Exam Statistics For Large Data

Anonymous

01/06/2023

Installing and Loading pagkages

In this section, install and load the relevant R packages

```
library(tidyverse)
library(lme4)
library(glmnet)


library(cluster)
library(caret)
```

Importing dataset

Running following code will load the dataset in the variable “ExamDataset”

```
ExamDataset <- read.csv("https://raw.githubusercontent.com/HuguesLortieForgues/Miscellaneous/master/ExamDataset.csv")
```

```
glimpse(ExamDataset)
```

```
## Rows: 500
## Columns: 36
## $ ID          <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,…
## $ Age         <dbl> 15.93979, 18.44808, 18.69428, 15.81369, 16.01864, 16.72467…
## $ YearGroup   <chr> "Grade12", "Grade14", "Grade14", "Grade12", "Grade12", "Gr…
## $ Q1_response <dbl> 150.64996, 78.52942, 61.94825, 187.03611, 166.07069, 33.98…
## $ Q1_1        <dbl> 12.758097, 14.079290, 21.234833, 15.282034, 15.517151, 21.…
## $ Q1_2        <dbl> 17.208631, 12.592429, 11.025206, 19.107140, 18.004245, 8.9…
## $ Q2_1        <int> 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0…
## $ Q2_2        <dbl> 17.680304, 18.322072, 14.355007, 3.998646, 18.272713, 8.90…
## $ Q2_response <int> 7, 10, 5, 4, 12, 4, 16, 7, 21, 3, 14, 3, 21, 9, 22, 4, 14,…
## $ Q3_1        <chr> "AA", "AA", "AB", "AB", "AB", "AB", "AB", "AB", "AB", "AB"…
## $ Q3_response <int> 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
## $ Q4_1        <dbl> 19.218223, -3.676130, -3.657018, 16.680656, 18.812126, -7.…
## $ Q4_2        <dbl> -17.983439, -3.426916, -1.469901, -18.578450, -19.747334, …
## $ Q4_3        <dbl> 8.557360, -2.355115, -3.227566, 6.459236, 9.746332, 21.863…
## $ Q5_1        <dbl> -2.17279758, -1.52332777, 3.25530722, 0.01589249, -2.27783…
## $ Q5_2        <dbl> -2.1661282, -0.8596832, 2.8388019, 0.7962889, 1.4306456, 6…
## $ Q5_3        <dbl> -2.0248880, -0.3845413, 3.6669233, 1.4096003, 0.5232128, 4…
## $ Q5_4        <dbl> -1.94278260, -1.51245430, 1.45173409, 1.04324468, -2.39486…
## $ Q5_5        <dbl> -1.594850308, -2.712633751, 1.016118844, 1.105262542, -1.0…
## $ Q5_6        <dbl> -0.14677622, -4.08977291, 0.88834752, 1.75049929, -3.64432…
## $ Q6_response <dbl> -0.106812856, -1.116337529, -2.397631435, -0.759893848, 1.…
## $ Q6_1        <dbl> -0.56047565, -0.23017749, 1.55870831, 0.07050839, 0.129287…
## $ Q6_2        <dbl> -0.60189285, -0.99369859, 1.02678506, 0.75106130, -1.50916…
## $ Q6_3        <dbl> -0.99579872, -1.03995504, -0.01798024, -0.13217513, -2.549…
## $ Q6_4        <dbl> -0.820986697, -0.307257233, -0.902098009, 0.627068743, 1.1…
## $ Q6_5        <dbl> -0.51160372, 0.23693788, -0.54158917, 1.21922765, 0.174135…
## $ Q6_6        <dbl> -0.67880762, 0.57431274, -0.70451453, -0.53398406, 0.77438…
## $ Q6_7        <dbl> -0.15030748, -0.32775713, -1.44816529, -0.69728458, 2.5984…
## $ Q6_8        <dbl> 1.47833446, -1.40678672, -1.88397213, -0.27736623, 0.43042…
## $ Q6_9        <dbl> 0.1965498, 0.6501132, 0.6710042, -1.2841578, -2.0261096, 2…
## $ Q6_10       <dbl> 0.8343715, -0.6984039, 1.3092405, -0.9801776, 0.7479851, 1…
## $ Q7_2        <chr> "a", "a", "a", "a", "a", "a", "a", "a", "a", "a", "a", "a"…
## $ Q7_1        <dbl> 44.39524, 47.69823, 65.58708, 50.70508, 51.29288, 67.15065…
## $ Q8_response <int> 12, 28, 19, 83, 81, 10, 6, 6, 10, 29, 13, 20, 14, 32, 4, 1…
## $ Q8_1        <dbl> 0.57515504, 1.57661027, 0.81795384, 1.76603481, 1.88093457…
## $ Q8_2        <chr> "GroupB", "GroupB", "GroupC", "GroupC", "GroupC", "GroupC"…
```

Question 1

```
linModel1=lm(Q1_response ~ Q1_1 + Q1_2, data = ExamDataset)
summary(linModel1)
```

```
##
## Call:
## lm(formula = Q1_response ~ Q1_1 + Q1_2, data = ExamDataset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.907  -7.969  -3.110   4.323  65.105
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -109.4445     3.0130 -36.324  < 2e-16 ***
## Q1_1           0.4894     0.1346   3.636 0.000306 ***
## Q1_2          15.1572     0.1395 108.666  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.6 on 497 degrees of freedom
## Multiple R-squared:  0.9596, Adjusted R-squared:  0.9595
## F-statistic:  5905 on 2 and 497 DF,  p-value: < 2.2e-16
```

Explain why it is necessary to interpret the results of this analysis with caution.

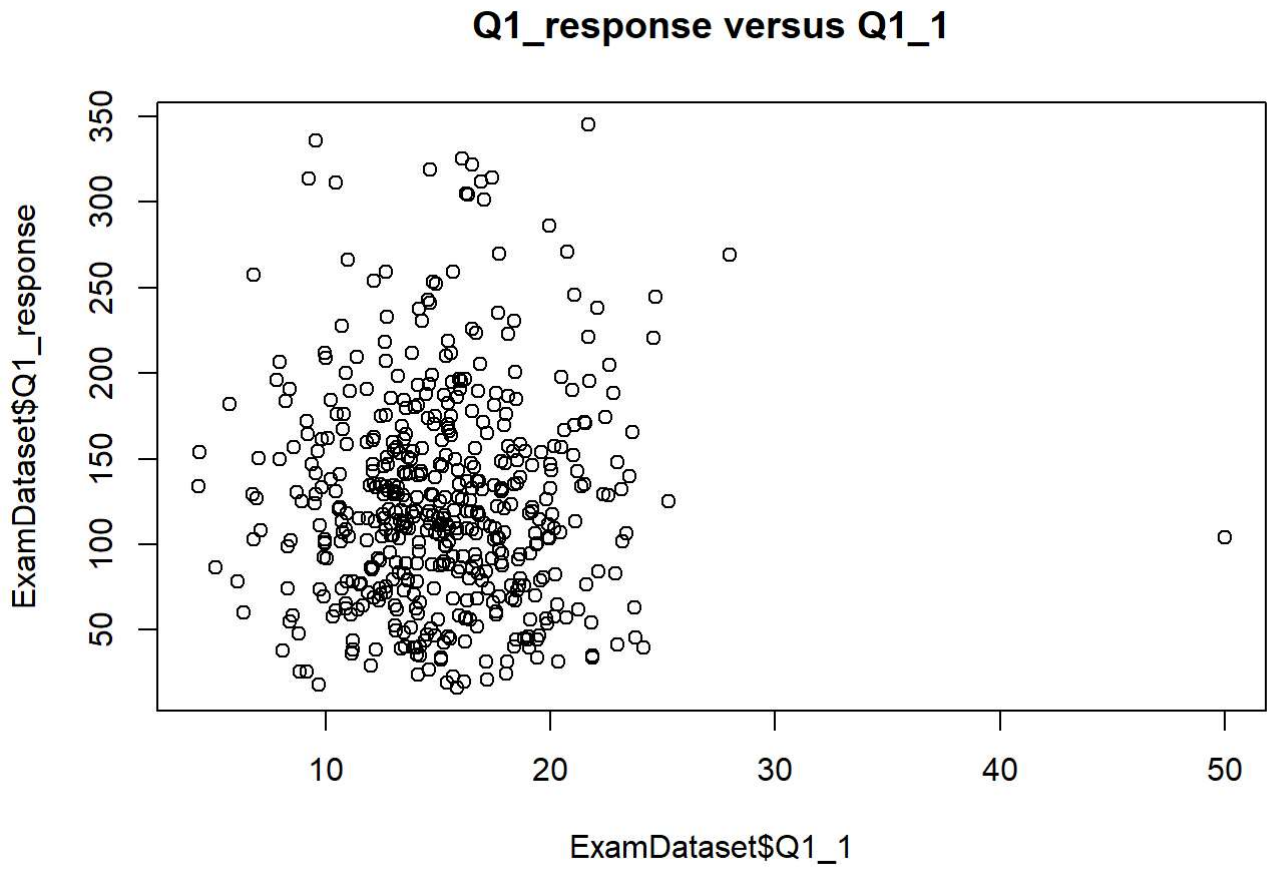
By interpreting the results, I can know the direction of the relationship between the term and the response by looking at the sign of the coefficients; I can do summaries to questions like how much difference is there among groups and how much of a change do we see in a response for each unit change in an input; I can do inference like if the regression does explain the observed variance in the response variable; and I can even make a guess of error on predictions.

In this example, the adjusted R2 is 0.9595, so roughly 95.95% of the variance found in the response variable can be explained by the predictors and which means the regression does explain the observed variance in the response variable since 0.9595 is so close to 1.

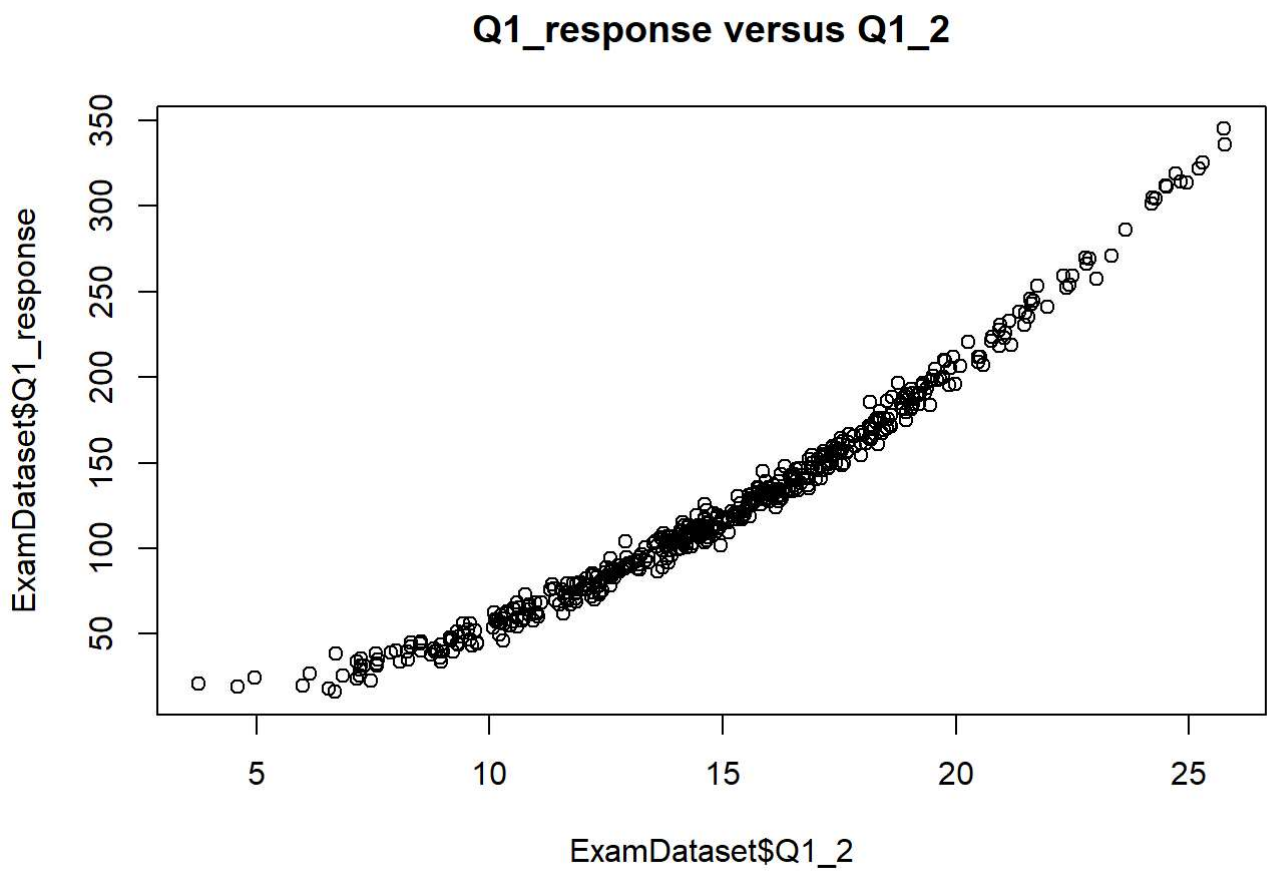
In this example, the Q1_response deviates from the regression line by approximately 12.6, on average. In other words, given that the mean of all Q1_response is -109.4445 and that the Residual Standard Error is 12.6 calculated with 497 degrees of freedom, I can say that the percentage error is (any prediction would still be off by) 11.51%.

