MSCI - 718

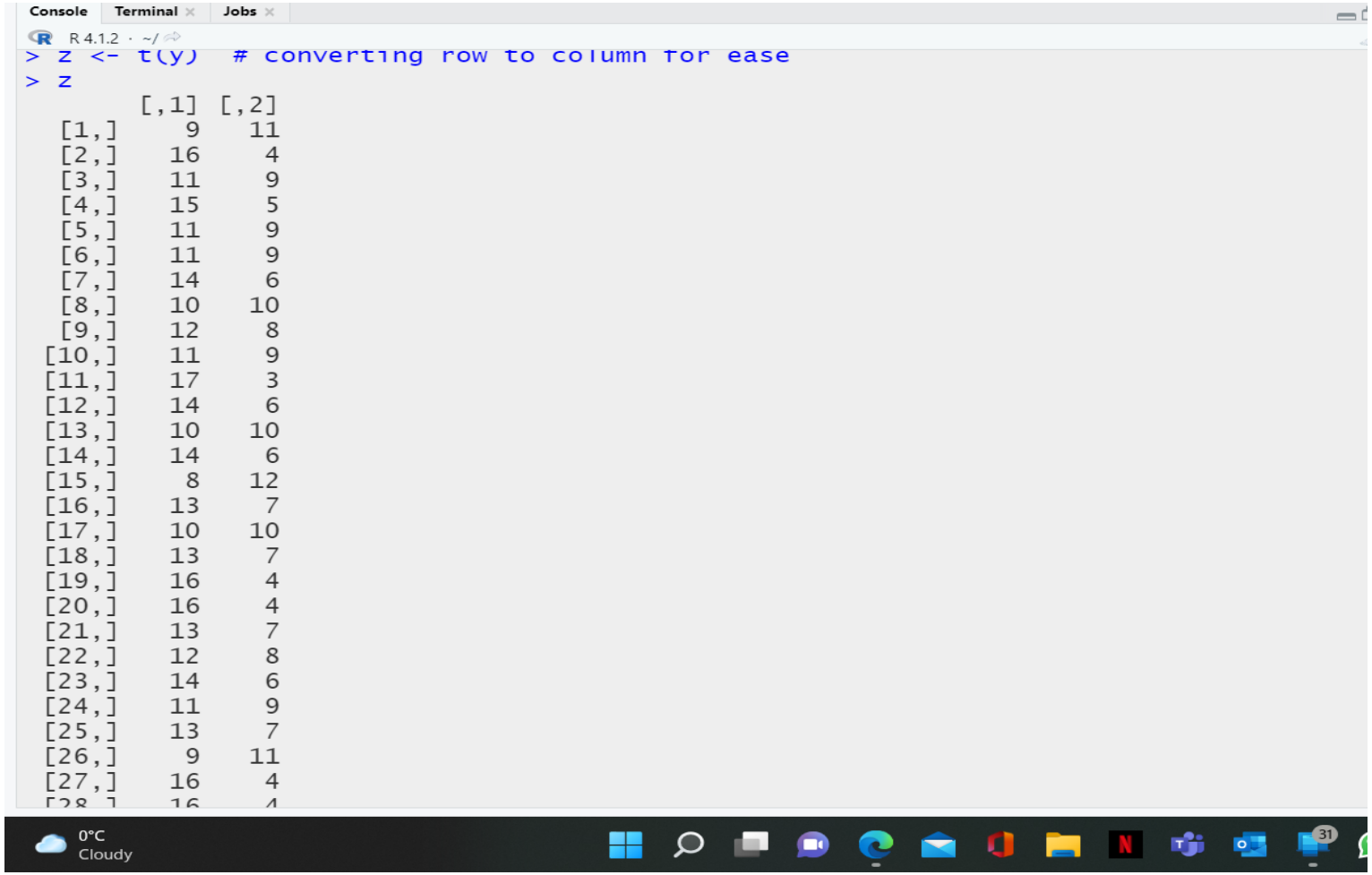
Juhi Vasudev Bachani

20979706

Section 1: R Programming Questions: Q-2

```
library(MASS)
library(fitdistrplus)
#### Just for reference given O for blue fish and 1 for Gold fish
b <- 0
g <- 1
Lake <- c (b, g)
Blue1 <- 60
Gold1 <- 40
Catch1 <- Blue1 + Gold1  # 1st catch where there were 60 blue & 40 gold fish

y <- rmultinom(100, size = 20, prob = c(0.6,0.4))
#generating random matrix where 20 fishes are picked with probability 0.6 and 0.4
z <- t(y)  # converting row to column for ease
z
```

