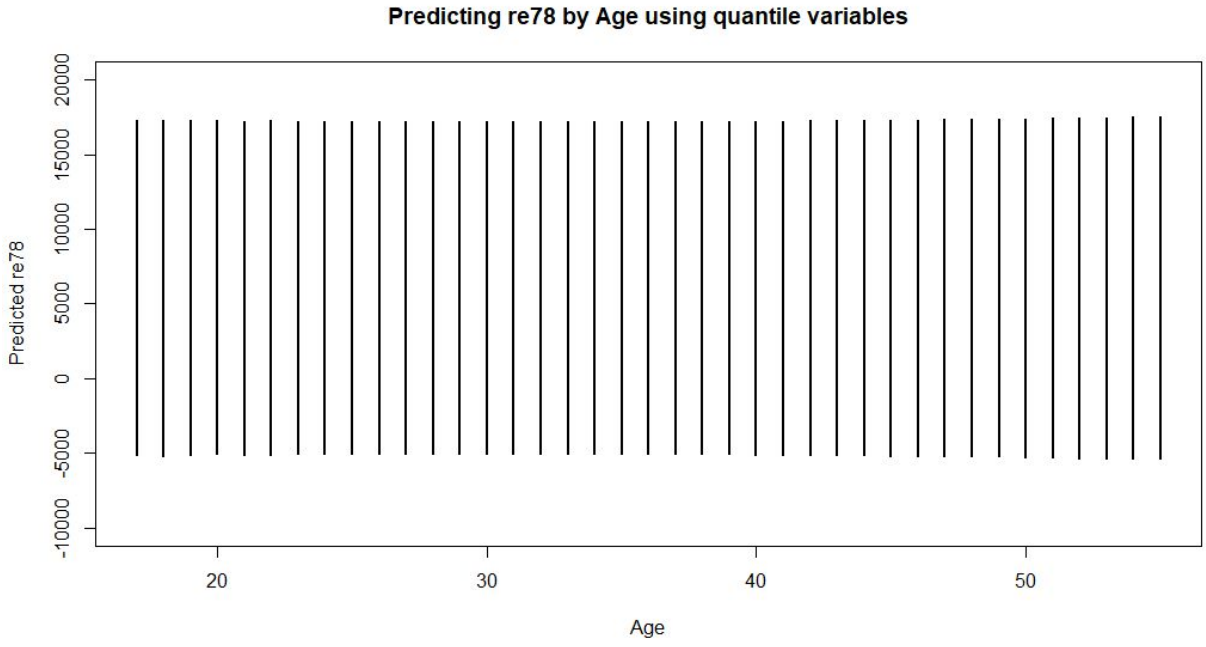
Question 2

A)