Additionally, discuss what steps can be taken to increase the confidence in the information produced by the model, and how these steps can address potential sources of bias or limitations of the model.

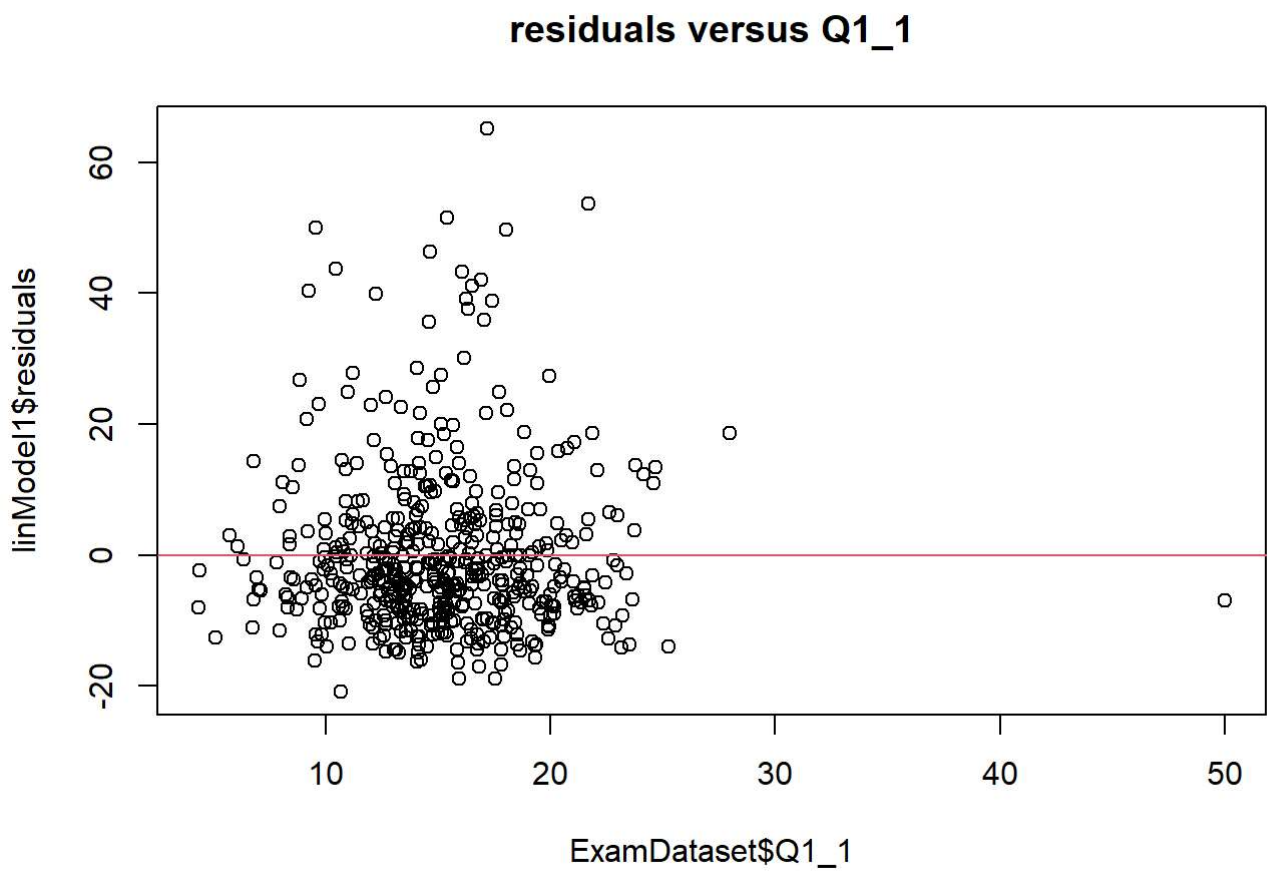
```
plot(ExamDataset$Q1_1,ExamDataset$Q1_response, main="Q1_response versus Q1_1")
```



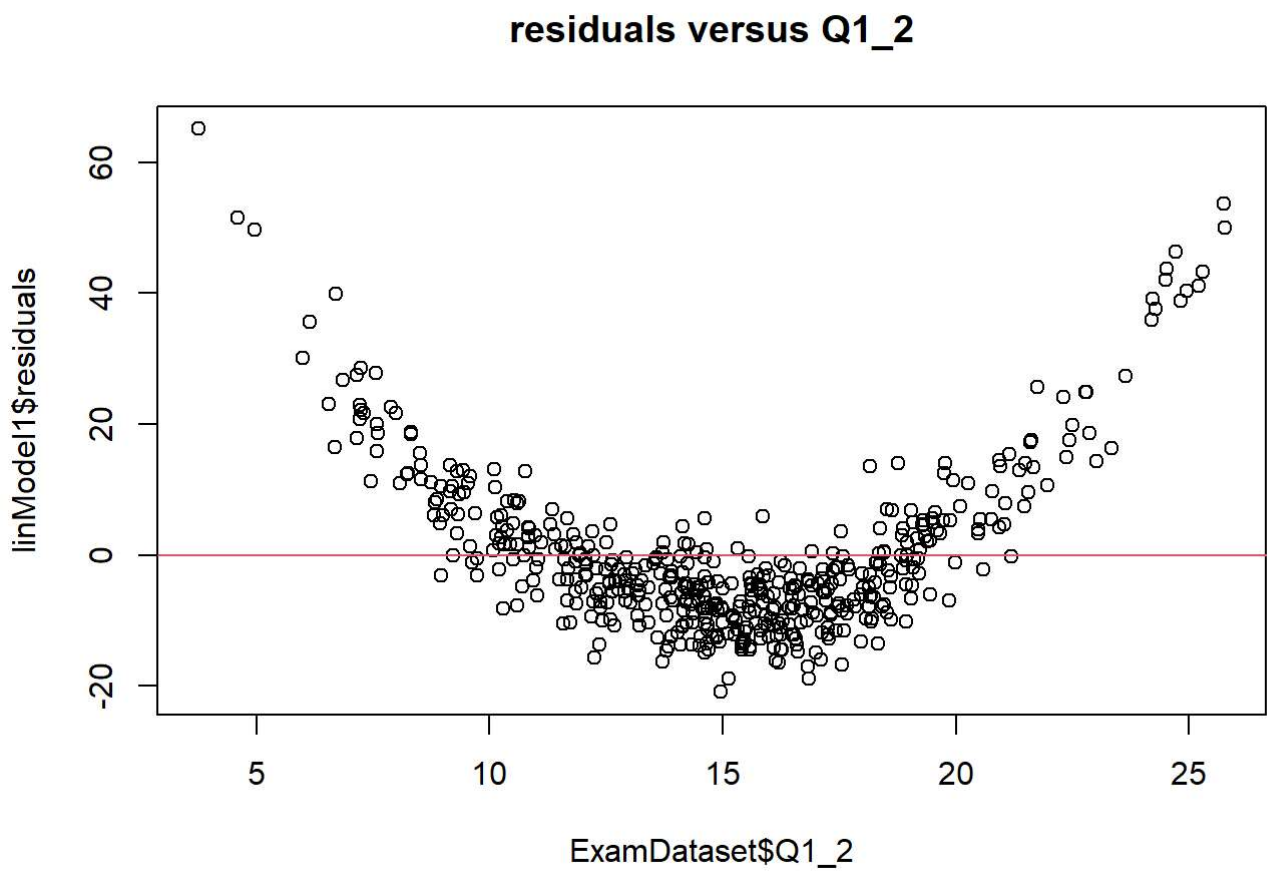

```
plot(ExamDataset$Q1_2,ExamDataset$Q1_response, main="Q1_response versus Q1_2")
```



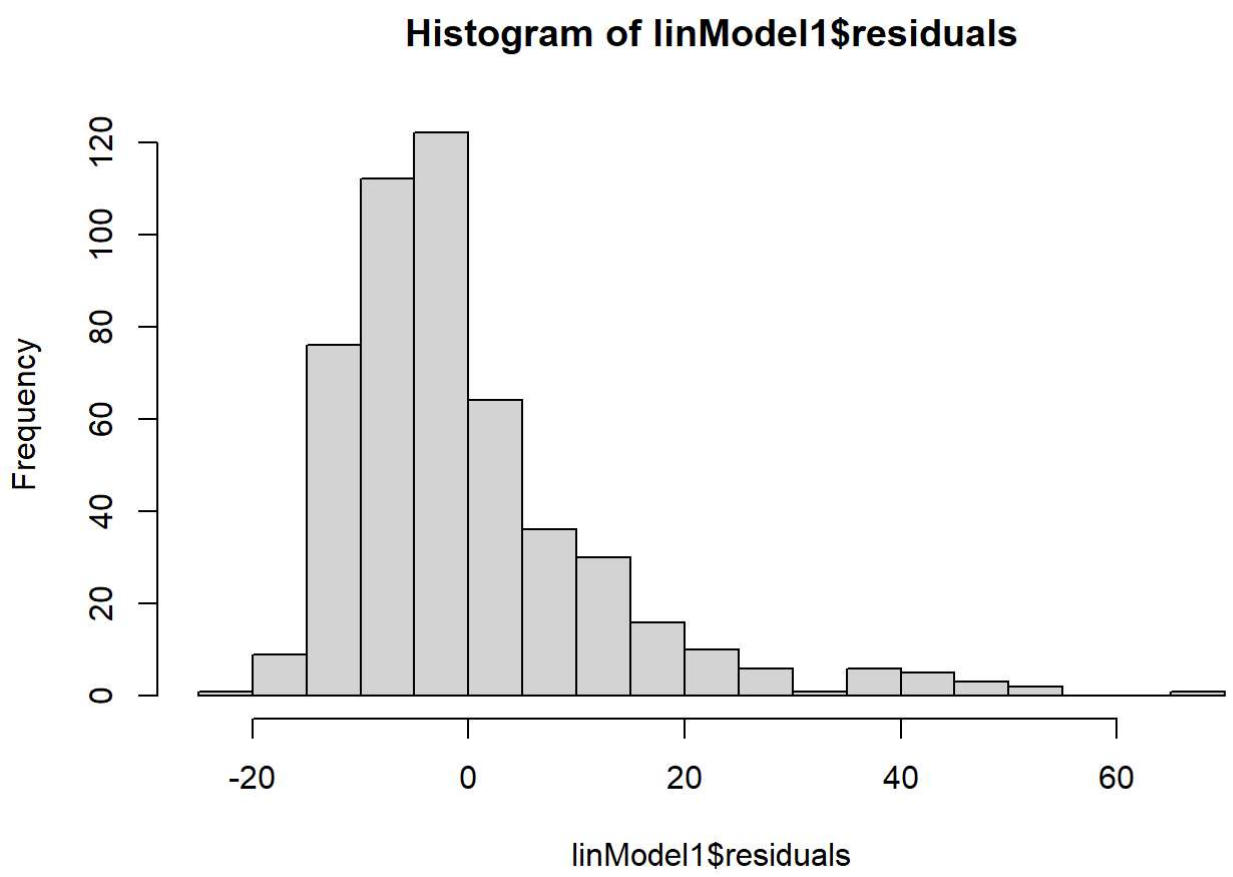

```
plot(ExamDataset$Q1_1,linModel1$residuals, main="residuals versus Q1_1")
abline(h=0,col=2)
```



