

MSCI - 718

Juhi Vasudev Bachani

20979706

Section 1: R Programming Questions: Q-2

```
library(MASS)
library(fitdistrplus)
#### Just for reference given 0 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 x Jobs x
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
[28,]	16	4

0°C Cloudy

Windows taskbar icons: Start, Search, Task View, Teams, Edge, Mail, OneDrive, File Explorer, Notepad, Visual Studio Code, Outlook, and a notification bubble with '31'.

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 x Jobs x
R 4.1.2 · ~/
# Combining z matrix and likelihood in one data
> 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

Windows taskbar icons: Start, Search, Task View, Teams, Edge, Mail, OneDrive, File Explorer, Notepad, Teams, Outlook.

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
> m1 <- f[which(likelihood == max(likelihood)), ]
> m1
      Nb Ng Likelihood
[1,]  7 13  4.229268
[2,]  7 13  4.229268
[3,]  7 13  4.229268
> |
```



Project (MScF - 718)

Section: I

(Q-1) 2)

Blue fish = 60
Gold fish = 40 } Catch 1 = 100

No } 20 } Catch = 2
New tagged

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

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

`z <- t(y)` # Converting row to column

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 N_b, N_g using maximum
likelihood.

II
(a) N_b and N_g ~~was~~ are given

hypothesis 1: $N_b = 280$ $N_g = 220$

hypothesis 2: $N_b = 290$, $N_g = 210$

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
```

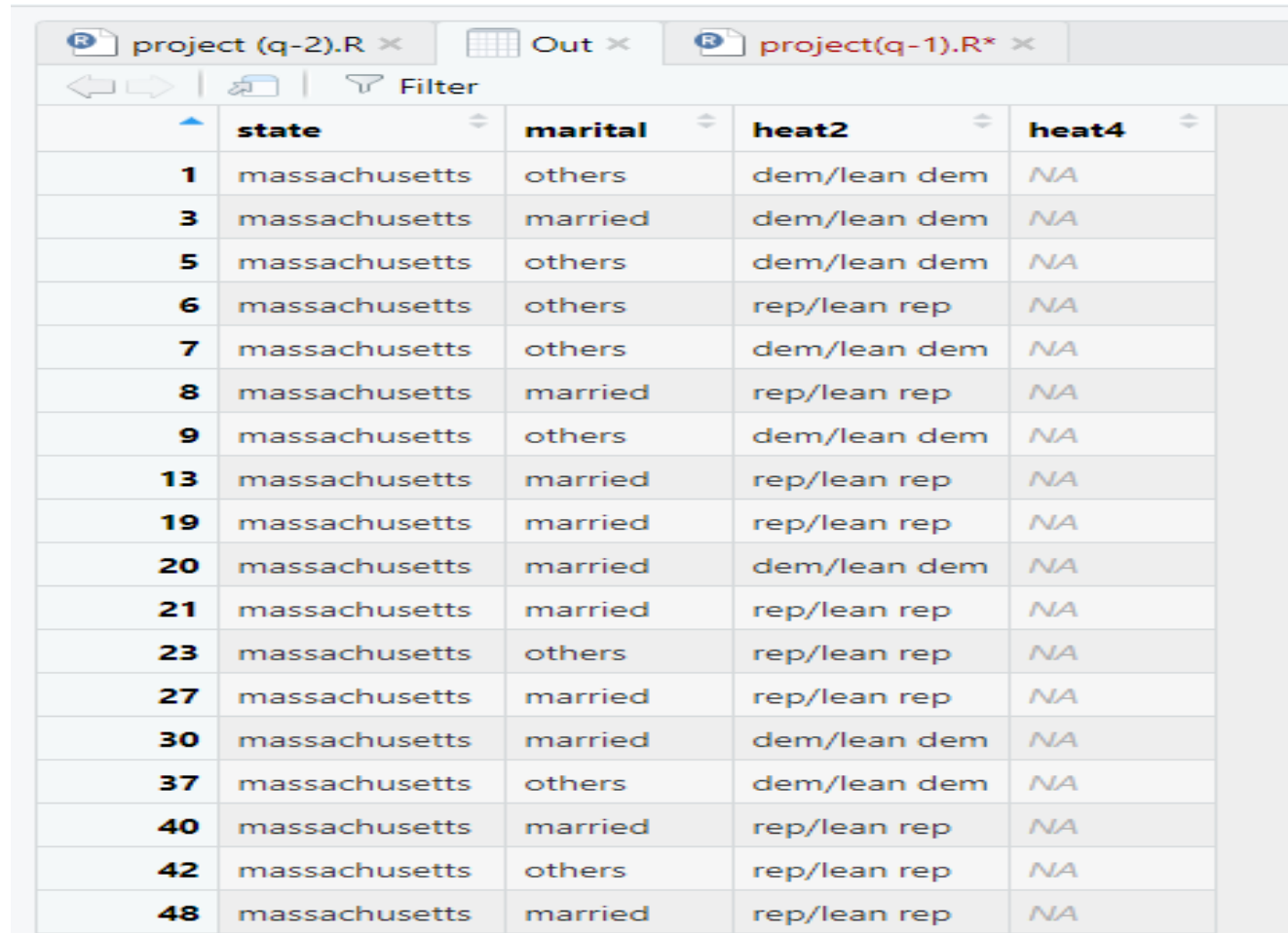
- #-----a-4) changing all values of marital column to "others" except "married"

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