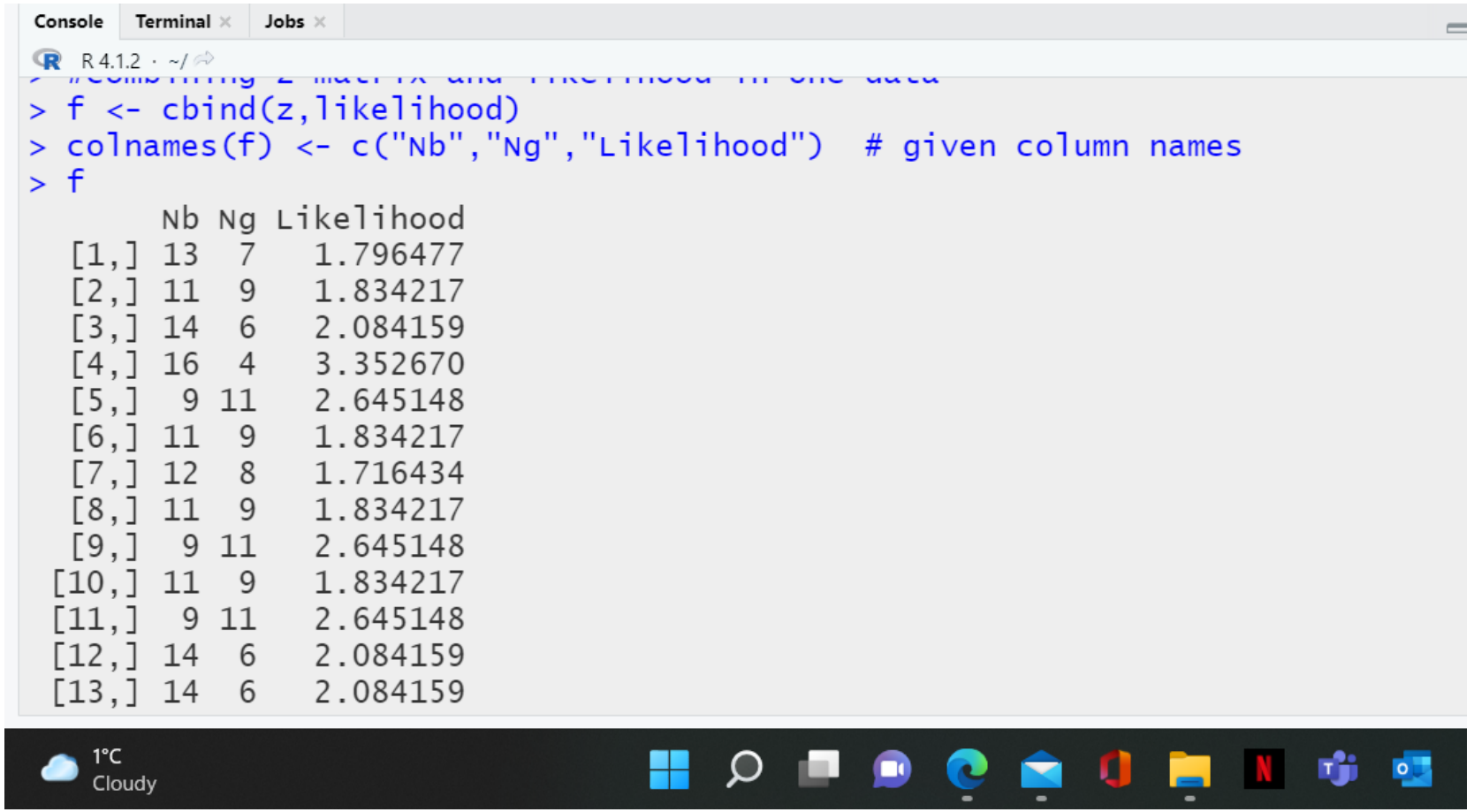
Output of z


Console
Terminal
Jobs
R 4.1.2 · ~/
> z <- t(y) # converting row to column for ease
> z
[,1] [,2]
[1,] 9 11
[2,] 16 4
[3,] 11 9
[4,] 15 5
[5,] 11 9
[6,] 11 9
[7,] 14 6
[8,] 10 10
[9,] 12 8
[10,] 11 9
[11,] 17 3
[12,] 14 6
[13,] 10 10
[14,] 14 6
[15,] 8 12
[16,] 13 7
[17,] 10 10
[18,] 13 7
[19,] 16 4
[20,] 16 4
[21,] 13 7
[22,] 12 8
[23,] 14 6
[24,] 11 9
[25,] 13 7
[26,] 9 11
[27,] 16 4
0°C
Cloudy

Creating likelihood

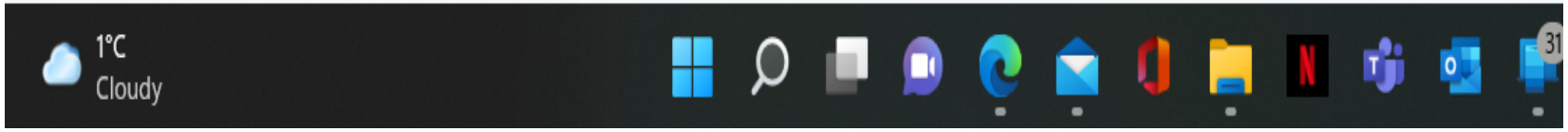
```
likelihood <- c()        #creating likelihood function empty list

#creating likelihood with previous random outputs generated
for(i in 1:nrow(z)){
  likelihood[i] <- -dmultinom(z[i,], prob=c(0.6,0.4),log =T)
}
#combining z matrix and likelihood in one data
f <- cbind(z,likelihood)
colnames(f) <- c("Nb","Ng","Likelihood")          # giving column names
f
max(likelihood)                                   #searching for max likelihood value
ml <- f[which(likelihood == max(likelihood)), ]
ml
```

Output of f


Console
Terminal
Jobs
R 4.1.2 · ~/
> f <- cbind(z,likelihood)
> colnames(f) <- c("Nb","Ng","Likelihood") # given column names
> f
Nb Ng Likelihood
[1,] 13 7 1.796477
[2,] 11 9 1.834217
[3,] 14 6 2.084159
[4,] 16 4 3.352670
[5,] 9 11 2.645148
[6,] 11 9 1.834217
[7,] 12 8 1.716434
[8,] 11 9 1.834217
[9,] 9 11 2.645148
[10,] 11 9 1.834217
[11,] 9 11 2.645148
[12,] 14 6 2.084159
[13,] 14 6 2.084159
1°C
Cloudy

Output of Likelihood

```
 [97,] 11  9   1.834217
 [98,] 11  9   1.834217
 [99,] 11  9   1.834217
[100,] 10 10   2.144372
> max(likelihood)    #searching for max likelihood value
[1] 4.229268
> ml <- f[which(likelihood == max(likelihood)), ]
> ml
     Nb Ng Likelihood
[1,]  7 13   4.229268
[2,]  7 13   4.229268
[3,]  7 13   4.229268
>
```



Julie Vasudev Bachani Id-No: 20979706

Project (MSCI-718)

Section: 1

(Q-1) a) Bluefish = 60
Goldfish = 40
} Catch 1 = 100

Nb
Ng
} 20 tagged } Catch = 2

a) Here I used rmultinom to generate random samples of size 20 using probability of 0.6 and

```
y= rmultinom (100, size = 20, ~~prob~~
                  prob= c(0.6,0.4))

z <- t(y)        # Converting row ~~and~~ to columns
```

b) Created likelihood function and passed ~~it to~~ previous random samples into it.

c) From using this, I got

4.229268 = max (likelihood)

Also, Combined z value and likelihood using cbind

Then ~~generated~~ tried to find value of Nb, Ng using maximum likelihood.

