STATS_204_HW5

Qi Wang

Question 1: There are both continuous variable and categorical variable in this data set. For pnr, it is just the ID of the observation, and three categories in type, 2 categories in preg and dead. The gvhd is the response variable and it is also categorical, so we need to fit a logistic regression. Here is the range and distribution of the continuous variable.

```
rm(list = ls())
library(ISwR)
library(car)
```

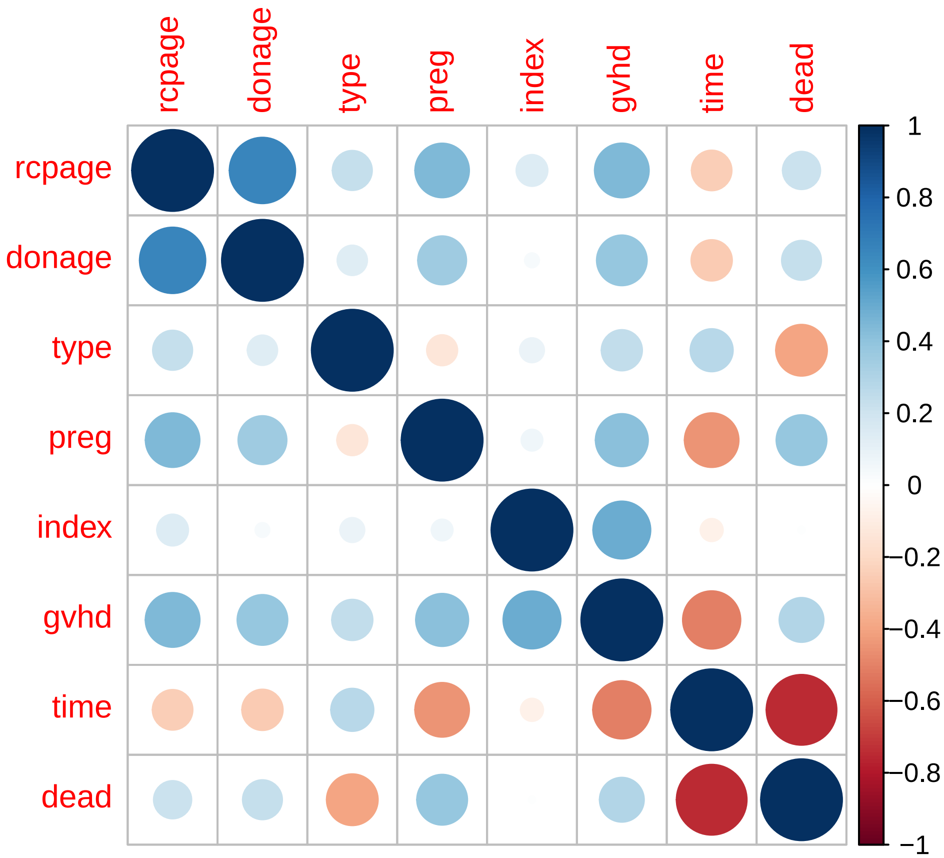
```
## Warning:   'car' R 4.1.2
```

```
##      carData
```

```
data1 <- graft.vs.host
summary(data1[,c(2,3,6,8)])
```

```
##      rcpage          donage          index             time       
##  Min.   :13.00   Min.   :14.00   Min.   : 0.270   Min.   :  41.0  
##  1st Qu.:20.00   1st Qu.:20.00   1st Qu.: 0.920   1st Qu.: 177.0  
##  Median :23.00   Median :23.00   Median : 2.010   Median : 667.0  
##  Mean   :25.43   Mean   :25.81   Mean   : 2.556   Mean   : 669.8  
##  3rd Qu.:29.00   3rd Qu.:34.00   3rd Qu.: 3.730   3rd Qu.:1105.0  
##  Max.   :43.00   Max.   :43.00   Max.   :10.110   Max.   :1504.0  
```

```
corrplot::corrplot(cor(data1[,2:ncol(data1)]))
```



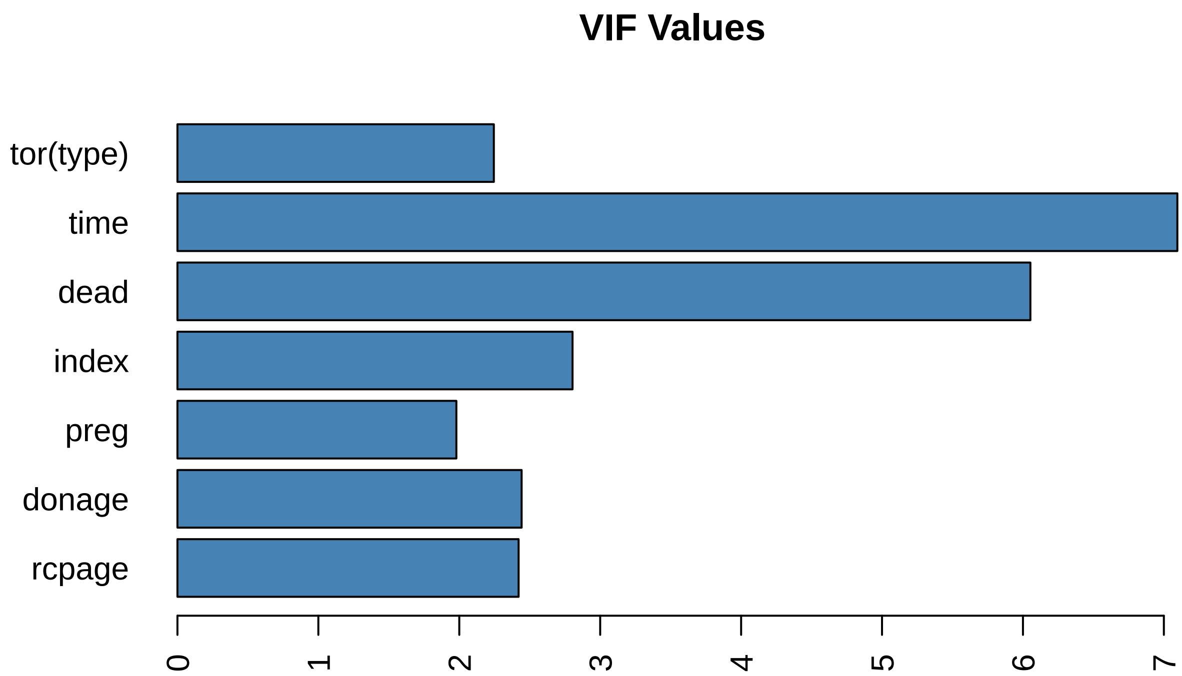
First, I will use VIF to check whether there are some correlations among the variables.