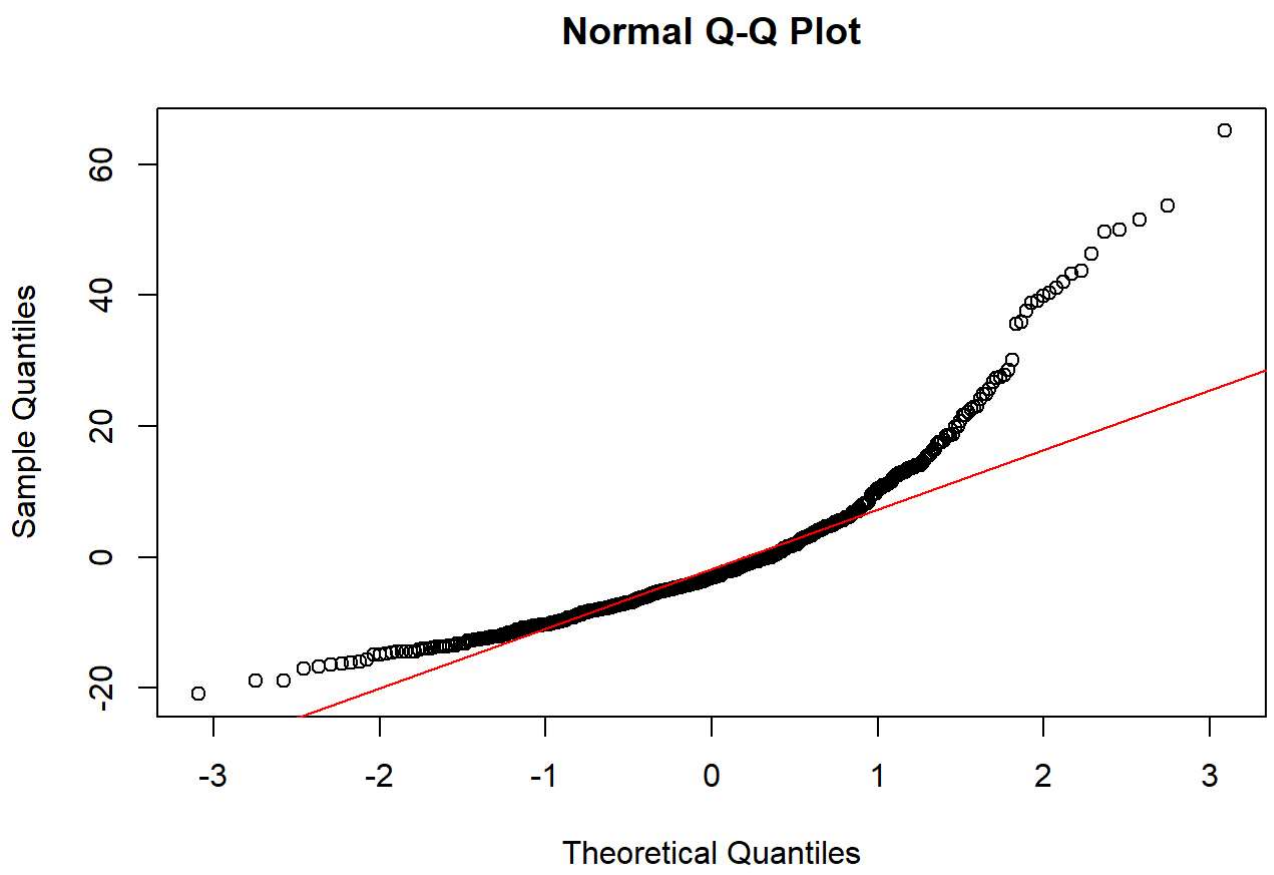

```
plot(ExamDataset$Q1_2,linModel1$residuals, main="residuals versus Q1_2")
abline(h=0,col=2)
```



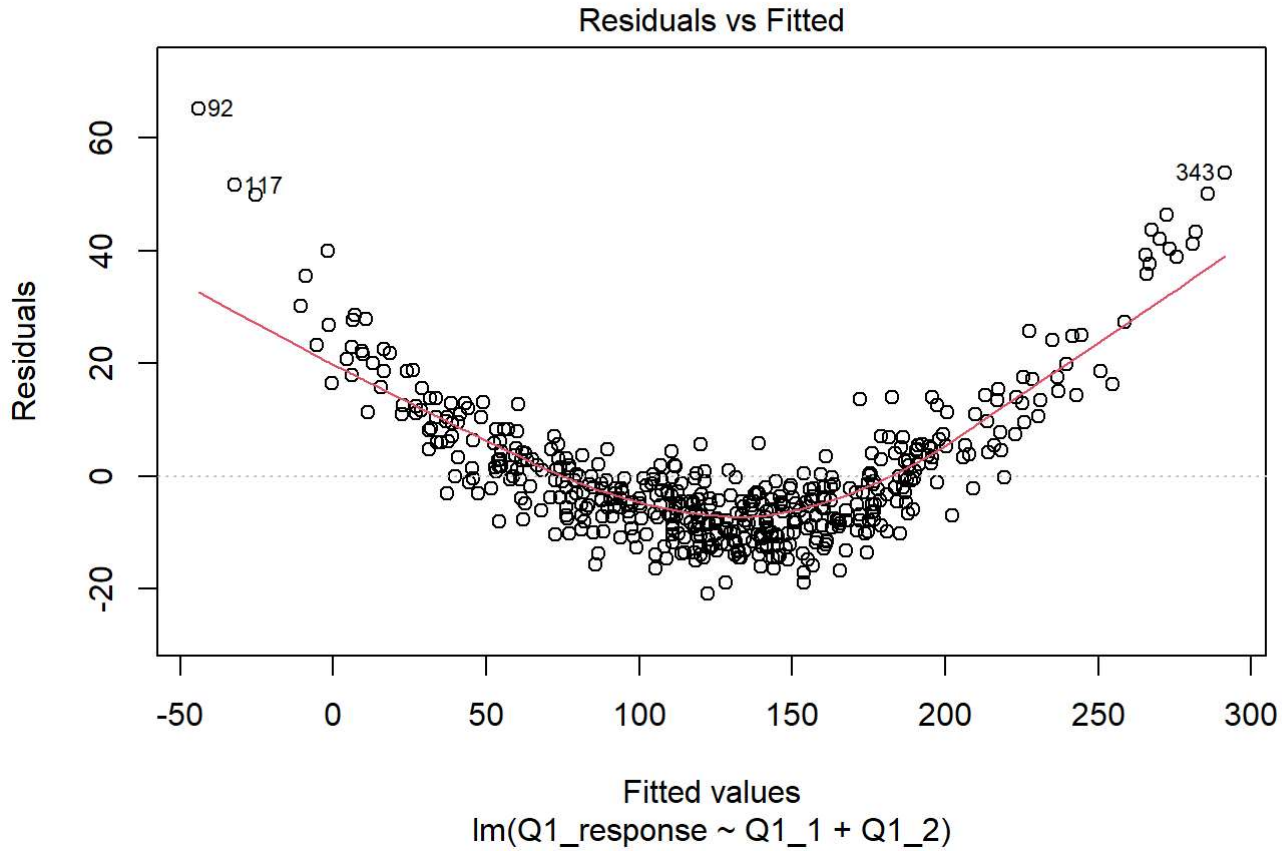

```
hist(linModel1$residuals,breaks=15)
```

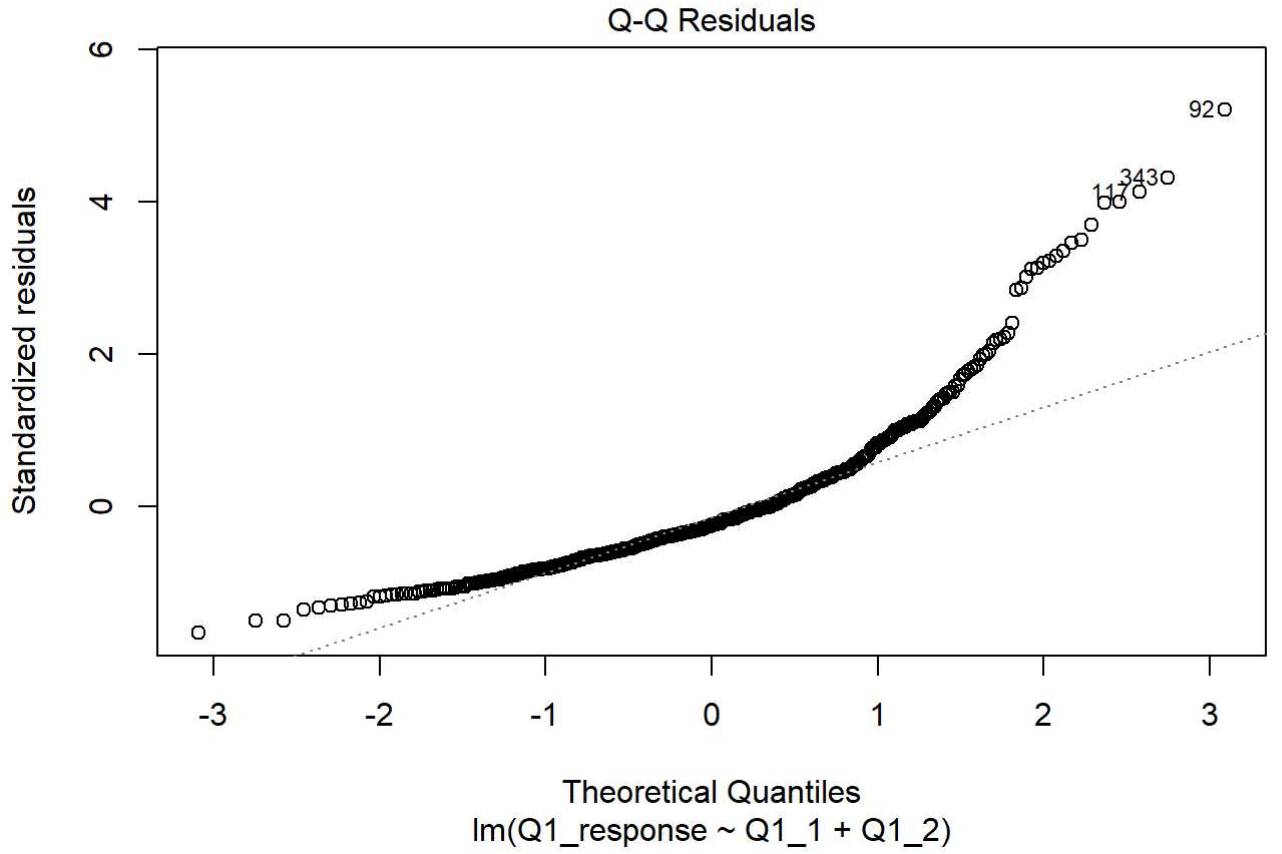


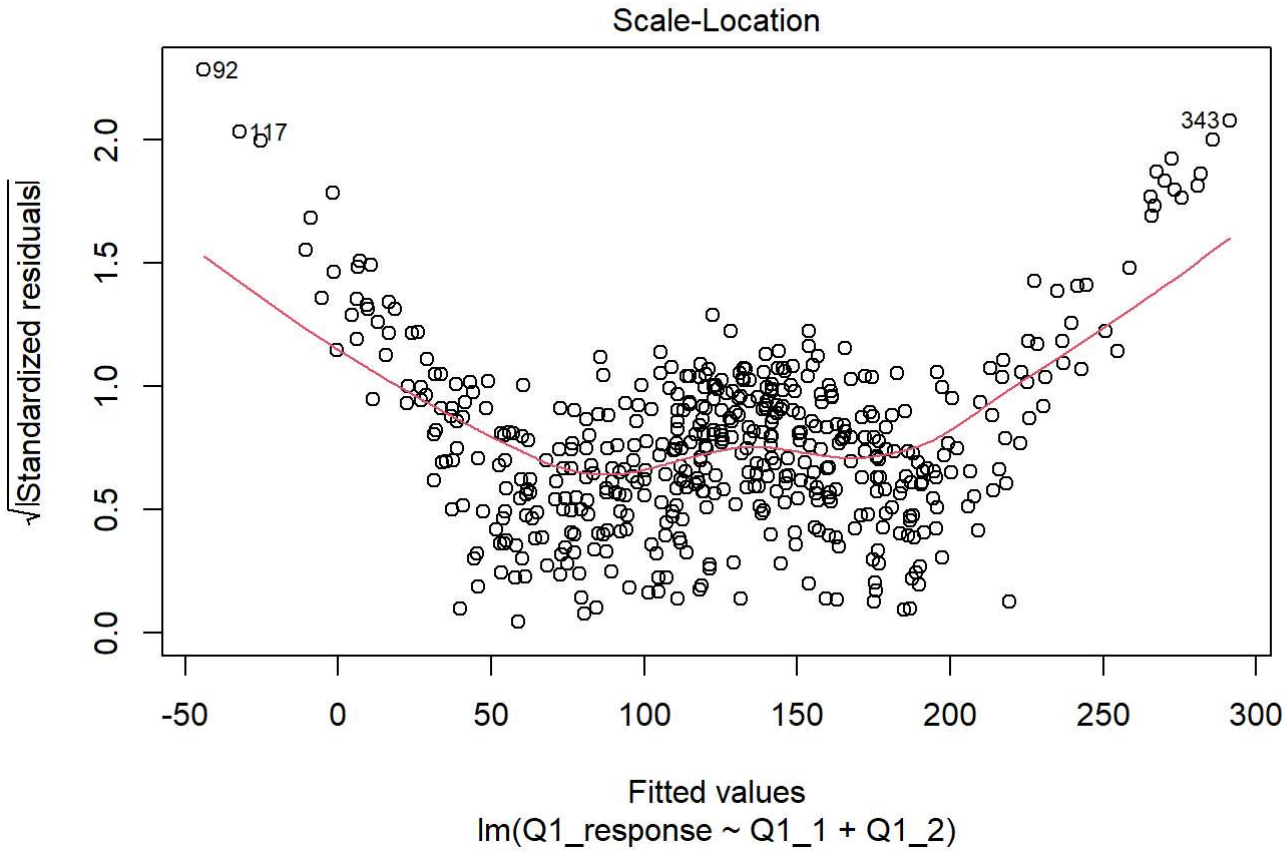

```
qqnorm(linModel1$residuals)
qqline(linModel1$residuals, col = "red")
```