(d) Nb and Ng ~~were~~ are given

hypothesis 1: Nb = 280 Ng = 220

hypothesis 2: Nb = 290, Ng = 210

According to me, chances of hypothesis 2 is more likely as we assumed probability of blue fish = 0.6 and of gold fish = 0.4 from our 1st catch.

```
s<- function(Nb,Ng){
  Ng1<- Ng/20
  print(Ng)
  Nb1<- Nb/20
  print(Nb)
  z[Nb1,Ng1]
}
s(280,220)
s(290,210)
### here it seems the 2st hypothesis is more likely to happen as it shows chance of dividing probability in 0.6 and 0.4 value.
```

Section 2: Data Analysis Questions (Q-2)

```
library(foreign)
#Loading the 2 datasets
Dat1 = read.dta("Q1Data1.dta")
Dat2 = read.csv("Q1Data2.csv", header =T, stringsAsFactors = F)
names(Dat1)    #to view all column headings name

#----- a-1) Creating Subset by Removing  the states named "Hawaii", "Alaska", and "Washington D.C"
x<-0
x <- subset(Dat1, state!= c("hawaii"))      #Removed hawaii
x <- subset(x, state!= c("alaska"))         #Removed alaska
x <- subset(x, state!= c("washington dc"))  #washington dc
x
# only four columns “state",“marital”, “heat2”, and “heat4”
Dat1_mod <- x[,c("state","marital", "heat2", "heat4")]
```

- #----- a-2) If heat 2 and heat4 ==NA , then remove that row

```
Dat1_mod_no_na<-subset(Dat1_mod,!(heat2 %in% NA & heat4 %in% NA))
#Removing all NA values from marital column
Dat1_mod_no_naa<-subset(Dat1_mod_no_na,!(marital %in% NA))
```

- #----- a-3) Heat2 has only 2 values after subsetting "dem/lean dem" & "rep/lean rep"

```
Dat1_mod_no_naa_h <- subset(Dat1_mod_no_naa, heat2 == c("dem/lean dem", "rep/lean rep"))
Out <- Dat1_mod_no_naa_h
```

- #-----a-4) changing all values of marital column to "others" except "married"

```
Out$marital <- ifelse(Out$marital != "married", Out$marital<-"others",Out$marital<-"married")
Out
View(Out)
```

Output after a-4 (out)

project (q-2).R | Out | project(q-1).R*

Filter

	state	marital	heat2	heat4
1	massachusetts	others	dem/lean dem	NA
3	massachusetts	married	dem/lean dem	NA
5	massachusetts	others	dem/lean dem	NA
6	massachusetts	others	rep/lean rep	NA
7	massachusetts	others	dem/lean dem	NA
8	massachusetts	married	rep/lean rep	NA
9	massachusetts	others	dem/lean dem	NA
13	massachusetts	married	rep/lean rep	NA
19	massachusetts	married	rep/lean rep	NA
20	massachusetts	married	dem/lean dem	NA
21	massachusetts	married	rep/lean rep	NA
23	massachusetts	others	rep/lean rep	NA
27	massachusetts	married	rep/lean rep	NA
30	massachusetts	married	dem/lean dem	NA
37	massachusetts	others	dem/lean dem	NA
40	massachusetts	married	rep/lean rep	NA
42	massachusetts	others	rep/lean rep	NA
48	massachusetts	married	rep/lean rep	NA

- #----b-1)

```
#for (i in 1:length(levels(Out$state))) {
group_by(Out, state) %>% mutate(ratio = heat2 / sum(heat2))
m<- Out$heat2
library(dplyr)
summarise_at(group_by(Out,state),funs(mean(.,na.rm=TRUE)))
```

- #---- b-2)

```
for(state in 1:length(marital)){
  marital <- Out %>%select(state, marital)
  sum(Out$married)/sum(Out$marital)
}
```

- #---- b-3)

```
group_by(Out, state) %>% mutate(ratio = marital$married / marital)
```

- #---- c-1) creating subset by removing three states,"Hawaii", "Alaska", and "District of Columbia“ for Dat2

```
x1 <- subset(Dat2, state != c("Hawaii", "Alaska" , "District of Columbia"))
```

#----c-2) Reducing columns to only two columns “state,” and “vote_Obama_pct”

```
Dat2_mod <- x1[,c(1,3)]      # choosing column number in c(1,3)
head(Dat2_mod)
```

head

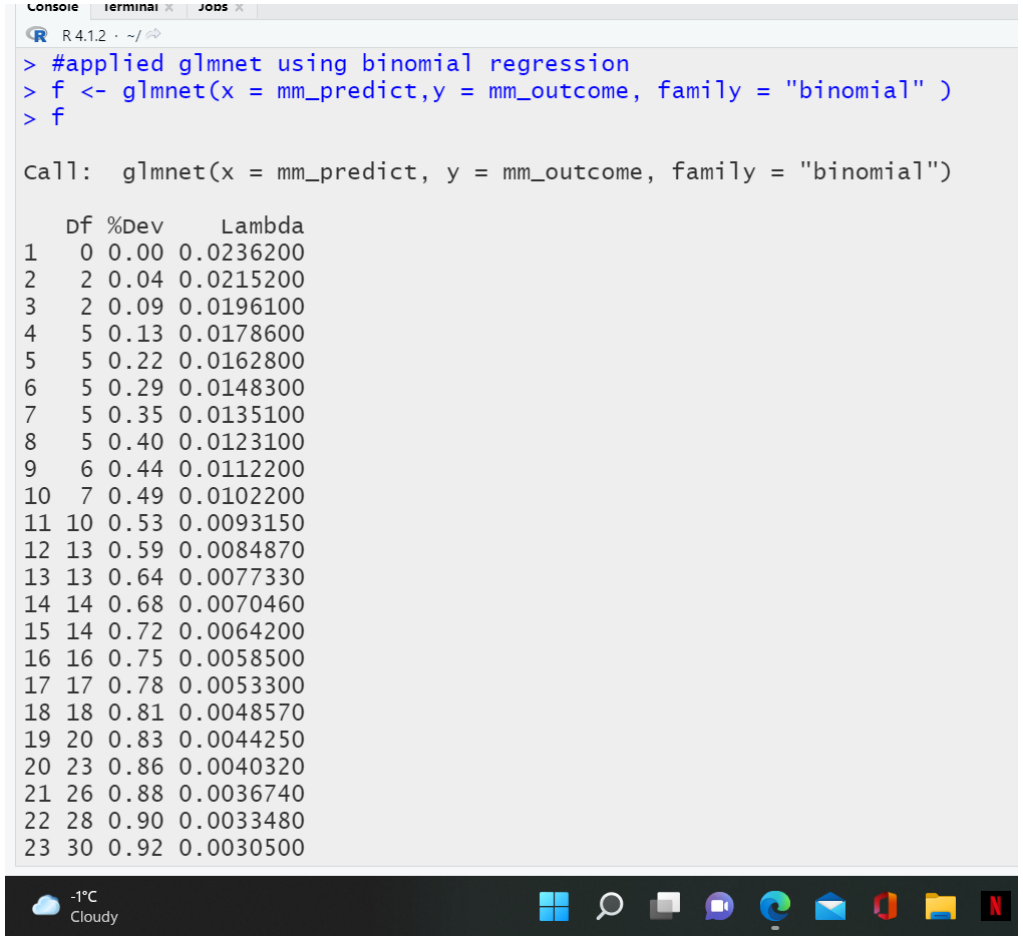
Output of c)

```
> Dat2_mod <- x1[,c(1,3)]
> head(Dat2_mod)
        state vote_Obama_pct
1     Alabama           38.8
3     Arizona           45.0
4    Arkansas           38.8
5  California           60.9
6    Colorado           53.5
7 Connecticut           60.5
> head
```

