STAT4198 Report
Min Yang

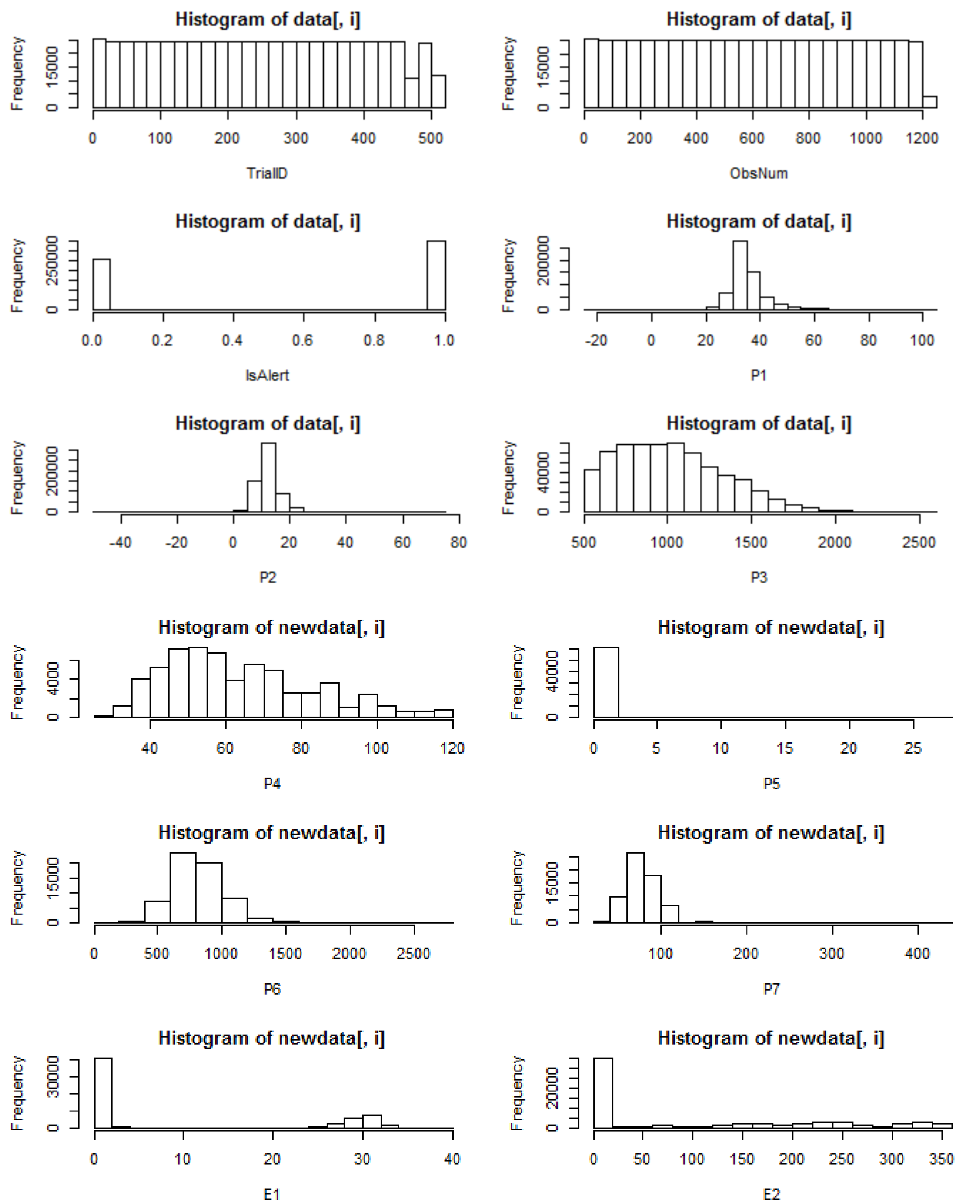
Project Report

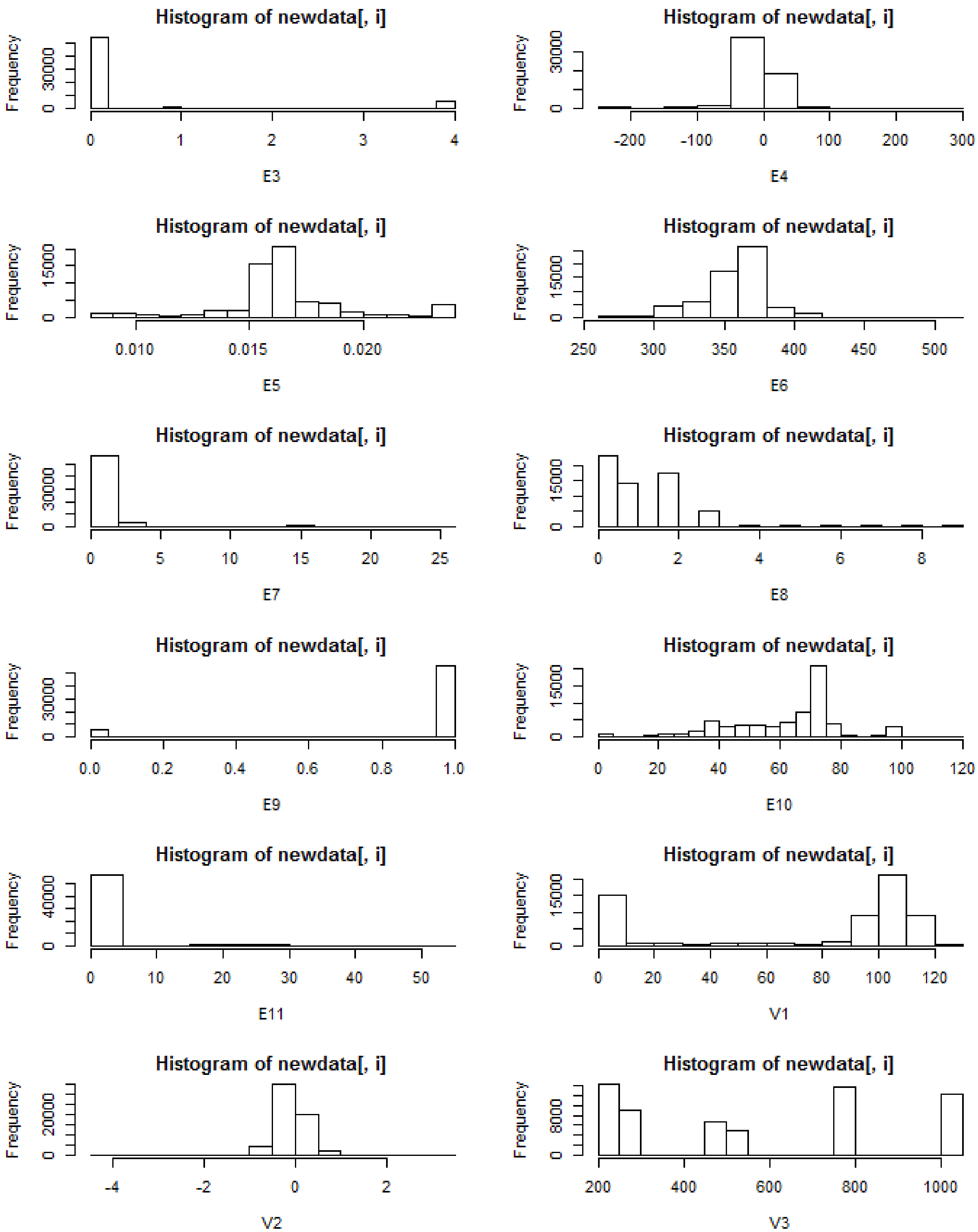
**Exploratory Data Analysis**

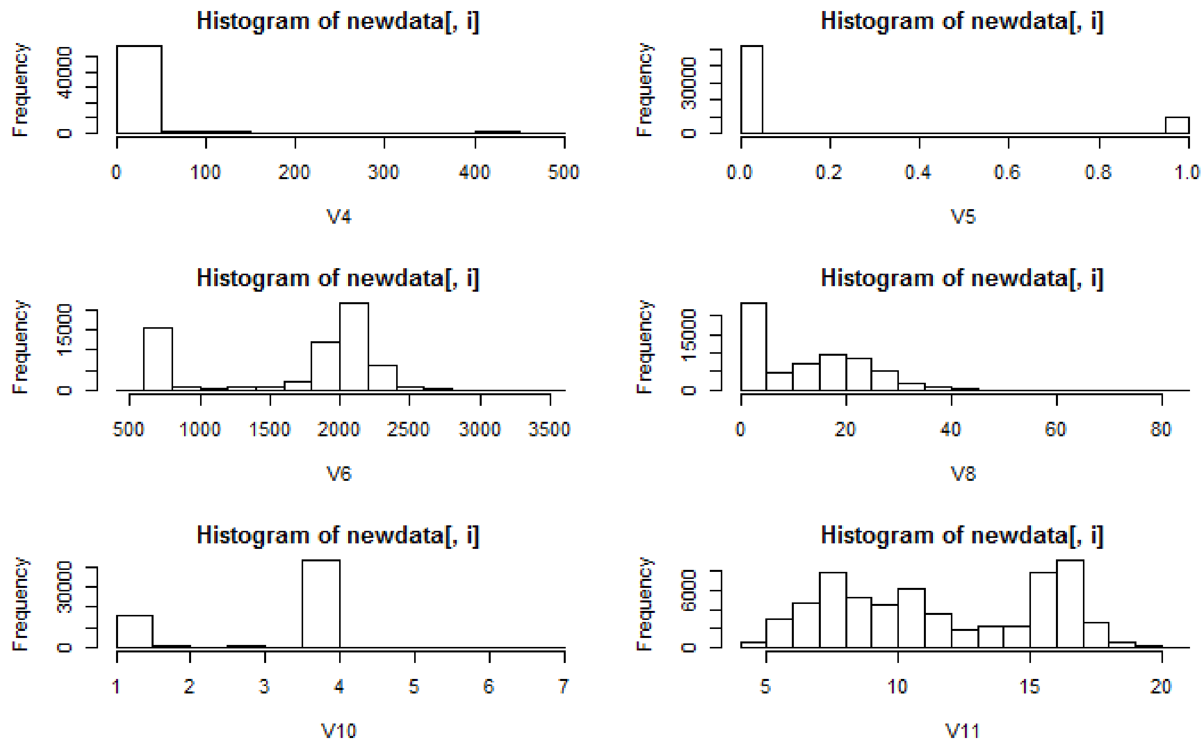
1. **Remove features which do no change their values :**

P8, V7, V9. All the variables are 0.

2. **Specify**




Histogram of newdata[, i]
Frequency
0 30000
0 1 2 3 4
E3
Histogram of newdata[, i]
Frequency
0 30000
-200 -100 0 100 200 300
E4
Histogram of newdata[, i]
Frequency
0 15000
0.010 0.015 0.020
E5
Histogram of newdata[, i]
Frequency
0 15000
250 300 350 400 450 500
E6
Histogram of newdata[, i]
Frequency
0 30000
0 5 10 15 20 25
E7
Histogram of newdata[, i]
Frequency
0 15000
0 2 4 6 8
E8
Histogram of newdata[, i]
Frequency
0 30000
0.0 0.2 0.4 0.6 0.8 1.0
E9
Histogram of newdata[, i]
Frequency
0 15000
0 20 40 60 80 100 120
E10
Histogram of newdata[, i]
Frequency
0 40000
0 10 20 30 40 50
E11
Histogram of newdata[, i]
Frequency
0 15000
0 20 40 60 80 100 120
V1
Histogram of newdata[, i]
Frequency
0 20000
-4 -2 0 2
V2
Histogram of newdata[, i]
Frequency
0 8000
200 400 600 800 1000
V3




| | |
|---|---|
| TrialID | Categorical, as stated in question, 511 categories as drivers |
| ObsNum | Generate the order of the observations for each subject, no meaning to interpret |
| IsAlert | Categorical, two category (0,1) |
| P1 | Continuous, symmetric with slight right skewed |
| P2 | Continuous, slightly right skewed |