Now I will first use the non-transformed index to fit the regression:

```
M1 <- glm(gvhd ~ rcpage + donage  + preg + index  + dead + time + factor(type) , family = binomial(link
```

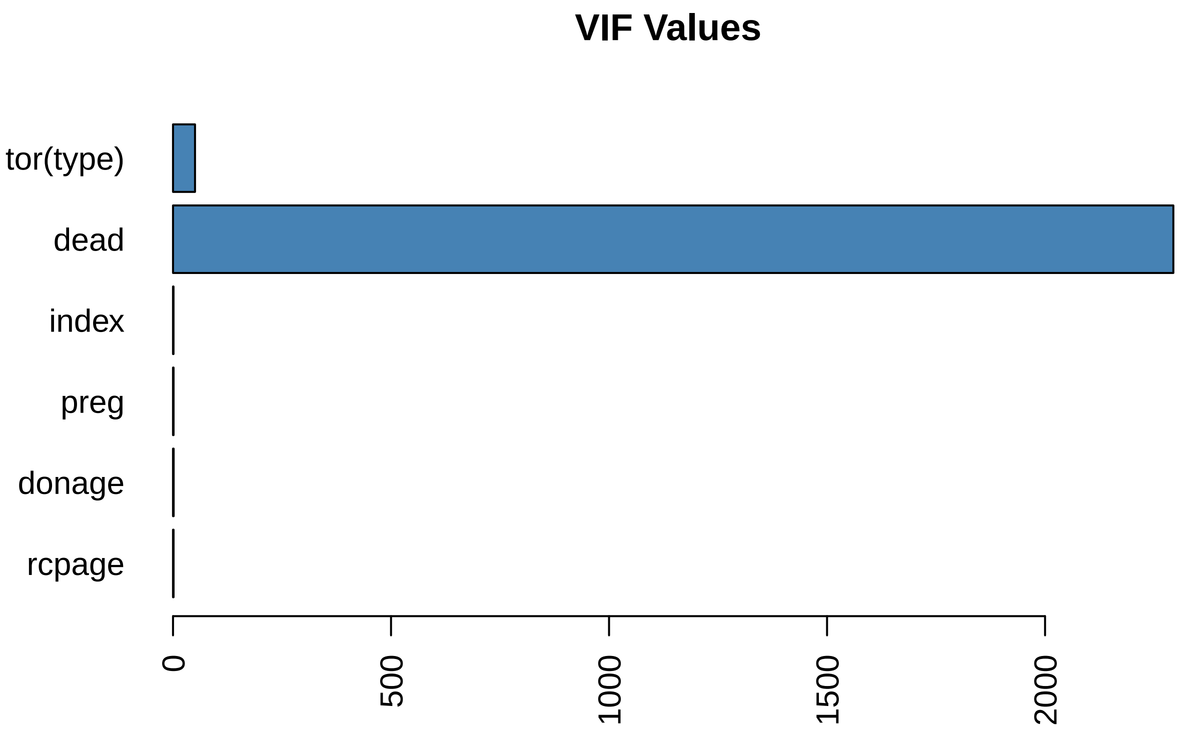
```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
barplot(vif(M1)[,3], main = "VIF Values", horiz = TRUE, col = "steelblue", las = 2)
```



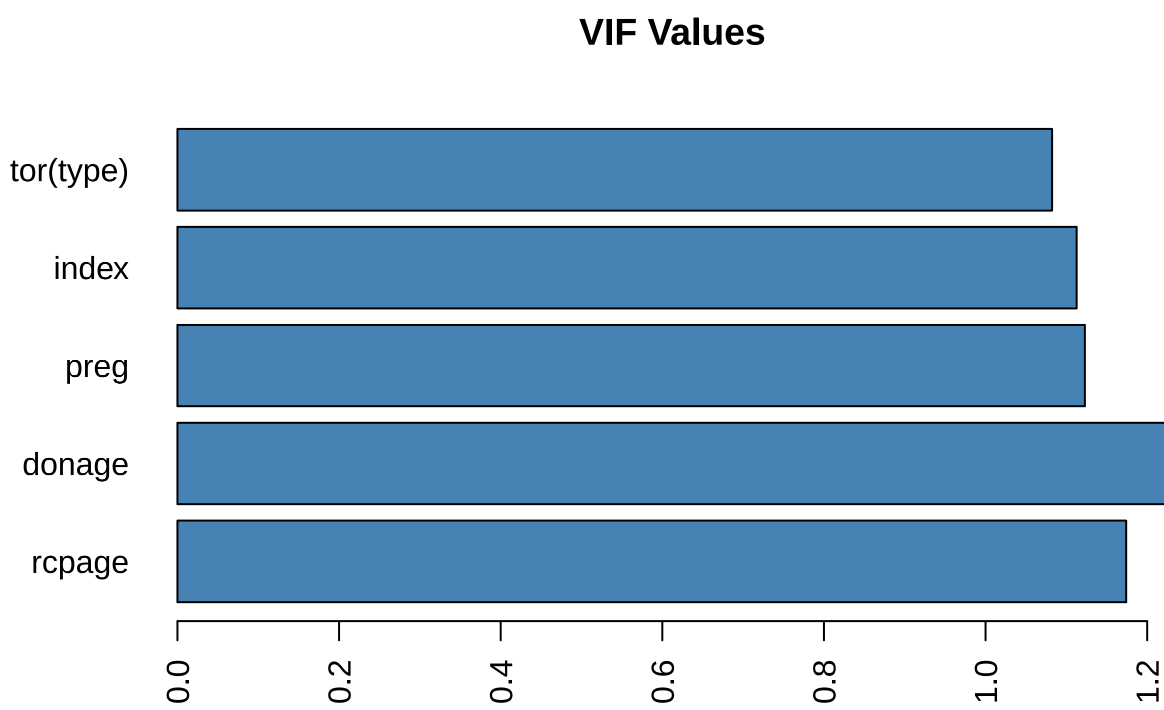
It is obvious that the time type and dead has large VIF, I will first delete the variable time from the model:

```
M1_time <- glm(gvhd ~ rcpage + donage  + preg + index + dead + factor(type), family = binomial(link = "l
barplot(vif(M1_time)[,3], main = "VIF Values", horiz = TRUE, col = "steelblue", las = 2)
```