#---- d) logistic regression

```
mm_outcome <-Out$marital   # y for model
mm_predict <- model.matrix(marital~state, data = Out)
#x predict value
library(glmnet)
#applied glmnet using binomial regression
f <- glmnet(x = mm_predict,y = mm_outcome, family = "binomial" )
```

Result after applying glmnet



- #Assumption 1) No pooling (means lambda = 0)

```
no_pooling <- glmnet(x = mm_predict, y = mm_outcome, family = "binomial",alpha = 0, lambda = 0)
coef(no_pooling)
```

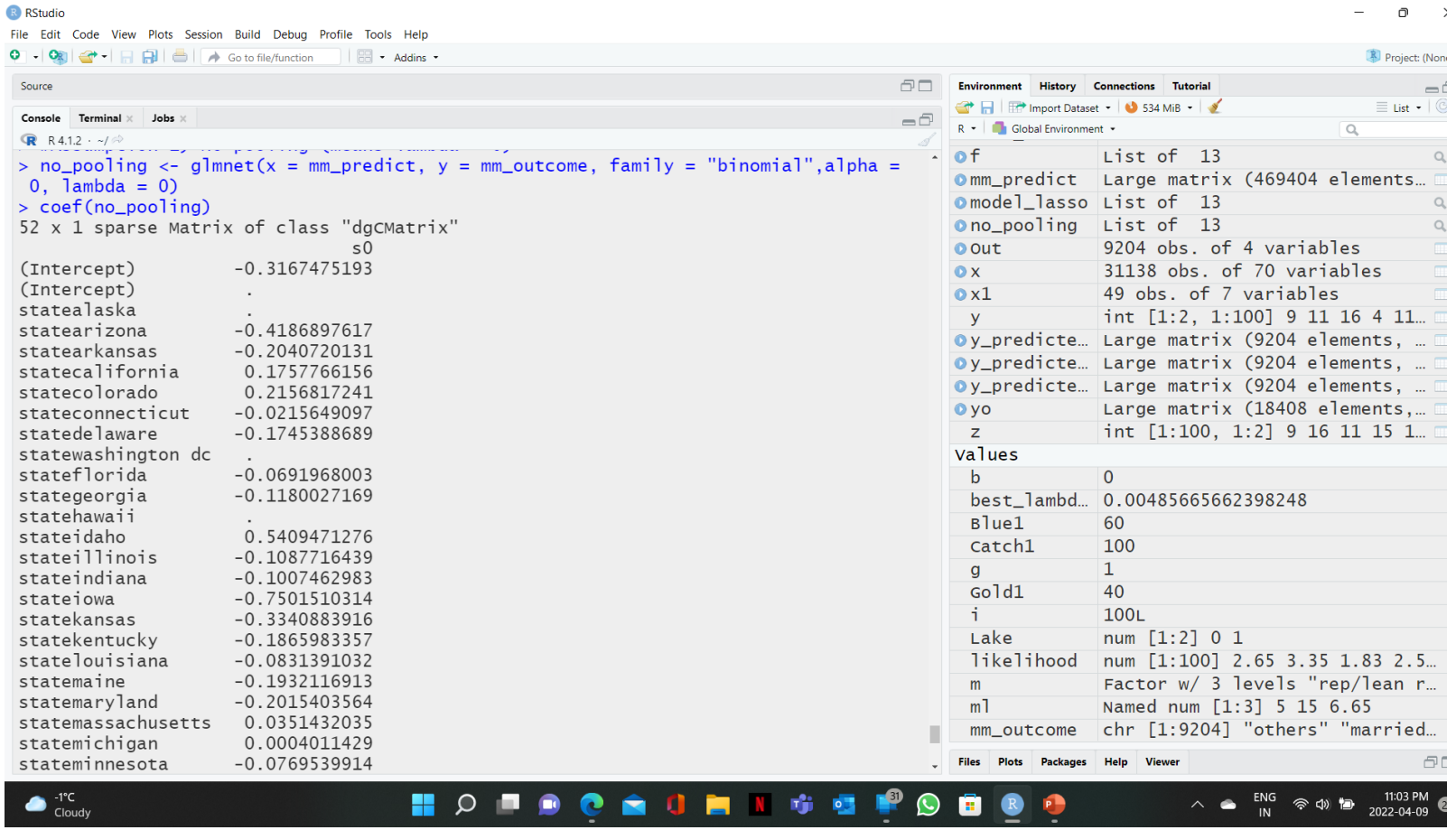
- #Assumption 2) Complete pooling (means lambda = high)

```
complete_pooling <- glmnet(x = mm_predict, y = mm_outcome,family = "binomial",alpha = 0, lambda = 10^5)
coef(complete_pooling)
```