Sr No	Age	Median Education	Median Re74	Median Re75	Median Lower Bound	Median Upper Bound	Quantile Education	Quantile Re74	Quantile Re75	Quantile Lower Bound	Quantile Upper Bound
1	17	10	0	0	-6548.624	14952.54	12	7628.052	4492.998	-5177.082	17331.08
2	18	10	0	0	-6620.141	14962.51	12	7628.052	4492.998	-5194.843	17310.82
3	19	10	0	0	-6579.221	14972.18	12	7628.052	4492.998	-5161.772	17293.6
4	20	10	0	0	-6734.703	14952.4	12	7628.052	4492.998	-5079.057	17285.98
5	21	10	0	0	-6760.829	14952.4	12	7628.052	4492.998	-5142.693	17208.41
6	22	10	0	0	-6773.604	14952.51	12	7628.052	4492.998	-5156.21	17259.98
7	23	10	0	0	-6760.234	14952.51	12	7628.052	4492.998	-5092.718	17242.92
8	24	10	0	0	-6747.95	14945.95	12	7628.052	4492.998	-5062.118	17202.37
9	25	10	0	0	-6747.95	14943.84	12	7628.052	4492.998	-5057.817	17231.23
10	26	10	0	0	-6735.53	14943.84	12	7628.052	4492.998	-5043.862	17205.81
11	27	10	0	0	-6745.883	14962.36	12	7628.052	4492.998	-5045.739	17214.17
12	28	10	0	0	-6734.173	14972.18	12	7628.052	4492.998	-5051.808	17208.38
13	29	10	0	0	-6739.017	14938.18	12	7628.052	4492.998	-5043.661	17226.67
14	30	10	0	0	-6757.055	14945.95	12	7628.052	4492.998	-5049.513	17234.41
15	31	10	0	0	-6763.294	14964.42	12	7628.052	4492.998	-5065.686	17228.22
16	32	10	0	0	-6760.877	14967.65	12	7628.052	4492.998	-5066.41	17219.81
17	33	10	0	0	-6759.36	14974.83	12	7628.052	4492.998	-5082.152	17212.15
18	34	10	0	0	-6757.007	14998.82	12	7628.052	4492.998	-5083.792	17211.24
19	35	10	0	0	-6760.387	14994.26	12	7628.052	4492.998	-5089.962	17214.28
20	36	10	0	0	-6766.608	14999.01	12	7628.052	4492.998	-5098.657	17218.78
21	37	10	0	0	-6767.239	15012.45	12	7628.052	4492.998	-5093.99	17215.85
22	38	10	0	0	-6764.735	15013.5	12	7628.052	4492.998	-5092.469	17225.93
23	39	10	0	0	-6760.387	15025.42	12	7628.052	4492.998	-5097.308	17229.92
24	40	10	0	0	-6759.444	15032.59	12	7628.052	4492.998	-5112.996	17238.75
25	41	10	0	0	-6753.369	15033.31	12	7628.052	4492.998	-5119.835	17246.29
26	42	10	0	0	-6759.573	15033.49	12	7628.052	4492.998	-5125.688	17255.34
27	43	10	0	0	-6767.774	15034.58	12	7628.052	4492.998	-5134.763	17268.64
28	44	10	0	0	-6764.097	15041.24	12	7628.052	4492.998	-5169.38	17285.05
29	45	10	0	0	-6773.693	15039.34	12	7628.052	4492.998	-5184.219	17300.45
30	46	10	0	0	-6778.328	15038.09	12	7628.052	4492.998	-5194.82	17317.18
31	47	10	0	0	-6782.436	15046.38	12	7628.052	4492.998	-5214.788	17340.23
32	48	10	0	0	-6785.573	15046.38	12	7628.052	4492.998	-5238.47	17356.58
33	49	10	0	0	-6788.748	15061.57	12	7628.052	4492.998	-5261.803	17384.42
34	50	10	0	0	-6796.957	15072.95	12	7628.052	4492.998	-5290.579	17403.6
35	51	10	0	0	-6809.476	15076.89	12	7628.052	4492.998	-5323.843	17431.97
36	52	10	0	0	-6817.91	15083.4	12	7628.052	4492.998	-5348.605	17461.8
37	53	10	0	0	-6820.448	15104.5	12	7628.052	4492.998	-5372.975	17489.64
38	54	10	0	0	-6830.929	15108.69	12	7628.052	4492.998	-5399.456	17521.27
39	55	10	0	0	-6833.026	15117.29	12	7628.052	4492.998	-5424.376	17547.23

B)


Predicting re78 by Age using median variables
Predicted re78
20000
15000
10000
5000
0
-5000
-10000
20
30
40
50
Age