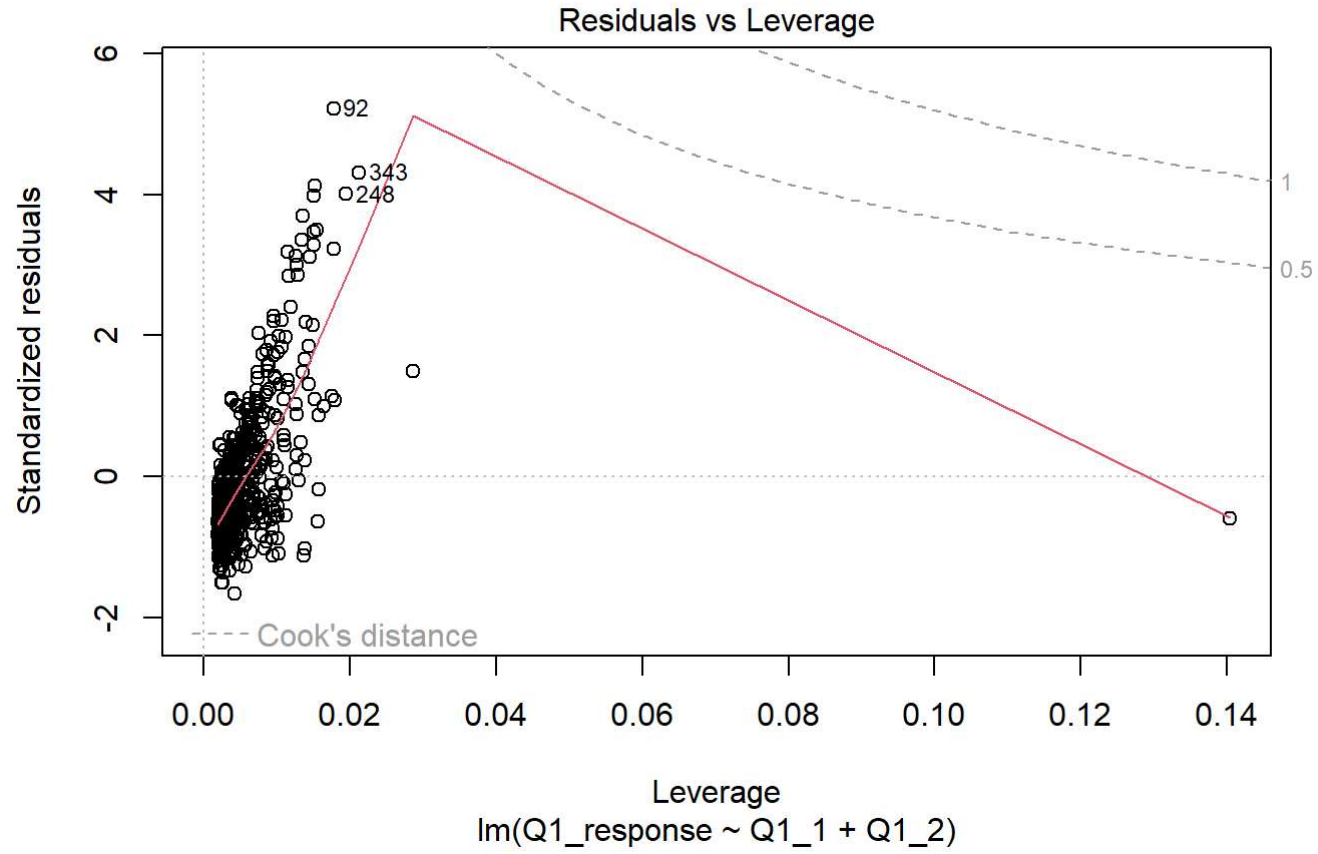



```
plot(linModel1)
```


Residuals vs Fitted
92
117
343
Residuals
60
40
20
0
-20
-50
0
50
100
150
200
250
300
Fitted values
lm(Q1_response ~ Q1_1 + Q1_2)


Q-Q Residuals
92
343
117
Standardized residuals
6
4
2
0
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(Q1_response ~ Q1_1 + Q1_2)


Scale-Location
92
117
343
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
-50
0
50
100
150
200
250
300
Fitted values
lm(Q1_response ~ Q1_1 + Q1_2)


Residuals vs Leverage
92
343
248
Standardized residuals
6
4
2
0
-2
1
0.5
Cook's distance
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
Leverage
lm(Q1_response ~ Q1_1 + Q1_2)

So the first linear model I have built is: Q1_response ~ Normal(−109.4 + 0.4894 × Q1_1 + 15.1572 × Q1_2).

looking at the scatterplots of “Q1_response versus Q1_1” and “Q1_response versus Q1_2”, I see “Q1_response versus Q1_2” is linear and the data in “Q1_response versus Q1_1” is more shrink to the center, and from “residuals vs fitted” also showed symmetric, and the “qqplot” is roughly straight, which increases the confidence of the model.

However, the limitations are also clear since “Q1_response versus Q1_1” has much more data under red line, “Q1_response versus Q1_2” is not even, the histogram of residual plot is not normally distributed and in “qqplot” the red line is not so fitting the main.

Question 2

The response variable for a linear model to predict should be continuous to avoid problems. However, Q2_response is integer and not continuous.

The problem is likely to happen as the predictor is added to the model, the response starts near zero and then diffusing outward. Hence, the relationship between the predictor and the response may not be linear. Such response data accepts only positive integer values, definitely no negative and not continuous then.

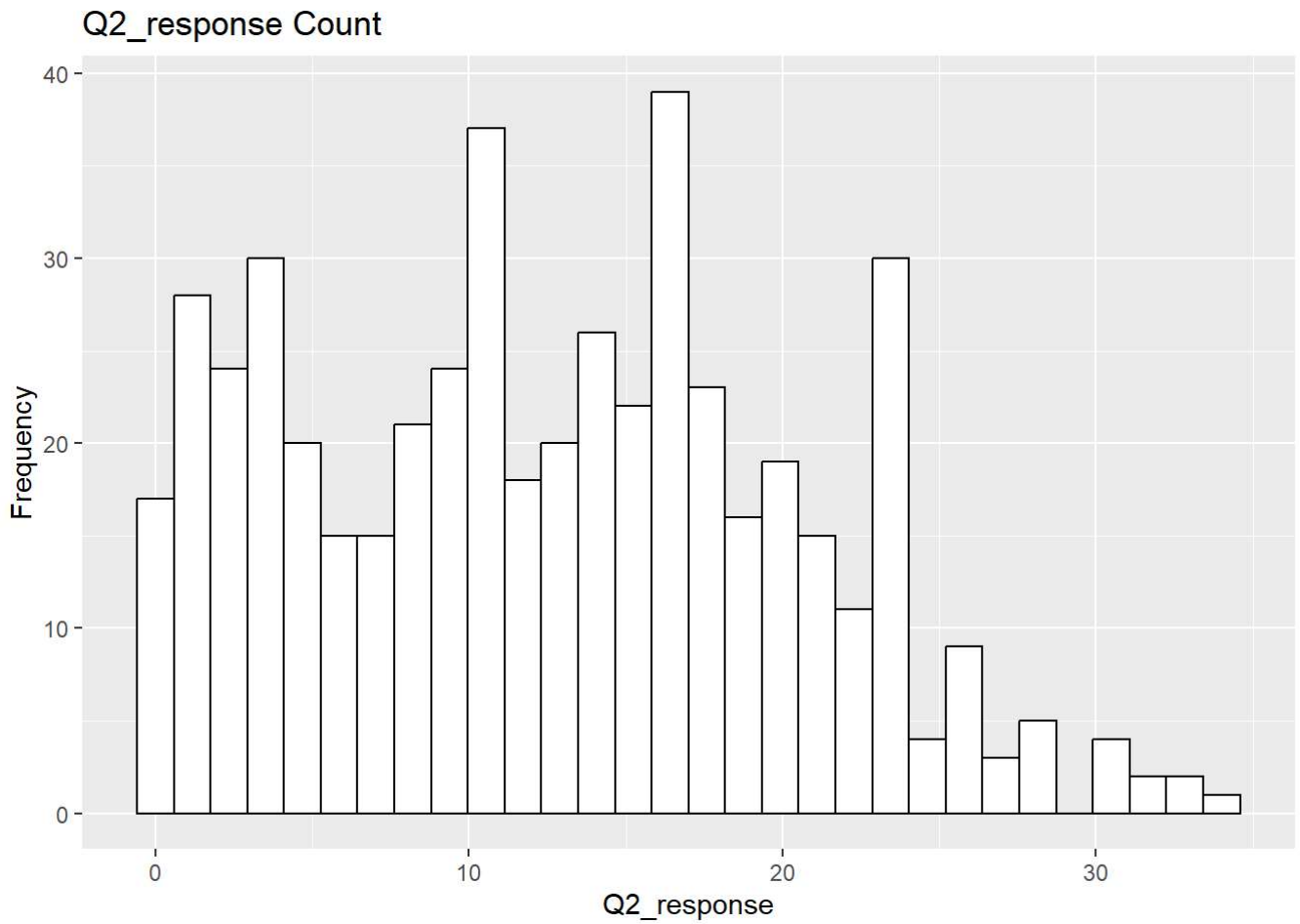
#Evidence: The linear model

```
linModel2 <- glm(Q2_response ~  Q2_1 + Q2_2, data = ExamDataset,family=gaussian)
summary(linModel2)
```

```
##
## Call:
## glm(formula = Q2_response ~ Q2_1 + Q2_2, family = gaussian, data = ExamDataset)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.02194    0.35824  -0.061    0.951
## Q2_1         0.98970    0.31852   3.107    0.002 **
## Q2_2         0.48362    0.01095  44.150   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 12.60862)
##
##     Null deviance: 30844.7  on 499  degrees of freedom
## Residual deviance:  6266.5  on 497  degrees of freedom
## AIC: 2691.1
##
## Number of Fisher Scoring iterations: 2
```

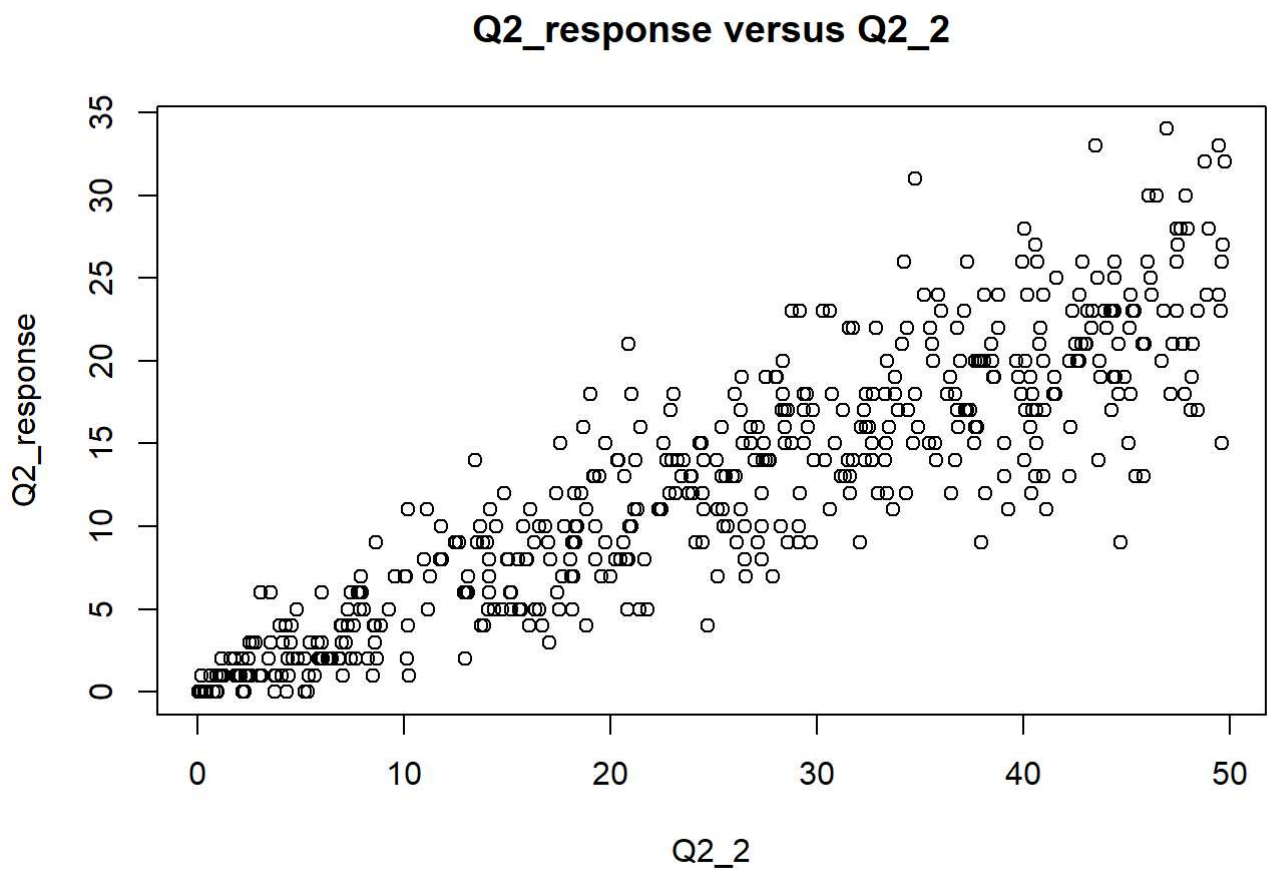
The Pt value here is high.

```
ggplot(data = ExamDataset) +geom_histogram(aes(x = Q2_response), fill = "white", color = "blac
k") +
  labs(x = "Q2_response", y = "Frequency", title = "Q2_response Count")
```