There are still co-linearity exists, and we need to delete the dead variable from the model:

```
M1_best_base <- glm(gvhd ~ rcpage + donage  + preg + index + factor(type), family = binomial(link = "lo
barplot(vif(M1_best_base)[,3], main = "VIF Values", horiz = TRUE, col = "steelblue", las = 2)
```



Here the VIF seems nice and almost no co-linearity exists in the model. Now I will use step function to make model selection based on the left variables, the left variables are index, preg and donage

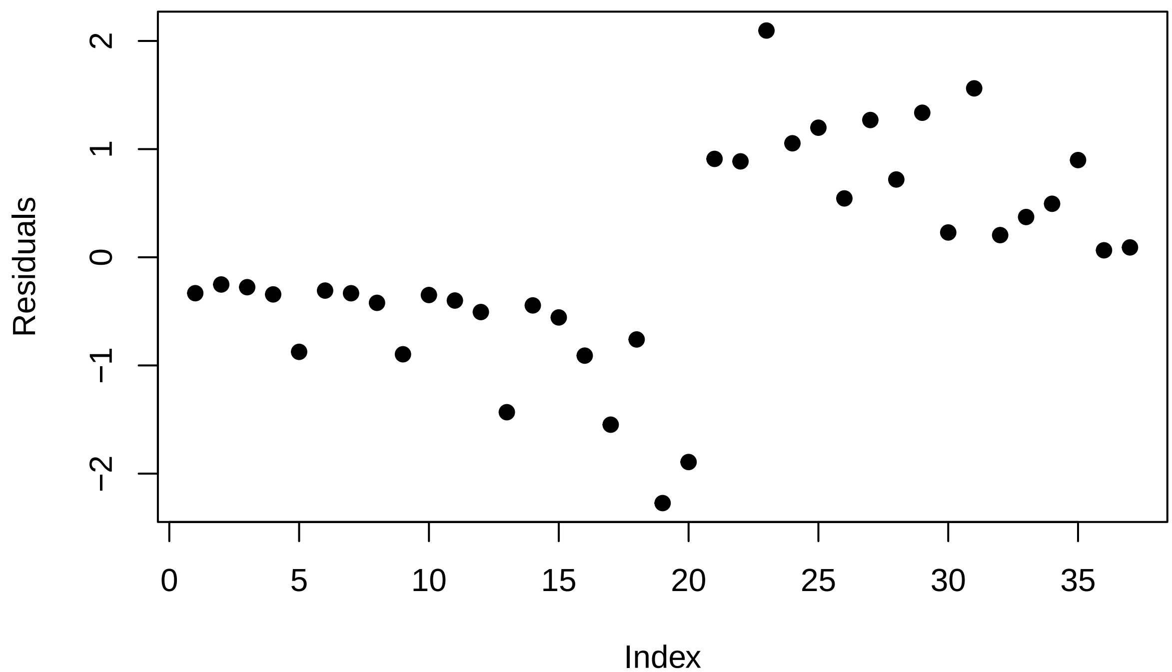
```
#step(M1_best_base)
M1_best <- glm(gvhd ~  donage + preg  + index, family = binomial(link = "logit"), data = data1, maxit =
summary(M1_best)
```

```
##
## Call:
## glm(formula = gvhd ~ donage + preg + index, family = binomial(link = "logit"),
##     data = data1, maxit = 100)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0716  -0.4978  -0.2732   0.6925   1.9978
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.88275    2.22347  -2.646  0.00815 **
## donage       0.11925    0.06261   1.905  0.05682 .
## preg         1.55904    1.01886   1.530  0.12597
## index        0.88989    0.37068   2.401  0.01637 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 51.049  on 36  degrees of freedom
## Residual deviance: 29.848  on 33  degrees of freedom
## AIC: 37.848
##
## Number of Fisher Scoring iterations: 5
```

In the residual plot as follows, we can see almost no trend exists in the model. The residual deviance is around 30 and not that big.

```
plot(rstudent(M1_best, type = "pearson"), pch = 19, ylab = "Residuals")
```