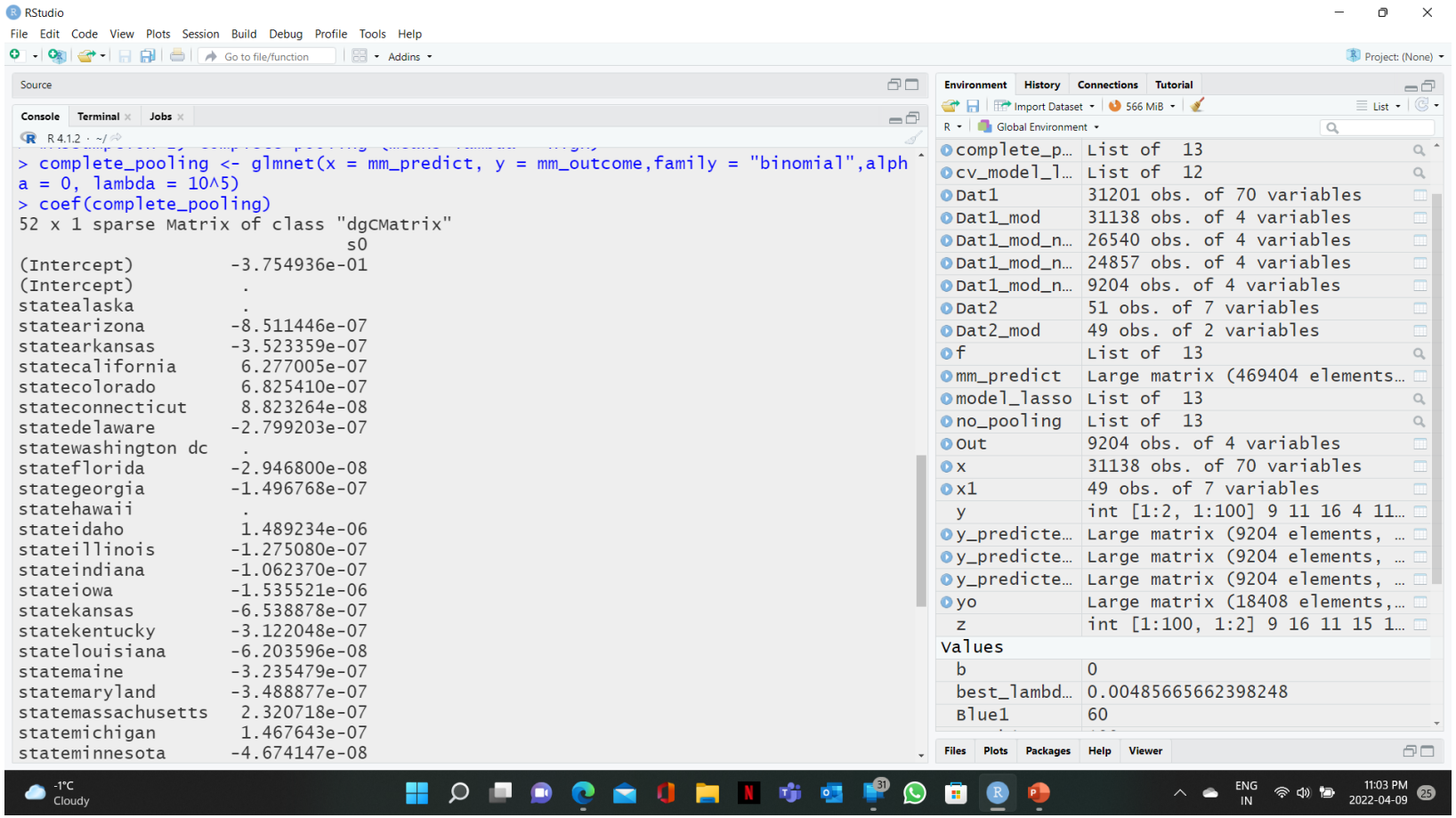
- #Assumption 3) Partial Pooling(Used lasso)

```
### will find value of lambda 1st using cross validation(k=10) and the will find glm output after applying best lambda
cv_model_lasso <- cv.glmnet(x = mm_predict, y = mm_outcome,family = "binomial", alpha = 1)
best_lambda_lasso <- cv_model_lasso$lambda.min
best_lambda_lasso
## best_lambda_lasso =0.005330175
model_lasso <- glmnet(x = mm_predict, y = mm_outcome,family = "binomial", alpha = 1, lambda =best_lambda_lasso)
coef(model_lasso )
```