| P3 | Continuous, symmetric, slightly right skewed |
| P4 | Continuous, symmetric, slightly right skewed |
| P5 | Continuous, right skewed |
| P6 | Continuous, symmetric, contain 14 missing values |
| P7 | Continuous, symmetric but somewhat right skewed, contain 4 missing values |
| E1 | Continuous, missing data in between |
| E2 | Continuous, missing data, the observation missing for E1 is also missed here |
| E3 | Categorical, integer, three categories (0,1,4), mode at 0 |
| E4 | Continuous, symmetric, slightly left-skewed |
| E5 | Continuous, symmetric |
| E6 | Continuous, symmetric, slightly left skewed |
| E7 | Categorical, integer, 22 categories, mode at 0 |
| E8 | Categorical, integer, 10 categories, mode at 0 |
| E9 | Categorical, two categories, mode at 1 |
| E10 | Continuous integer, left skewed |
| E11 | Continuous, multimodality, one peak around 40, the other around 70 |
| V1 | Continuous, multimodality, two peaks, one around 0, the other one around 100 |
| V2 | Continuous, symmetric |

| | |
|---|---|
| V3 | Categorical, 28 categories, integer, mode is around 240 and 800, contain 14 missing data |
| V4 | Continuous, right-skewed |
| V5 | Categorical, 2 categories, mode is 0 |
| V6 | Continuous, multimodality, two peaks. One around 500, one around 2000 |
| V8 | Continuous, right skewed |
| V10 | Categorical, 5 categories, mode at 4 |
| V11 | Continuous, multimodality, two peaks, one around7, one around 16 |