```
Out
```

```
View(Out)
```

Output after
a-4 (out)



The screenshot shows an RStudio window with a data table. The table has 5 columns: an index column, 'state', 'marital', 'heat2', and 'heat4'. The data is filtered to show only rows where 'state' is 'massachusetts'. The 'heat4' column contains 'NA' for all rows.

	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)

```
> x1 <- subset(Dat2, state != c("Maine",  
> 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
```

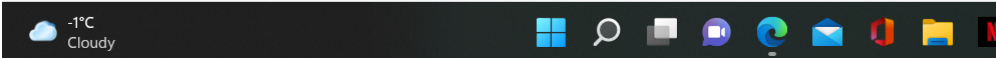
```
f <- glmnet(x = mm_predict, y = mm_outcome, family = "binomial" )
```


Result after applying glmnet

```
Console | terminal x | Jobs x
R 4.1.2 · ~/
> #applied glmnet using binomial regression
> f <- glmnet(x = mm_predict, y = mm_outcome, family = "binomial" )
> f

call: glmnet(x = mm_predict, y = mm_outcome, family = "binomial")

  Df %Dev   Lambda
1  0 0.00 0.0236200
2  2 0.04 0.0215200
3  2 0.09 0.0196100
4  5 0.13 0.0178600
5  5 0.22 0.0162800
6  5 0.29 0.0148300
7  5 0.35 0.0135100
8  5 0.40 0.0123100
9  6 0.44 0.0112200
10 7 0.49 0.0102200
11 10 0.53 0.0093150
12 13 0.59 0.0084870
13 13 0.64 0.0077330
14 14 0.68 0.0070460
15 14 0.72 0.0064200
16 16 0.75 0.0058500
17 17 0.78 0.0053300
18 18 0.81 0.0048570
19 20 0.83 0.0044250
20 23 0.86 0.0040320
21 26 0.88 0.0036740
22 28 0.90 0.0033480
23 30 0.92 0.0030500
```



- #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 $\lambda = \text{high}$)

```
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 λ 1st using cross validation($k=10$) and then will find glm output after applying best λ

```
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

The screenshot displays the RStudio interface. The console shows the execution of a `glmnet` model with `alpha = 0` and `lambda = 0`, resulting in a `dgCMatrix` of size 52 x 1. The coefficients for the model are listed, including the intercept and coefficients for various states and variables. The environment pane on the right shows the objects created during the session, including `f`, `mm_predict`, `model_lasso`, `no_pooling`, `out`, `x`, `x1`, `y`, `y_predicte...`, `yo`, and `z`.