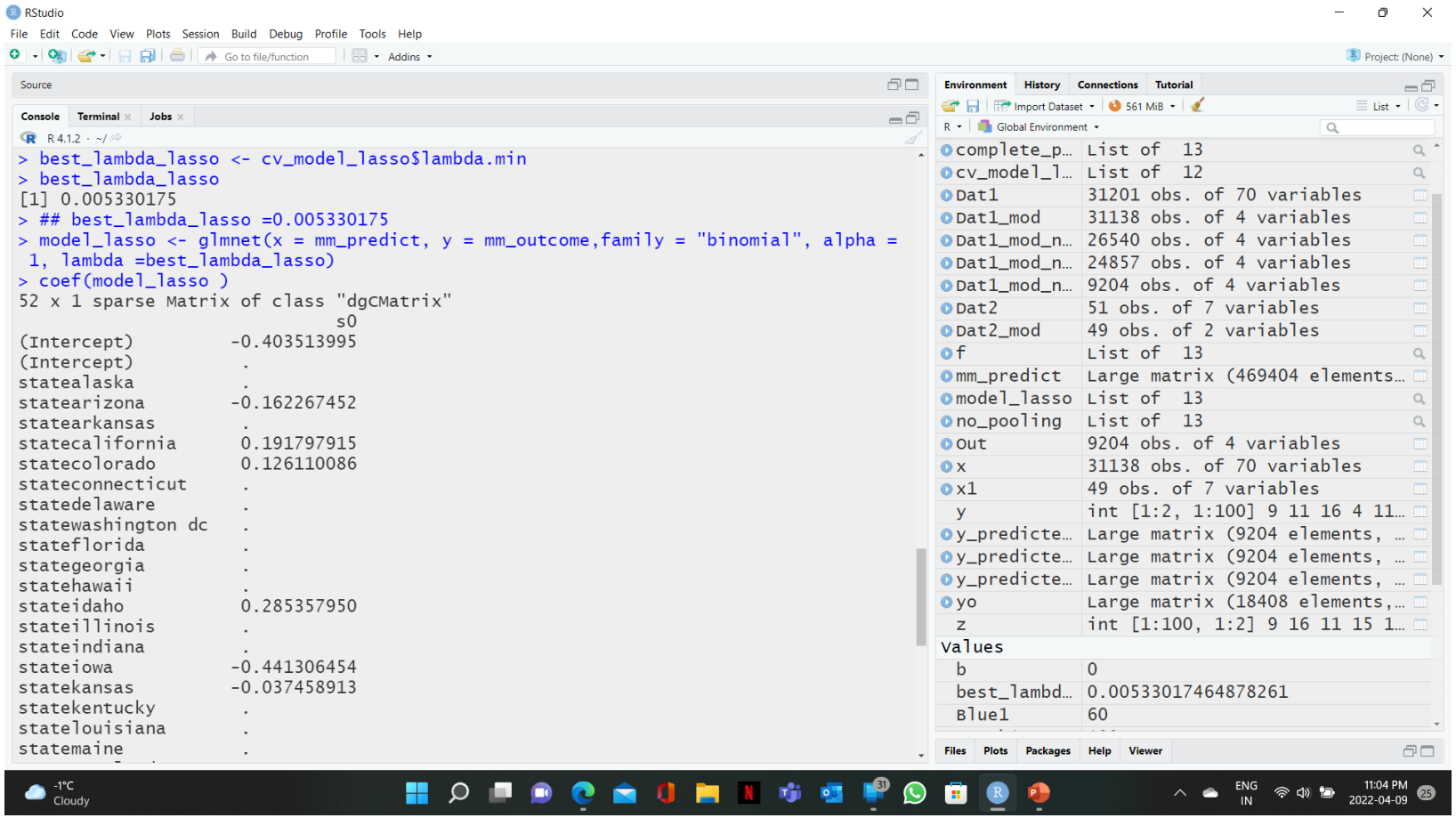
Assumption : 1 No pooling



Assumption :2) Complete pooling



Assumption :3) partial pooling lasso



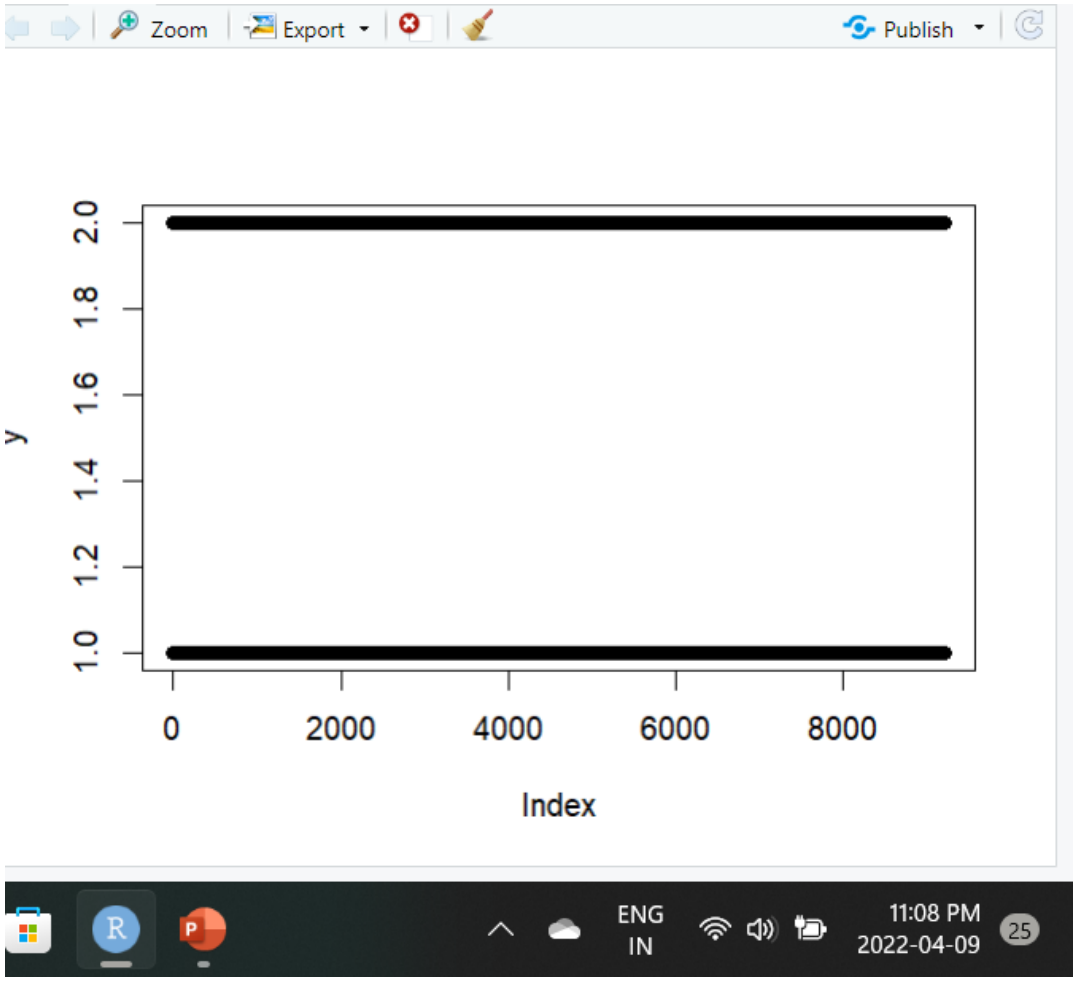
Prediction

```
## prediction using mm_predict as data now
y_predicted_lasso <- predict(model_lasso, s = best_lambda_lasso, newx = mm_predict)
y_predicted_lasso

y_predicted_no_pooling <- predict(no_pooling, s = 0, newx = mm_predict)
y_predicted_complete_pooling <- predict(complete_pooling, s = 0, newx = mm_predict)
```