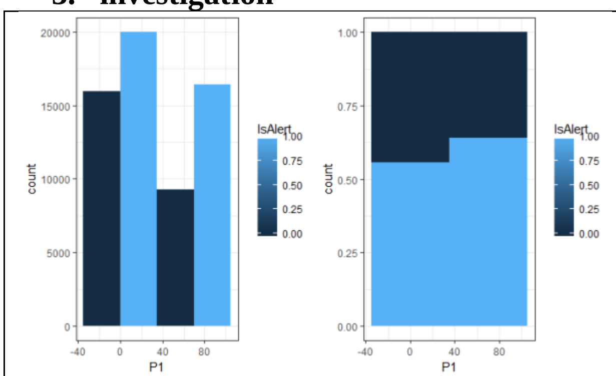
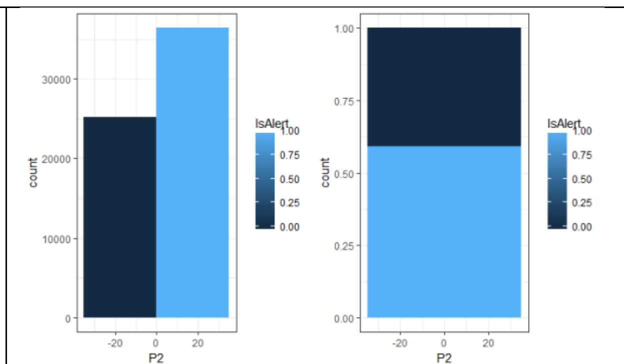
```{r}
n_occur <- data.frame(table(newdata$P1))
n_occur[n_occur$Freq > 1,]

```

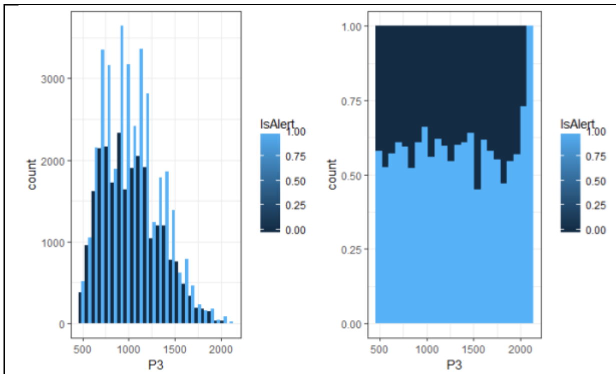
| | Var1 <fctr> | Freq <int> |
|---|---|---|
| 10 | 22.6409 | 2 |
| 59 | 23.1269 | 2 |
| 124 | 23.3417 | 2 |
| 167 | 23.4787 | 2 |
| 180 | 23.5446 | 2 |
| 193 | 23.5765 | 2 |
| 243 | 23.7381 | 3 |
| 262 | 23.7773 | 2 |
| 308 | 23.9073 | 2 |
| 348 | 24.042 | 2 |

1-10 of 14,242 rows Previous 1 2 3 4 5 6 ... 100 Next

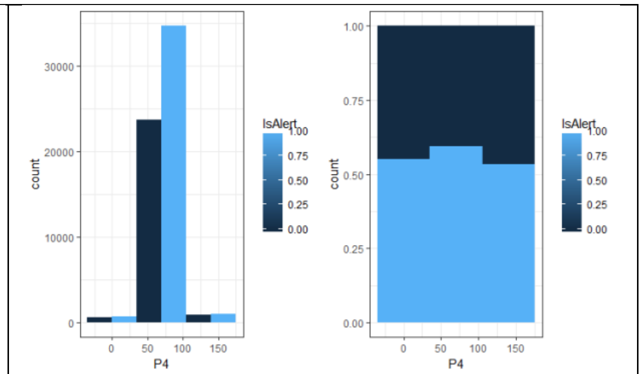
From the table, we can see that the data have repeated observations on the same subject.

### 3. investigation

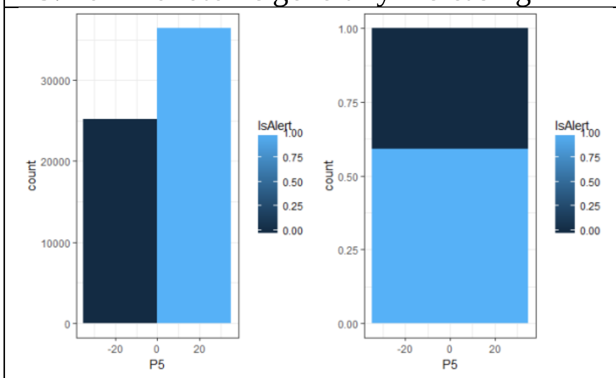
| | |
|---|---|
|  |  |
| P1: monotonic increasing | P2: remain constant, no relationship |



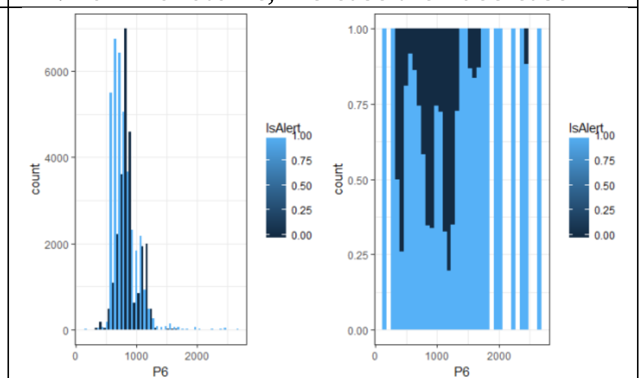
P3: non-monotonic generally increasing



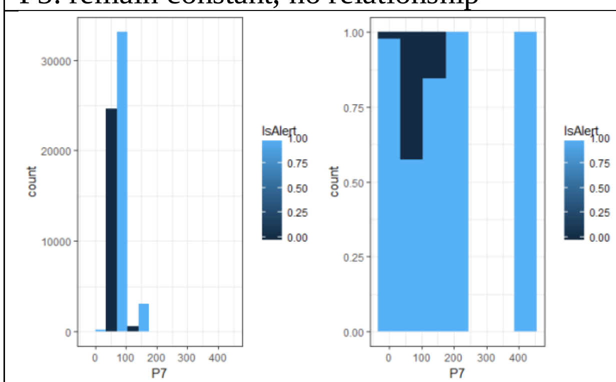
P4: non-monotonic, increase then decrease



