# Logistic Regression Assignment–Audit

*Saixiong Han (sah178)*

*January 31, 2017*

Task: Analyze dataset analyze dataset audit.csv. The objective is to predict the binary (TARGET_Adjusted) and continuous (RISK_Adjustment) target variables.

## 1. Identify and report response variable and predictor

```
audit <- read.csv("C:/Users/daisy/OneDrive/Study/DM/week3/audit.csv", header = TRUE,
    sep = ",", stringsAsFactors = TRUE)
head(audit)
```

```
##        ID Age Employment Education   Marital Occupation    Income Gender
## 1 1004641  38    Private    College Unmarried    Service  81838.00 Female
## 2 1010229  35    Private Associate    Absent  Transport  72099.00   Male
## 3 1024587  32    Private     HSgrad  Divorced   Clerical 154676.74   Male
## 4 1038288  45    Private   Bachelor   Married     Repair  27743.82   Male
## 5 1044221  60    Private    College   Married  Executive   7568.23   Male
## 6 1047095  74    Private     HSgrad   Married    Service  33144.40   Male
##   Deductions Hours RISK_Adjustment TARGET_Adjusted
## 1          0    72               0               0
## 2          0    30               0               0
## 3          0    40               0               0
## 4          0    55            7298               1
## 5          0    40           15024               1
## 6          0    30               0               0
```

RISK_Adjustment and TARGET_Adjusted are the response variable. RISK_Adjustment is numeric, so we have to use linear and non-linear regression to predict it. While TARGET_Adjusted is binary, thus we have to use logistic regression to predict it. The rest predictors includes Age, Employment, Education, Marital, Occupation, Income, Gender, Deductions,Hours.

## 2.Explore data and generate summary

### -Data Preparation

There are some missing value in the data and useless variable in the dataset. Before generate summary, we need to deal with the useless variables and missing value first. Since ID in the dataset is useless. we can delete this variable from dataset first.

```
audit1 <- audit[, 2:12]
dim(audit1)
```

```
## [1] 2000   11
```

```
summary(audit1)
```

```
##       Age             Employment       Education
##  Min.   :17.00   Private    :1411   HSgrad    :660
##  1st Qu.:28.00   Consultant: 148   College   :442
```

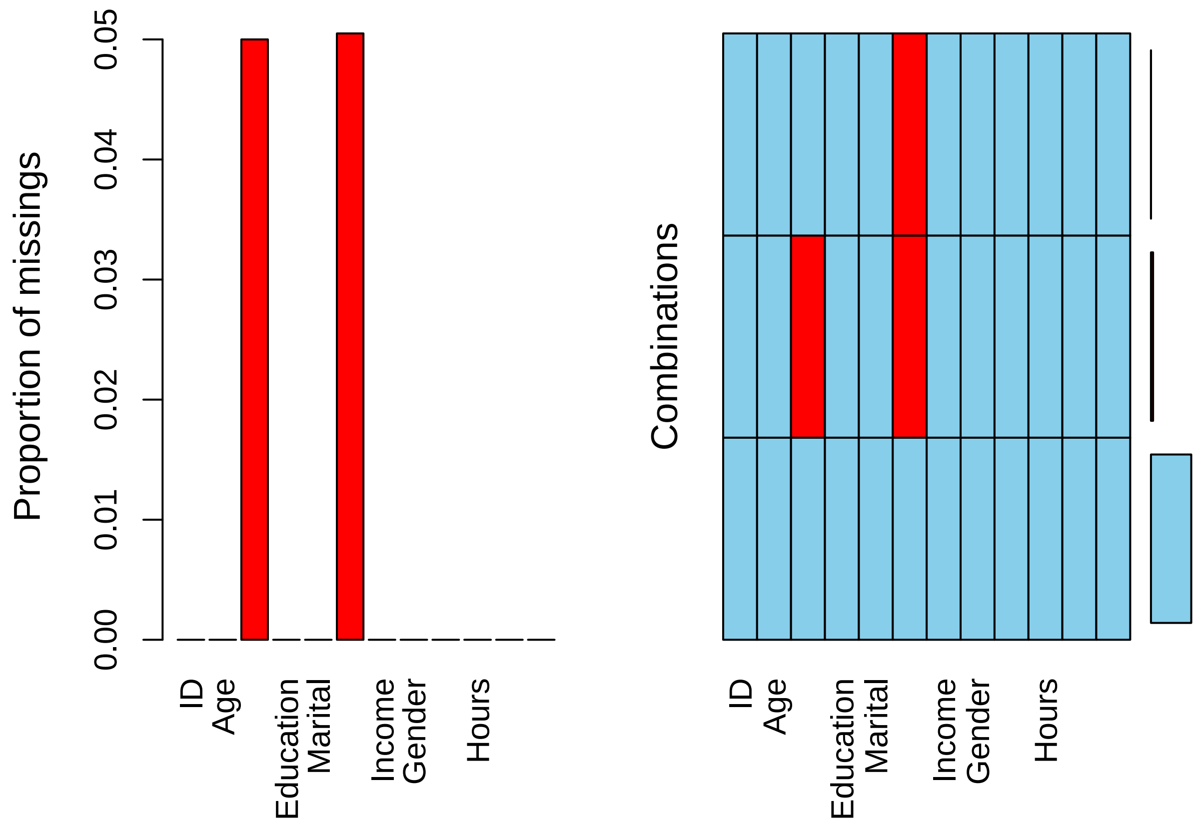
```
##  Median :37.00   PSLocal   : 119   Bachelor  :345  
##  Mean   :38.62   SelfEmp   :  79   Master    :102  
##  3rd Qu.:48.00   PSState   :  72   Vocational: 86  
##  Max.   :90.00   (Other)   :  71   Yr11      : 74  
##                  NA's      : 100   (Other)   :291  
##                   Marital            Occupation      Income        
##  Absent               :669   Executive   :289   Min.   :   609.7  
##  Divorced             :266   Professional:247   1st Qu.: 34433.1  
##  Married              :917   Clerical    :232   Median : 59768.9  
##  Married-spouse-absent: 22   Repair      :225   Mean   : 84688.5  
##  Unmarried            : 67   Service     :210   3rd Qu.:113842.9  
##  Widowed              : 59   (Other)     :696   Max.   :481259.5  
##                              NA's        :101                     
##     Gender       Deductions          Hours       RISK_Adjustment 
##  Female: 632   Min.   :   0.00   Min.   : 1.00   Min.   : -1453  
##  Male  :1368   1st Qu.:   0.00   1st Qu.:38.00   1st Qu.:     0  
##                Median :   0.00   Median :40.00   Median :     0  
##                Mean   :  67.57   Mean   :40.07   Mean   :  2021  
##                3rd Qu.:   0.00   3rd Qu.:45.00   3rd Qu.:     0  
##                Max.   :2904.00   Max.   :99.00   Max.   :112243  
##                                                                  
##  TARGET_Adjusted 
##  Min.   :0.0000  
##  1st Qu.:0.0000  
##  Median :0.0000  
##  Mean   :0.2315  
##  3rd Qu.:0.0000  
##  Max.   :1.0000  
## 
```

The summary shows Age, Income, Deductions, Hours, RISK_Adjustment are all numerical variables. Employment, Education, Marital, Occupation, Gerder and TARGET_Adjusted are categotical variables. In aqddition, there are about 200 missing value in Employment and Occupation. We can generate a new level for the misssing value in Employment and Occupation.

### -Deal with missing data

In order to know the distribution of missing data, the first thing I would like yo do is spelling the pattern of missing data.

```r
library(VIM)
library(mice)
aggr(audit)
```




The graph shows Employment has 100 missing value and Occupation has 101 missing value. Since the missing value are shown as NA, we can add a new level for the missing value since we don't know the employment or occupation situation. In Employment, we add "NewEmploy" as a new level, while in Occupation, we add "NewOccupy" as a new level.

```
audit2 = audit1

levels(audit2$Employment) = c(levels(audit2$Employment), "NewEmploy")
audit2$Employment[is.na(audit2$Employment)] = "NewEmploy"
summary(audit2$Employment)

levels(audit2$Occupation) = c(levels(audit2$Occupation), "NewOccupy")
audit2$Occupation[is.na(audit2$Occupation)] = "NewOccupy"
summary(audit2$Occupation)
```

#### a–generate the summary table

For each numeric variable, list:name, mean, median, 1st quartile, 3rd quartile, standard deviation. From the summary we can know, Age, Income, Deductions, Hours, RISK_Adjustment are numerical variables. The summary table is as following.

```
library(knitr)
Age = c(summary(audit2$Age), sd(audit2$Age))
Income = c(summary(audit2$Income), sd(audit2$Income))
Deductions = c(summary(audit2$Deductions), sd(audit2$Deductions))
Hours = c(summary(audit2$Hours), sd(audit2$Hours))
RISK_Adjustment = c(summary(audit2$RISK_Adjustment), sd(audit2$RISK_Adjustment))
```

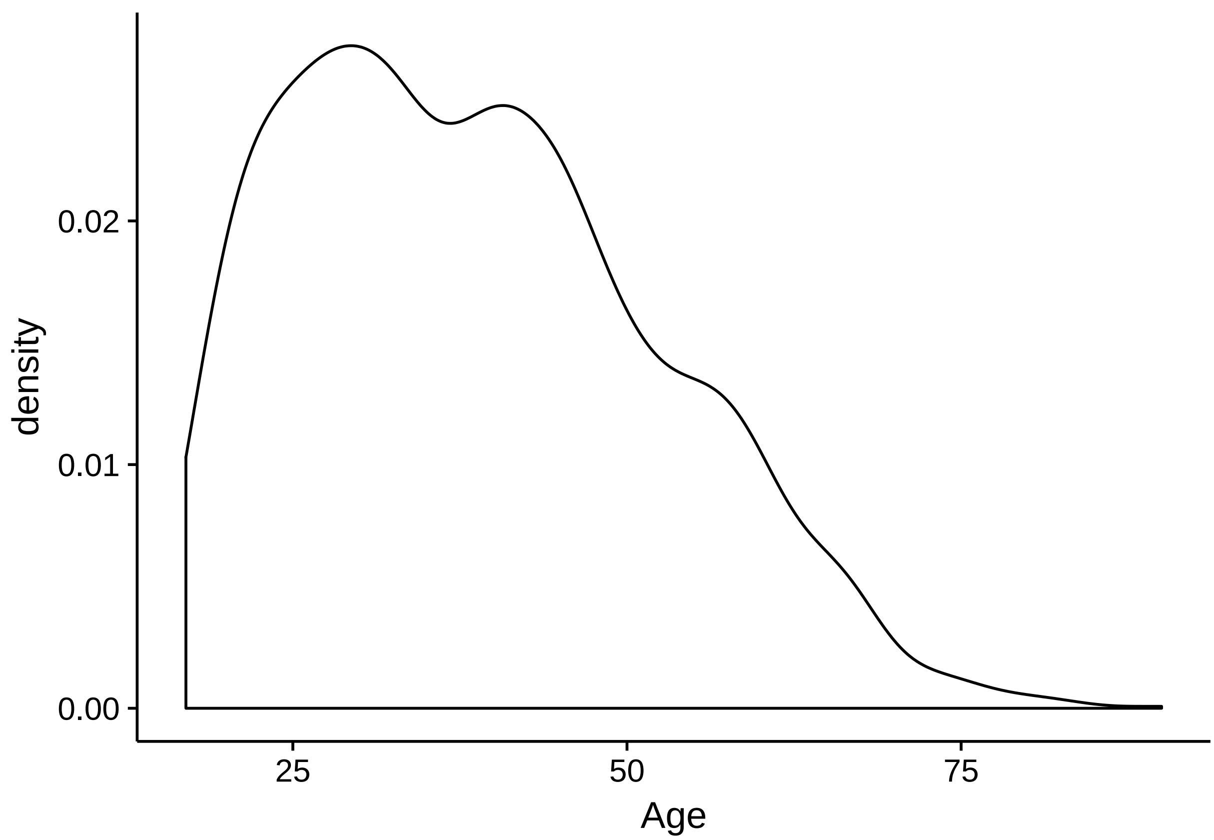
```
result = rbind(Age, Income, Deductions, Hours, RISK_Adjustment)
result = as.data.frame(result)
colnames(result)[7] = c("sd")
kable(result, caption = "Table 1: Summary of attributes")
```

Table 1: Table 1: Summary of attributes

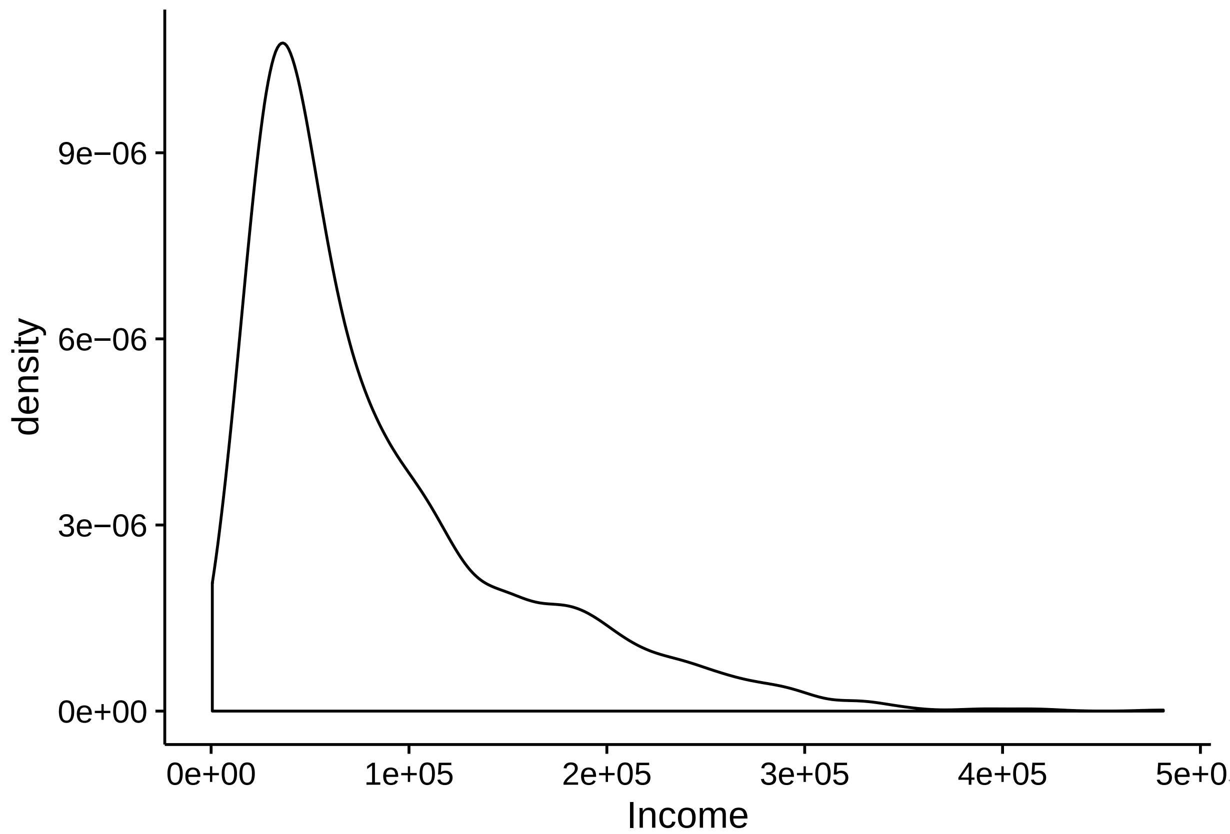
| | Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. | sd |
|---|---|---|---|---|---|---|---|
| Age | 17.0 | 28 | 37 | 38.62 | 48 | 90 | 13.58475 |
| Income | 609.7 | 34430 | 59770 | 84690.00 | 113800 | 481300 | 69621.64450 |
| Deductions | 0.0 | 0 | 0 | 67.57 | 0 | 2904 | 340.70470 |
| Hours | 1.0 | 38 | 40 | 40.07 | 45 | 99 | 12.15372 |
| RISK_Adjustment | -1453.0 | 0 | 0 | 2021.00 | 0 | 112200 | 8341.87229 |