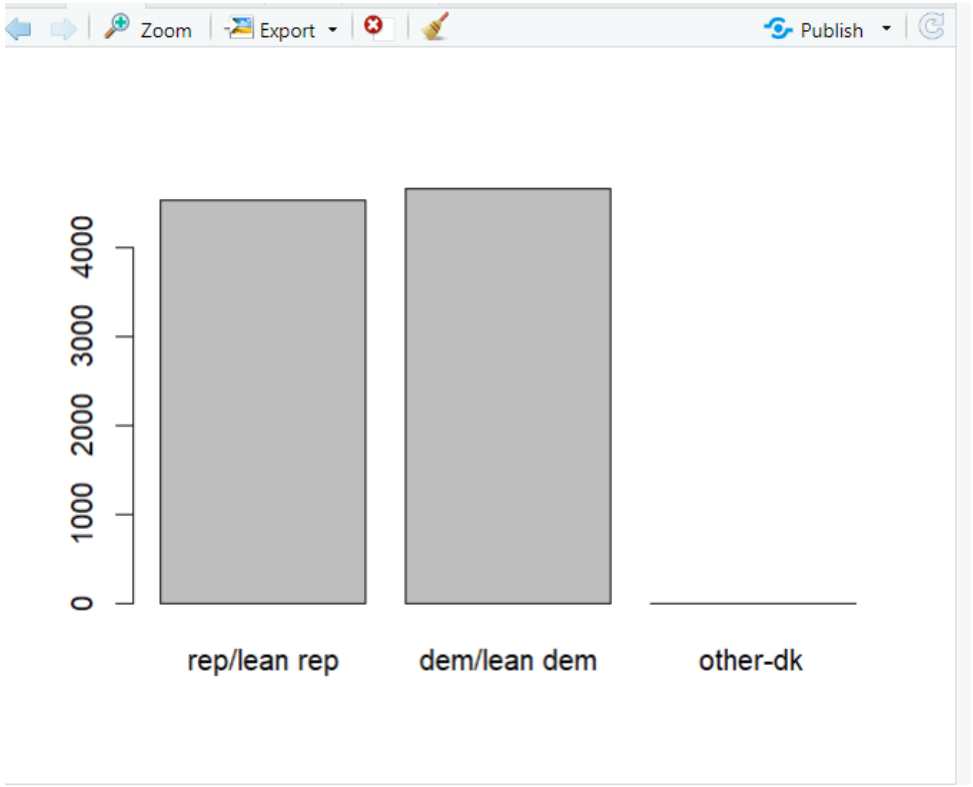
Plots

```
library(glmnet)
#fit = glmnet(as.matrix(y_predicted_lasso[1],model_lasso[-1]))
#plot(fit, xvar='lambda')
plot(Out$heat2)
plot(model_lasso, type = "b")
plot(y_predicted_lasso)

plot(no_pooling, type = "b")
plot(y_predicted_no_pooling)
plot(complete_pooling, type = "b")
plot(y_predicted_complete_pooling)
```

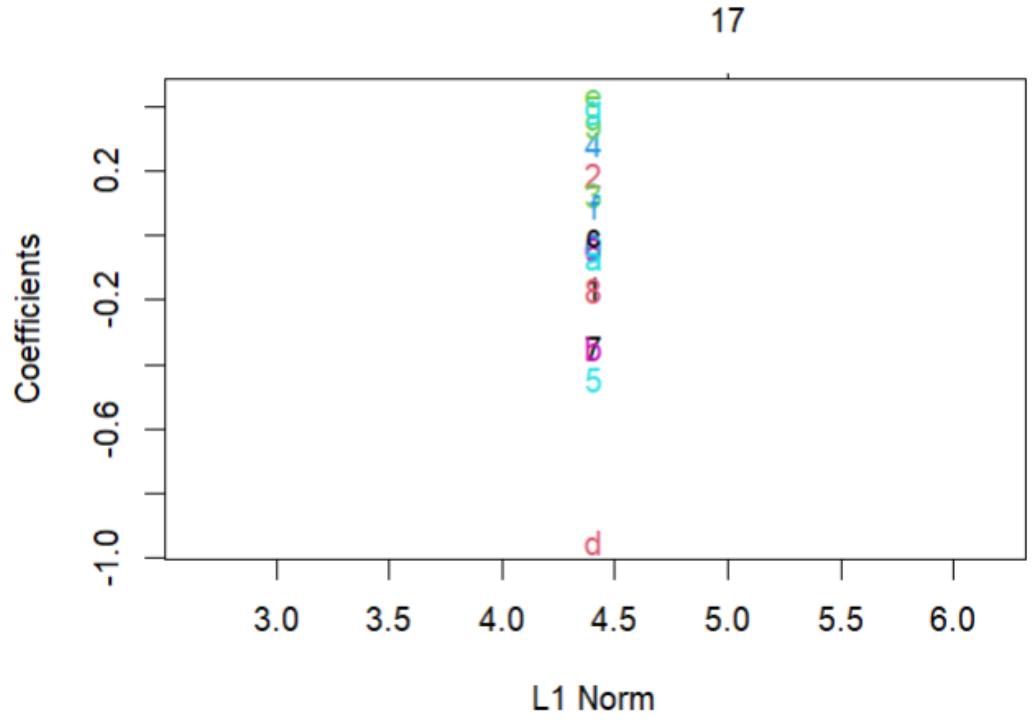
plot(Out$heat2,fitted.values(no_pooling))