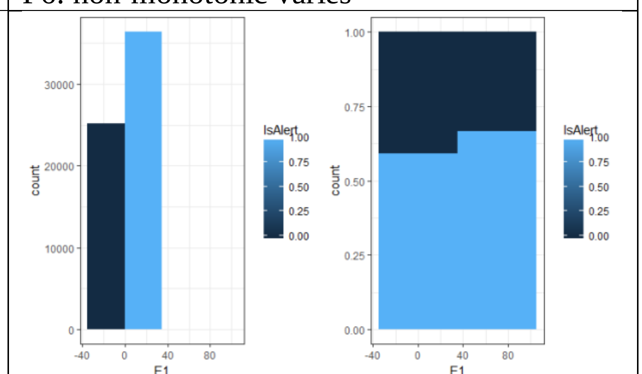
P5: remain constant, no relationship



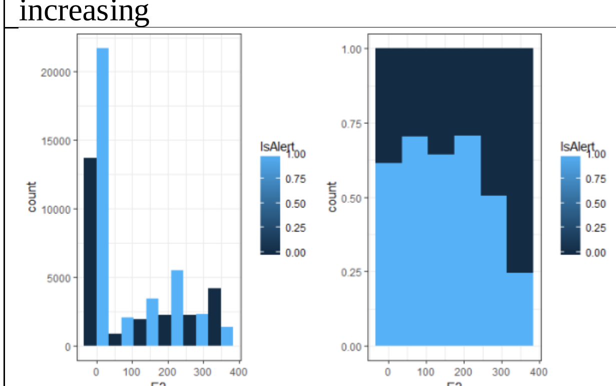
P6: non-monotonic varies



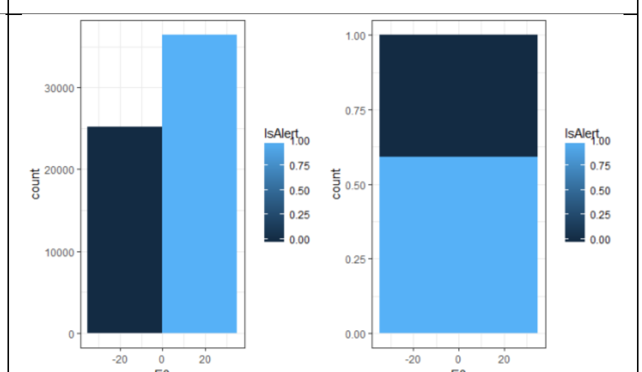
P7: non-monotonic decreasing then increasing