```
> no_pooling <- glmnet(x = mm_predict, y = mm_outcome, family = "binomial", alpha = 0, lambda = 0)
> coef(no_pooling)
52 x 1 sparse Matrix of class "dgCMatrix"
      s0
(Intercept)      -0.3167475193
(Intercept)      .
statealaska      .
statearizona     -0.4186897617
statearkansas    -0.2040720131
statecalifornia  0.1757766156
statecolorado    0.2156817241
stateconnecticut -0.0215649097
statedelaware    -0.1745388689
statewashington dc .
stateflorida     -0.0691968003
stategeorgia     -0.1180027169
statehawaii      .
stateidaho       0.5409471276
stateillinois    -0.1087716439
stateindiana     -0.1007462983
stateiowa        -0.7501510314
statekansas      -0.3340883916
statekentucky    -0.1865983357
statelouisiana   -0.0831391032
statemaine       -0.1932116913
statemaryland    -0.2015403564
statemassachusetts 0.0351432035
statemichigan    0.0004011429
stateminnesota   -0.0769539914
```

Environment

Object	Description
f	List of 13
mm_predict	Large matrix (469404 elements...)
model_lasso	List of 13
no_pooling	List of 13
out	9204 obs. of 4 variables
x	31138 obs. of 70 variables
x1	49 obs. of 7 variables
y	int [1:2, 1:100] 9 11 16 4 11...
y_predicte...	Large matrix (9204 elements, ...)
y_predicte...	Large matrix (9204 elements, ...)
y_predicte...	Large matrix (9204 elements, ...)
yo	Large matrix (18408 elements, ...)
z	int [1:100, 1:2] 9 16 11 15 1...

Values

Variable	Value
b	0
best_lambda...	0.00485665662398248
Blue1	60
Catch1	100
g	1
Gold1	40
i	100L
Lake	num [1:2] 0 1
likelihood	num [1:100] 2.65 3.35 1.83 2.5...
m	Factor w/ 3 levels "rep/lean r...
m1	Named num [1:3] 5 15 6.65
mm_outcome	chr [1:9204] "others" "married...

Assumption :2) Complete pooling

The screenshot displays the RStudio interface. The console shows the execution of the following R code:

```
> complete_pooling <- glmnet(x = mm_predict, y = mm_outcome, family = "binomial", alpha = 0, lambda = 10^5)
> coef(complete_pooling)
52 x 1 sparse Matrix of class "dgCMatrix"
      s0
(Intercept) -3.754936e-01
statealaska .
statearizona -8.511446e-07
statearkansas -3.523359e-07
statecalifornia 6.277005e-07
statecolorado 6.825410e-07
stateconnecticut 8.823264e-08
statedelaware -2.799203e-07
statewashington dc .
stateflorida -2.946800e-08
stategeorgia -1.496768e-07
statehawaii .
stateidaho 1.489234e-06
stateillinois -1.275080e-07
stateindiana -1.062370e-07
stateiowa -1.535521e-06
statekansas -6.538878e-07
statekentucky -3.122048e-07
statelouisiana -6.203596e-08
statemaine -3.235479e-07
statemaryland -3.488877e-07
statemassachusetts 2.320718e-07
statemichigan 1.467643e-07
stateminnesota -4.674147e-08
```

The Environment pane on the right shows the objects created during the execution:

Object	Description
complete_p...	List of 13
cv_model_l...	List of 12
Dat1	31201 obs. of 70 variables
Dat1_mod	31138 obs. of 4 variables
Dat1_mod_n...	26540 obs. of 4 variables
Dat1_mod_n...	24857 obs. of 4 variables
Dat1_mod_n...	9204 obs. of 4 variables
Dat2	51 obs. of 7 variables
Dat2_mod	49 obs. of 2 variables
f	List of 13
mm_predict	Large matrix (469404 elements...)
model_lasso	List of 13
no_pooling	List of 13
Out	9204 obs. of 4 variables
x	31138 obs. of 70 variables
x1	49 obs. of 7 variables
y	int [1:2, 1:100] 9 11 16 4 11...
y_predicte...	Large matrix (9204 elements, ...)
y_predicte...	Large matrix (9204 elements, ...)
y_predicte...	Large matrix (9204 elements, ...)
yo	Large matrix (18408 elements, ...)
z	int [1:100, 1:2] 9 16 11 15 1...

The Values pane shows the following values:

Variable	Value
b	0
best_lambda...	0.00485665662398248
Blue1	60

Assumption :3) partial pooling lasso

The screenshot displays the RStudio interface with the following components:

- Source:** Empty editor pane.
- Console:** Contains the following R code and output:

```
> best_lambda_lasso <- cv_model_lasso$lambda.min
> best_lambda_lasso
[1] 0.005330175
> ## 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)
52 x 1 sparse Matrix of class "dgCMatrix"
              s0
(Intercept) -0.403513995
(Intercept) .
statealaska .
statearizona -0.162267452
statearkansas .
statecalifornia 0.191797915
statecolorado 0.126110086
stateconnecticut .
statedelaware .
statewashington dc .
stateflorida .
stategeorgia .
statehawaii .
stateidaho 0.285357950
stateillinois .
stateindiana .
stateiowa -0.441306454
statekansas -0.037458913
statekentucky .
statelouisiana .
statemaine .
```
- Environment:** Lists objects in the Global Environment:
 - complete_p...: List of 13
 - cv_model_l...: List of 12
 - Dat1: 31201 obs. of 70 variables
 - Dat1_mod: 31138 obs. of 4 variables
 - Dat1_mod_n...: 26540 obs. of 4 variables
 - Dat1_mod_n...: 24857 obs. of 4 variables
 - Dat1_mod_n...: 9204 obs. of 4 variables
 - Dat2: 51 obs. of 7 variables
 - Dat2_mod: 49 obs. of 2 variables
 - f: List of 13
 - mm_predict: Large matrix (469404 elements, ...)
 - model_lasso: List of 13
 - no_pooling: List of 13
 - Out: 9204 obs. of 4 variables
 - x: 31138 obs. of 70 variables
 - x1: 49 obs. of 7 variables
 - y: int [1:2, 1:100] 9 11 16 4 11...
 - y_predicte...: Large matrix (9204 elements, ...)
 - y_predicte...: Large matrix (9204 elements, ...)
 - y_predicte...: Large matrix (9204 elements, ...)
 - yo: Large matrix (18408 elements, ...)
 - z: int [1:100, 1:2] 9 16 11 15 1...
- Values:** A table showing the values of selected variables:

Variable	Value
b	0
best_lambda...	0.00533017464878261
Blue1	60
- Files, Plots, Packages, Help, Viewer:** Empty panes.

The Windows taskbar at the bottom shows the system clock as 11:04 PM on 2022-04-09.

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)
```