**b–plot density distribution for for numeric variables**

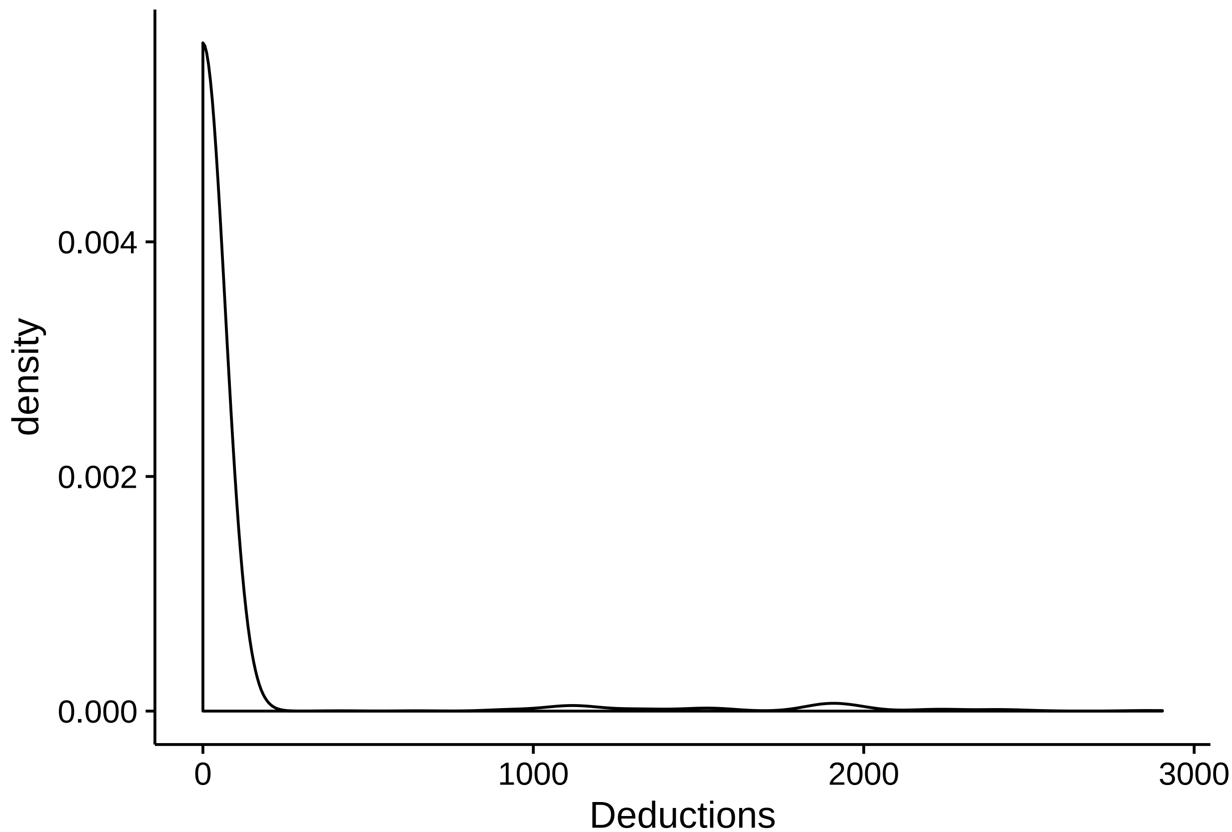
```
library(cowplot)
ggplot(data = audit2, aes(x = Age)) + geom_density()
```



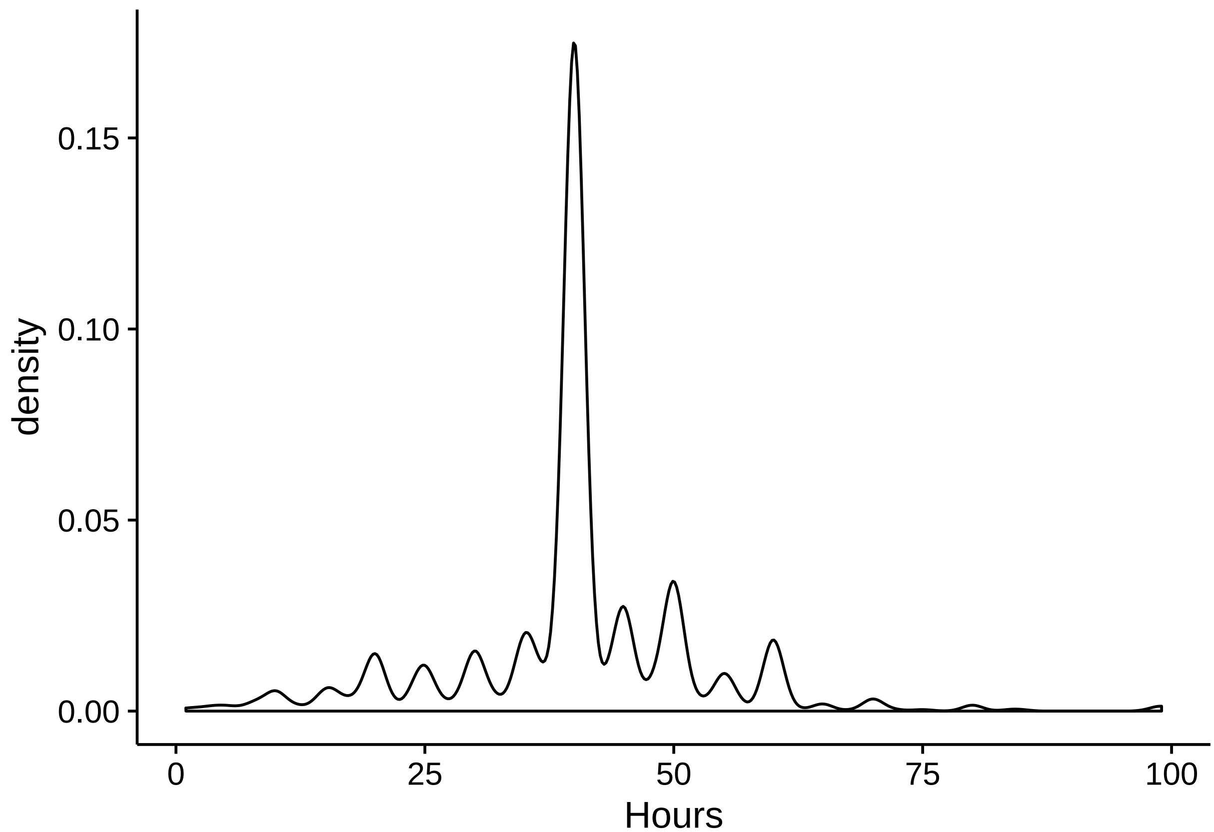

```
library(cowplot)
ggplot(data = audit2, aes(x = Income)) + geom_density()
```



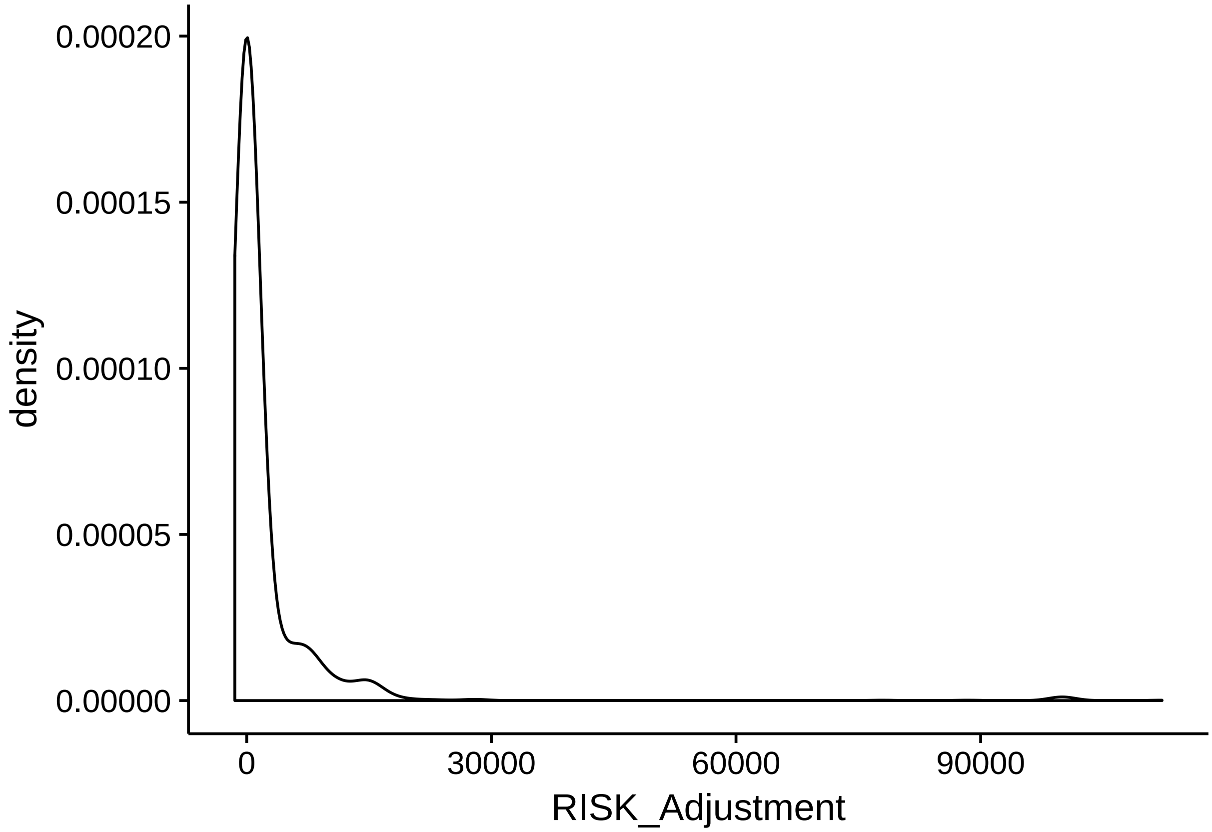

```r
library(cowplot)
ggplot(data = audit2, aes(x = Deductions)) + geom_density()
```




```
library(cowplot)
ggplot(data = audit2, aes(x = Hours)) + geom_density()
```




```r
library(cowplot)
ggplot(data = audit2, aes(x = RISK_Adjustment)) + geom_density()
```




From the graphs we can see, Except the hours, all the other numeric attributes are skewed to the right. We can use some other methods to test normality. We can test the skewness of these numerical variables as following.

```
library(e1071)
skewness(audit2$Age)
```

```
## [1] 0.4990696
```

```
skewness(audit2$Income)
```

```
## [1] 1.488821
```

```
skewness(audit2$Deductions)
```

```
## [1] 5.249432
```

```
skewness(audit2$RISK_Adjustment)
```

```
## [1] 9.591535
```

The skewness of Income, Deductions and RISK_Adjustment are al larger than one, which means they are highly skewed to the right, especially Deductions and RISK_Adjustment. ONly age's skewness is less than 0.5, which is with tolerance.

Perform Shapiro-Wilktest, and reject the null hypothesis (normality) if p-value is significant.

```
shapiro.test(audit2$Age)
```

```
##
##  Shapiro-Wilk normality test
```

```
##
## data:  audit2$Age
## W = 0.96698, p-value < 2.2e-16
```

```
shapiro.test(audit2$Income)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  audit2$Income
## W = 0.84983, p-value < 2.2e-16
```

```
shapiro.test(audit2$Deductions)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  audit2$Deductions
## W = 0.19809, p-value < 2.2e-16
```

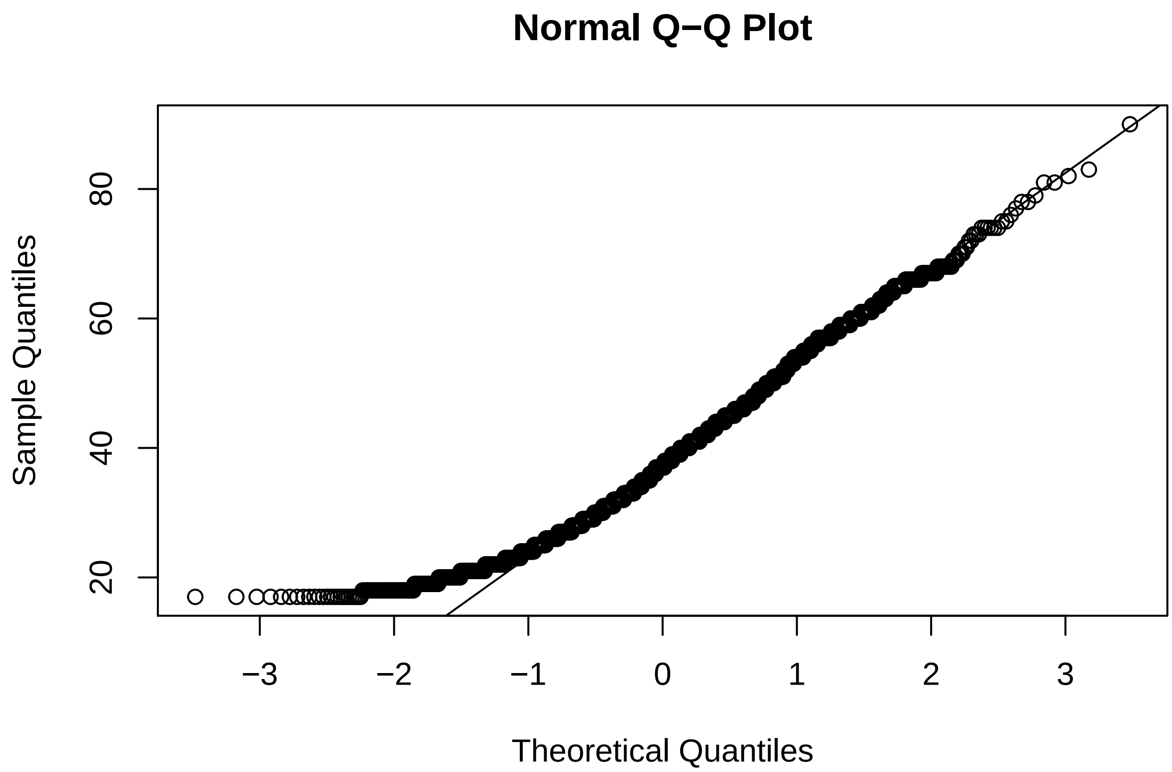
```
shapiro.test(audit2$RISK_Adjustment)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  audit2$RISK_Adjustment
## W = 0.23081, p-value < 2.2e-16
```