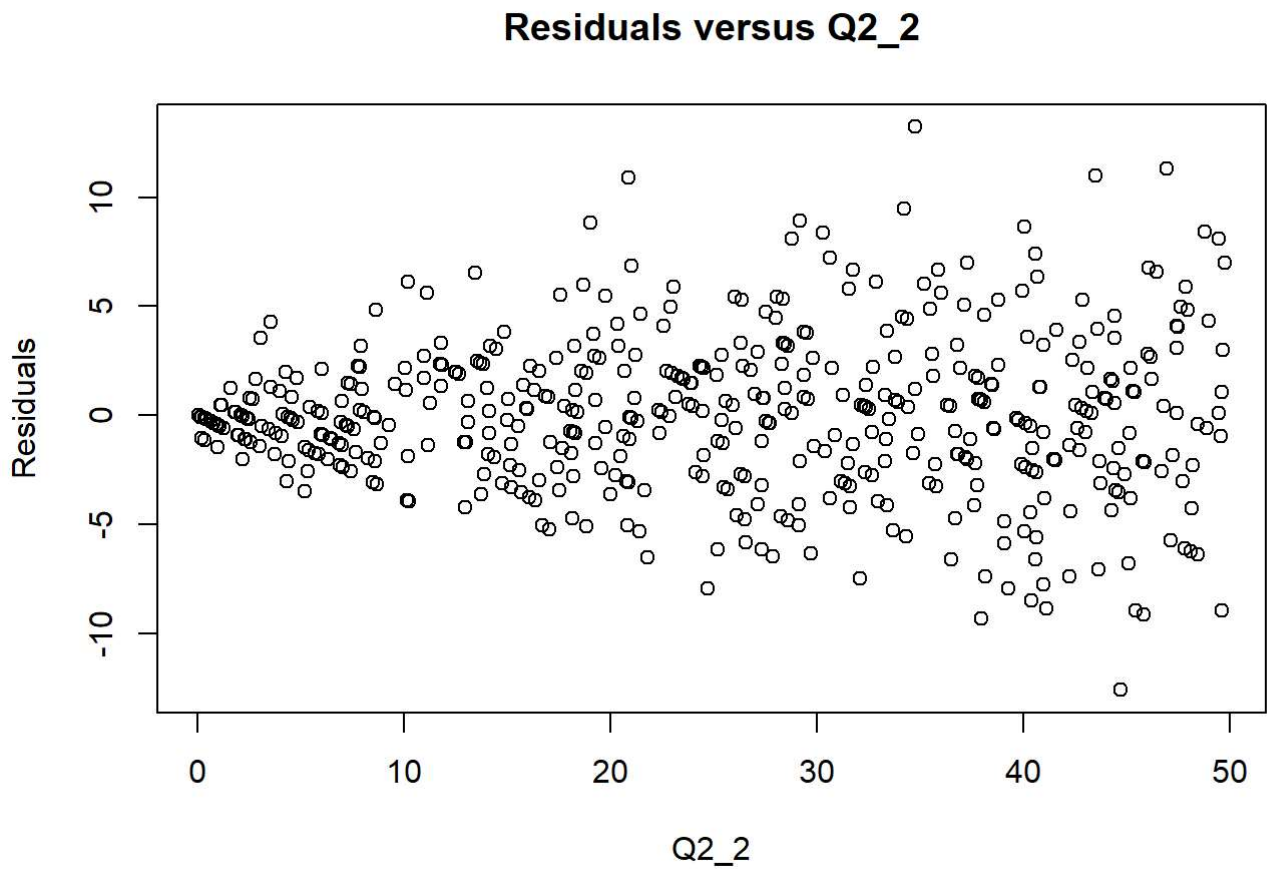


From above, the distribution of Q2_response is not normally distributed and positive count then.

```
plot(ExamDataset$Q2_2, ExamDataset$Q2_response, ylab = "Q2_response", xlab = "Q2_2", main = "Q2_
response versus Q2_2")
```




```
plot(ExamDataset$Q2_2, linModel2$residuals, xlab = "Q2_2", ylab = "Residuals", main = "Residuals
versus Q2_2" )
```



From the above plots, I can see although it shows a linear pattern roughly, but which is very loose-fitting and the data is getting spread out as to the right and so I can see the variance is definitely not constant(also showed in residual plot) and so which proves to not use a linear model here.

And for such response, the poisson and quassi poisson models were better choices for better predictions.

Poisson model

```
POIModel2 <- glm(Q2_response ~  Q2_1 + Q2_2, family=poisson(), data = ExamDataset)
```

Question 3

Lets take a look at without and with random effect at first.

Q3_response should be a factor(0 or 1) and we use binomial here

Without random effect (Q3_1 as a fixed effect)

```
Q3fit <- glm(Q3_response ~ Q3_1, data = ExamDataset,family=binomial)
summary(Q3fit)
```

```
##
## Call:
## glm(formula = Q3_response ~ Q3_1, family = binomial, data = ExamDataset)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.262e-14  1.414e+00   0.000    1.000
## Q3_1AB      -1.946e+00  1.773e+00  -1.098    0.272
## Q3_1AC       1.757e+01  1.251e+03   0.014    0.989
## Q3_1AD       5.107e-14  2.000e+00   0.000    1.000
## Q3_1AE      -1.099e+00  1.633e+00  -0.673    0.501
## Q3_1AF       1.386e+00  1.620e+00   0.856    0.392
## Q3_1AG       1.757e+01  2.797e+03   0.006    0.995
## Q3_1AH      -1.099e+00  1.633e+00  -0.673    0.501
## Q3_1AI      -1.386e+00  1.620e+00  -0.856    0.392
## Q3_1AJ      -1.757e+01  2.797e+03  -0.006    0.995
## Q3_1AK       1.099e+00  1.633e+00   0.673    0.501
## Q3_1AL       5.336e-14  1.549e+00   0.000    1.000
## Q3_1AM       5.183e-14  2.000e+00   0.000    1.000
## Q3_1AN       5.108e-01  1.592e+00   0.321    0.748
## Q3_1AO       5.346e-14  1.549e+00   0.000    1.000
## Q3_1AP       1.757e+01  2.797e+03   0.006    0.995
## Q3_1AQ       1.099e+00  1.633e+00   0.673    0.501
## Q3_1AR      -8.473e-01  1.574e+00  -0.538    0.590
## Q3_1AS       5.471e-14  2.000e+00   0.000    1.000
## Q3_1AT       5.353e-14  1.581e+00   0.000    1.000
## Q3_1AU      -2.197e+00  1.764e+00  -1.246    0.213
## Q3_1AV      -1.757e+01  2.797e+03  -0.006    0.995
## Q3_1AW       5.108e-01  1.592e+00   0.321    0.748
## Q3_1AX      -8.473e-01  1.574e+00  -0.538    0.590
## Q3_1AY      -1.757e+01  2.797e+03  -0.006    0.995
## Q3_1AZ      -1.946e+00  1.773e+00  -1.098    0.272
## Q3_1BA       4.055e-01  1.555e+00   0.261    0.794
## Q3_1BB       5.251e-14  2.000e+00   0.000    1.000
## Q3_1BC      -1.099e+00  1.633e+00  -0.673    0.501
## Q3_1BD       2.197e+00  1.764e+00   1.246    0.213
## Q3_1BE       5.436e-14  2.000e+00   0.000    1.000
## Q3_1BF       5.263e-14  1.581e+00   0.000    1.000
## Q3_1BG       1.386e+00  1.620e+00   0.856    0.392
## Q3_1BH       1.757e+01  2.797e+03   0.006    0.995
## Q3_1BI       5.296e-14  1.581e+00   0.000    1.000
## Q3_1BJ       8.473e-01  1.574e+00   0.538    0.590
## Q3_1BK       5.386e-14  2.000e+00   0.000    1.000
## Q3_1BL      -5.108e-01  1.592e+00  -0.321    0.748
## Q3_1BM       5.024e-14  1.549e+00   0.000    1.000
## Q3_1BN       6.873e-14  2.000e+00   0.000    1.000
## Q3_1BO      -1.946e+00  1.773e+00  -1.098    0.272
## Q3_1BP       5.253e-14  1.549e+00   0.000    1.000
## Q3_1BQ       4.516e-14  2.000e+00   0.000    1.000
## Q3_1BR       1.946e+00  1.773e+00   1.098    0.272
## Q3_1BS       2.197e+00  1.764e+00   1.246    0.213
## Q3_1BT      -1.757e+01  2.797e+03  -0.006    0.995
```

```
## Q3_1BU      -5.108e-01  1.592e+00  -0.321    0.748
## Q3_1BV      -8.473e-01  1.574e+00  -0.538    0.590
## Q3_1BW       5.041e-14  2.000e+00   0.000    1.000
## Q3_1BX       5.413e-14  1.581e+00   0.000    1.000
## Q3_1BY       5.284e-14  1.549e+00   0.000    1.000
## Q3_1BZ       6.192e-14  2.000e+00   0.000    1.000
## Q3_1CA       5.416e-14  1.581e+00   0.000    1.000
## Q3_1CB       2.197e+00  1.764e+00   1.246    0.213
## Q3_1CC      -1.757e+01  2.797e+03  -0.006    0.995
## Q3_1CD       1.946e+00  1.773e+00   1.098    0.272
## Q3_1CE      -1.386e+00  1.620e+00  -0.856    0.392
## Q3_1CF      -1.757e+01  2.797e+03  -0.006    0.995
## Q3_1CG       5.181e-14  1.581e+00   0.000    1.000
## Q3_1CH      -4.055e-01  1.555e+00  -0.261    0.794
## Q3_1CI       1.757e+01  2.797e+03   0.006    0.995
## Q3_1CJ      -1.099e+00  1.633e+00  -0.673    0.501
## Q3_1CK       4.055e-01  1.555e+00   0.261    0.794
## Q3_1CL       5.279e-14  2.000e+00   0.000    1.000
## Q3_1CM      -1.946e+00  1.773e+00  -1.098    0.272
## Q3_1CN      -8.473e-01  1.574e+00  -0.538    0.590
## Q3_1CO       1.757e+01  2.797e+03   0.006    0.995
## Q3_1CP       5.108e-01  1.592e+00   0.321    0.748
## Q3_1CQ       1.386e+00  1.620e+00   0.856    0.392
## Q3_1CR       1.757e+01  2.797e+03   0.006    0.995
## Q3_1CS       5.108e-01  1.592e+00   0.321    0.748
## Q3_1CT      -1.757e+01  1.251e+03  -0.014    0.989
## Q3_1CU      -1.757e+01  2.797e+03  -0.006    0.995
## Q3_1CV       5.378e-14  1.581e+00   0.000    1.000
## Q3_1CW      -8.473e-01  1.574e+00  -0.538    0.590
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 692.95  on 499  degrees of freedom
## Residual deviance: 523.61  on 425  degrees of freedom
## AIC: 673.61
##
## Number of Fisher Scoring iterations: 16
```

```
exp(Q3fit$coefficients)
```

```
##  (Intercept)       Q3_1AB       Q3_1AC       Q3_1AD       Q3_1AE       Q3_1AF
## 1.000000e+00 1.428571e-01 4.254481e+07 1.000000e+00 3.333333e-01 4.000000e+00
##       Q3_1AG       Q3_1AH       Q3_1AI       Q3_1AJ       Q3_1AK       Q3_1AL
## 4.254481e+07 3.333333e-01 2.500000e-01 2.350463e-08 3.000000e+00 1.000000e+00
##       Q3_1AM       Q3_1AN       Q3_1AO       Q3_1AP       Q3_1AQ       Q3_1AR
## 1.000000e+00 1.666667e+00 1.000000e+00 4.254481e+07 3.000000e+00 4.285714e-01
##       Q3_1AS       Q3_1AT       Q3_1AU       Q3_1AV       Q3_1AW       Q3_1AX
## 1.000000e+00 1.000000e+00 1.111111e-01 2.350463e-08 1.666667e+00 4.285714e-01
##       Q3_1AY       Q3_1AZ       Q3_1BA       Q3_1BB       Q3_1BC       Q3_1BD
## 2.350463e-08 1.428571e-01 1.500000e+00 1.000000e+00 3.333333e-01 9.000000e+00
##       Q3_1BE       Q3_1BF       Q3_1BG       Q3_1BH       Q3_1BI       Q3_1BJ
## 1.000000e+00 1.000000e+00 4.000000e+00 4.254481e+07 1.000000e+00 2.333333e+00
##       Q3_1BK       Q3_1BL       Q3_1BM       Q3_1BN       Q3_1BO       Q3_1BP
## 1.000000e+00 6.000000e-01 1.000000e+00 1.000000e+00 1.428571e-01 1.000000e+00
##       Q3_1BQ       Q3_1BR       Q3_1BS       Q3_1BT       Q3_1BU       Q3_1BV
## 1.000000e+00 7.000000e+00 9.000000e+00 2.350463e-08 6.000000e-01 4.285714e-01
##       Q3_1BW       Q3_1BX       Q3_1BY       Q3_1BZ       Q3_1CA       Q3_1CB
## 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 9.000000e+00
##       Q3_1CC       Q3_1CD       Q3_1CE       Q3_1CF       Q3_1CG       Q3_1CH
## 2.350463e-08 7.000000e+00 2.500000e-01 2.350463e-08 1.000000e+00 6.666667e-01
##       Q3_1CI       Q3_1CJ       Q3_1CK       Q3_1CL       Q3_1CM       Q3_1CN
## 4.254481e+07 3.333333e-01 1.500000e+00 1.000000e+00 1.428571e-01 4.285714e-01
##       Q3_1CO       Q3_1CP       Q3_1CQ       Q3_1CR       Q3_1CS       Q3_1CT
## 4.254481e+07 1.666667e+00 4.000000e+00 4.254481e+07 1.666667e+00 2.350463e-08
##       Q3_1CU       Q3_1CV       Q3_1CW
## 2.350463e-08 1.000000e+00 4.285714e-01
```