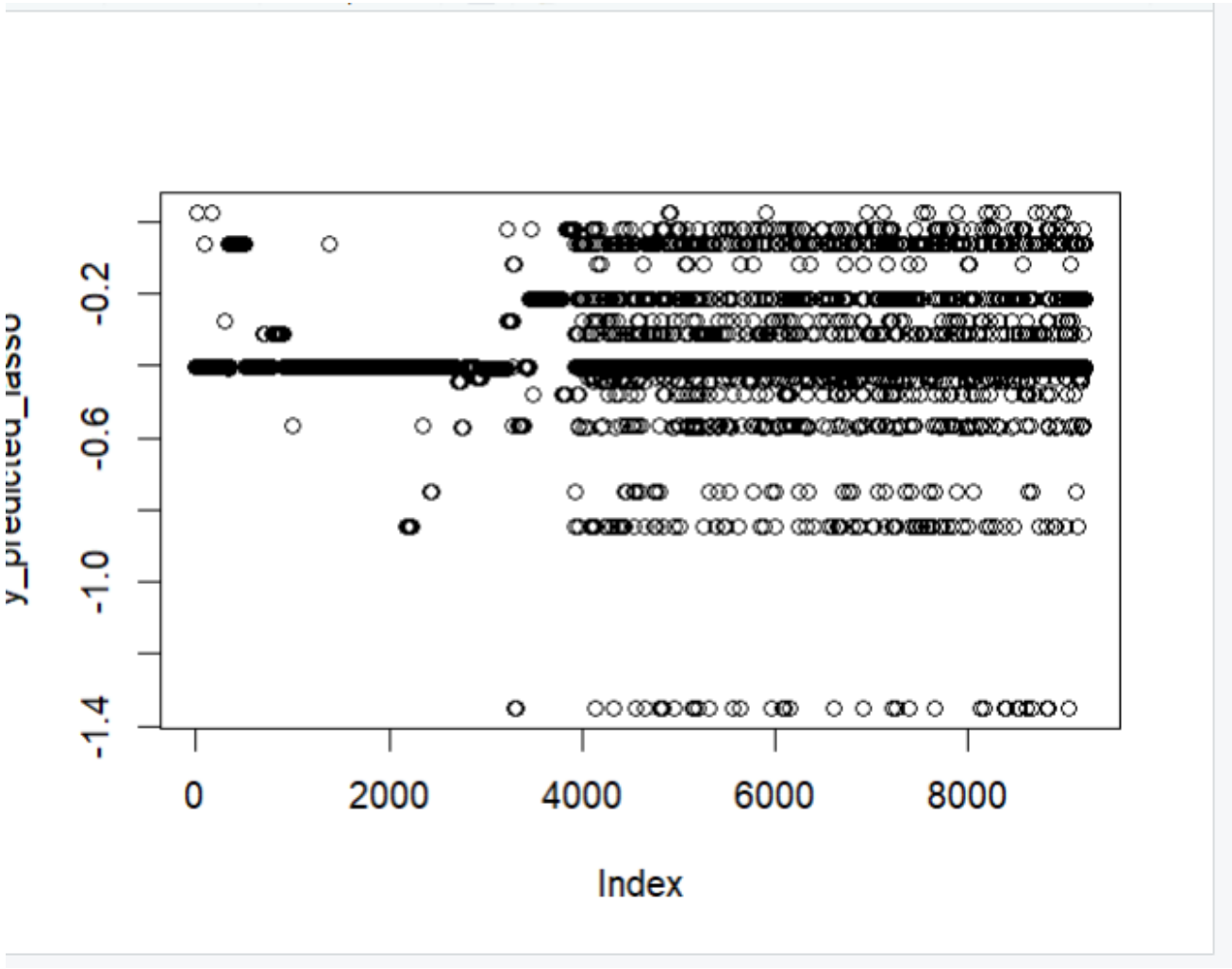
plot(Out$heat2)



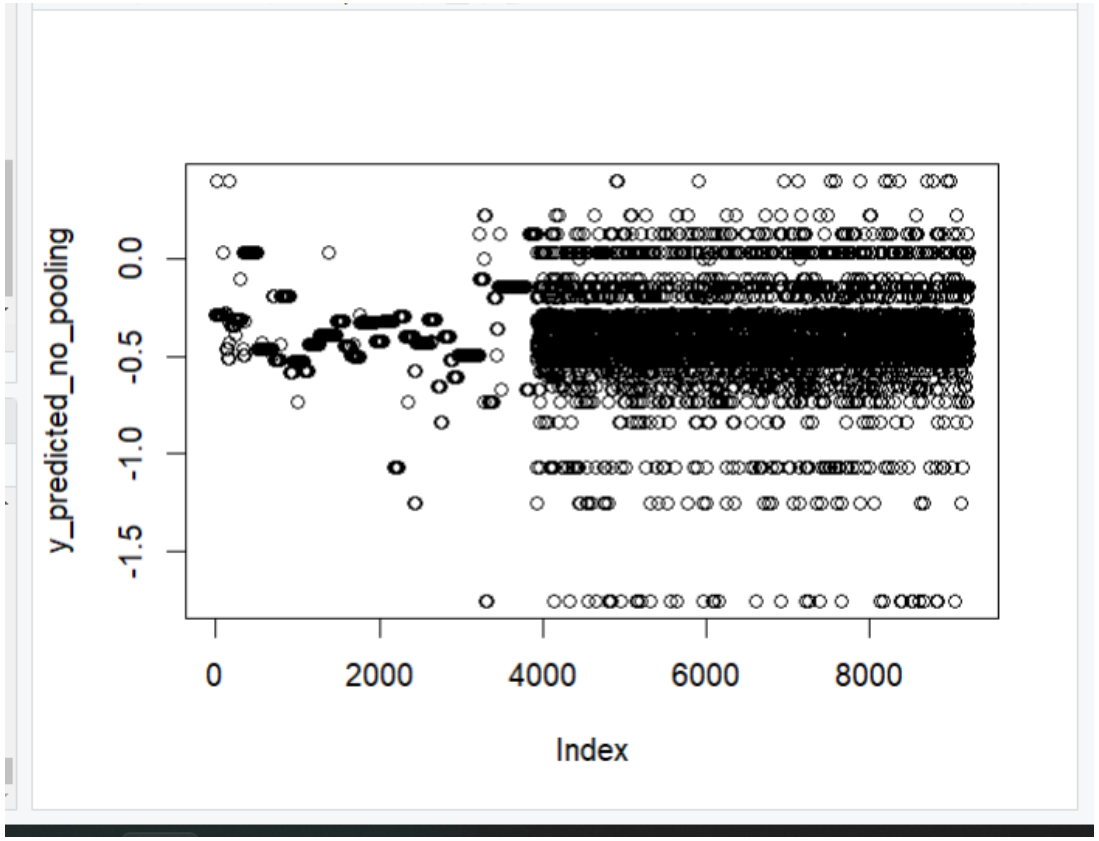
Plot of lasso



Graph of predicted lasso



Graph of predicted no pooling



Grapg of predicted complete Pooling