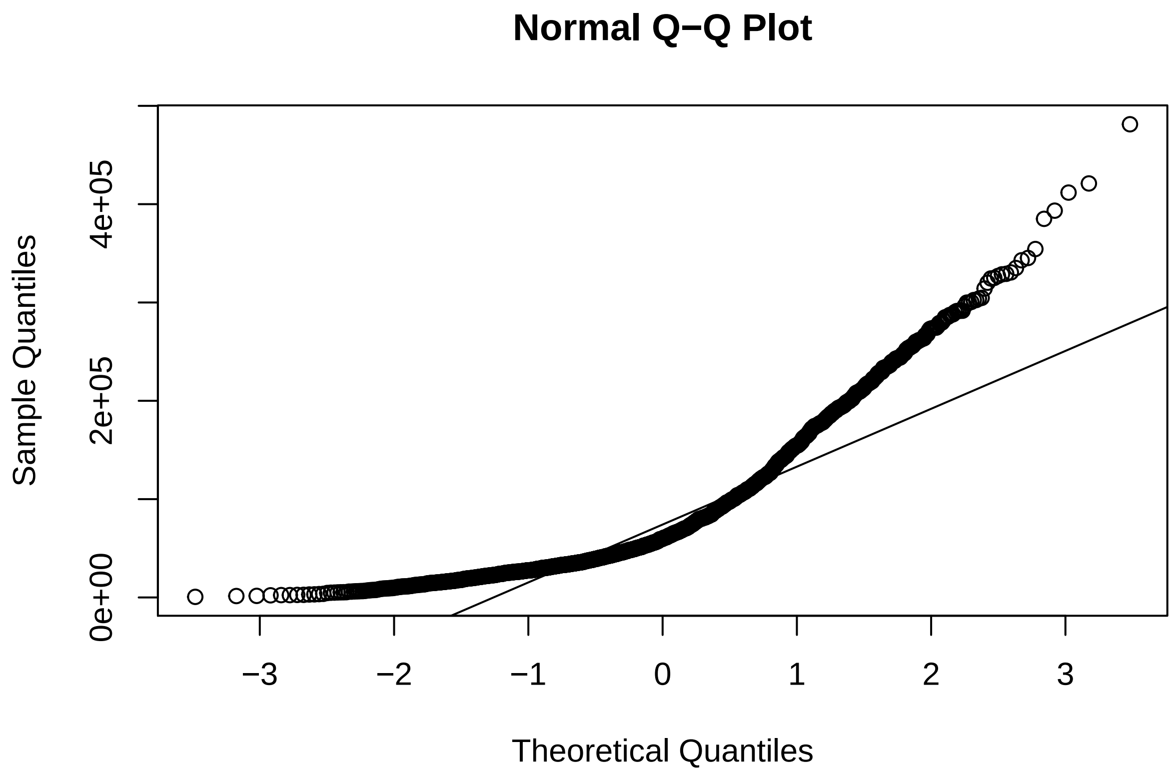
If we set the significance level as 0.05, we can see that all p-values are significant (less than 0.05), which implies that we can reject the null hypothesis and claim that all attributes except hours are not normal distribution.

Draw a normal probability plot (q-q plot), and check if the distribution is approximately forms a straight line.

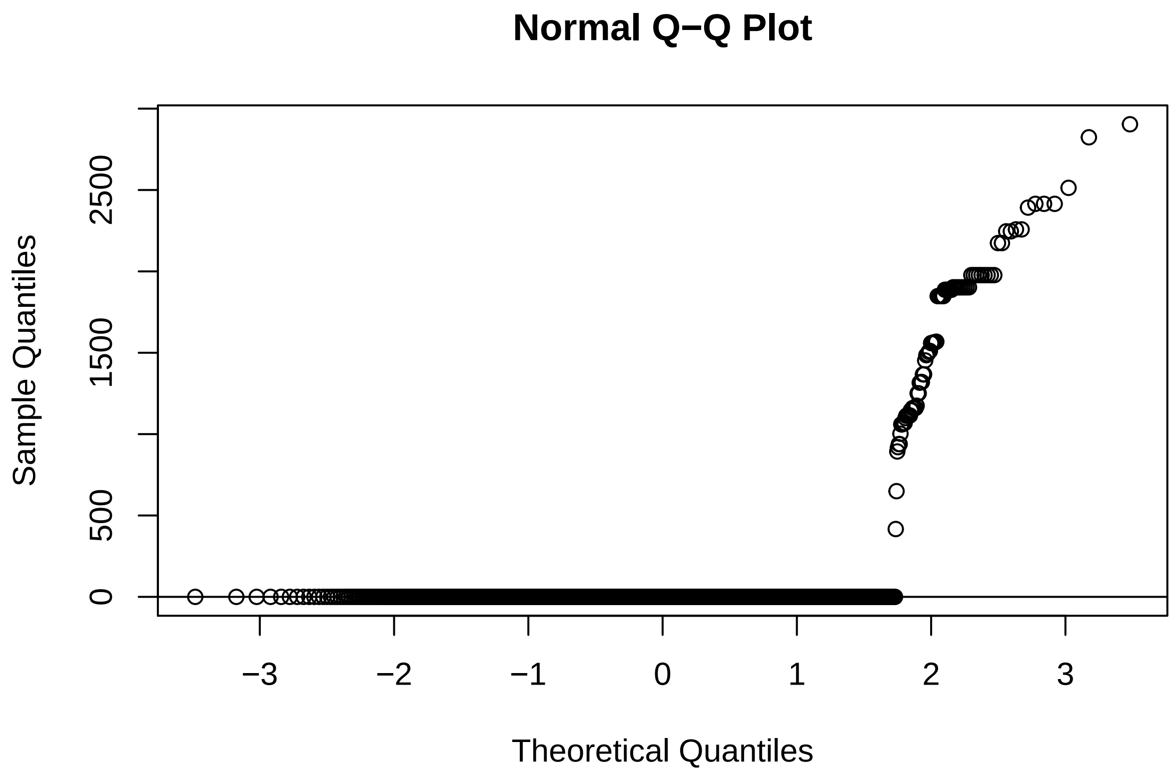
```
qqnorm(audit2$Age)
qqline(audit2$Age)
```



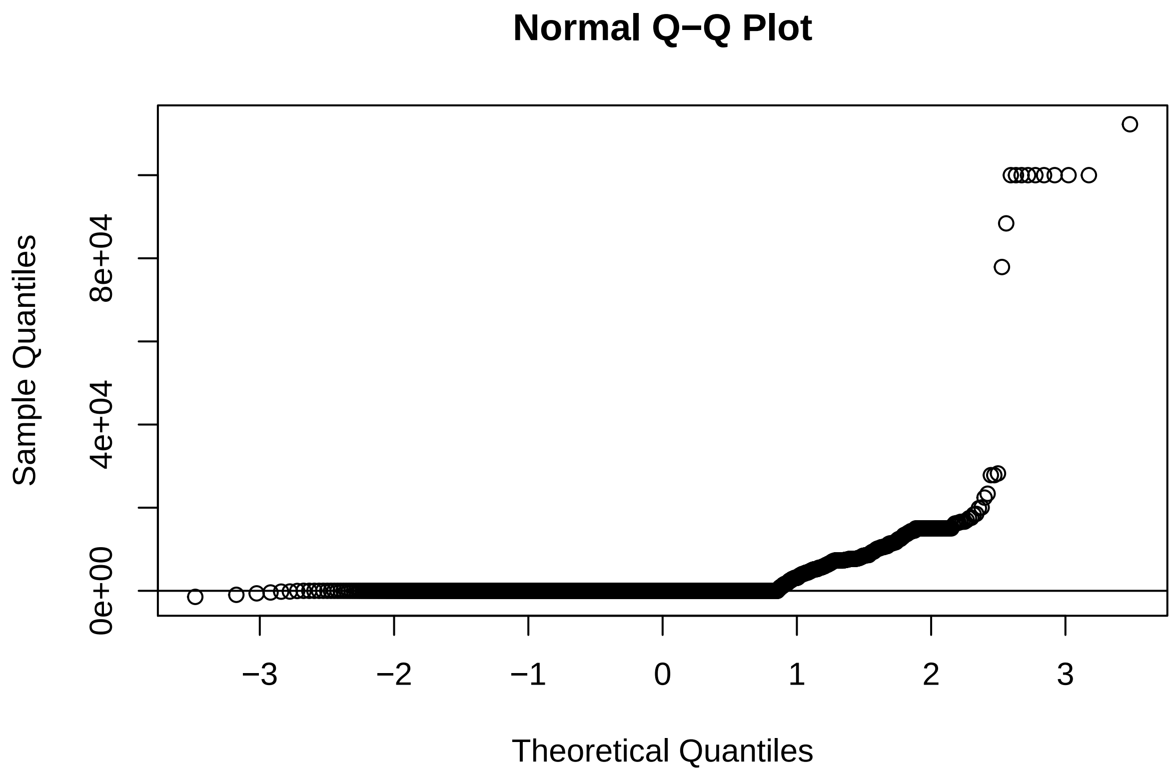

```
qqnorm(audit2$Income)
qqline(audit2$Income)
```




```r
qqnorm(audit2$Deductions)
qqline(audit2$Deductions)
```




```r
qqnorm(audit2$RISK_Adjustment)
qqline(audit2$RISK_Adjustment)
```



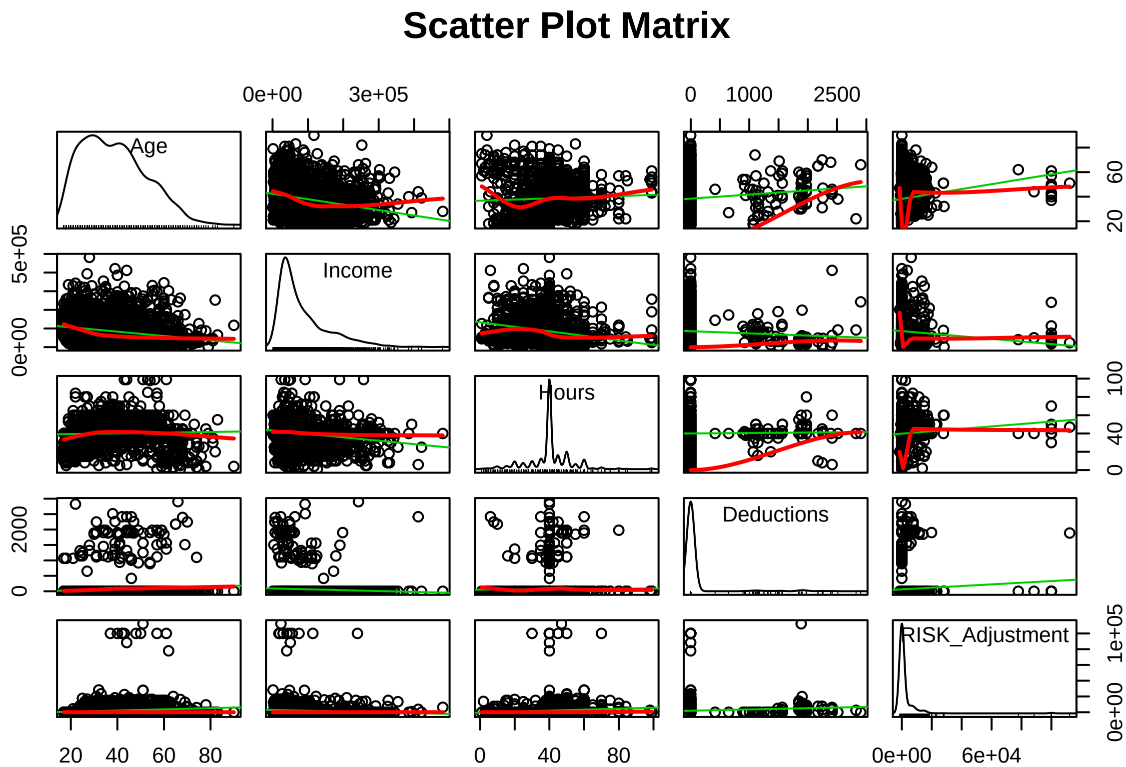

From q-q plots, we can see that the points for Deductions and RISK_Adjustment do not fall on the straight line which clearly voilate the normality assumption.They are skewed to the right.