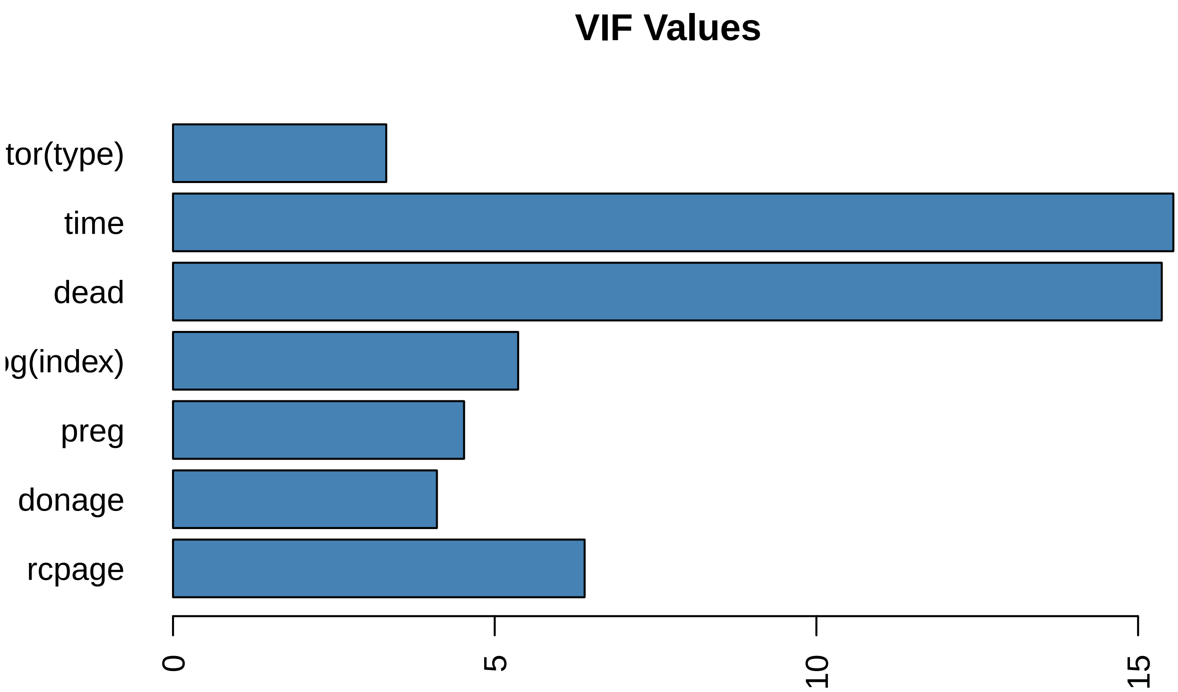


Now let's try the logarithm of index. First steps are similar since we still need to delete the variables that have strong co-linearity.

```
M2 <- glm(gvhd ~ rcpage + donage  + preg + log(index)  + dead + time + factor(type) , family = binomial
```

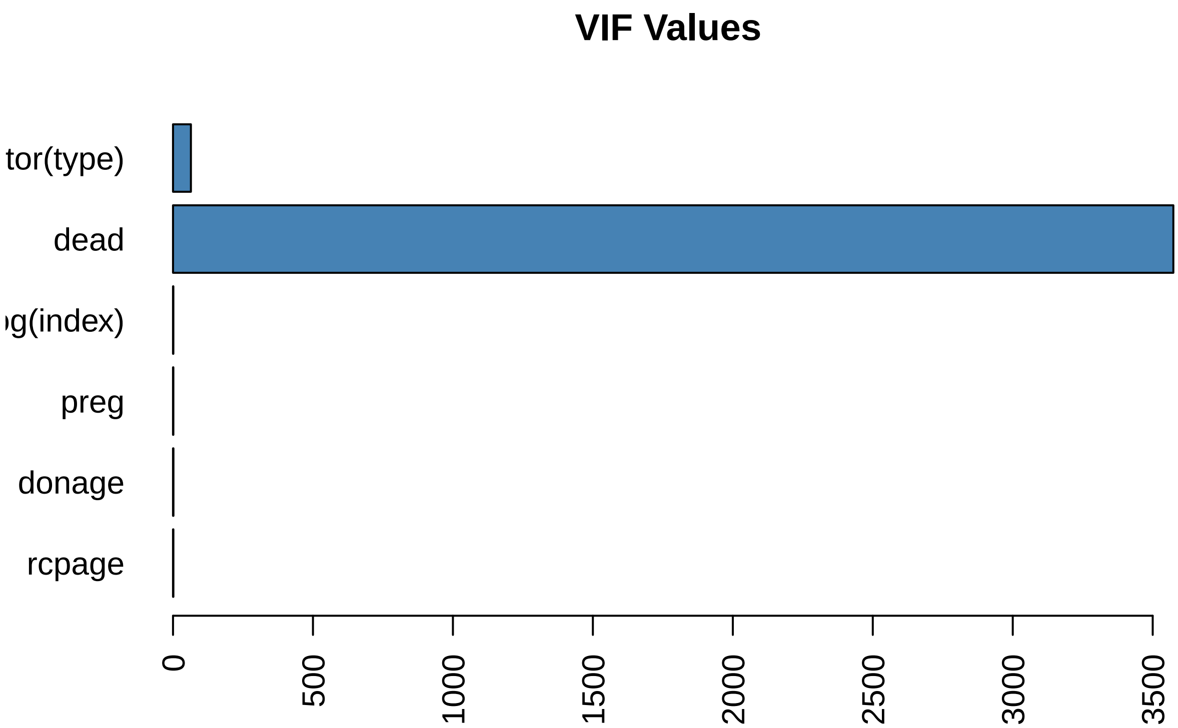
```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
barplot(vif(M2)[,3], main = "VIF Values", horiz = TRUE, col = "steelblue", las = 2)
```



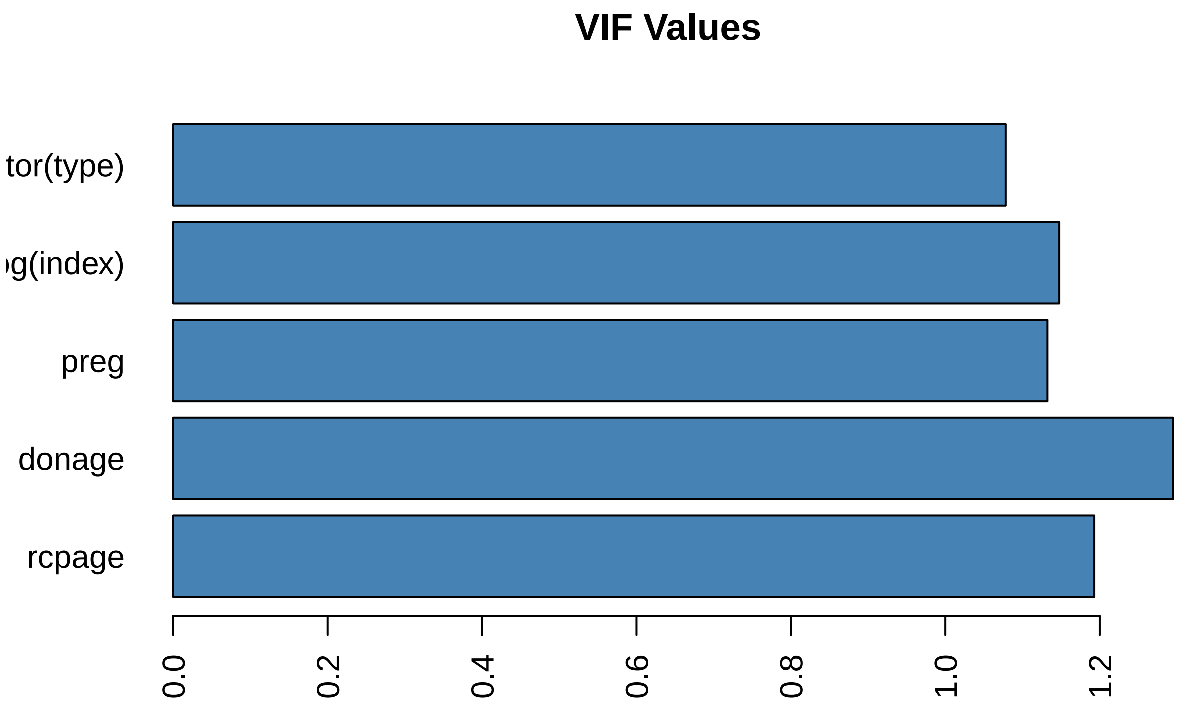
We still need to delete the time variable:

```
M2_time <- glm(gvhd ~ rcpage + donage  + preg + log(index)  + dead + factor(type) , family = binomial(l
barplot(vif(M2_time)[,3], main = "VIF Values", horiz = TRUE, col = "steelblue", las = 2)
```