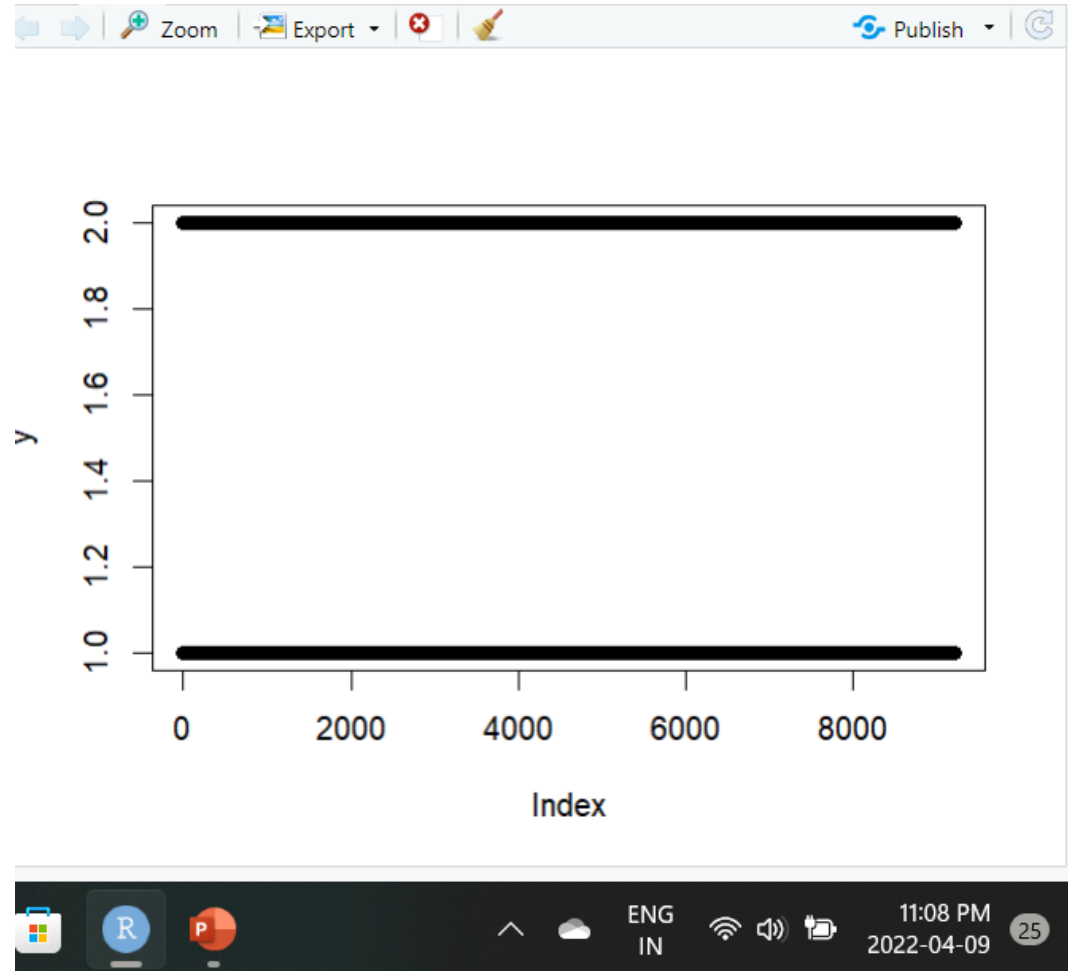
```
y_predicted_complete_pooling <- predict(complete_pooling, s = 0, newx = mm_predict)
```