**- For each numerical predictor, describe its relationship with the response variable through correlation and scatterplot.**

```r
library(car)
dt = audit2[, c("Age", "Income", "Hours", "Deductions", "RISK_Adjustment")]
cor(dt)
```

```
##                         Age      Income       Hours  Deductions
## Age              1.00000000 -0.22686777  0.04236487  0.08399899
## Income          -0.22686777  1.00000000 -0.21269065 -0.05734147
## Hours            0.04236487 -0.21269065  1.00000000  0.01365124
## Deductions       0.08399899 -0.05734147  0.01365124  1.00000000
## RISK_Adjustment  0.12274079 -0.08339021  0.09060735  0.06559720
##                 RISK_Adjustment
## Age                  0.12274079
## Income              -0.08339021
## Hours                0.09060735
## Deductions           0.06559720
## RISK_Adjustment      1.00000000
```

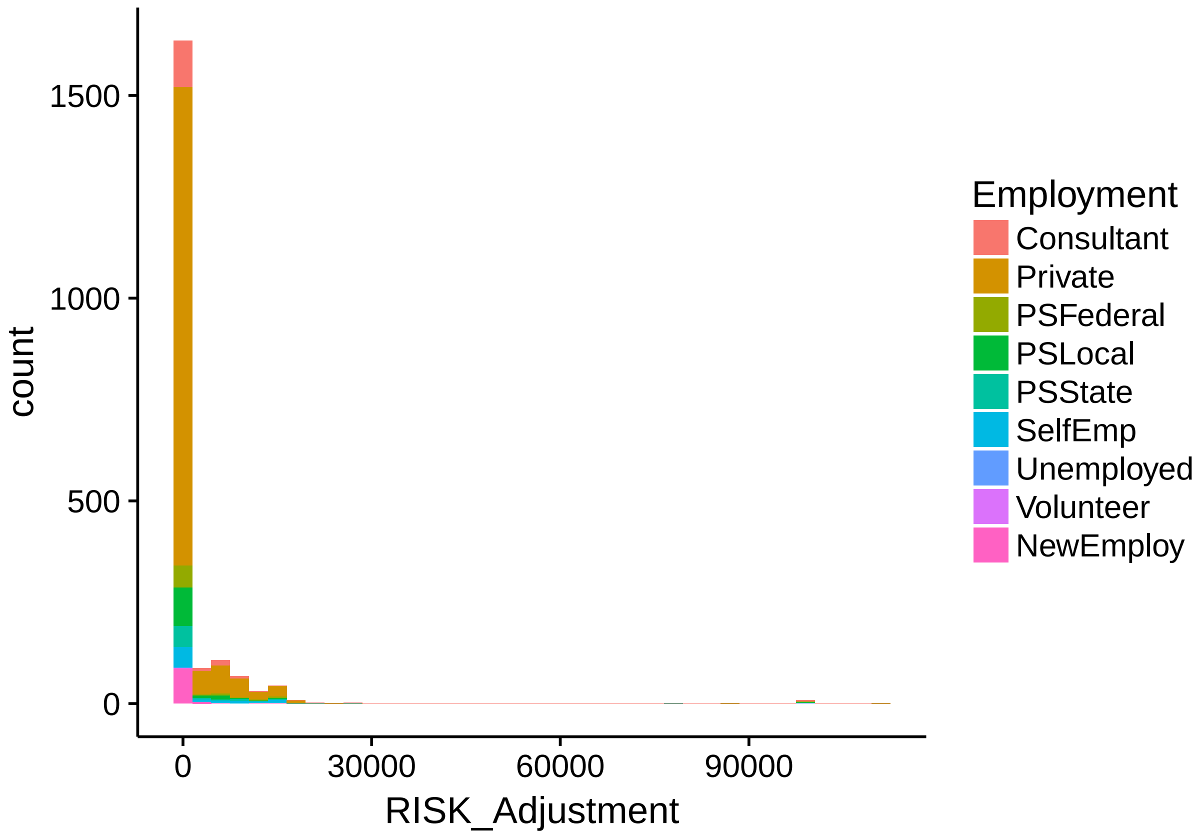
```r
scatterplotMatrix(dt, spread = FALSE, lty.smooth = 2, main = "Scatter Plot Matrix")
```



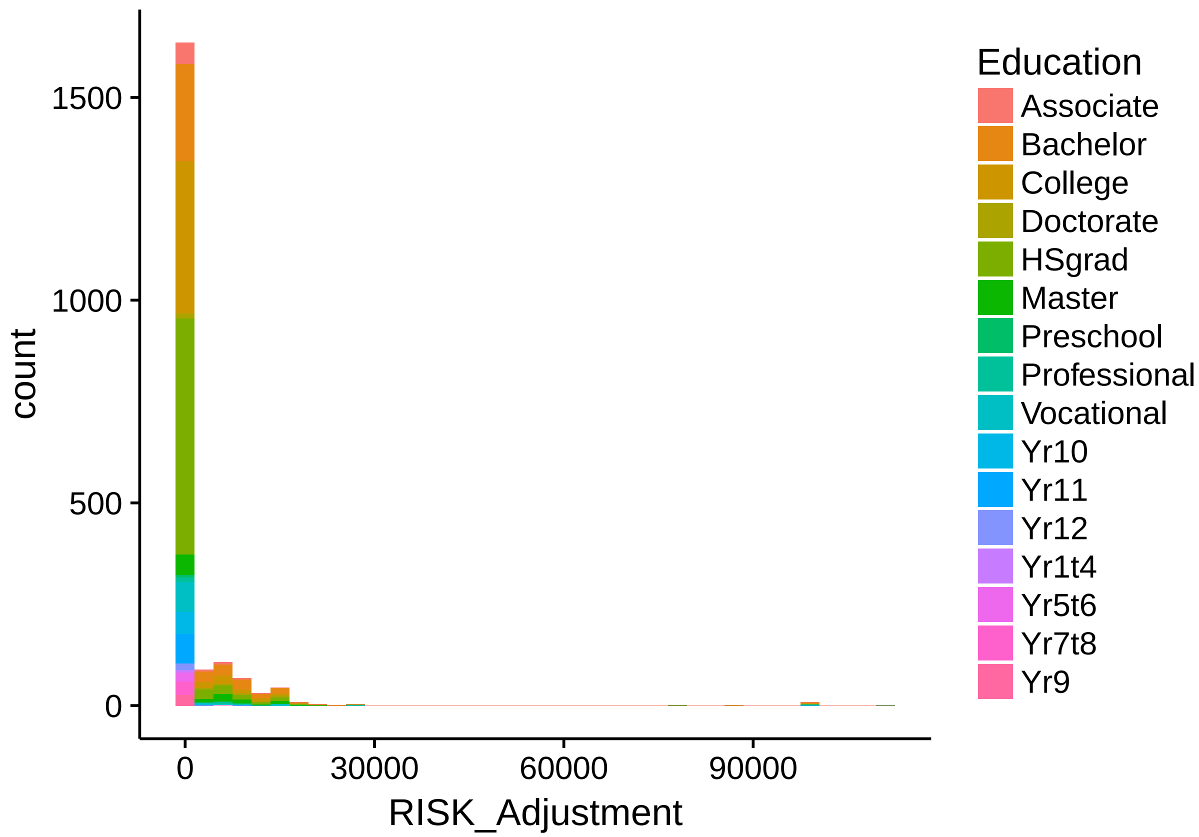

By examining the correlations (bottom-left four values) and scatterplots (bottom-left four figures) between predictors and response, we can see that RISK_Adjustment doesn't show very obvious correlation with the other predictors. We might need to combine these feature so that they can be someway related.

**c–For each categorical predictor, generate the conditional histogram plot of response variable.**

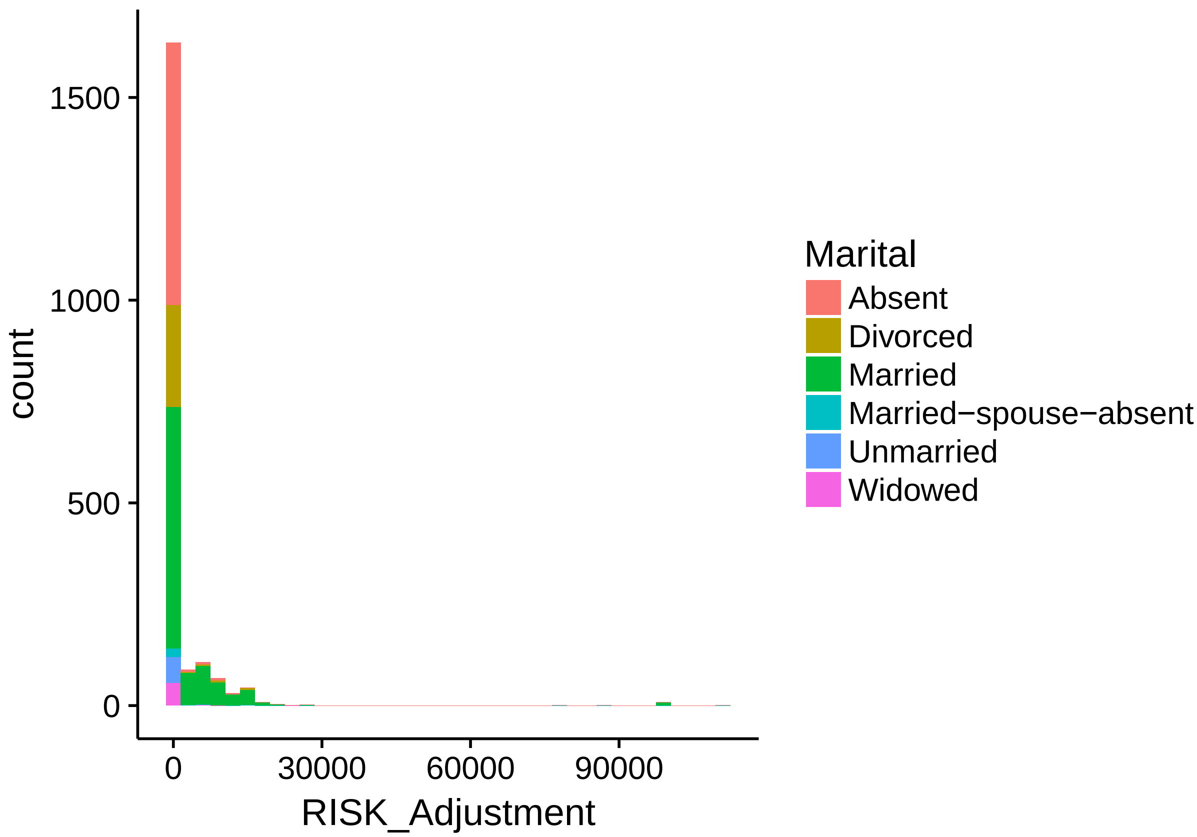
```
ggplot(audit2, aes(x = RISK_Adjustment, fill = Employment)) + geom_histogram(binwidth = 3000)
```



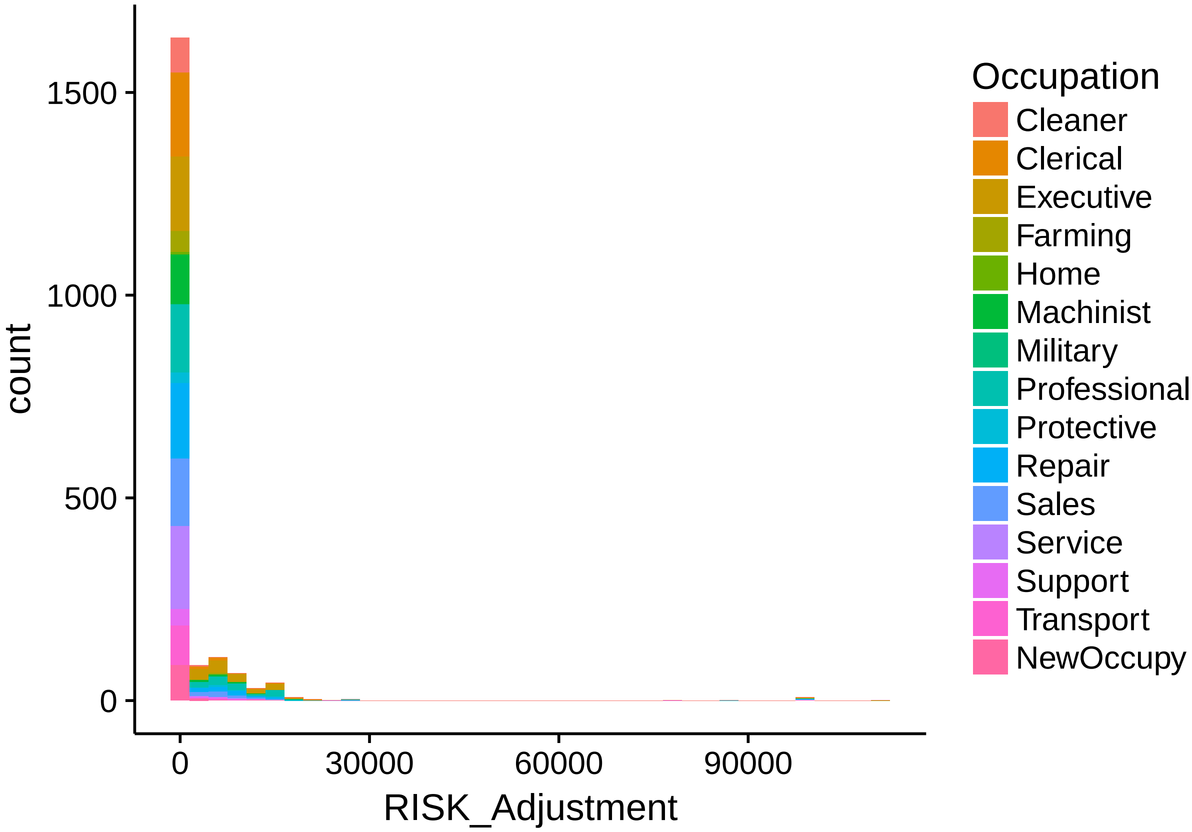

```
ggplot(audit2, aes(x = RISK_Adjustment, fill = Education)) + geom_histogram(binwidth = 3000)
```



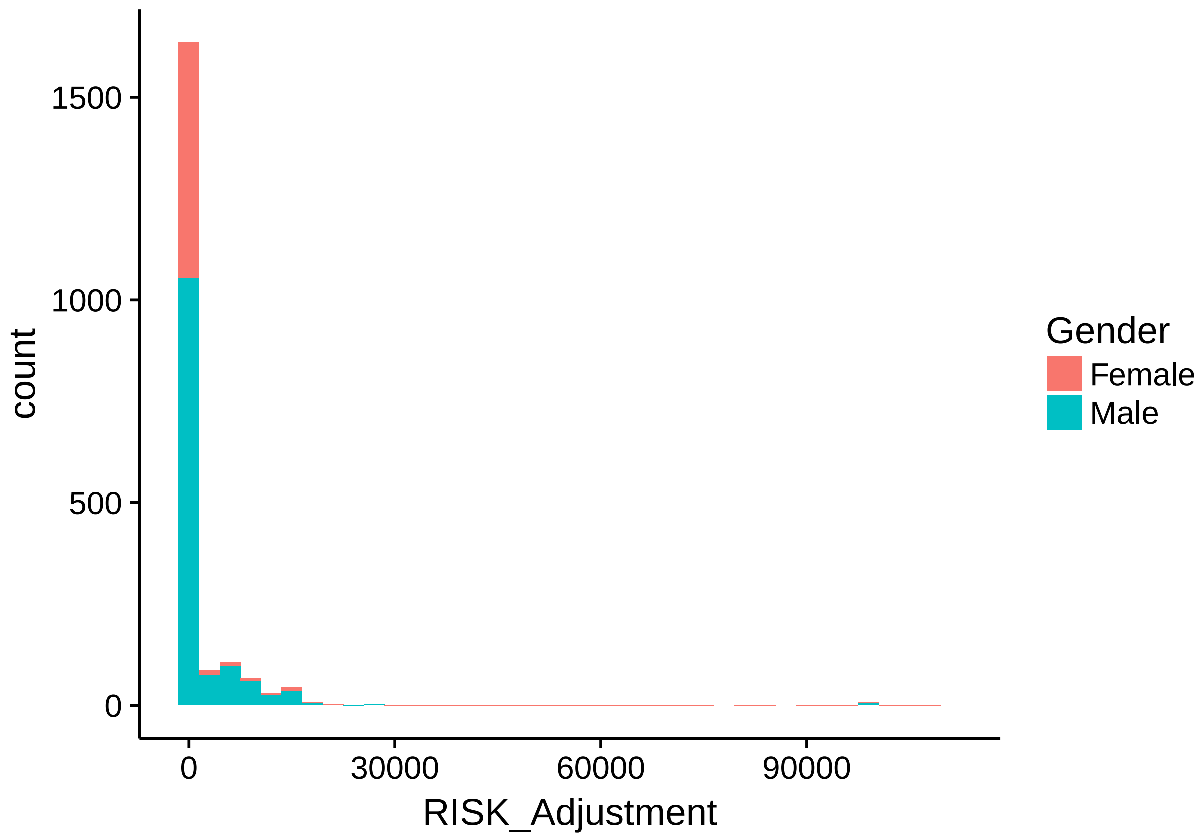

```r
ggplot(audit2, aes(x = RISK_Adjustment, fill = Marital)) + geom_histogram(binwidth = 3000)
```




```
ggplot(audit2, aes(x = RISK_Adjustment, fill = Occupation)) + geom_histogram(binwidth = 3000)
```




```
ggplot(audit2, aes(x = RISK_Adjustment, fill = Gender)) + geom_histogram(binwidth = 3000)
```




## 3–Apply logistic regression analysis to predict TARGET_Adjusted. Evaluate the models through cross-validation and on holdout samples.