Then we should delete the dead variable, too.

```
M2_time_dead <- glm(gvhd ~ rcpage + donage  + preg + log(index) + factor(type) , family = binomial(link
barplot(vif(M2_time_dead)[,3], main = "VIF Values", horiz = TRUE, col = "steelblue", las = 2)
```



Now it shows no coliearity, so let's do the model selection:

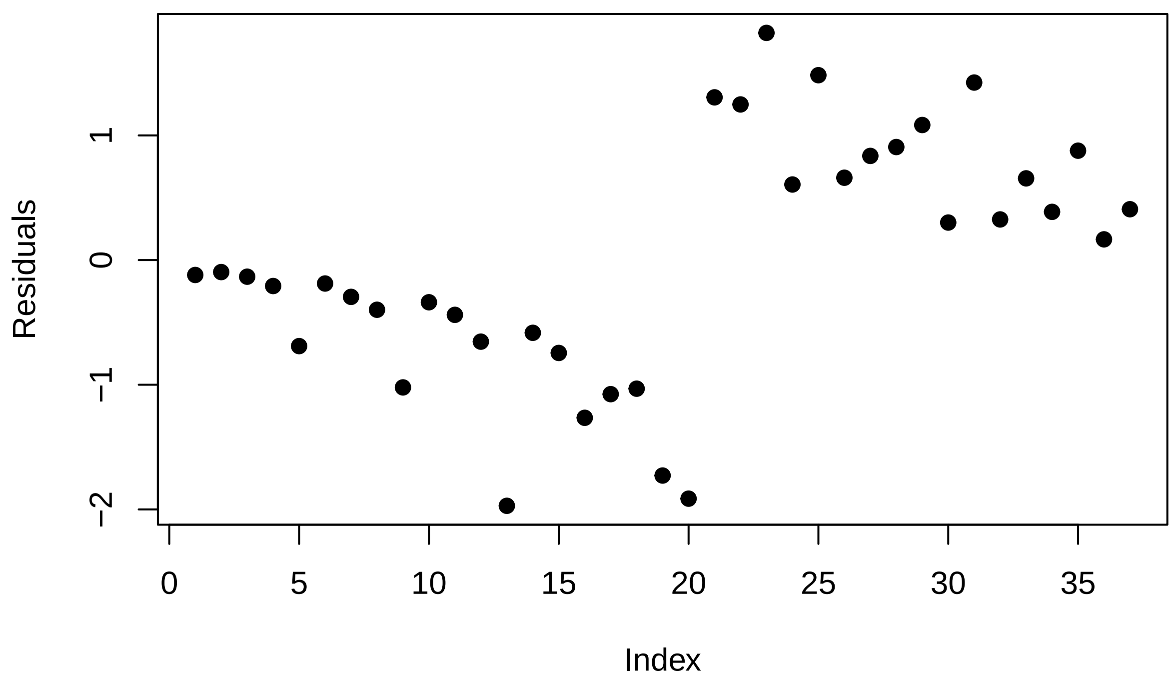
```
#step(M2_time_dead)
M2_best <- glm(gvhd ~ donage  + log(index) , family = binomial(link = "logit"), data = data1, maxit = 10
summary(M2_best)
```

```
##
## Call:
## glm(formula = gvhd ~ donage + log(index), family = binomial(link = "logit"),
##     data = data1, maxit = 100)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8298  -0.6412  -0.1189   0.6440   1.7503
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.45399    2.08147  -2.620  0.00879 **
## donage       0.14594    0.06465   2.257  0.02399 *
## log(index)   2.17773    0.78986   2.757  0.00583 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 51.049  on 36  degrees of freedom
```

```
## Residual deviance: 31.068  on 34  degrees of freedom
## AIC: 37.068
##
## Number of Fisher Scoring iterations: 5
```

This model has smaller AIC than the before one, then let's check the residuals:

```
plot(rstudent(M2_best, type = "pearson"), pch = 19, ylab = "Residuals")
```



There still seems to be no trend and the residual deviance is close to the degrees of freedom.

Question 2:

(a)