With a random effect (Q3_1 as a fixed effect)

```
fitrandom <- glmer(Q3_response ~ (1|Q3_1), data = ExamDataset,family = binomial)
summary(fitrandom)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Q3_response ~ (1 | Q3_1)
##    Data: ExamDataset
##
##      AIC      BIC   logLik deviance df.resid
##    668.1    676.5   -332.0    664.1      498
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.7511 -0.7547 -0.4695  0.8657  1.7912
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Q3_1   (Intercept) 0.8063   0.898
## Number of obs: 500, groups:  Q3_1, 75
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.05603    0.14890  -0.376    0.707
```

```
nrow(ExamDataset)
```

```
## [1] 500
```

Random effects are estimated with partial pooling, while fixed effects are not. And this improves efficiency. By sharing information across groups, partial pooling can lead to more efficient estimates. It allows for borrowing of strength from other groups, resulting in more precise and reliable parameter estimates. And also this Stabilizes estimates, especially for smaller groups or subgroups with limited data.

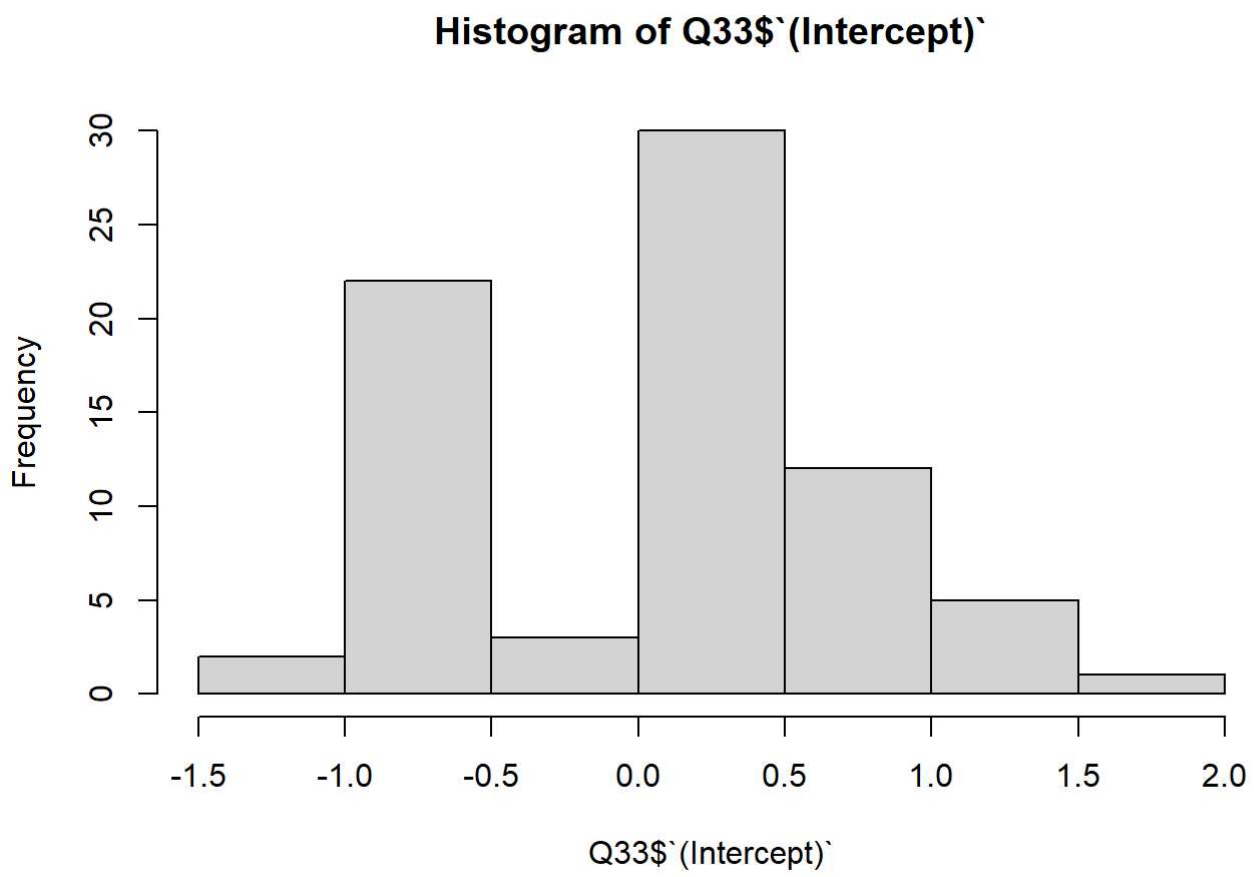
In Q3_1, there are so many subgroups, however, there are only 500 data in Q3_response which is fairly limited as consider so many subgroups. Hence, treating the variable as a random effect is definitely better in Q3.

```
Q33<-ranef(fitrandom)$Q3_1
Q33
```

```
##    (Intercept)
## AA  0.01609589
## AB -0.93715957
## AC  1.51764892
## AD  0.01609589
## AE -0.59636340
## AF  0.86746688
## AG  0.59437385
## AH -0.59636340
## AI -0.79686345
## AJ -0.56404557
## AK  0.66291629
## AL  0.03744783
## AM  0.01609589
## AN  0.34442791
## AO  0.03744783
## AP  0.59437385
## AQ  0.66291629
## AR -0.50691400
## AS  0.01609589
## AT  0.03458099
## AU -1.10971064
## AV -0.56404557
## AW  0.34442791
## AX -0.50691400
## AY -0.56404557
## AZ -0.93715957
## BA  0.30567471
## BB  0.01609589
## BC -0.59636340
## BD  1.17654360
## BE  0.01609589
## BF  0.03458099
## BG  0.86746688
## BH  0.59437385
## BI  0.03458099
## BJ  0.57997654
## BK  0.01609589
## BL -0.27590180
## BM  0.03744783
## BN  0.01609589
## BO -0.93715957
## BP  0.03744783
## BQ  0.01609589
## BR  1.00020450
## BS  1.17654360
## BT -0.56404557
## BU -0.27590180
## BV -0.50691400
## BW  0.01609589
## BX  0.03458099
## BY  0.03744783
```

```
## BZ  0.01609589
## CA  0.03458099
## CB  1.17654360
## CC -0.56404557
## CD  1.00020450
## CE -0.79686345
## CF -0.56404557
## CG  0.03458099
## CH -0.23122647
## CI  0.59437385
## CJ -0.59636340
## CK  0.30567471
## CL  0.01609589
## CM -0.93715957
## CN -0.50691400
## CO  0.59437385
## CP  0.34442791
## CQ  0.86746688
## CR  0.59437385
## CS  0.34442791
## CT -1.45624029
## CU -0.56404557
## CV  0.03458099
## CW -0.50691400
```

```
hist(Q33$`(Intercept)`,breaks=10)
```



With above evidence, I conclude using random effects is better.

Question 4

```
NumData = ExamDataset %>% select(Q4_1,Q4_2,Q4_3)
NumData <- scale(NumData)
distData <- dist(NumData, method = "manhattan")
NumData_scaled<-scale(as.matrix(NumData))
```

Then, I look for an optimal number of clusters using silhouette distance.

```
silav2<-2:15
for(i in 2:15){
sili=silhouette(kmeans(NumData_scaled,centers=i)$cluster,distData)
silav2[i]<-mean(sili[,3])
}
plot(2:15,silav2[2:15],type="l",xlab = "Number of Clusters", ylab = "Mean Silhouette Score", mai
n ="Silhouette plot")
```

I see that the maximum occurs at 3 with mean width, respectively, 0.95.

Create 3 clusters using K-means

```
kclust3 <- kmeans(NumData, centers = 3)
ExamDataset$cluster_kmean3 <- as.factor(kclust3$cluster)
```

Extract the location of the 3 centroids

```
kclust3$centers
```

```
##         Q4_1       Q4_2       Q4_3
## 1  0.2530641 -0.5513916  0.6050636
## 2 -0.4761810  1.1088726 -1.1581237
## 3 -0.5492126  1.1100984 -1.2893600
```

the total within-cluster sum of squares associated with 3 clusters

```
kclust3$tot.withinss
```

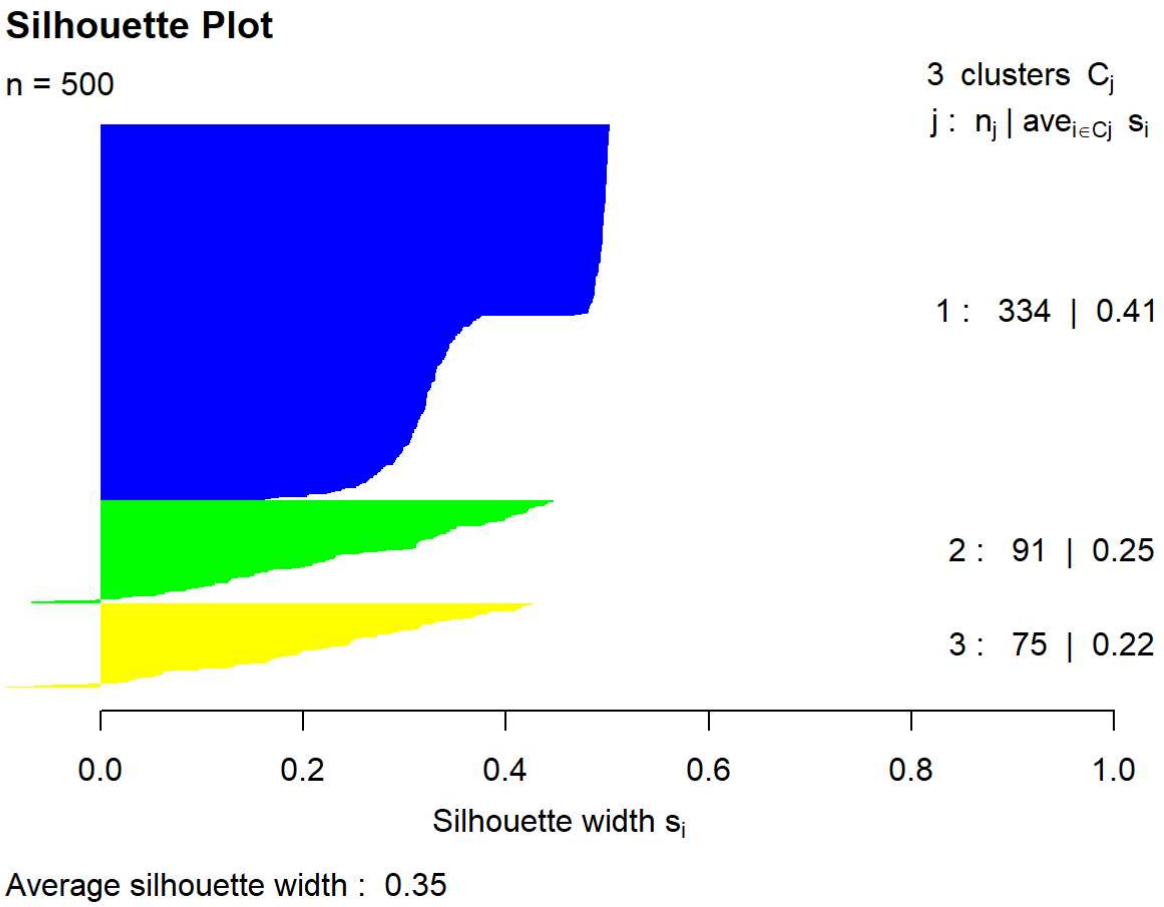
```
## [1] 757.4737
```

Investigate what these look like using aggregate and table

```
aggregate(NumData,list(kclust3$cluster),median)
```

```
##   Group.1       Q4_1      Q4_2      Q4_3
## 1       1 -0.6483497 -1.194318  1.007913
## 2       2 -0.4887885  1.115609 -1.163978
## 3       3 -0.5495192  1.112593 -1.277799
```