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)
```

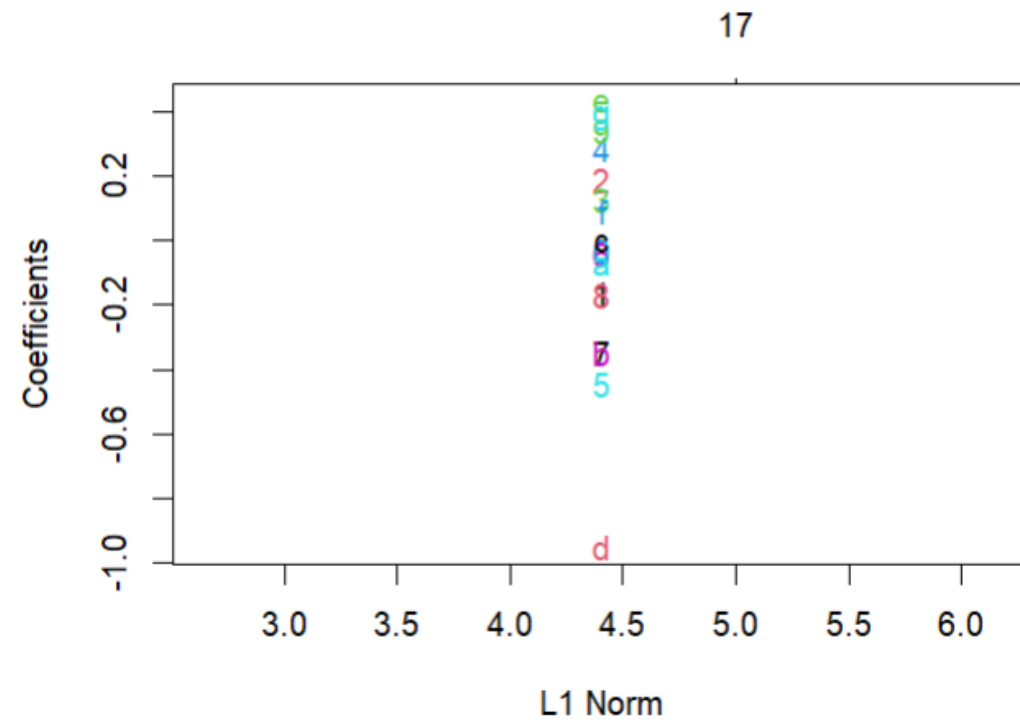
```
plot(Out$heat2,fitted.values(no_pooling))
```



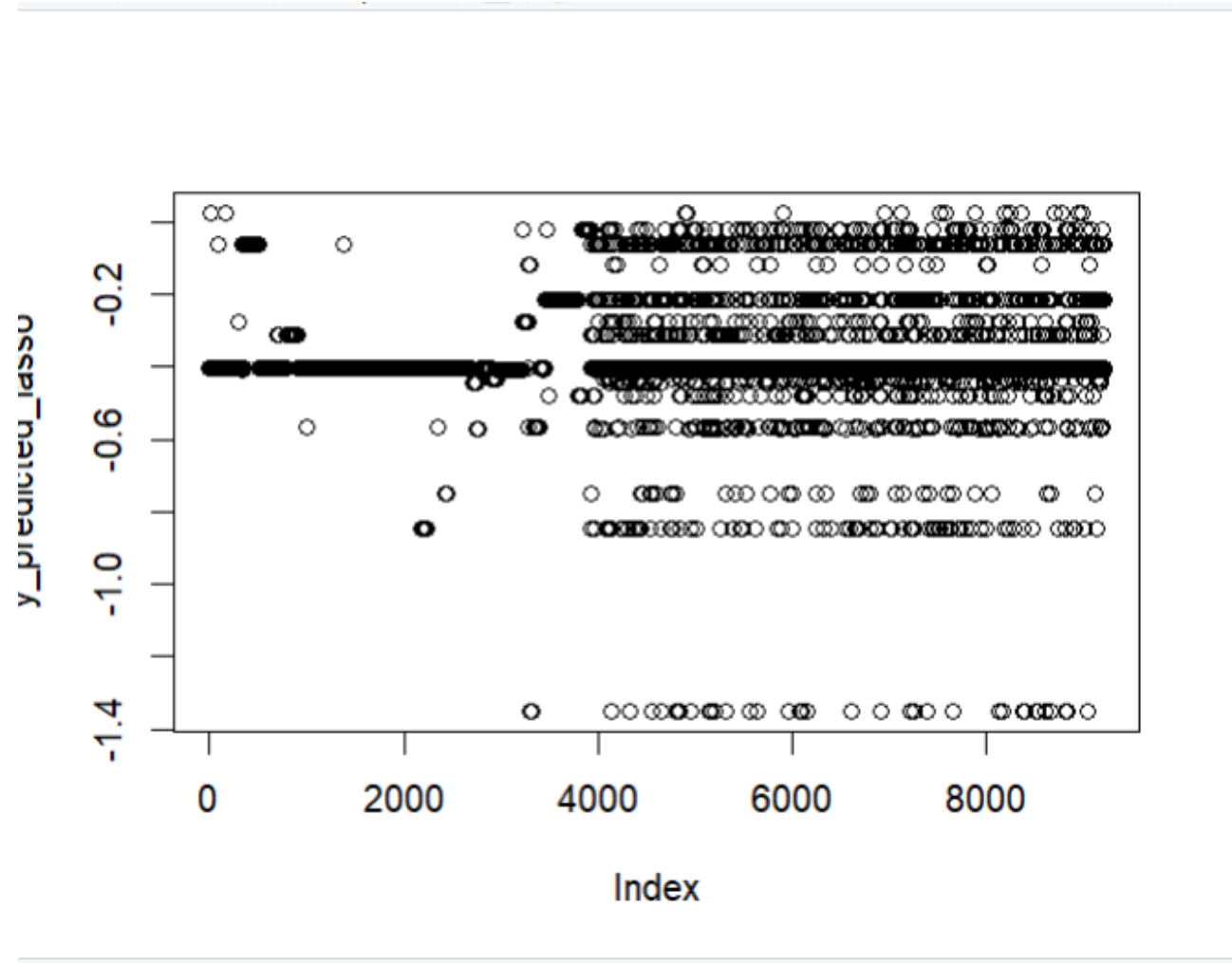