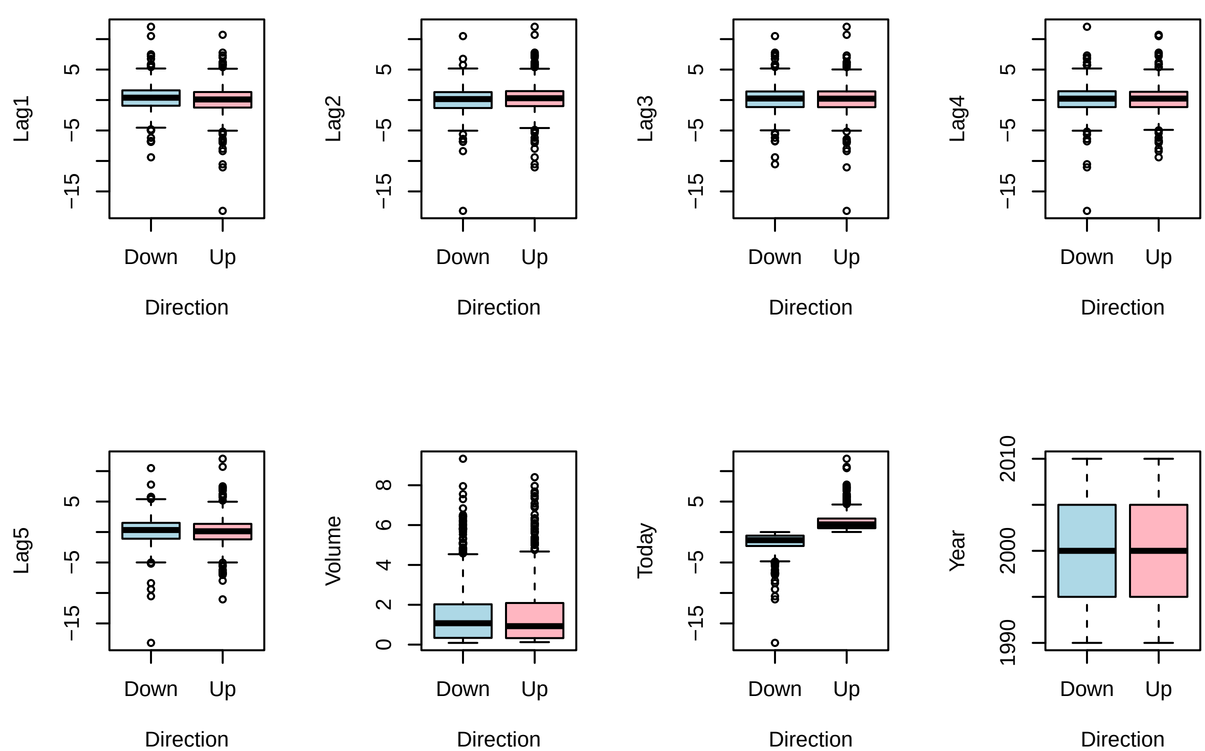
```
library(ISLR)
data2 <- Weekly
summary(Weekly)
```

```
##       Year          Lag1               Lag2               Lag3
##  Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950
##  1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580
##  Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410
##  Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472
##  3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090
##  Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260
```

```
##       Lag4               Lag5              Volume             Today
##  Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747   Min.   :-18.1950
##  1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202   1st Qu.: -1.1540
##  Median :  0.2380   Median :  0.2340   Median :1.00268   Median :  0.2410
##  Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462   Mean   :  0.1499
##  3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373   3rd Qu.:  1.4050
##  Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821   Max.   : 12.0260
##  Direction
##  Down:484
##  Up  :605
##
##
##
##
```

For variable Volume, the maximum is too large and far from the median and even 3rd quantile.

```
par(mfrow = c(2,4))
boxplot(Lag1 ~ Direction, data = data2, col = c("lightblue", "lightpink"))
boxplot(Lag2 ~ Direction, data = data2, col = c("lightblue", "lightpink"))
boxplot(Lag3 ~ Direction, data = data2, col = c("lightblue", "lightpink"))
boxplot(Lag4 ~ Direction, data = data2, col = c("lightblue", "lightpink"))
boxplot(Lag5 ~ Direction, data = data2, col = c("lightblue", "lightpink"))
boxplot(Volume ~ Direction, data = data2, col = c("lightblue", "lightpink"))
boxplot(Today ~ Direction, data = data2, col = c("lightblue", "lightpink"))
boxplot(Year ~ Direction, data = data2, col = c("lightblue", "lightpink"))
```



From the box plot, there mean of the group up are higher than that of the group down. For other variables the difference are not that significant.

(b)

```r
M_week <- glm(Direction ~ .- Year - Today, data = data2, family = binomial(link = "logit"))
summary(M_week)
```

```
##
## Call:
## glm(formula = Direction ~ . - Year - Today, family = binomial(link = "logit"),
##     data = data2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6949  -1.2565   0.9913   1.0849   1.4579
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## Lag1        -0.04127    0.02641  -1.563   0.1181
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
## Lag5        -0.01447    0.02638  -0.549   0.5833
## Volume      -0.02274    0.03690  -0.616   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

It seems that the Lag2 is significant at 0.05 level, but other variables seem not to be significant.

(c)

```r
library(caret)
```

```
## Warning:   'caret' R 4.1.2
```

```
##      ggplot2
```

```
##      lattice
```

```
prob <- predict(M_week, type = "response")
pre_dir <- ifelse(prob >= 0.5, "Up", "Down")
attach(data2)
table(pre_dir, Direction)
```

```
##        Direction
## pre_dir Down  Up
##    Down   54  48
##    Up    430 557
```

So the true fraction is:

```
mean(pre_dir == data2$Direction)
```

```
## [1] 0.5610652
```

The model does not perform well, it predicts most of the probability over 0.5 and give most of the predictions "Up". The right upper number 48 means that there are 48 wrong predictions whose true value is "Up" but the model predicts them as "down". The lower left number 430 means that there are 430 wrong predictions whose tru value is "Down", but our model predicted it as "Up".

(d)

```
dat_tr <- data2[which(data2$Year >= 1990 & data2$Year <= 2008), c(3,9)]
dat_te <- data2[which(data2$Year >= 2009 & data2$Year <= 2010), c(3,9)]
M_tr <- glm(Direction ~ Lag2, family = binomial(link = "logit"), data = dat_tr)
pre_prob <- predict(M_tr, newdata = data.frame(Lag2 = dat_te$Lag2), type = "response")
pre_direction <- ifelse(pre_prob >= 0.5, "Up", "Down")
attach(dat_te)
```

```
## The following objects are masked from data2:
##
##     Direction, Lag2
```

```
table(pre_direction, Direction)
```

```
##              Direction
## pre_direction Down Up
##          Down    9  5
##          Up     34 56
```

The overall fraction of correct predictions are:

```
mean(pre_direction == Direction)
```

```
## [1] 0.625
```

The rate is rising a little.

Question 3: (a)

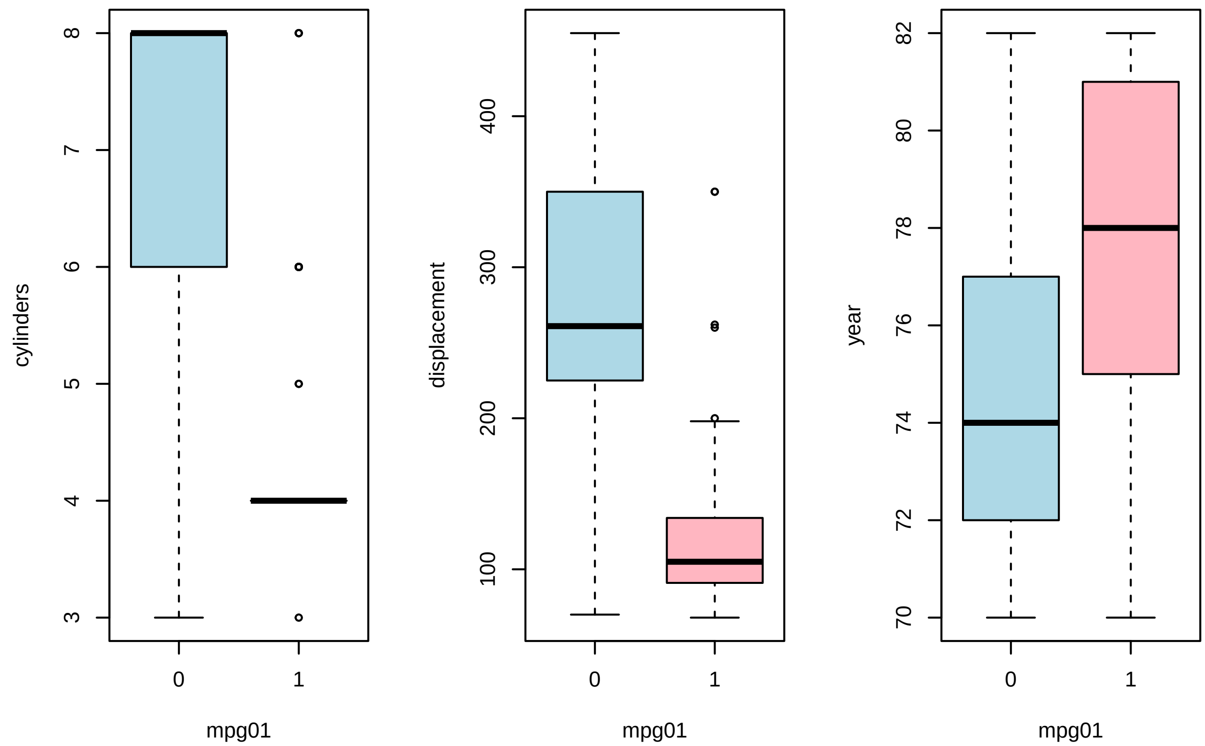
```
data3 <- Auto
mpg01 <- ifelse(data3$mpg >= median(data3$mpg), 1, 0)

data3_new <- data.frame(mpg01 = mpg01, cylinders = Auto$cylinders, displacement = Auto$displacement,
                        horsepower = Auto$horsepower, weight = Auto$weight, acceleration = Auto$accelera
                        year = Auto$year, origin = Auto$origin, name = Auto$name)
```

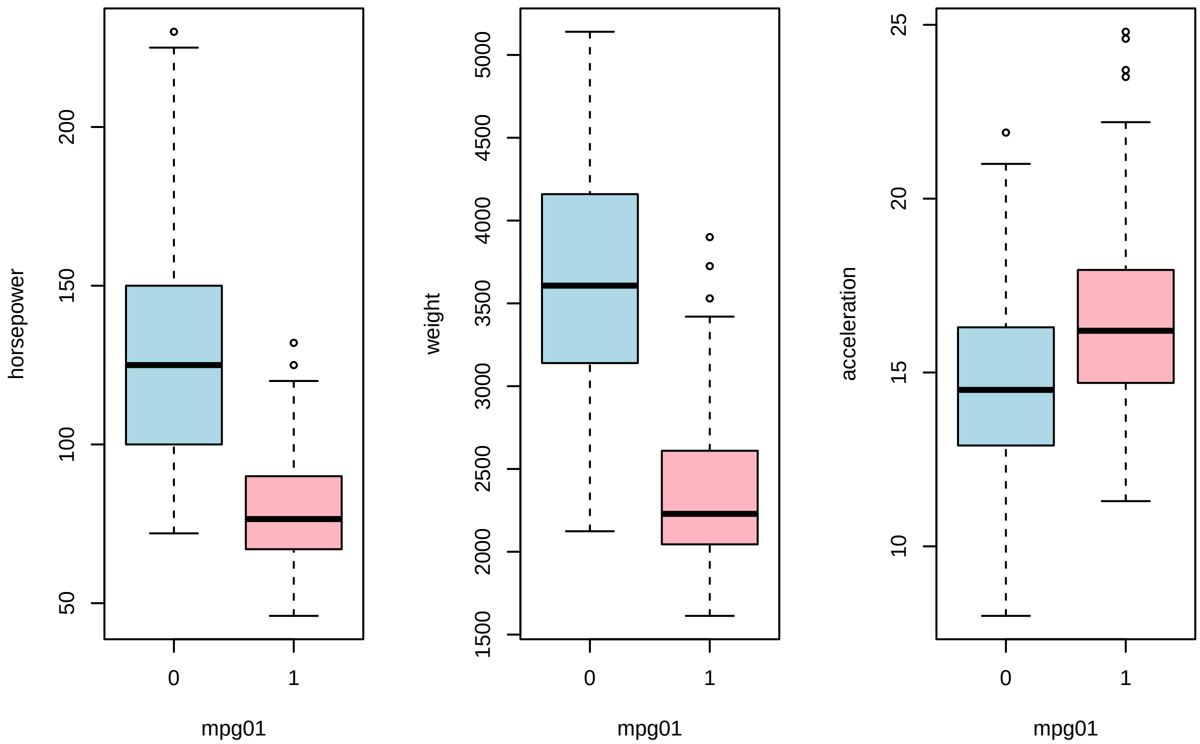
(b)

```
par(mfrow = c(1,3))

boxplot(cylinders ~ mpg01, data = data3_new, col = c("lightblue","lightpink"))
boxplot(displacement~ mpg01, data = data3_new, col = c("lightblue","lightpink"))
boxplot(year~ mpg01, data = data3_new, col = c("lightblue","lightpink"))
```




```
par(mfrow = c(1,3))
boxplot(horsepower ~ mpg01, data = data3_new, col = c("lightblue","lightpink"))
boxplot(weight ~ mpg01, data = data3_new, col = c("lightblue","lightpink"))
boxplot(acceleration ~ mpg01, data = data3_new, col = c("lightblue","lightpink"))
```



From the box plot, we can see that for those cars which mpg is less than the median, they tend to have more cylinders, more displacements, more horsepower more weight and less acceleration. I believe cylinders, displacements, horsepower and weight may be most useful ones in predicting the mpg.

(c)

I will randomly select 80% of the data without replacement as the training data set and the rest are the test data set.