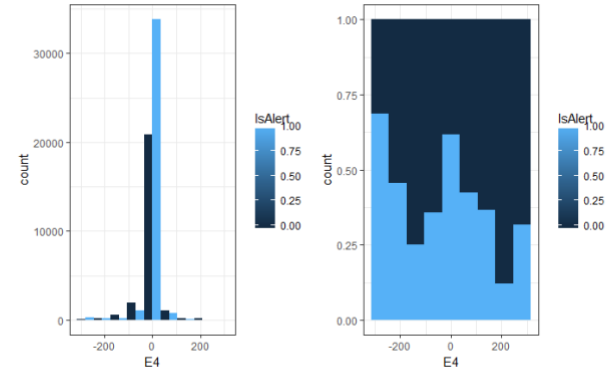
E1:monotonic increasing



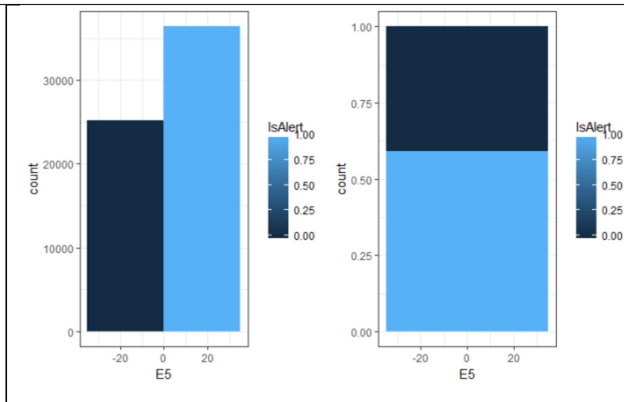
E2: decreasing



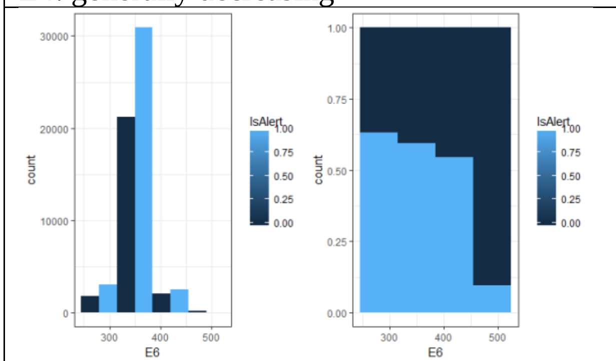
E3: remain constant, categorical



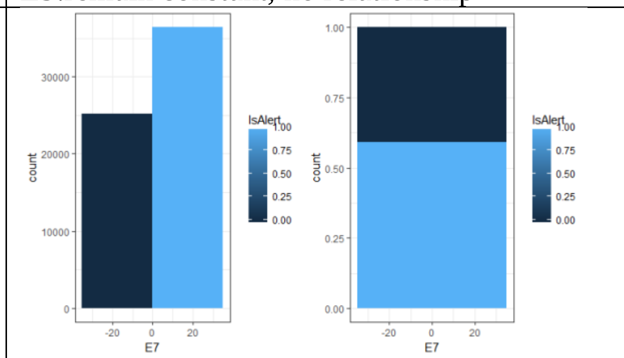

E4: generally decreasing



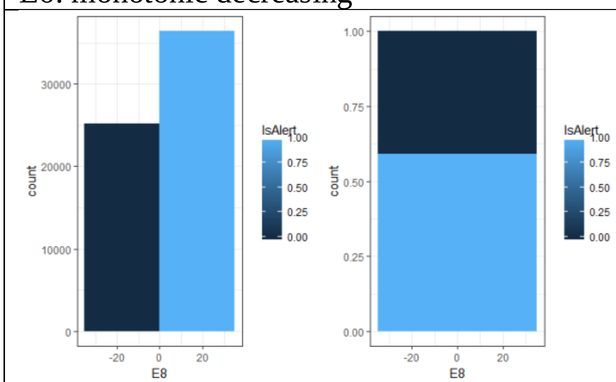

E5:remain constant, no relationship



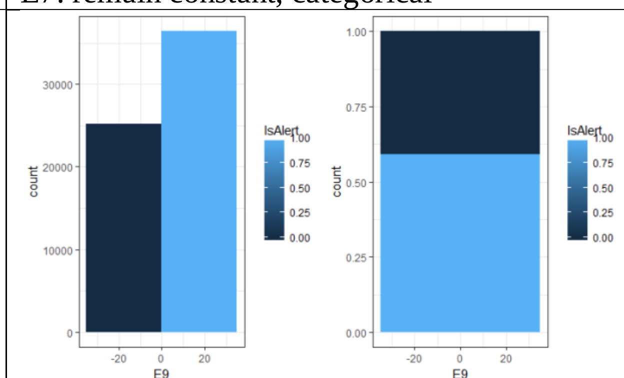

E6: monotonic decreasing



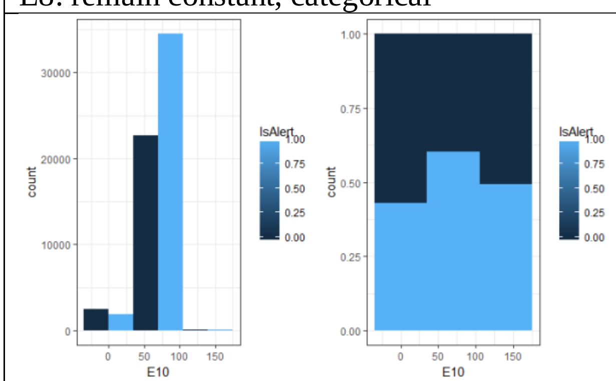

E7: remain constant, categorical



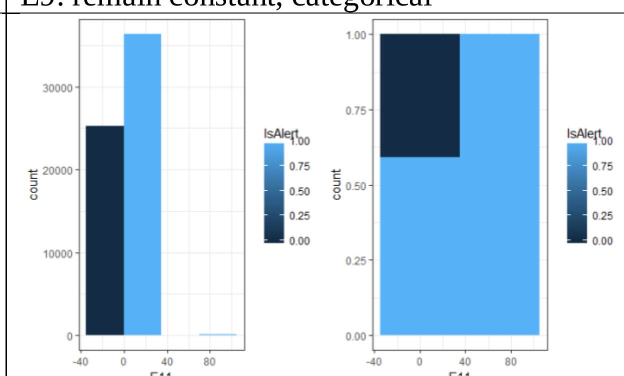

E8: remain constant, categorical



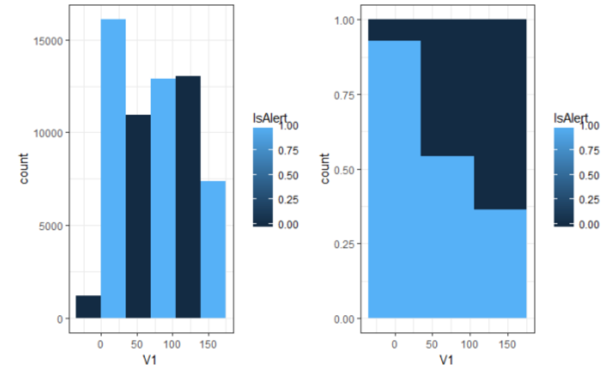
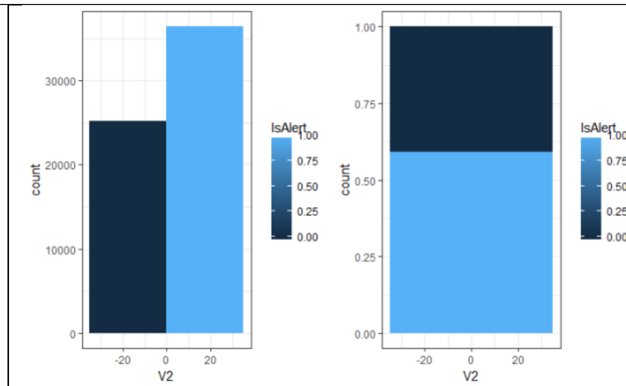
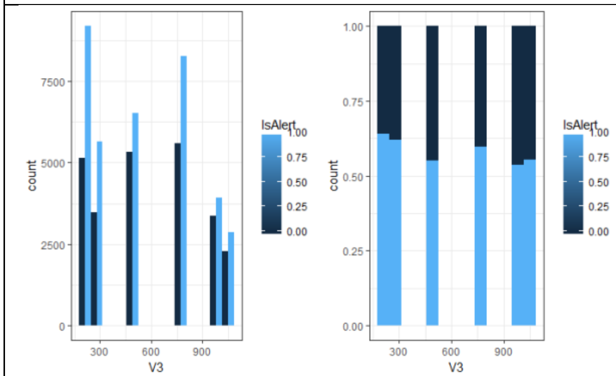
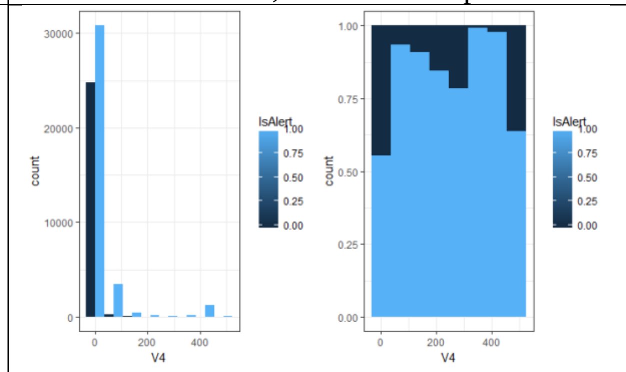
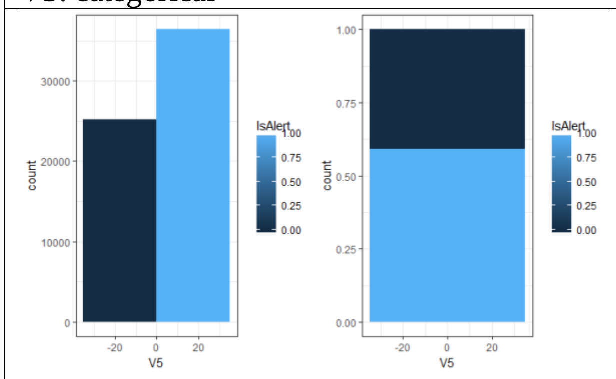
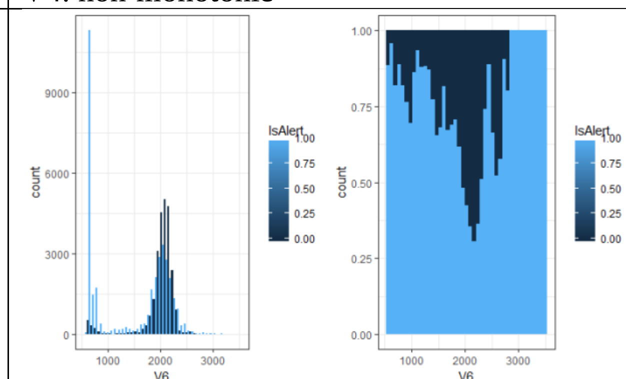
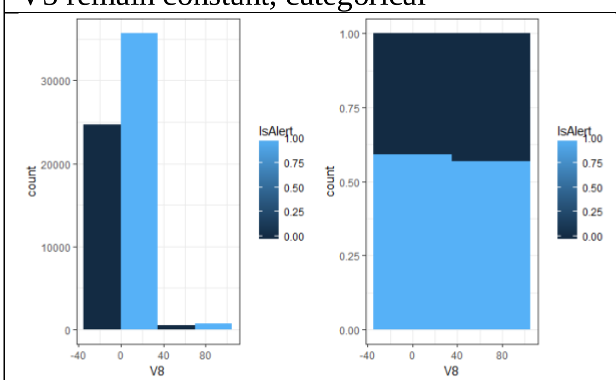
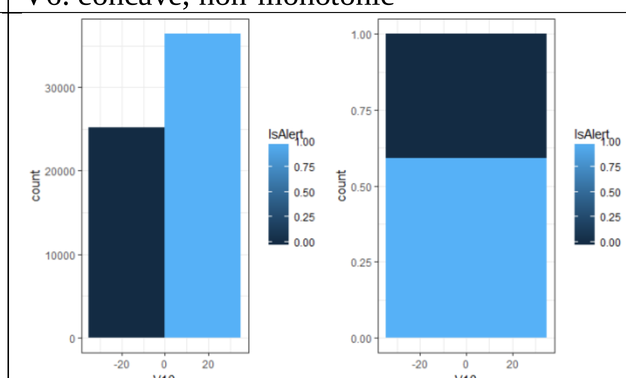