### a–Implement logistic regression

Implement a 10-fold cross-validation scheme by splitting the data into training and testing sets. Use the training set to train a logistic regression model to predict the response variable. Examine the performance of different models by varing the number of predictors. Report the performance of the models on testing set using proper measures (accuracy, precision, recall, F1, AUC) and plots (ROC, lift).

Check the TARGET_Adjusted distribution.

```
table(audit2$TARGET_Adjusted)
```

```
##
##    0    1
## 1537  463
```

Define a function for logistic regression with 10-fold cross valiadation and evaluation.

```
library(caret)
library(pROC)
crossvalid <- function(data) {
    Xdel = model.matrix(TARGET_Adjusted ~ ., data = audit3)[, -1]
    n.total = length(audit3$RISK_Adjustment)
    n.train = floor(n.total * (0.9))
    n.test = n.total - n.train
```

```
    error = dim(10)
    accuracy = dim(10)
    precision = dim(10)
    recall = dim(10)
    f1_score = dim(10)
    auc = dim(10)
    for (k in 1:10) {
        train = sample(1:n.total, n.train)
        xtrain = Xdel[train, ]
        xtest = Xdel[-train, ]
        ytrain = audit3$TARGET_Adjusted[train]
        ytest = audit3$TARGET_Adjusted[-train]
        m1 = glm(TARGET_Adjusted ~ ., family = binomial, data = data.frame(TARGET_Adjusted = ytrain,
            xtrain))

        ptest = predict(m1, newdata = data.frame(xtest), type = "response")
        btest = floor(ptest + 0.5)
        conf.matrix = table(ytest, btest)
        accuracy[k] = (conf.matrix[1, 1] + conf.matrix[2, 2])/n.test
        error[k] = 1 - accuracy[k]
        precision[k] = conf.matrix[1, 1]/(conf.matrix[1, 1] + conf.matrix[1,
            2])
        recall[k] = conf.matrix[1, 1]/(conf.matrix[1, 1] + conf.matrix[2, 1])
        f1_score[k] = (2 * precision * recall)/(precision + recall)
        auc[k] = auc(btest, ptest)

    }
    acc_avg = mean(accuracy)
    error_avg = mean(error)
    prec_avg = mean(precision)
    rec_avg = mean(recall)
    f1_avg = mean(f1_score)
    auc_avg = mean(auc)
    cat("accuracy:", acc_avg, "\n")
    cat("error: ", error_avg, "\n")
    cat("precision: ", prec_avg, "\n")
    cat("recall: ", rec_avg, "\n")
    cat("F1_score: ", f1_avg, "\n")
    cat("AUC: ", auc_avg, "\n")
    return(conf.matrix)
}
```

Define a function for generating Lift charts.

```
liftcharts <- function(data) {
    Xdel = model.matrix(TARGET_Adjusted ~ ., data = audit3)[, -1]
    n.total = length(audit3$RISK_Adjustment)
    n.train = floor(n.total * (0.9))
    n.test = n.total - n.train
    baserate = dim(10)
    for (k in 1:10) {
        train = sample(1:n.total, n.train)
        xtrain = Xdel[train, ]
        xtest = Xdel[-train, ]
```

```
        ytrain = audit3$TARGET_Adjusted[train]
        ytest = audit3$TARGET_Adjusted[-train]
        m1 = glm(TARGET_Adjusted ~ ., family = binomial, data = data.frame(TARGET_Adjusted = ytrain,
            xtrain))

        ptest = predict(m1, newdata = data.frame(xtest), type = "response")
        btest = floor(ptest + 0.5)
        df = cbind(ptest, ytest)
        rank.df = as.data.frame(df[order(ptest, decreasing = TRUE), ])
        colnames(rank.df) = c("predicted", "actual")
        baserate[k] = mean(ytest)
    }
    ax = dim(n.test)
    ay.base = dim(n.test)
    ay.pred = dim(n.test)
    ax[1] = 1
    ay.base[1] = mean(baserate)
    ay.pred[1] = rank.df$actual[1]
    for (i in 2:n.test) {
        ax[i] = i
        ay.base[i] = (mean(baserate)) * i  ## uniformly increase with rate xbar
        ay.pred[i] = ay.pred[i - 1] + rank.df$actual[i]
    }
    df = cbind(rank.df, ay.pred, ay.base)

    plot(ax, ay.pred, xlab = "number of cases", ylab = "number of successes",
        main = "Lift: Cum successes sorted by pred val/success prob")
    points(ax, ay.base, type = "l")
    return(0)
}
```

Define a function for generating ROC charts.

```
library(ROCR)
ROCcharts <- function(data) {
    Xdel = model.matrix(TARGET_Adjusted ~ ., data = audit3)[, -1]
    n.total = length(audit3$RISK_Adjustment)
    n.train = floor(n.total * (0.9))
    n.test = n.total - n.train
    sensi = dim(10)
    speci = dim(10)
    for (k in 1:10) {
        train = sample(1:n.total, n.train)
        xtrain = Xdel[train, ]
        xtest = Xdel[-train, ]
        ytrain = audit3$TARGET_Adjusted[train]
        ytest = audit3$TARGET_Adjusted[-train]
        m1 = glm(TARGET_Adjusted ~ ., family = binomial, data = data.frame(TARGET_Adjusted = ytrain,
            xtrain))

        ptest = predict(m1, newdata = data.frame(xtest), type = "response")
        btest = floor(ptest + 0.5)
        cut = 1/2
        gg1 = floor(ptest + (1 - cut))
```

```
        truepos = ytest == 1 & ptest >= cut
        trueneg = ytest == 0 & ptest < cut
        sensi[k] = sum(truepos)/sum(ytest == 1)
        speci[k] = sum(trueneg)/sum(ytest == 0)
        data = data.frame(predictions = ptest, labels = ytest)
        pred <- prediction(data$predictions, data$labels)
        perf <- performance(pred, "sens", "fpr")
    }
    plot(perf)
    cat("Specificity:", mean(speci), "\n")
    cat("Sensitivity:", mean(sensi), "\n")
    return(0)
}
```

Examine the performance of different models by varing the number of predictors.

Using all the predictors to train the model.

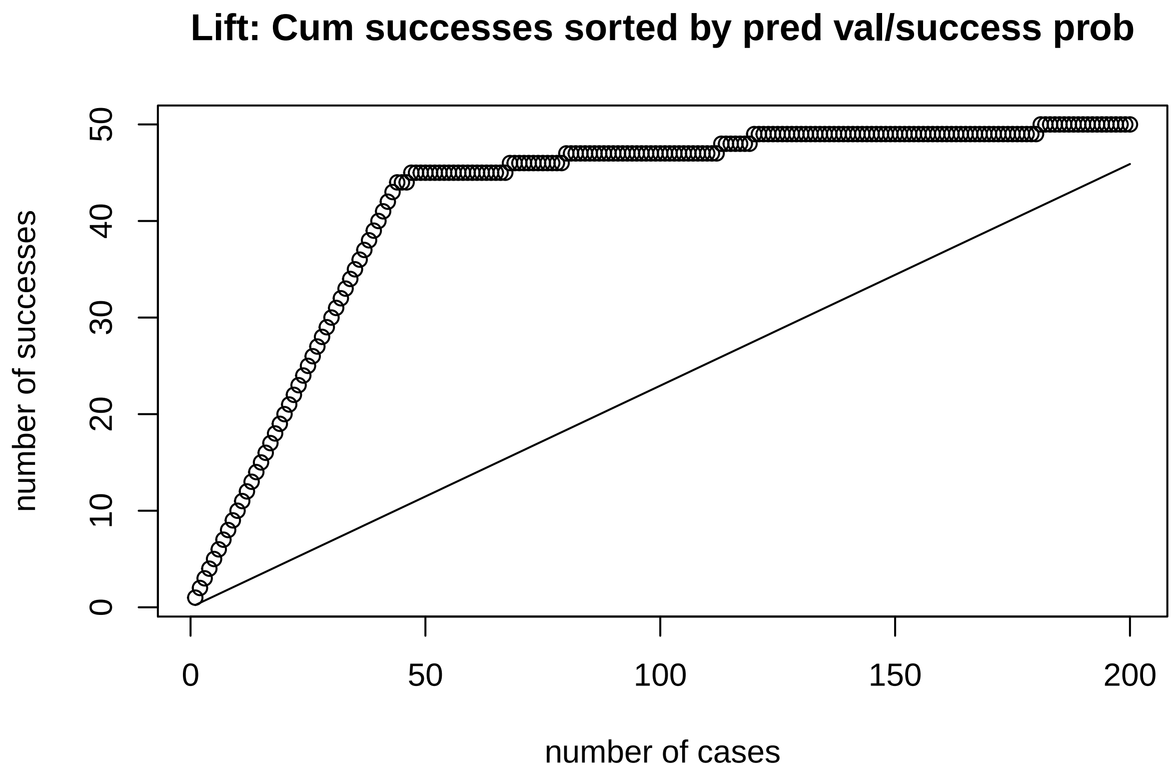
```
audit3 = audit2
crossvalid(audit3)
```

```
## accuracy: 0.9615
## error:  0.0385
## precision:  0.998021
## recall:  0.9537518
## F1_score:  0.970297
## AUC:  1
```

```
##      btest
## ytest   0   1
##     0 150   1
##     1   6  43
```

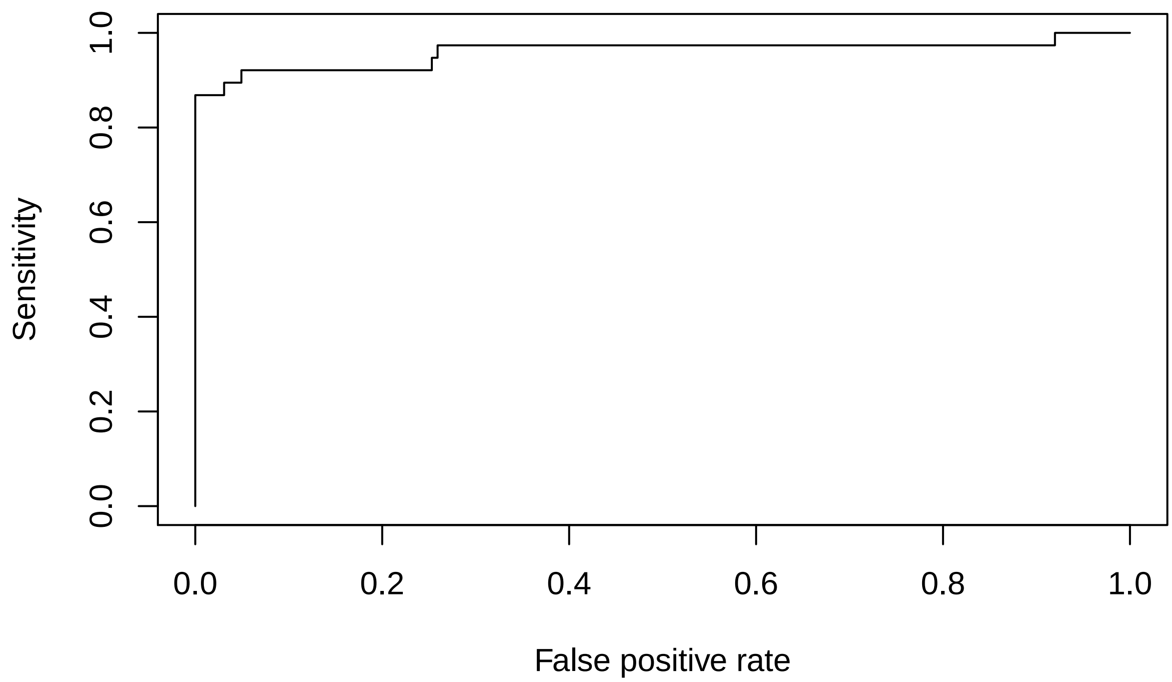
The result of 10-fold cross valiadation is not very stable. Thus, there might be some difference between the real result and result I saw in the console. The accuracy by using all predictors is 0.962 and the precision is 0.9967.

```
liftcharts(audit3)
```




```
## [1] 0
```

```
ROCcharts(audit3)
```




```
## Specificity: 0.9981168
## Sensitivity: 0.8236349
```

```
## [1] 0
```

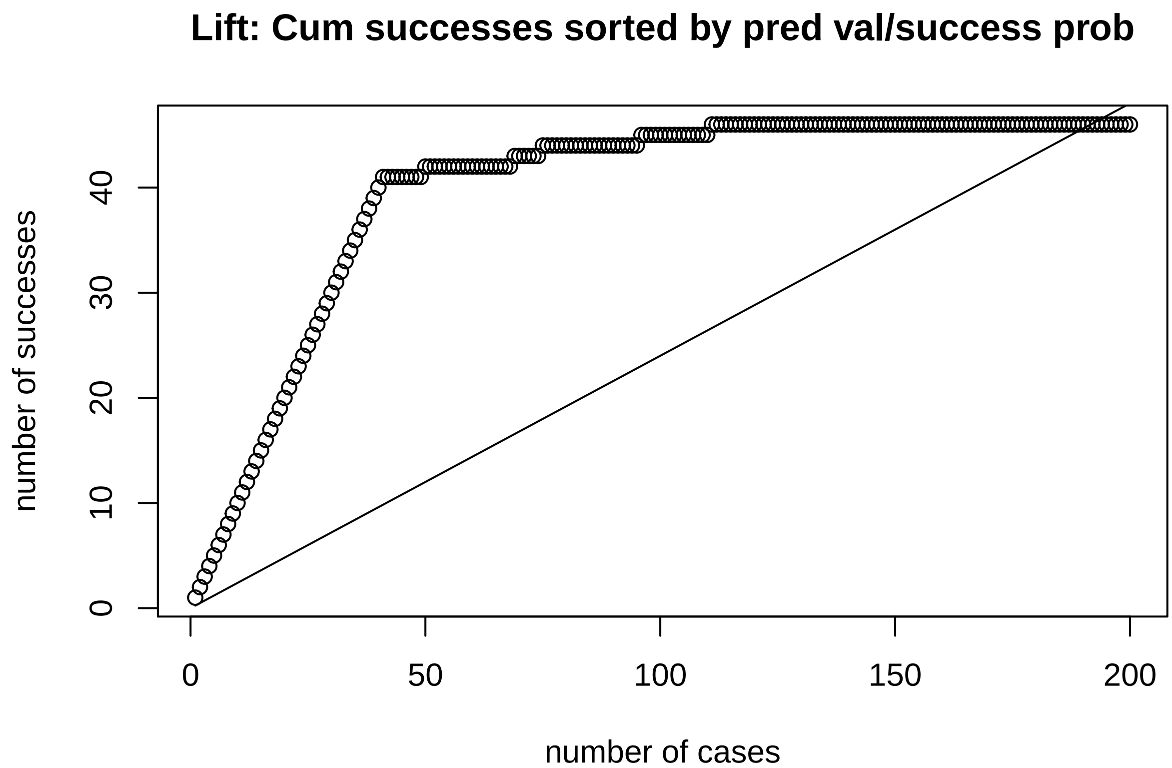
Drop Education varicable.

```
audit3 = audit2[c(-3)]
crossvalid(audit3)
```

```
## accuracy: 0.9615
## error:  0.0385
## precision:  0.9986237
## recall:  0.9524833
## F1_score:  0.9756098
## AUC:  1
```