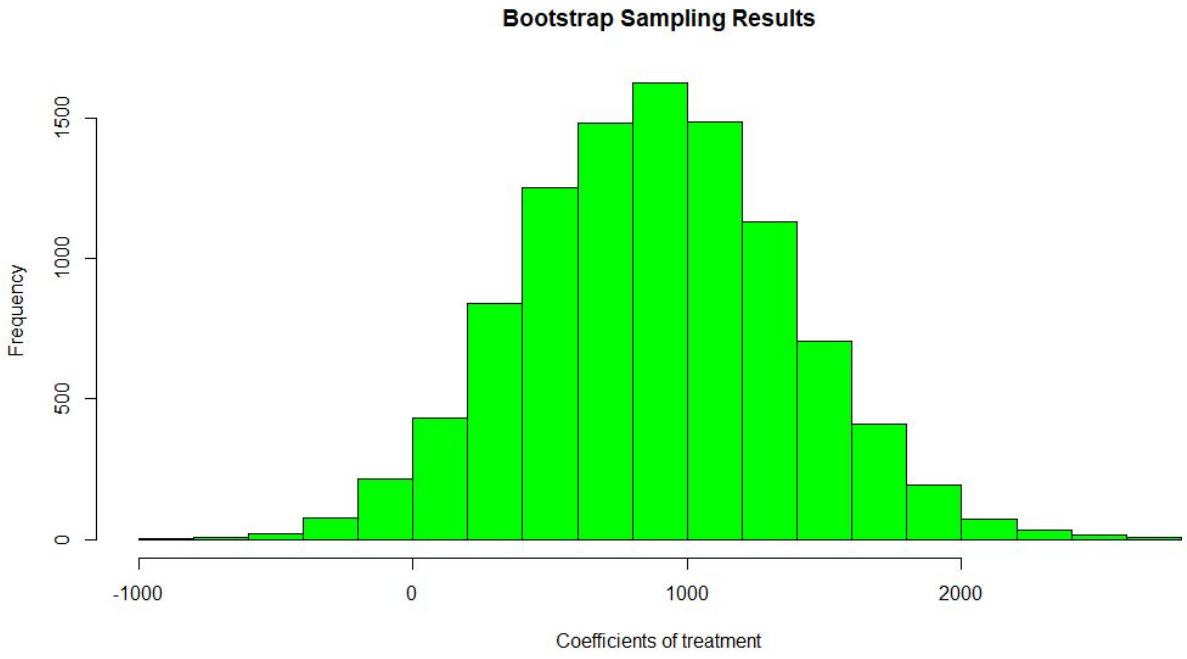

Predicting re78 by Age using quantile variables
Predicted re78
20000
15000
10000
5000
0
-5000
-10000
20
30
40
50
Age

Question 3

A)

```
        ci_using_formula ci_using_bootstrap
2.5 %          -40.52635          -46.86176
97.5 %        1813.13380         1854.75949
```

B)



C) We can see that using bootstrap to calculate the confidence interval does lead to really close results as using the inbuilt functions and methods. This indicates that further optimizing the bootstrap method will lead to better predictions. Also using bootstrap allows us to generate more data from a limited number of observations, which in turn help us predict with better accuracy.

Question 4

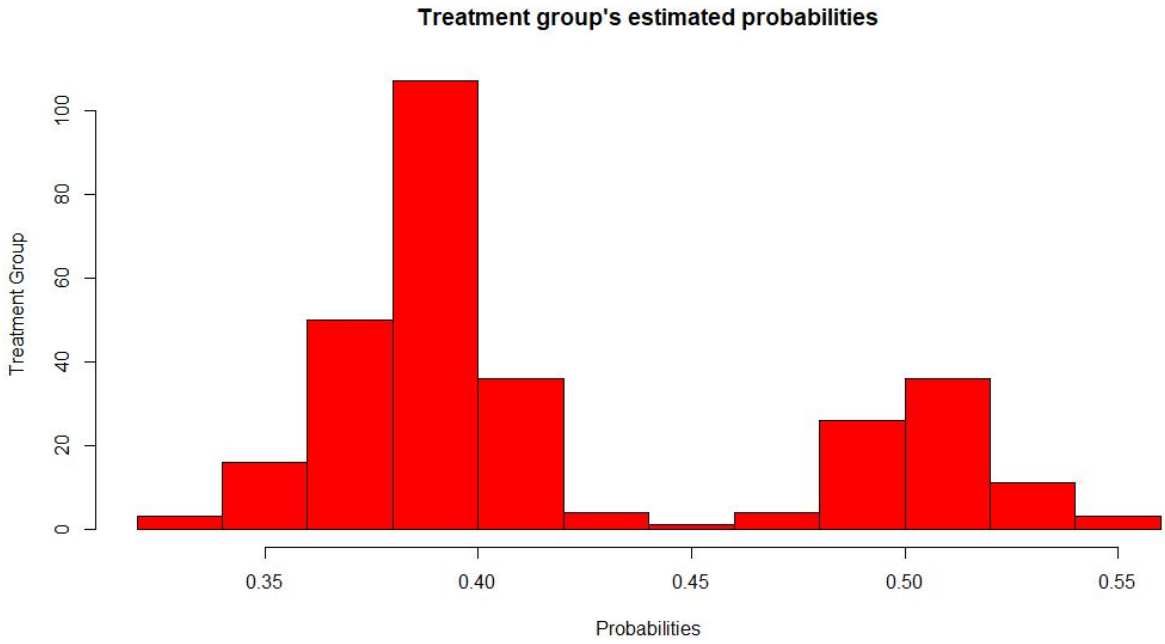
```
rss <- vector() # residual sum of squares
tss <- vector() # total sum of squares

rsquared = function(pred_y, actual_y){ #creating a function for R
Squared
  rss <- sum((pred_y - actual_y) ^ 2)
  tss <- sum((actual_y - mean(actual_y)) ^ 2)
  rsq <- 1 - rss/tss
  rsq
}

pred_y = predict(model1)
actual_y = datas$re78
function_rsq_value <- rsquared(pred_y, actual_y)

#comparing the R^2 values from the summary of the model1 from
question 3,
#vs the rsquared value from question 4 to check the accuracy
all.equal(summary(model1)$r.squared, function_rsq_value, tolerance =
1.5e-14)
```