E9: remain constant, categorical

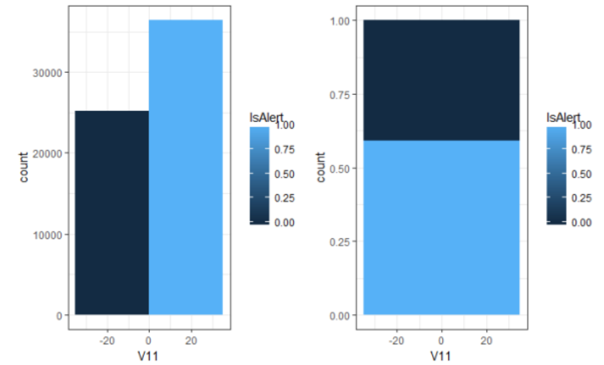



E10: increase then decrease



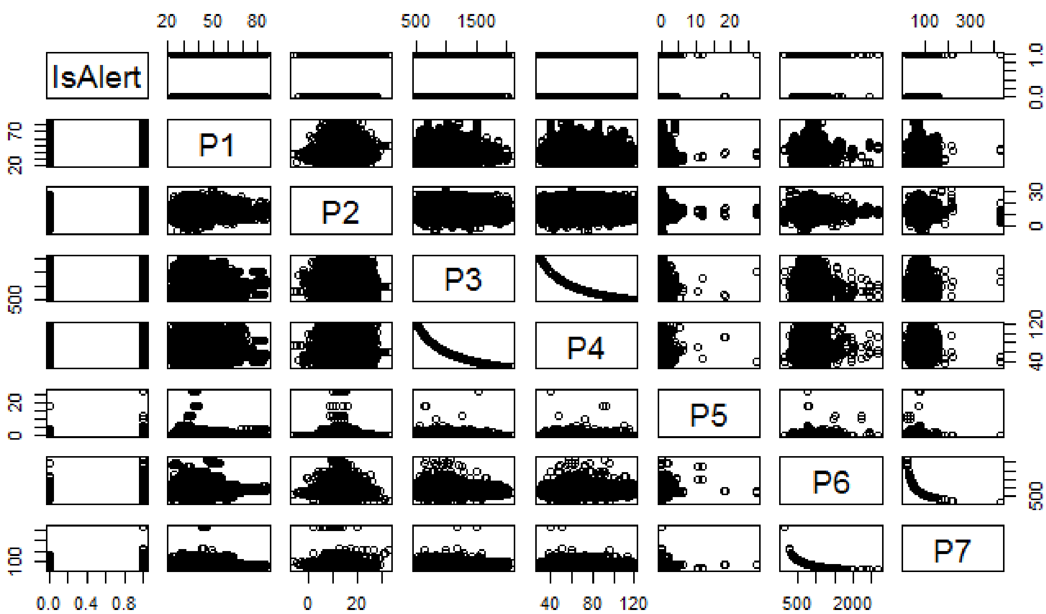

E11:monotonic increasing

| | |
|---|---|
|  |  |
| V1 monotonic and linear | V2 remain constant, no relationship |
|  |  |
| V3: categorical | V4: non-monotonic |
|  |  |
| V5 remain constant, categorical | V6: concave, non-monotonic |
|  |  |
| V8: slightly decreaing | V10 remain constant, categorical |