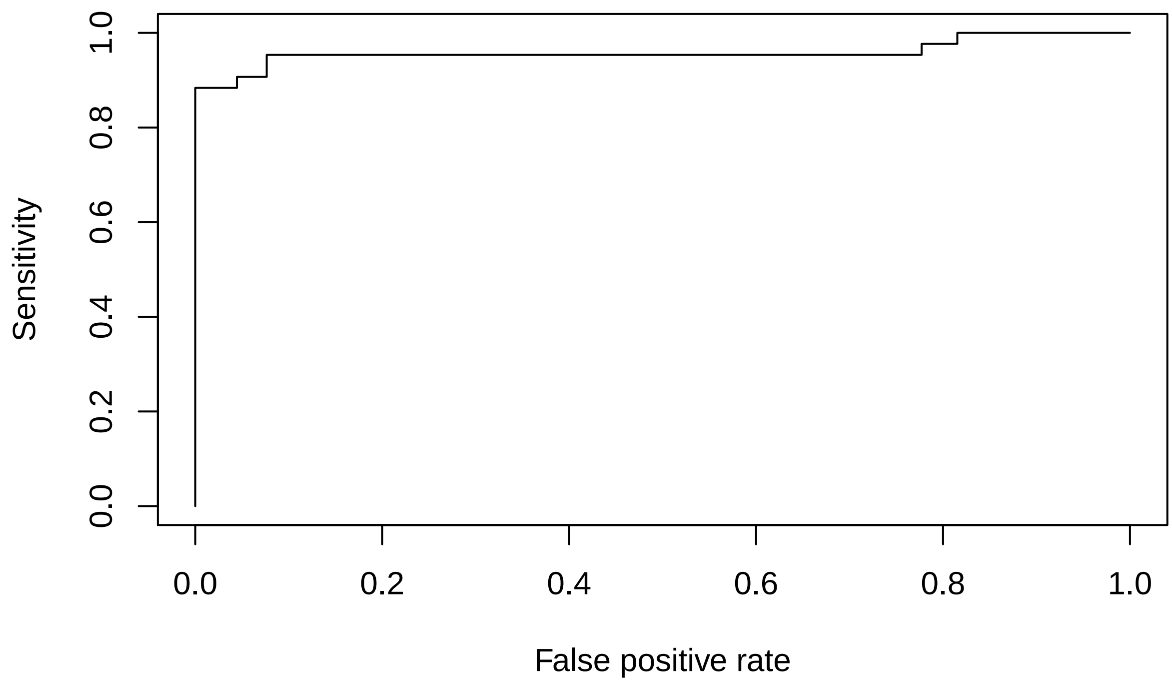
```
##       btest
## ytest   0   1
##     0 135   1
##     1  14  50
```

```
liftcharts(audit3)
```




```
## [1] 0
```

```r
ROCcharts(audit3)
```




```
## Specificity: 0.9993631
## Sensitivity: 0.8291966
```

```
## [1] 0
```

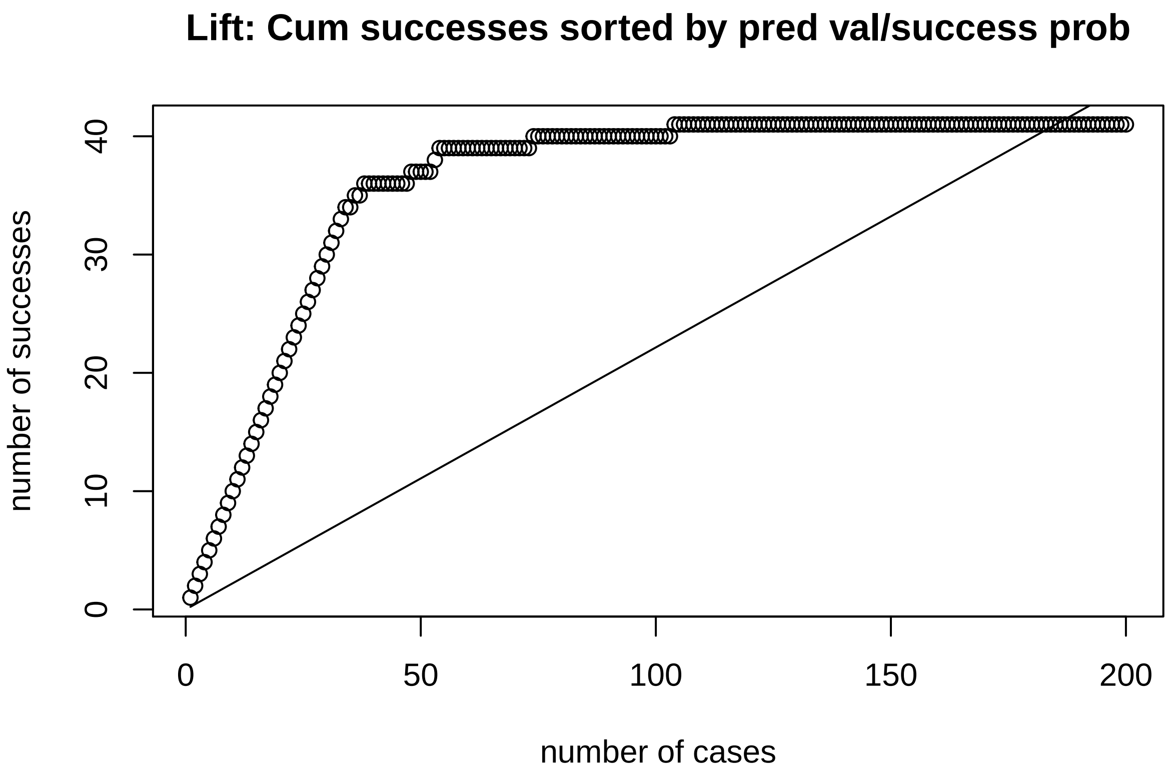
Drop Marital variable.

```
audit3 = audit2[c(-4)]
crossvalid(audit3)
```

```
## accuracy: 0.965
## error:  0.035
## precision:  0.9993548
## recall:  0.9573261
## F1_score:  0.9833887
## AUC:  1
```

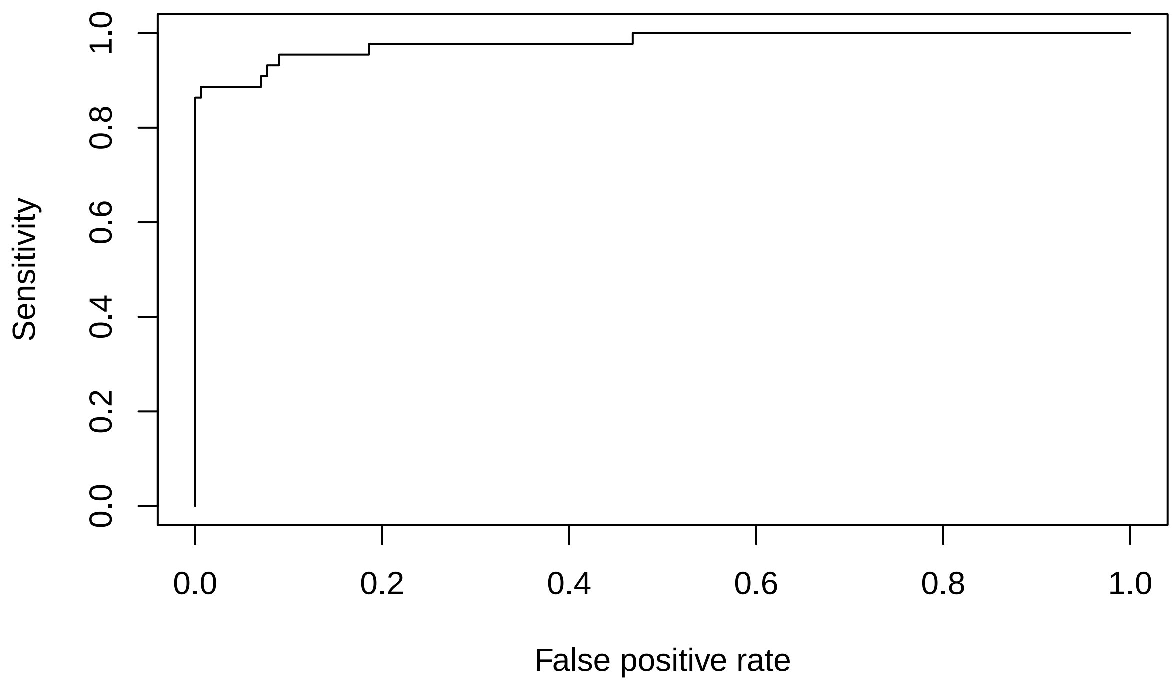
```
##      btest
## ytest   0   1
##     0 154   1
##     1   7  38
```

```
liftcharts(audit3)
```




```
## [1] 0
```

```
ROCcharts(audit3)
```




```
## Specificity: 0.9980476
## Sensitivity: 0.8477581
```

```
## [1] 0
```

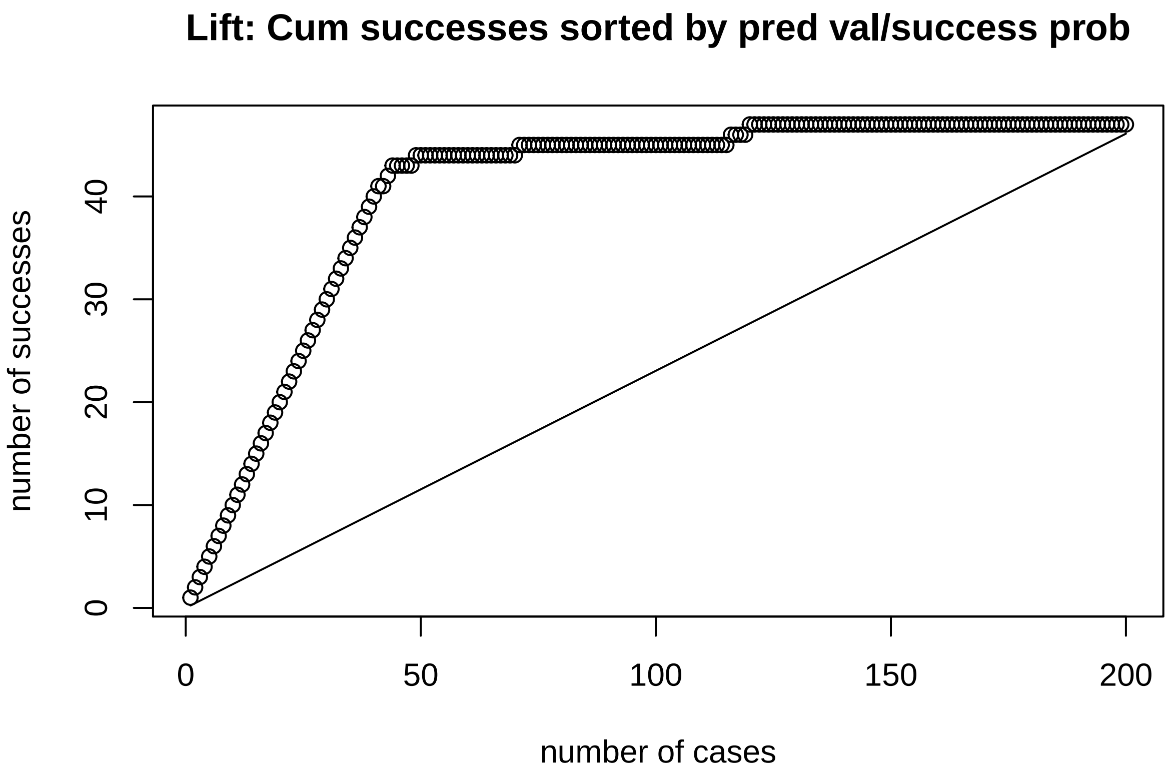
Drop Income Variable.

```
audit3 = audit2[c(-6)]
crossvalid(audit3)
```

```
## accuracy: 0.957
## error:  0.043
## precision:  0.9948595
## recall:  0.9508453
## F1_score:  0.9704918
## AUC:  1
```

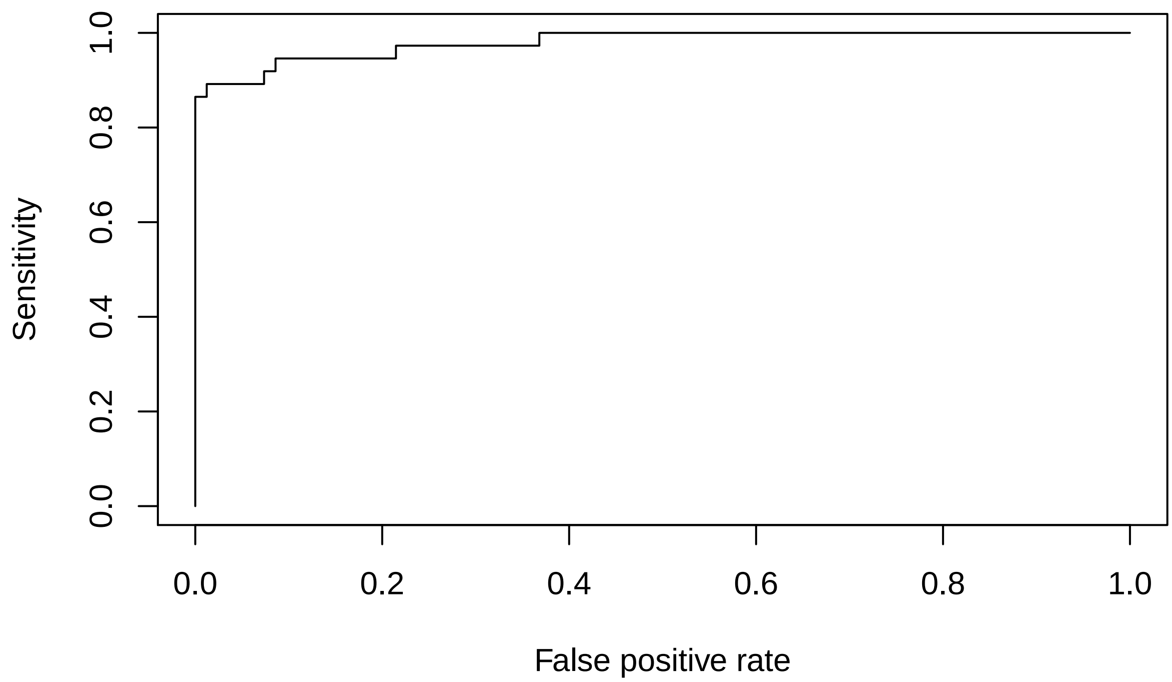
```
##      btest
## ytest   0   1
##     0 157   3
##     1  10  30
```

```
liftcharts(audit3)
```




```
## [1] 0
```

```r
ROCcharts(audit3)
```