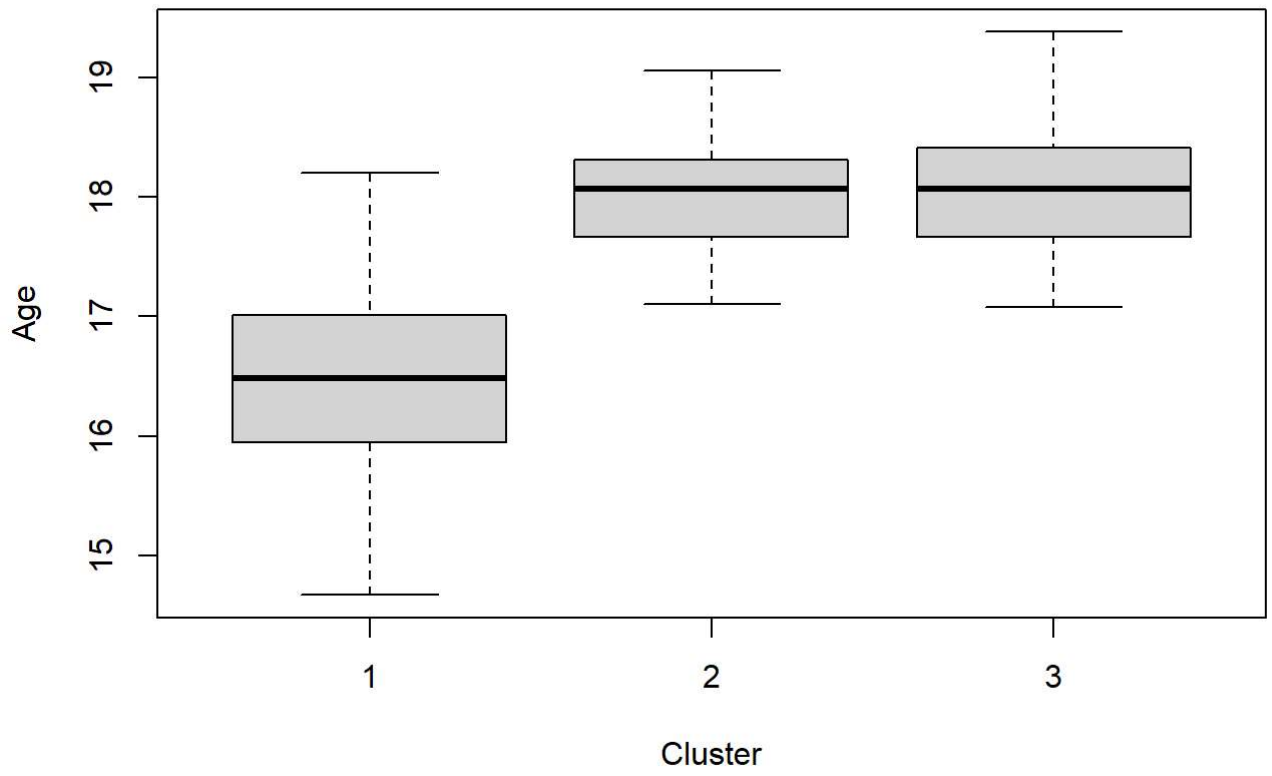
```
sildd <- silhouette(as.numeric(ExamDataset$cluster_kmean3), distData)
plot(sildd, col = c("blue", "green", "yellow"), border = NA, main = "Silhouette Plot")
```



Seems good. So the clusters are identified.

Explore how the variables Age and YearGroup are associated with each cluster

```
boxplot(ExamDataset$Age ~ kclust3$cluster, xlab = "Cluster", ylab = "Age", title="Age versus Clu
ster")
```




```
prop.table(table(ExamDataset$YearGroup, kclust3$cluster),1)
```

```
##
##                   1         2         3
##   Grade12 1.0000000 0.0000000 0.0000000
##   Grade13 1.0000000 0.0000000 0.0000000
##   Grade14 0.0000000 0.5481928 0.4518072
```

From above, clearly, the data is just assigned to each cluster probably by different Grade(Grade12,13,14) since I see that in table, there just 1 in each cluster and also in box plot, the mean is vary at 18, 16, 17 which proves my former guess.

Question 5

Undertake the principal component analysis centering and scaling the data.

```
Q5_CAC <- ExamDataset %>%
                select(Q5_1,Q5_2,Q5_3,Q5_4,Q5_5,Q5_6)
Q5 <- prcomp(Q5_CAC, center = TRUE, scale = TRUE)
```

Summarise the result

```
summary(Q5)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5    PC6
## Standard deviation     1.6672 1.6139 0.45265 0.40675 0.35516 0.3456
## Proportion of Variance 0.4633 0.4341 0.03415 0.02757 0.02102 0.0199
## Cumulative Proportion  0.4633 0.8973 0.93150 0.95908 0.98010 1.0000
```

From the above table, I can see that PC1 and PC2 is equivalent to almost 1.6 of the original variables, whereas others are only under 5% of one of the original variables; the PC1 and PC2 already by itself explain nearly half of the overall variance in the data, and others just decreases. Hence it indicates 1 or 2 dimensions in this data. And I check the proportion, With only 2 PC, we can explain close to 89.73% of the variance in the numerical data, but With 1, we explain 46.33%. Hence, I would summarise this by saying that there are really only about 2 dimensions in this data, with the remaining 4 representing noise.

Therefore, the answers given to the set of six questions can be effectively reduced to fewer dimensions.

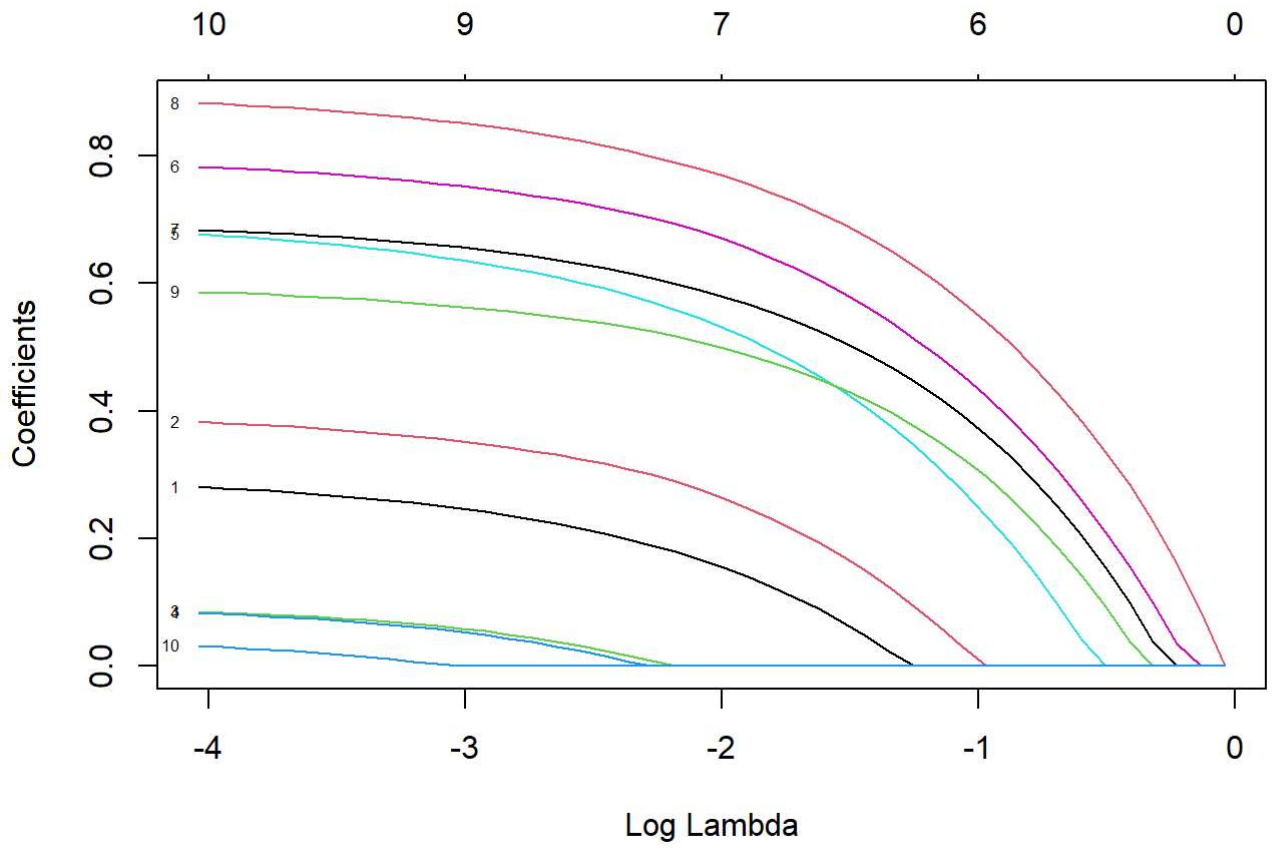
Question 6

```
Q6s<-as.vector(ExamDataset$Q6_response)
Q6predict<-model.matrix(~.-1,ExamDataset[,c(22:31)])
```

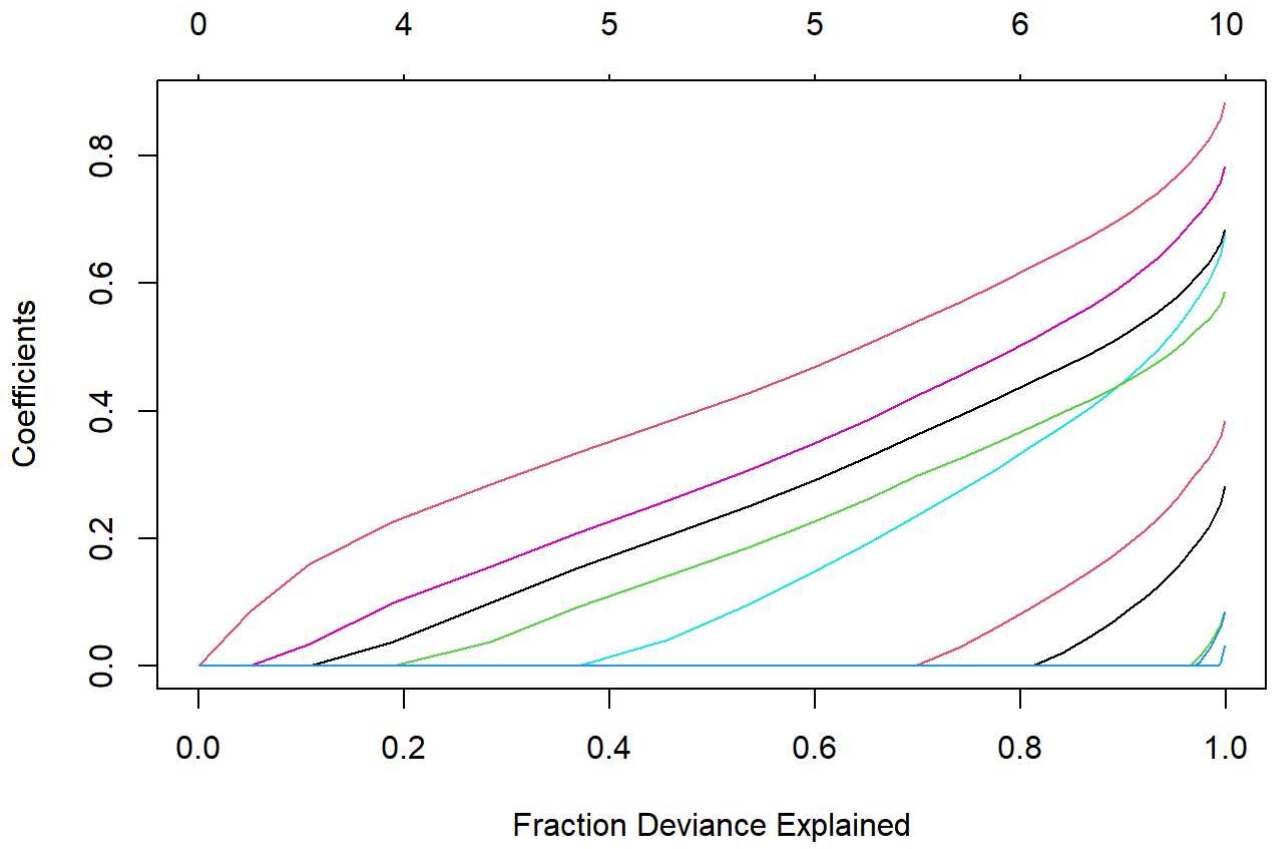
fit the model

```
Q6fit<-glmnet(Q6predict,Q6s)
```

```
plot(Q6fit,xvar = "lambda", label = TRUE)
```