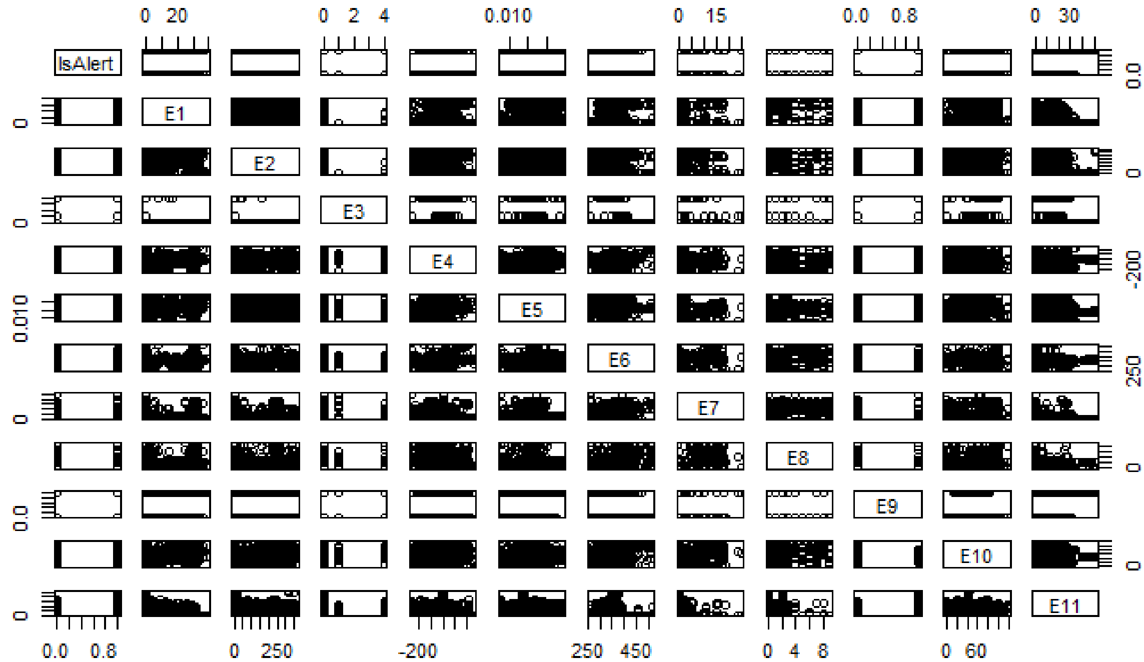
| | |
|---|---|
|  | |
| V11: remain constant, no relationship | |



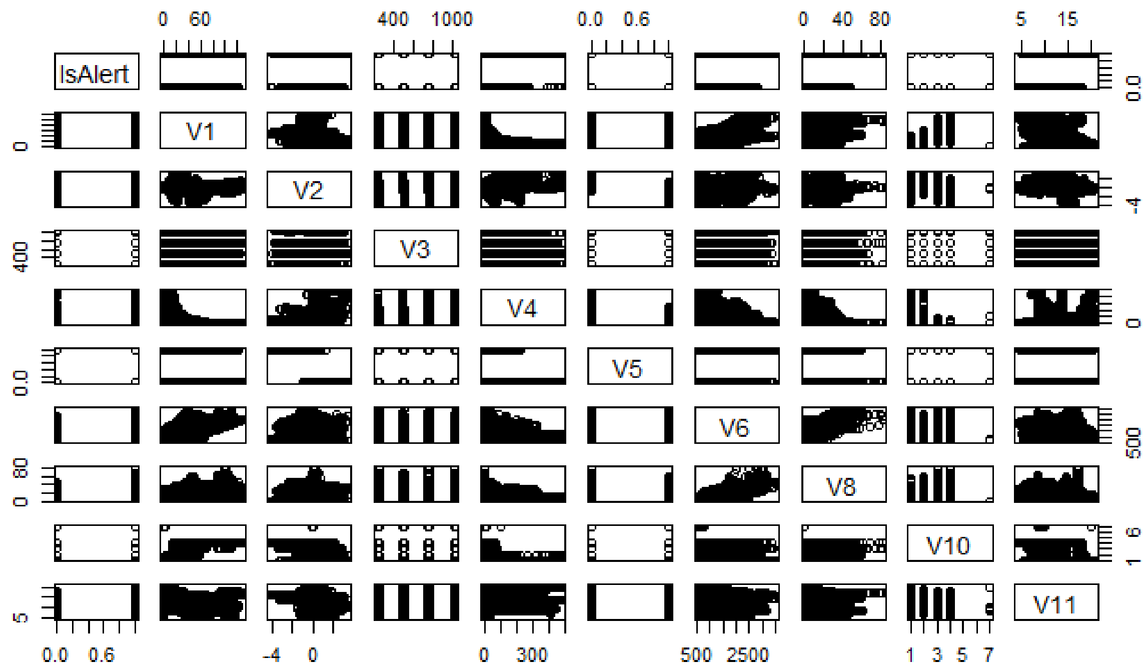
**4. Identify pairs of features which are related to each other**




P1-P7: Pairs related to each other: P3 and P4, P6 and P7.




E1-E11: there is no clear relationship, it is expected as many of them are categorical variables




V1-V11: there is linear relationship between V6 and V8, V1 and V4 has a non-linear relationship.

5. **Dimension Reduction**
   a) remove the features which have no relationship with the response
   From analysis above, P2, P5, E5, V2, V11 do not have relationship on response and are not categorical variables, so I decided to remove them.
   b) remove a feature that has a perfect relationship to another one.
   I remove P4 since it has perfect relationship with P3, but does not have strong relationship with response as P3 does.
   c) transform a set of highly correlated features into a smaller set.
   P6 and P7 is highly correlated, so I use log transformation for them.
   d) V3 has 28 levels which is large, so I applied binning on it and split the data into four groups

```
binned.V3 <-
1*(newdata2$V3<255)+2*(newdata2$V3<511)*(newdata2$V3>254)+3*(newdata2$V3<767)*(newdata2$V3>510)+4*(newdata2$V3>766)
newdata2$V3 <-factor(binned.V3,label=c("level1","level2","level3","level4"))
```

   e) V1 and V4 are significantly related to each other in non-linear way, so I remove V4 since it has a weaker relationship to response.

   Notation:
   newdata is the original dataset generate from the dataset
   newdata1 remove the feature not change values
   newdata2 remove the five features have no relationship on response
   newdata3 is the cleaned data with all transformation, binned down. I will use this for all the modeling

**Basic Model Building**

I split 75% of the data to training, and 25% for testing.

1) **Neural Network**
   Notation: newdat is same as newdata3, as I use newdat to create transform categorical feature into dummy variables.