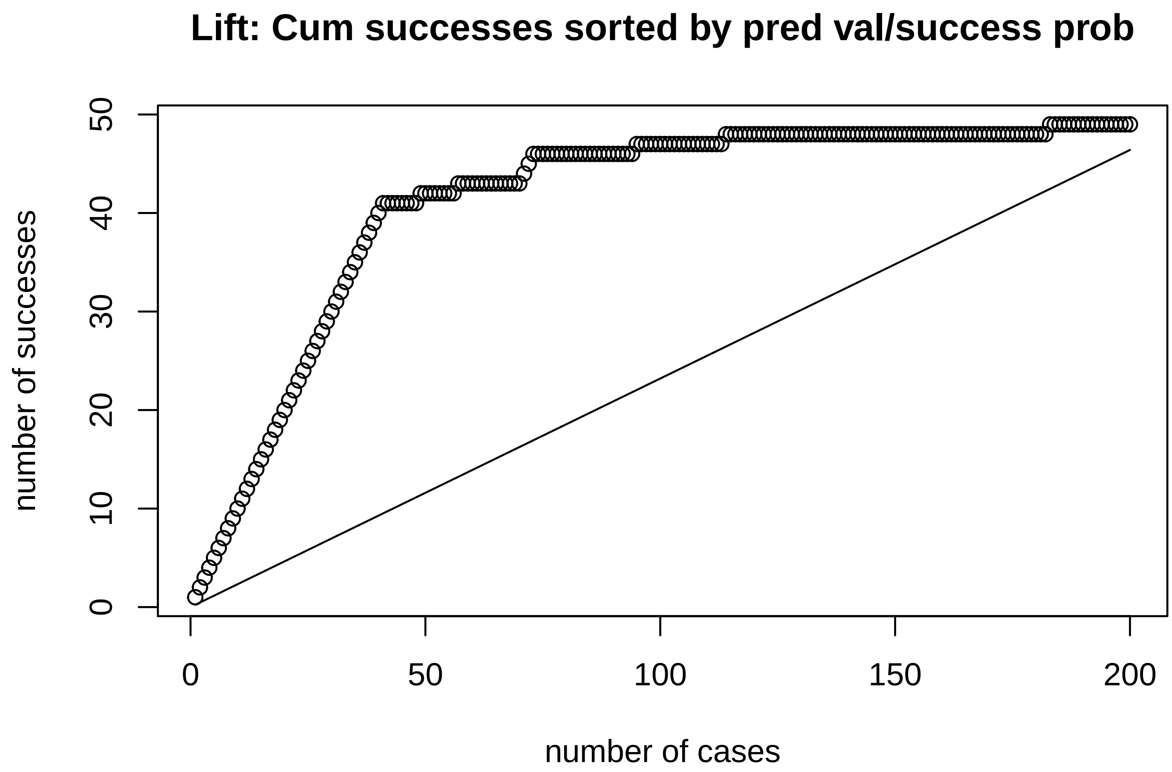

```
## Specificity: 0.9968774
## Sensitivity: 0.8325532

## [1] 0
```

Drop Marital and Income variables.

```
audit3 = audit2[c(-4, -6)]
crossvalid(audit3)
```

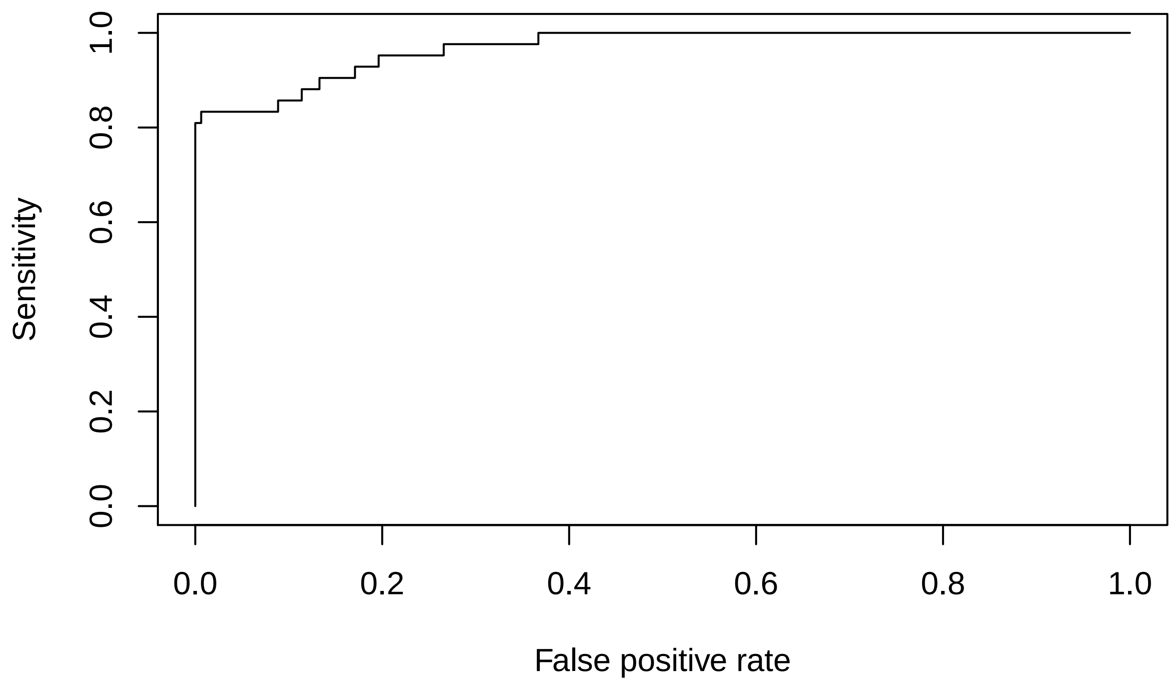
```
## accuracy: 0.954
## error:  0.046
## precision:  0.9986746
## recall:  0.9443194
## F1_score:  0.9716088
## AUC:  1

##      btest
## ytest   0   1
##     0 155   0
##     1   8  37
```

```
liftcharts(audit3)
```




```
## [1] 0
```

```
ROCcharts(audit3)
```



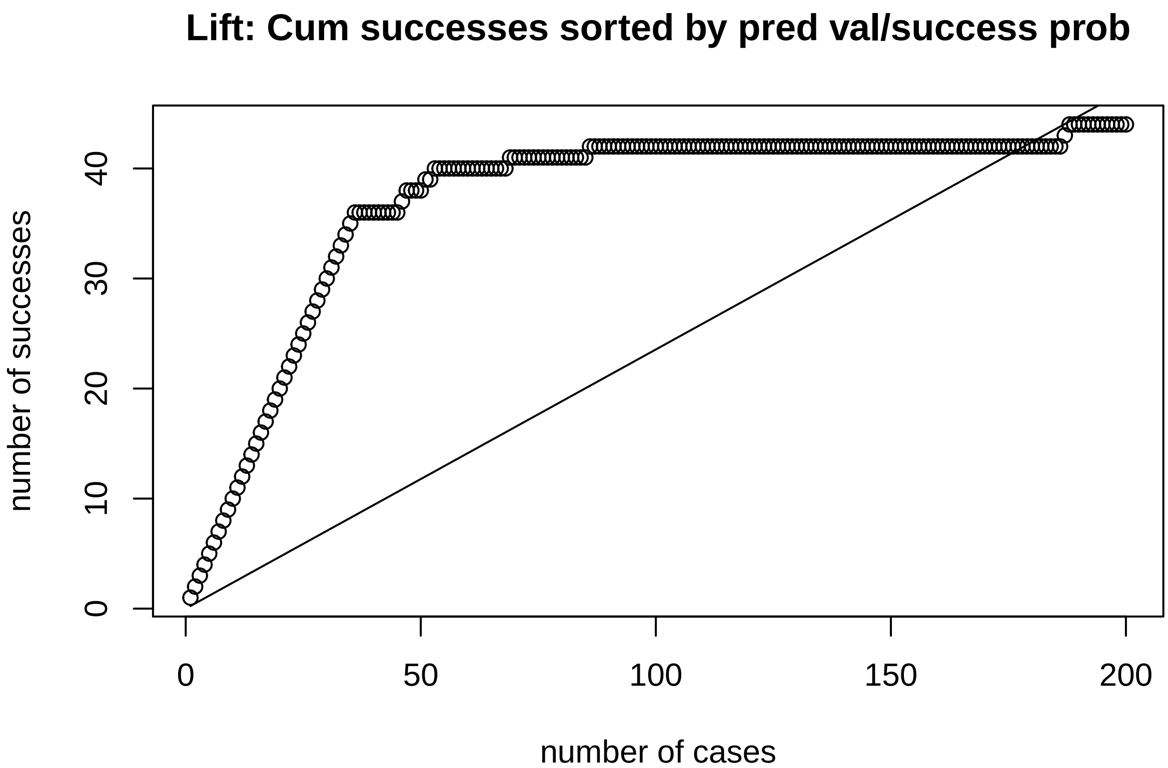

```
## Specificity: 0.9967681
## Sensitivity: 0.8430793

## [1] 0
```

Drop Educarion, Marital and Income variables.

```
audit3 = audit2[c(-3, -4, -6)]
crossvalid(audit3)
```

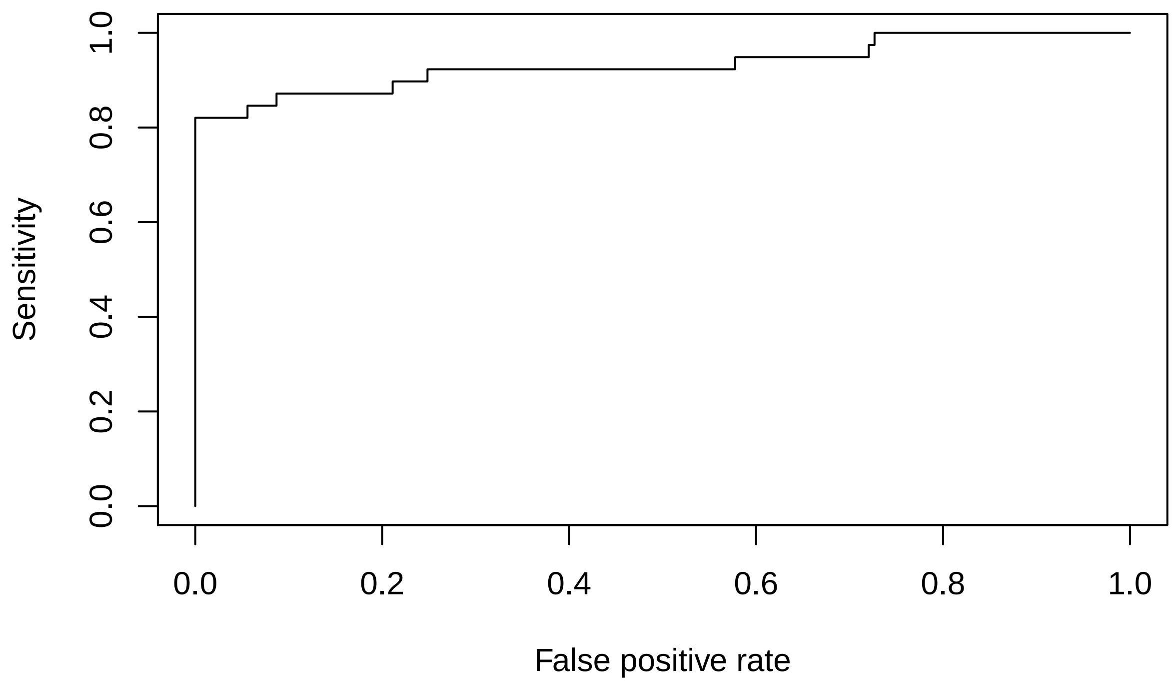
```
## accuracy: 0.965
## error:  0.035
## precision:  1
## recall:  0.956109
## F1_score:  0.9847095
## AUC:  1

##      btest
## ytest   0   1
##     0 140   0
##     1  13  47
```

```
liftcharts(audit3)
```




```
## [1] 0
```

```r
ROCcharts(audit3)
```




```
## Specificity: 0.9987419
## Sensitivity: 0.8405309

## [1] 0
```

From the above resulet we can see, we get the best accuracy from model 5, which dropped Marital and Income variables. This is the best result we can get from logistic regression model. We can also use Naive Bayesian and Desicion Tree model to predict TARGET_Adjusted.