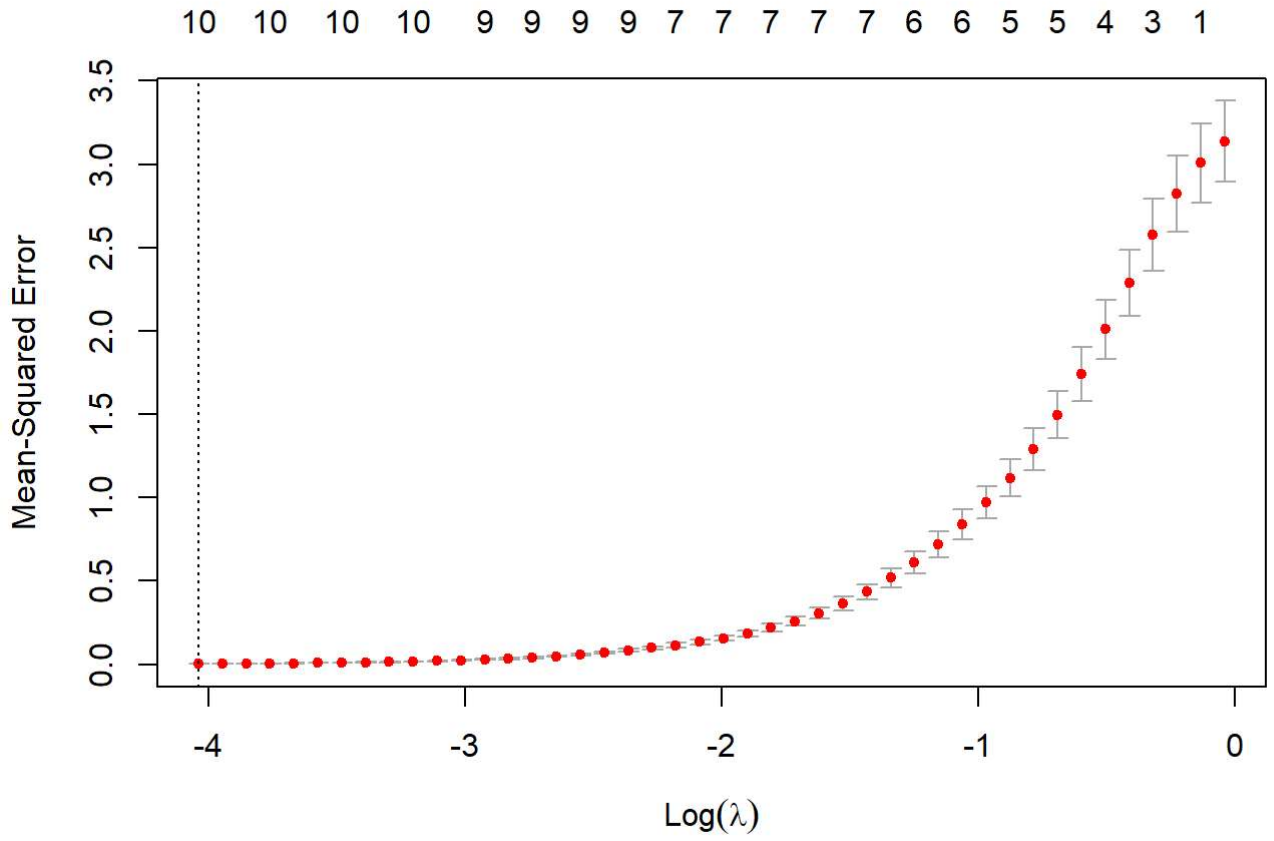



```
plot(Q6fit,xvar="dev")
```



Choosing lambda and variables with cv.glmnet

```
Q6cv<-cv.glmnet(Q6predict,Q6s)
plot(Q6cv)
```



From above plots, I can see there is no problem if I choose all predictors from Q6_1 to Q6_10 since 10 nonzero coefficients explains 100% of the deviance and minimize the error also.

But I can see that it is also reasonable to pick 6 predictors since 6 nonzero coefficients explains 80% of the deviance and the error is not too high.

To find these 6 predictors, I can see the log lambda is -1.2.

```
Q6_coef6<-coef(Q6fit, s=exp(-1.2))
Q6_coef6@Dimnames[[1]][1+Q6_coef6@i]
```

```
## [1] "(Intercept)" "Q6_2"        "Q6_5"        "Q6_6"        "Q6_7"
## [6] "Q6_8"        "Q6_9"
```

But lets back to 10 predictors.

Fitting and evaluating the final model.

To evaluate model quality, we will want to build the model on a training set and test it on a testing set.

```
set.seed(321)
training.samples <- createDataPartition(ExamDataset$Q6_response, p = 0.8, list = FALSE)
train.data <- ExamDataset[training.samples, ]
test.data <- ExamDataset[-training.samples, ]
```

Now we create the model on the training data.

```
train.model <- lm(Q6_response ~ Q6_1+Q6_2+Q6_3+Q6_4+Q6_5+Q6_6+Q6_7+Q6_8+Q6_9+Q6_10,train.data)
```

Now we will make predictions on the testing set.

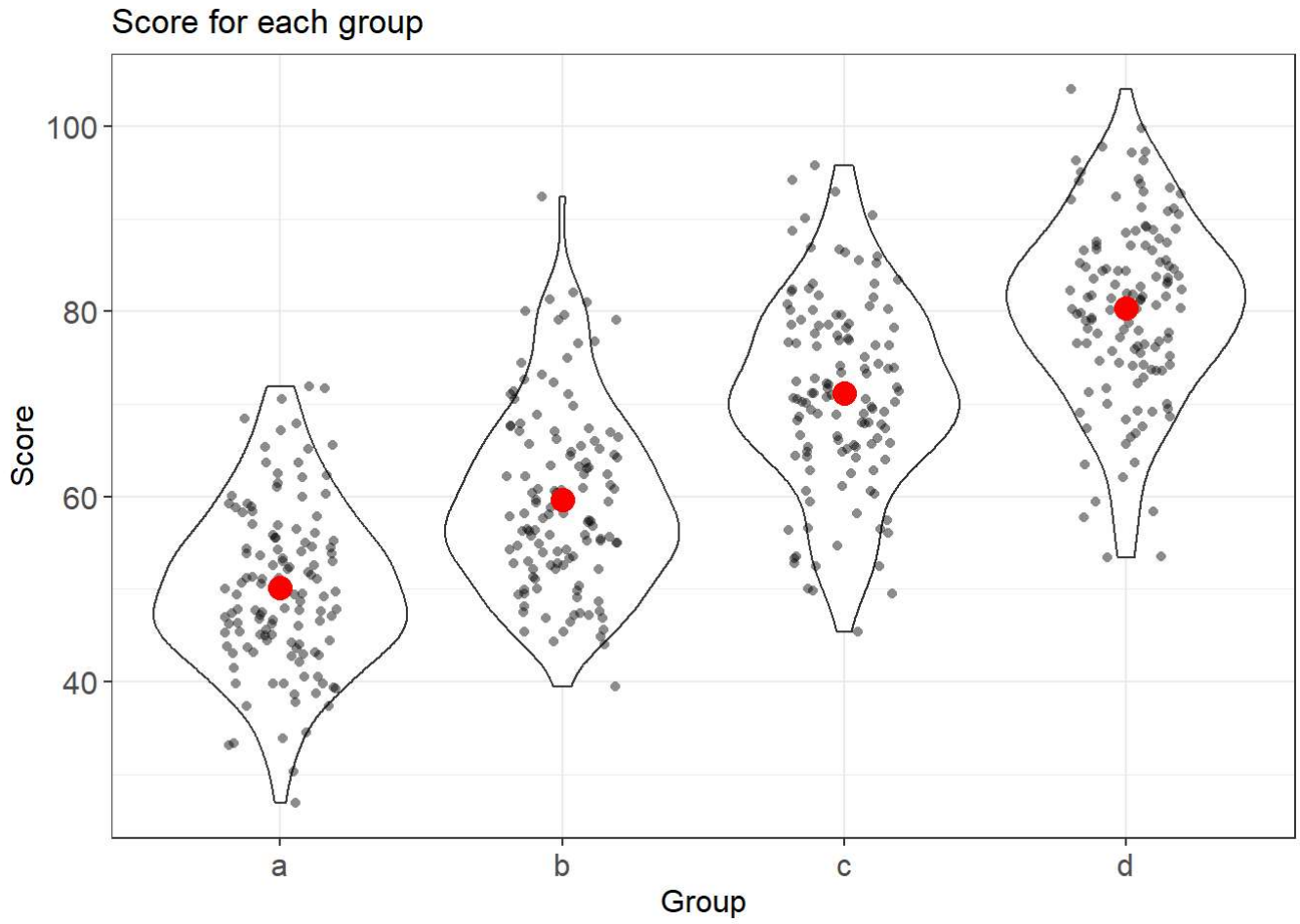
```
predictions <- predict(train.model, test.data)
data.frame( R2 = R2(predictions, test.data$Q6_response),
            RMSE = RMSE(predictions, test.data$Q6_response),
            MAE = MAE(predictions, test.data$Q6_response))
```

```
##   R2         RMSE          MAE
## 1  1 3.494025e-15 2.654647e-15
```

So this is pretty good performance–the R2 is 1, which means there is a very good correlation between the predicted Q6_response and the actual Q6_response on the testing set.

Question 7

```
ggplot(ExamDataset, aes(x = Q7_2, y = Q7_1)) +
  geom_violin() +
  geom_jitter(shape = 16 , position=position_jitter(0.2), alpha = 0.45) +
  labs(x = "Group",y = "Score", title = "Score for each group") +
  theme_bw() +
  theme(plot.title = element_text(size = 13),
        axis.text.x = element_text(size = 12),
        axis.text.y = element_text(size = 12),
        axis.title = element_text(size = 12)) +
  stat_summary(fun.data = mean_sdl,geom = "point", color = "red", size = 4)
```



Question 8

```
Q8 <- glm(Q8_response ~ Q8_1 * Q8_2, family=poisson(link="log"),data=ExamDataset)
summary(Q8)
```

```
##
## Call:
## glm(formula = Q8_response ~ Q8_1 * Q8_2, family = poisson(link = "log"),
##     data = ExamDataset)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      1.36316    0.07563  18.023  < 2e-16 ***
## Q8_1             0.57998    0.05575  10.402  < 2e-16 ***
## Q8_2GroupB       0.23776    0.09125   2.605  0.00917 **
## Q8_2GroupC       0.55993    0.08372   6.688 2.27e-11 ***
## Q8_1:Q8_2GroupB  0.42586    0.06641   6.413 1.43e-10 ***
## Q8_1:Q8_2GroupC  0.80788    0.06103  13.237  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 8157.49  on 499  degrees of freedom
## Residual deviance:  512.79  on 494  degrees of freedom
## AIC: 2748
##
## Number of Fisher Scoring iterations: 4
```

```
exp(Q8$coefficients)
```

```
##     (Intercept)            Q8_1      Q8_2GroupB      Q8_2GroupC Q8_1:Q8_2GroupB
##        3.908511        1.786004        1.268401        1.750543        1.530906
## Q8_1:Q8_2GroupC
##        2.243157
```

In this formula, there is an iteration term, which is the product of Q8_1 and groups in Q8_2. The term shows the influence of Q8_1 on Q8_response to differ between groups in Q8_2.

The formula is E(Q8_response) = b0 + b1 × Q8_1 + b2 × Q8_2 + b3 × Q8_1 × Q8_2.

The base here is Group A from Q8_2.

Question: For each group,determine the impact of a change of 1 unit in Q8_1 on the expected count of Q8_response.

Changing Q8_1 by 1 unit will change the expected counts of goals by 78.6%, given group A in Q8_2.

Changing Q8_1 by 1 unit will change the expected counts of goals by 131.69% = 78.6%+53.09%, given group B in Q8_2.

Changing Q8_1 by 1 unit will change the expected counts of goals by 202.9% = 78.6%+124.3%, given group C in Q8_2.