```
index <- sample(1:nrow(data3_new), round(0.8*nrow(data3_new)), replace = F)
data3_new_tr <- data3_new[index,]
data3_new_te <- data3_new[-index,]
```

(f) I will include all the variables and then use AIC criteria to do model selection.

```
M_mpg <- glm(mpg01 ~ cylinders + displacement + horsepower + weight + acceleration + year, data = data3_
summary(M_mpg)
```

```
##
## Call:
## glm(formula = mpg01 ~ cylinders + displacement + horsepower +
##     weight + acceleration + year, family = binomial(link = "logit"),
##     data = data3_new_tr)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -2.1071  -0.1251  -0.0009   0.2391   3.2765
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -13.370129   6.086750  -2.197    0.028 *
## cylinders     -0.254048   0.481738  -0.527    0.598
## displacement  -0.003529   0.011469  -0.308    0.758
## horsepower    -0.047912   0.026249  -1.825    0.068 .
## weight        -0.003201   0.001140  -2.807    0.005 **
## acceleration  -0.041249   0.156542  -0.263    0.792
## year           0.385659   0.076897   5.015  5.3e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 435.09  on 313  degrees of freedom
## Residual deviance: 133.17  on 307  degrees of freedom
## AIC: 147.17
##
## Number of Fisher Scoring iterations: 7
```

And I will use the step wise AIC criteria to do the model selection:

```
#step(M_mpg)
M_mpg_tr <- glm(mpg01 ~ displacement + horsepower + weight  + year, data = data3_new_tr, family = binomi
summary(M_mpg_tr)
```

```
##
## Call:
## glm(formula = mpg01 ~ displacement + horsepower + weight + year,
##     family = binomial(link = "logit"), data = data3_new_tr)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0498  -0.1362  -0.0007   0.2337   3.2792
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.461e+01  5.076e+00  -2.878  0.00401 **
## displacement -7.992e-03  7.105e-03  -1.125  0.26065
## horsepower   -4.529e-02  1.715e-02  -2.641  0.00826 **
## weight       -3.286e-03  9.992e-04  -3.289  0.00101 **
## year          3.862e-01  7.645e-02   5.051 4.38e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 435.09  on 313  degrees of freedom
## Residual deviance: 133.51  on 309  degrees of freedom
## AIC: 143.51
```

```
##
## Number of Fisher Scoring iterations: 7
```

Here is the confusion matrix:

```r
pred_prob <- predict(M_mpg_tr, newdata = data.frame(displacement = data3_new_te$displacement, horsepowe
                                                    weight = data3_new_te$weight, year = data3_new_te$ye
pred_mpg <- ifelse(pred_prob >= 0.5, 1, 0)

Real <- data3_new_te$mpg01
table(Predicted = pred_mpg, Real)
```

```
##          Real
## Predicted  0  1
##         0 35  6
##         1  0 37
```

Therefore, the percentage of correctly predicted data are:

```r
mean(pred_mpg == Real)
```

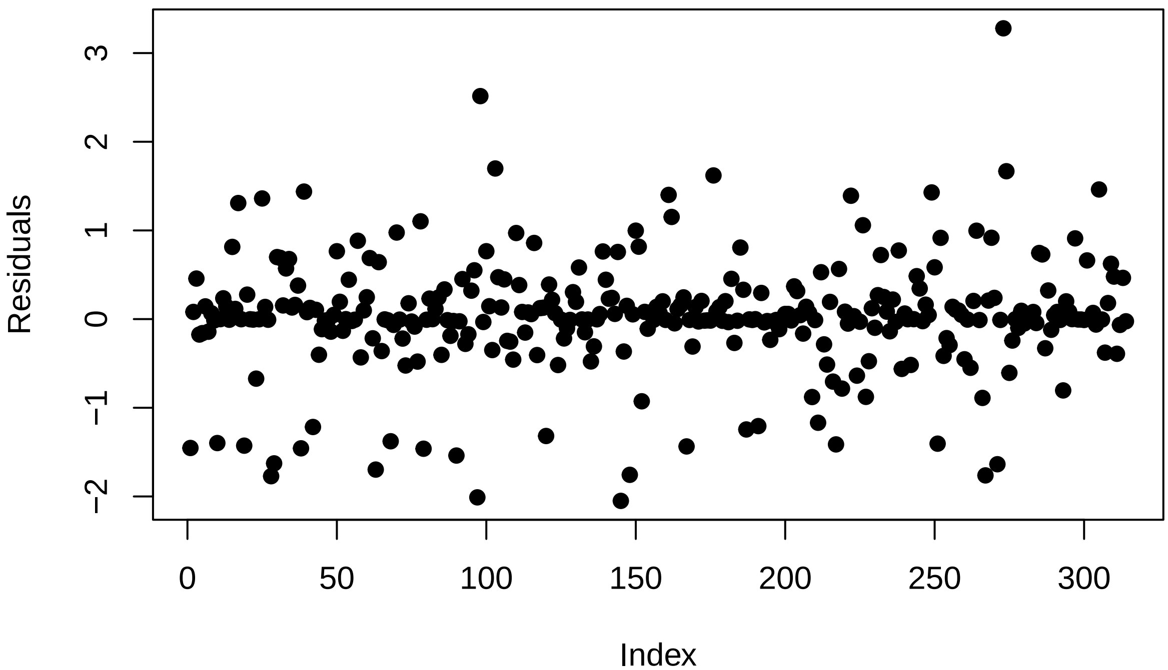
```
## [1] 0.9230769
```

Therefore, the error rate fot the prediction is:

```r
1-mean(pred_mpg == Real)
```

```
## [1] 0.07692308
```

```r
plot(residuals(M_mpg_tr), pch = 19, ylab = "Residuals")
```



Residuals seem nice. And prediction is nice.