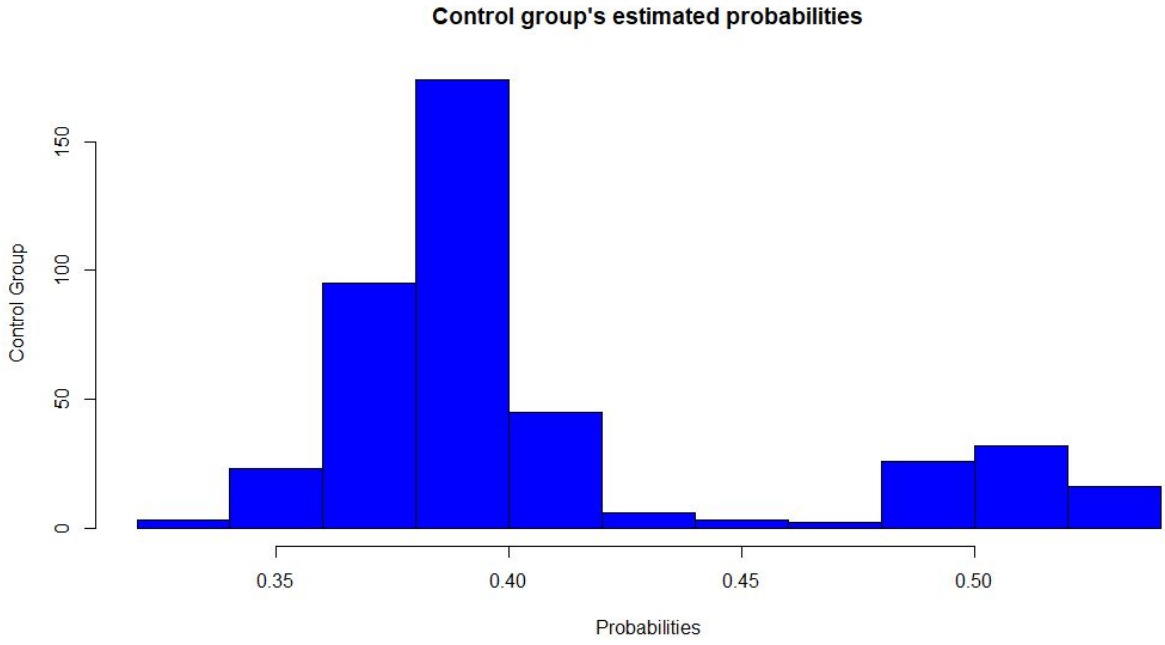
Output

```
> function_rsq_value
[1] 0.004871571
> summary(model1)$r.squared
[1] 0.004871571
>all.equal(summary(model1)$r.squared, function_rsq_value, tolerance =
1.5e-14)
[1] TRUE
```

Question 5

A)


Treatment group's estimated probabilities
Treatment Group
0
20
40
60
80
100
0.35
0.40
0.45
0.50
0.55
Probabilities


Control group's estimated probabilities
Control Group
0
50
100
150
0.35
0.40
0.45
0.50
Probabilities

B) Looking at the histograms, the probability distributions look really identical, suggesting that the probabilities are not affected by the outcomes of being in a specific group. This might be because the assignment of people in different groups was really random.