```
plot(Out$heat2)
```



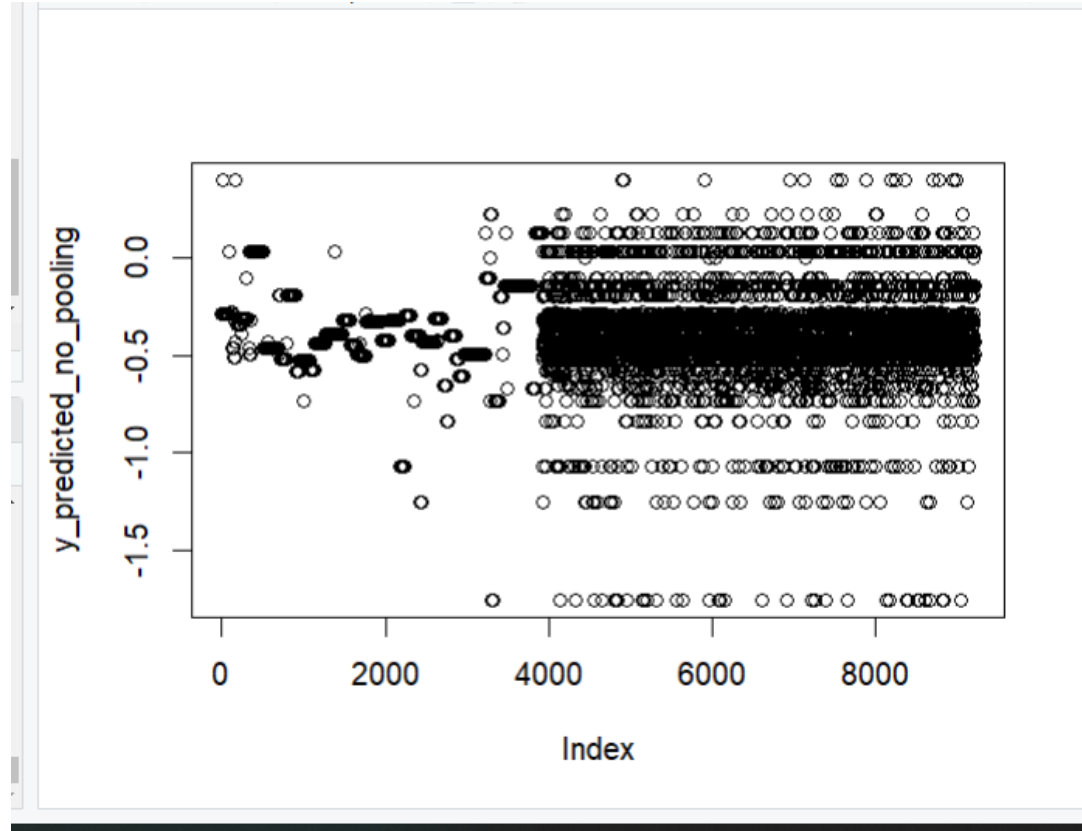
Plot of lasso



Graph of predicted lasso



Graph of predicted no pooling



Grapg of predicted complete Pooling