   I tried size=5,10,20. Decay=0,0.2,0.5,0.8. maxit=100,200,500,1000 to fit the best neural network with the lowest misclassification rate.
   With maxit=100,200 all the iteration stopped without converged, so I need a larger iteration as the lowest value is not reached yet.
   After several tries, I end up test size=5,10,15, maxit=500,1000, decay=0,0.2,0.5.
   Here are all models that converged:

| Node | maxit | decay | Iteration value | Train | Test |
|---|---|---|---|---|---|
| 5 | 500 | 0 | 5014.25 | 0.1362 | 0.1327 |
| 5 | 500 | 0.2 | 4133.28 | 0.0957 | 0.0923 |
| 5 | 500 | 0.5 | 4648.2 | 0.1054 | 0.1035 |
| 5 | 1000 | 0.2 | 4366.48 | 0.1029 | 0.0997 |
| 10 | 1000 | 0.2 | 3693.94 | 0.0751 | 0.0725 |
| 5 | 1000 | 0.5 | 4801.61 | 0.1077 | 0.1034 |
| 10 | 1000 | 0.5 | 4442.68 | 0.0868 | 0.0842 |

Among all, the best neural network model is the one with lowest iteration value, which is node=10, maxit=1000, decay=0.2,with a training misclassification rate of 0.0751, and a testing misclassification rate of 0.0725. So the model fits very well as the misclassificataion rate is very small.
The training AUC is 0.96, the testing AUC is 0.96

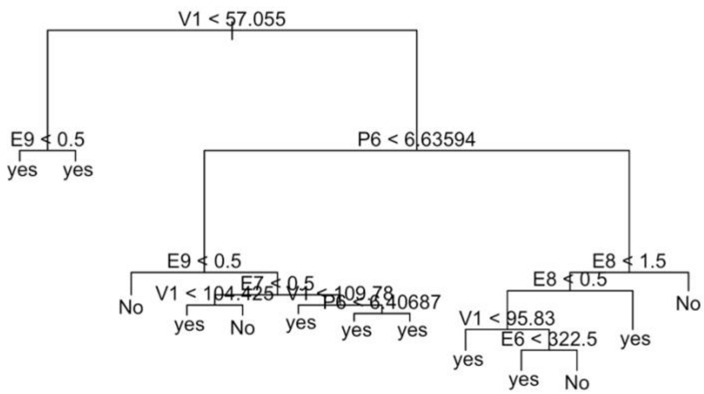
2) **Logistic Regression**
The variables are continuous variables and dummy variable for categorical variables
First, I run a glm with all variables, and turns out P1, E4, V5, V10, E3, E7, E8, E9, V3, V5, V10 are not significant compare to other variables, so I removed them from the model and run a glm model with the rest of the variables.
The second try gives me P6 are not significant, so I removed them and did again.

```
Call:
glm(formula = IsAlert ~ P3 + P6 + P7 + E1 + E2 + E6 + E10 + E11 +
    V1 + V8, family = "binomial", data = trainSet)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.5444  -0.7096   0.2512   0.6380   2.1595

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.881e+01  3.004e+00 -16.249  < 2e-16 ***
P3           1.367e-04  3.956e-05   3.456 0.000548 ***
P6           6.266e+00  3.874e-01  16.173  < 2e-16 ***
P7           1.523e-01  5.303e-03  28.721  < 2e-16 ***
E1           1.975e-02  1.523e-03  12.970  < 2e-16 ***
E2          -2.373e-03  1.701e-04 -13.952  < 2e-16 ***
E6          -4.177e-03  4.710e-04  -8.869  < 2e-16 ***
E10          5.520e-03  7.845e-04   7.037 1.97e-12 ***
E11         -2.580e-02  3.092e-03  -8.342  < 2e-16 ***
V1          -3.901e-02  4.957e-04 -78.694  < 2e-16 ***
V8          -8.744e-03  1.264e-03  -6.919 4.54e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

It turns out that all variables are significant at significance level 0+. So the best fit logistic regression model is with variables P3, P6, P7, E1, E2, E6, E10, E11, V1, V6, V8

This model’s training AUC is 0.88, and its testing AUC is 0.88
The logistic Regression does not perform as well as the neural network model, as its AUC is lower, but still in a good range.

3) **Decision Tree**
I use 75% of data to be the training set, and then I use tree function tree.fit to fit the best tree model into the data.
From training, the variable we choose are V1, E9, P6, E8, E6, E7

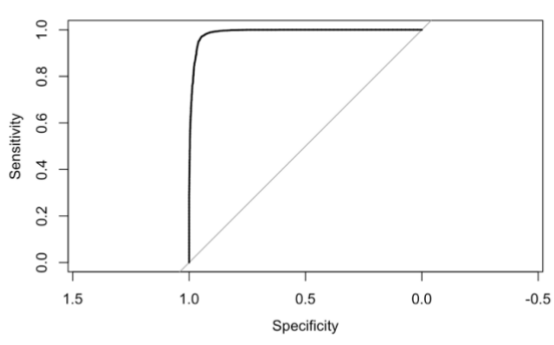


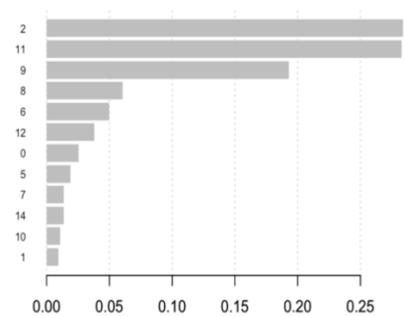

The training AUC is 0.93, the testing AUC is 0.93. So the decision tree model fit the data pretty well, better than logistic regression and slightly lower than neural network model.

**Application of learning technique**

**Bagging for decision tree**
I applied bagging to decision tree model and get a training AUC of 1, and testing AUC of 0.99, which is improved a lot compare to original decision tree's testing AUC of 0.93

**Boosting for logistic regression** (the code worked once, but not later, so I did not include this part in code, but still want to keep my output)




I choose to apply boosting to the logistic regression model.
The model improved, as the training AUC is 0.9909, and the testing AUC is 0.9892. while the original regression model has a training AUC of 0.88, and a testing AUC of 0.88. So the boosted model improves a lot. Since boost is used to reduce bias, the sensitivity graph and frequency graph shows a non-monotonic pattern, so there are bias in the data, thus the boosting helps to reduce the bias, and the model gets improved. With boosting, the boosted logistic regression model become the best fitted model among the three models we analyzed.

**Combine the models using ensemble**

For the multiple model, the predictor I choose are P3, P6, P7, E1, E2, E6, E10, E11, V1, V6,V8 given by the best fitted logistic regression model
The dependent variable is IsAlert

I used random forest and logistic regression model as my two models for multiple model. For random forest, the testing accuracy is 0.98. For logistic regression, the testing accuracy is 0.83

I choose gbm and glm as my top layor models and run twice to see which top layor model provides a better result.

When use generalized boosted model as the top layor model. The training AUC is around 0.99, the testing AUC is around 0.99. Both AUC show that the model fit the data very well.

When use logistic regression model as the top layor model. The training AUC is around 0.99, the testing AUC is around 0.99.  Both AUC show that the model fit the data very well.

So using gbm as top layor model perform slightly better.