Appendix

Gist - https://gist.github.com/Soumik0833/f4e653d6cb58478bf4e9e56c9b0a9e5d

```
Question 1
library(ggplot2) #ggplot2 helps us to plot 2 regression lines in the
same plot
set.seed(301) #this makes sure that the same results can be
reproduced on any computer
x <- c(1:99) #generating the initial set of n random numbers. Here n
= 99
y <- 0.15*x+5 + rnorm(99,0,1) #generating the y variables based on
the x random numbers
data <- data.frame(x, y) #creating the data set of x and y
reg1 <- lm(y~x, data=data) #creating the first regression line for
our data
summary(reg1) #summary of reg1
#outlier <-  data.frame(x=c(0), y=c(-1500)) #generating an outlier to
the data
updatedData <- rbind(data, c(120,-200)) #adding the outlier to the
existing dataset to generate a new one
reg2 <- lm(y~x, data=updatedData) #creating the second regression
line using the new dataset
summary(reg2) #summary of reg2
plot(updatedData, ylim=c(-250, 80), xlab="X Values", ylab = "Y
Values", main = "Regression lines with different data points")
#plotting the dataset
abline(reg1, col='red', lwd=1) #plotting the first reg line
abline(reg2, col='blue', lwd=1) #plotting the second reg line

Question 2
#loading prereqs
library("arm")
data(lalonde)
set.seed(20)
attach(data_control_only)

data_control_only <- lalonde[which(lalonde$treat==0),] #getting the
```

```
data that only has control and not treatment
#creating the linear model for the control group data on the given
variables
linear_model <- lm(re78 ~ age + educ + re74 + re75 + educ*re74 +
educ*re75 + age*re74 + age*re75 + re74*re75, data_control_only)

#simulations
reps = 10000
simulation <- sim(linear_model, reps)
simulation

#medians
med_edu <- median(educ)
med_re74 <- median(re74)
med_re75 <- median(re75)

#creating a loop
mat1 <- matrix(, nrow = 39, ncol = 3)
store_prediction <- vector()
for (age in c(17:55)){
    Xs <- c(1, age, med_edu, med_re74, med_re75, med_edu*med_re74,
med_edu*med_re75, age*med_re74, age*med_re75, med_re74*med_re75)
    for (i in 1:reps){
      prediction <- sum((simulation@coef[i,])*Xs) + rnorm(1, 0,
simulation@sigma[i])
      store_prediction <- c(store_prediction, prediction )

    }
    mat1[age-16,] <- c(age,  quantile(store_prediction, probs =
c(0.025,0.975)))
}
#store_prediction
#mat1

plot(x = c(1:100), y = c(1:100), type = "n", xlim = c(17,55), ylim =
c(-10000,20000),
     main = "Predicting re78 by Age using median variables", xlab =
```

```
"Age",
     ylab = "Predicted re78")

for (age in 17:55) {
  segments(
    x0 = age,
    y0 = mat1[age-16, 2],
    x1 = age,
    y1 = mat1[age-16, 3],
    lwd = 2)
}

#quantiles
quan_edu <- quantile(educ, probs = 0.9)
quan_re74 <- quantile(re74, probs = 0.9)
quan_re75 <- quantile(re75, probs = 0.9)

#creating a loop
mat2 <- matrix(, nrow = 39, ncol = 3)
store_prediction2 <- vector()
for (age in c(17:55)){
  Xz <- c(1, age, quan_edu, quan_re74, quan_re75, quan_edu*quan_re74,
quan_edu*quan_re75, age*quan_re74, age*quan_re75,
quan_re74*quan_re75)
  for (i in 1:reps){
    prediction2 <- sum((simulation@coef[i,])*Xz) + rnorm(1, 0,
simulation@sigma[i])
    store_prediction2 <- c(store_prediction2, prediction2 )

  }
  mat2[age-16,] <- c(age,  quantile(store_prediction2, probs =
c(0.025,0.975)))
}
store_prediction2
mat2

plot(x = c(1:100), y = c(1:100), type = "n", xlim = c(17,55), ylim =
```

```
c(-10000,20000),
     main = "Predicting re78 by Age using quantile variables", xlab =
"Age",
     ylab = "Predicted re78")

for (age in 17:55) {
  segments(
    x0 = age,
    y0 = mat2[age-16, 2],
    x1 = age,
    y1 = mat2[age-16, 3],
    lwd = 2)
}


table5 <- data.frame(mat1[,1], med_edu, med_re74, med_re75, mat1[,2],
mat1[,3], quan_edu, quan_re74, quan_re75, mat2[,2], mat2[,3])
colnames(table5) <- c("Age", "Median Education", "Median Re74",
"Median Re75", "Prediction Bound A", "Prediction Bound B", "Quantile
Education", "Quantile Re74", "Quantile Re75", "Prediction A using
Quantile", "Prediction B using Quantile")
```

Question 3

```
#seeting up vars
library(foreign)
datas <- read.dta("nsw.dta")
set.seed(123)

#model generation
model1 <- lm(re78 ~ treat, datas)
model1
summary(model1)
ci_using_formula <- confint(model1, level = 0.95)[2,]
ci_using_formula

#ci using bootstrap
```

```
store <- vector()
for (i in 1:10000){
  indexing <- sample(1:nrow(datas), nrow(datas), replace = T)
  bootstrap <- datas[indexing, ]
  bootstrapping_ci <- lm(re78 ~ treat, bootstrap)
  store <- c(store, bootstrapping_ci$coef[2])
}
ci_using_bootstrap <- quantile(store, probs = c(0.025,0.975))
ci_using_bootstrap

#histogram
hiss <- hist(store, xlab = "Coefficients of treatment", main =
"Bootstrap Sampling Results", col = "green")
hiss

#table with relevant results
summary_table <- data.frame(ci_using_formula, ci_using_bootstrap)
summary_table

Question 4
rss <- vector() # residual sum of squares
tss <- vector() # total sum of squares

rsquared = function(pred_y, actual_y){ #creating a function for R
Squared
  rss <- sum((pred_y - actual_y) ^ 2)
  tss <- sum((actual_y - mean(actual_y)) ^ 2)
  rsq <- 1 - rss/tss
  rsq
}

pred_y = predict(model1)
actual_y = datas$re78
function_rsq_value <- rsquared(pred_y, actual_y)
function_rsq_value
summary(model1)$r.squared
```

```
#comparing the R^2 values from the summary of the model1 from
question 3,
#vs the rsquared value from question 4 to check the accuracy
all.equal(summary(model1)$r.squared, function_rsq_value, tolerance =
1.5e-14)

Question 5
#setting up vars
library(foreign)
datas <- read.dta("nsw.dta")
set.seed(123)

#building the models
glm_fitting<-
glm(treat~age+education+black+hispanic+married+nodegree+re75,
data=datas, family=binomial)
glm_probabilities <- predict(glm_fitting, type="response")
View(glm_probabilities)

control_set <- vector()
treatment_set <- vector()
for (i in 1:nrow(datas)){
  if (datas$treat[i] == 0) {
    control_set <- c(control_set, glm_probabilities[i])
  }
  else {
    treatment_set <- c(treatment_set, glm_probabilities[i])
  }
}

histogram_a <- hist(treatment_set, col = "red", xlab =
"Probabilities", ylab = "Treatment Group", main = "Treatment group's
estimated probabilities")
histogram_b <- hist(control_set, col = "blue", xlab =
"Probabilities", ylab = "Control Group", main = "Control group's
estimated probabilities")
```