### Naive Bayesian Model

Split the data into training and test dataset.

```
n = length(audit2$TARGET_Adjusted)
n1 = floor(n * (0.9))
n2 = n - n1
train = sample(1:n, n1)
```

determining marginal probabilities

```
response = audit2$TARGET_Adjusted
tttt = cbind(audit2$Employment[train], audit2$Education[train], audit2$Marital[train],
    audit2$Occupation[train], audit2$Gender[train], response[train])
tttrain0 = tttt[tttt[, 6] < 0.5, ]
tttrain1 = tttt[tttt[, 6] > 0.5, ]
```

Prior probabilities

```r
tdel = table(response[train])
tdel = tdel/sum(tdel)
tdel
```

```
##
##         0         1
## 0.7711111 0.2288889
```

```r
ts0 = table(tttrain0[, 1])
ts0 = ts0/sum(ts0)
ts0
```

```
##
##            1            2            3            4            5
## 0.0698847262 0.7255043228 0.0309798271 0.0576368876 0.0317002882
##            6            7            9
## 0.0280979827 0.0007204611 0.0554755043
```

```r
ts1 = table(tttrain1[, 1])
ts1 = ts1/sum(ts1)
ts1
```

```
##
##          1          2          3          4          5          6
## 0.08495146 0.63834951 0.03640777 0.07524272 0.04611650 0.08009709
##          9
## 0.03883495
```

```r
tc0 = table(tttrain0[, 2])
tc0 = tc0/sum(tc0)
tc0
```

```
##
##           1           2           3           4           5           6
## 0.033141210 0.124639769 0.238472622 0.006484150 0.360230548 0.030979827
##           7           8           9          10          11          12
## 0.004322767 0.005043228 0.047550432 0.034582133 0.047550432 0.008645533
##          13          14          15          16
## 0.004322767 0.013688761 0.023054755 0.017291066
```

```r
tc1 = table(tttrain1[, 2])
tc1 = tc1/sum(tc1)
tc1
```

```
##
##           1           2           3           4           5           6
## 0.043689320 0.317961165 0.165048544 0.041262136 0.211165049 0.126213592
##           8           9          10          11          12          14
## 0.036407767 0.033980583 0.009708738 0.004854369 0.002427184 0.004854369
##          16
## 0.002427184
```

```r
td0 = table(tttrain0[, 3])
td0 = td0/sum(td0)
td0
```

```
##
##          1          2          3          4          5          6
```

```
## 0.42507205 0.15561960 0.32997118 0.01224784 0.04106628 0.03602305
```

```r
td1 = table(tttrain1[, 3])
td1 = td1/sum(td1)
td1
```

```
##
##           1           2           3           4           5           6
## 0.067961165 0.038834951 0.868932039 0.004854369 0.009708738 0.009708738
```

```r
to0 = table(tttrain0[, 4])
to0 = to0/sum(to0)
to0
```

```
##
##            1            2            3            4            5
## 0.0569164265 0.1296829971 0.0943804035 0.0338616715 0.0036023055
##            6            7            8            9           10
## 0.0806916427 0.0007204611 0.0943804035 0.0172910663 0.1152737752
##           11           12           13           14           15
## 0.1059077810 0.1340057637 0.0230547550 0.0540345821 0.0561959654
```

```r
to1 = table(tttrain1[, 4])
to1 = to1/sum(to1)
to1
```

```
##
##           1           2           3           4           6           8
## 0.009708738 0.072815534 0.283980583 0.014563107 0.043689320 0.230582524
##           9          10          11          12          13          14
## 0.033980583 0.101941748 0.099514563 0.016990291 0.021844660 0.031553398
##          15
## 0.038834951
```

```r
tw0 = table(tttrain0[, 5])
tw0 = tw0/sum(tw0)
tw0
```

```
##
##         1         2
## 0.3681556 0.6318444
```

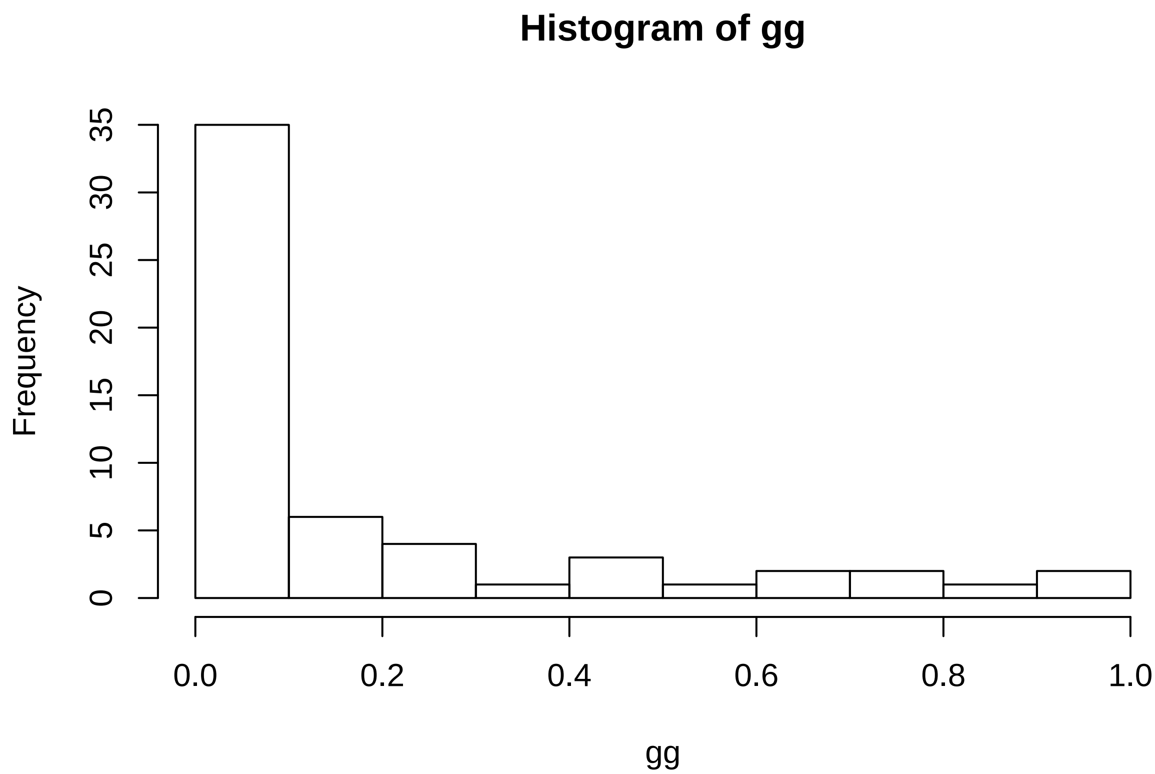
```r
tw1 = table(tttrain1[, 5])
tw1 = tw1/sum(tw1)
tw1
```

```
##
##         1         2
## 0.1456311 0.8543689
```

Create test dataset and predictions.

```r
tt = cbind(audit2$Employment[-train], audit2$Education[-train], audit2$Marital[-train],
    audit2$Occupation[-train], audit2$Gender[-train], response[-train])

p0 = ts0[tt[, 1]] * tc0[tt[, 2]] * td0[tt[, 3]] * to0[tt[, 4]] * tw0[tt[, 5] +
    1]
p1 = ts1[tt[, 1]] * tc1[tt[, 2]] * td1[tt[, 3]] * to1[tt[, 4]] * tw1[tt[, 5] +
    1]
```

```r
gg = (p1 * tdel[2])/(p1 * tdel[2] + p0 * tdel[1])
hist(gg)
```




Generate the

```r
gg1 = floor(gg + 0.5)
ttt = table(response[-train], gg1)

confusionMatrix(response[-train], gg1)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 45  4
##          1  4  4
##
##                Accuracy : 0.8596
##                  95% CI : (0.7421, 0.9374)
##     No Information Rate : 0.8596
##     P-Value [Acc > NIR] : 0.5928
##
##                   Kappa : 0.4184
##  Mcnemar's Test P-Value : 1.0000
##
##             Sensitivity : 0.9184
##             Specificity : 0.5000
```

```
##          Pos Pred Value : 0.9184
##          Neg Pred Value : 0.5000
##              Prevalence : 0.8596
##          Detection Rate : 0.7895
##    Detection Prevalence : 0.8596
##       Balanced Accuracy : 0.7092
##
##        'Positive' Class : 0
##
```

```
error = (ttt[1, 2] + ttt[2, 1])/n2
error
```

```
## [1] 0.04
```

```
precision = ttt[1, 1]/(ttt[1, 1] + ttt[1, 2])
precision
```

```
## [1] 0.9183673
```

```
recall = ttt[1, 1]/(ttt[1, 1] + ttt[2, 1])
recall
```

```
## [1] 0.9183673
```

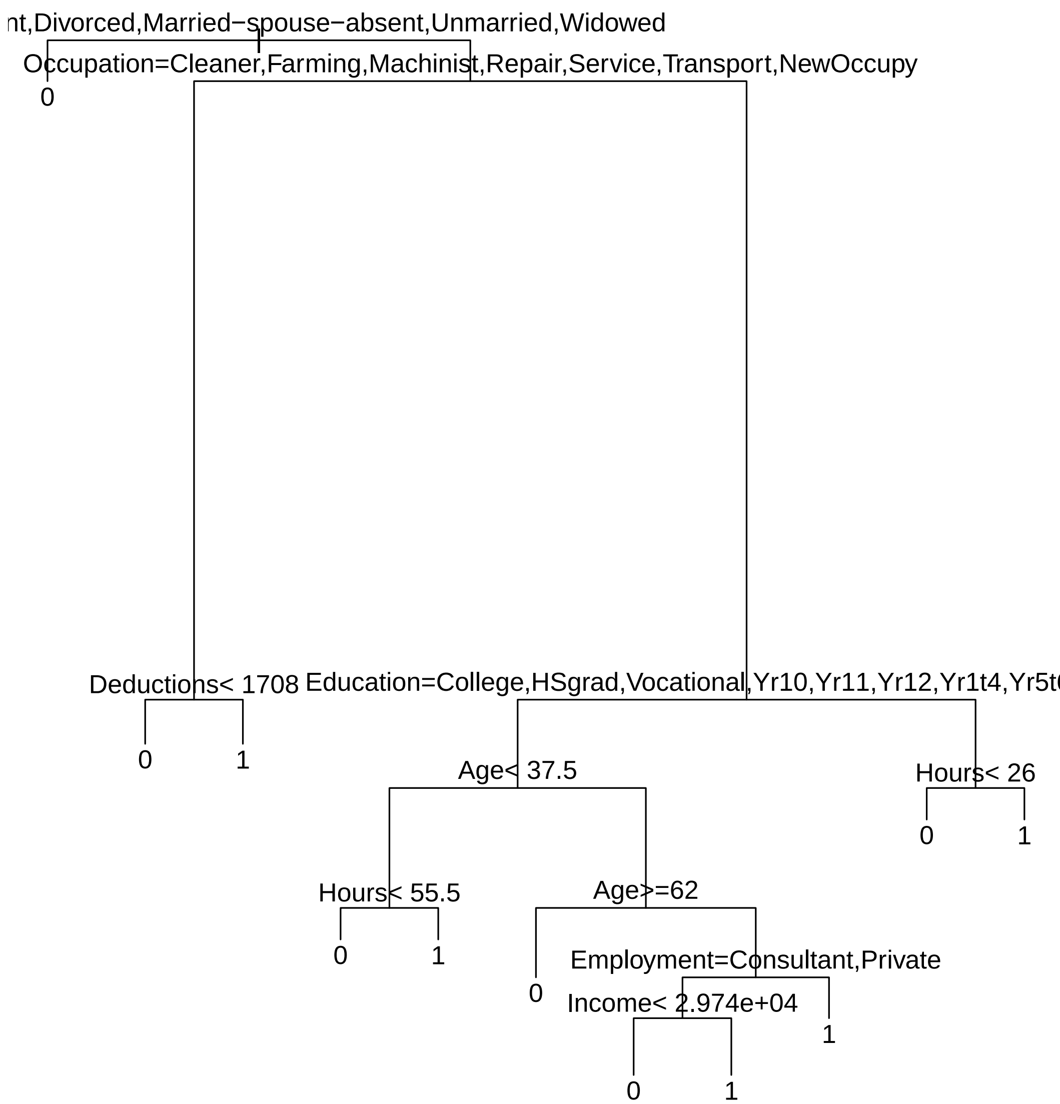
```
f1_score = (2 * precision * recall)/(precision + recall)
f1_score
```

```
## [1] 0.9183673
```

From the accuracy we can see, Naive Bayesina didn't give us a better result than logistic regression. This is because we can only use categorical variables in the model. However, some numeric variables are also important in this model. Thus, Naive Bayesian didn't show a good performance here.

### Desicion Trees

```
library(MASS)
library(tree)
library(rpart)

set.seed(1)
train <- sample(1:nrow(audit2), 0.9 * nrow(audit2))

auditTree <- rpart(TARGET_Adjusted ~ . - RISK_Adjustment + Income + Deductions +
    Hours + Age, data = audit2[train, ], method = "class")

plot(auditTree)
text(auditTree, pretty = 0)
```




```
summary(auditTree)
```

```
## Call:
## rpart(formula = TARGET_Adjusted ~ . - RISK_Adjustment + Income +
##     Deductions + Hours + Age, data = audit2[train, ], method = "class")
##   n= 1800
##
##           CP nsplit rel error    xerror       xstd
## 1 0.11835749      0 1.0000000 1.0000000 0.04312660
```

```
## 2 0.03985507      2 0.7632850 0.7850242 0.03941859
## 3 0.02657005      4 0.6835749 0.7415459 0.03854451
## 4 0.01690821      5 0.6570048 0.7077295 0.03783152
## 5 0.01207729      6 0.6400966 0.7004831 0.03767479
## 6 0.01086957      8 0.6159420 0.6980676 0.03762223
## 7 0.01000000     10 0.5942029 0.7222222 0.03814076
##
## Variable importance
##    Marital Occupation     Income        Age  Education     Gender
##         26         18         17         10          9          8
##      Hours Employment Deductions
##          8          3          2
##
## Node number 1: 1800 observations,    complexity param=0.1183575
##   predicted class=0  expected loss=0.23  P(node) =1
##     class counts:  1386   414
##    probabilities: 0.770 0.230
##   left son=2 (966 obs) right son=3 (834 obs)
##   Primary splits:
##       Marital    splits as  LLRLLL, improve=124.89010, (0 missing)
##       Education  splits as  LRLRLRLRLLLLLLLL, improve= 75.12717, (0 missing)
##       Occupation splits as  LLRLLLLRRLLLLLL, improve= 65.85001, (0 missing)
##       Income     < 64708.42 to the right, improve= 51.83670, (0 missing)
##       Age        < 32.5     to the left,  improve= 48.04789, (0 missing)
##   Surrogate splits:
##       Income     < 61251.76 to the right, agree=0.824, adj=0.620, (0 split)
##       Gender     splits as  LR, agree=0.678, adj=0.305, (0 split)
##       Age        < 32.5     to the left,  agree=0.644, adj=0.233, (0 split)
##       Occupation splits as  LLRLLRLLRRLLLRL, agree=0.627, adj=0.194, (0 split)
##       Hours      < 40.5     to the left,  agree=0.603, adj=0.143, (0 split)
##
## Node number 2: 966 observations
##   predicted class=0  expected loss=0.05693582  P(node) =0.5366667
##     class counts:   911    55
##    probabilities: 0.943 0.057
##
## Node number 3: 834 observations,    complexity param=0.1183575
##   predicted class=0  expected loss=0.4304556  P(node) =0.4633333
##     class counts:   475   359
##    probabilities: 0.570 0.430
##   left son=6 (404 obs) right son=7 (430 obs)
##   Primary splits:
##       Occupation splits as  LRRL-L-RRLRLRLL, improve=59.77851, (0 missing)
##       Education  splits as  RRLRLRLRLLLLLLLL, improve=54.88437, (0 missing)
##       Deductions < 1708     to the left,  improve=22.99535, (0 missing)
##       Age        < 32.5     to the left,  improve=13.25911, (0 missing)
##       Hours      < 41.5     to the left,  improve=11.28206, (0 missing)
##   Surrogate splits:
##       Education  splits as  RRRRLRLRRLLLLLLL, agree=0.711, adj=0.403, (0 split)
##       Employment splits as  LRRRRR-LL, agree=0.573, adj=0.119, (0 split)
##       Hours      < 41.5     to the left,  agree=0.571, adj=0.114, (0 split)
##       Age        < 32.5     to the left,  agree=0.553, adj=0.077, (0 split)
##       Income     < 19817.39 to the left,  agree=0.536, adj=0.042, (0 split)
##
```

```
## Node number 6: 404 observations,    complexity param=0.01690821
##   predicted class=0  expected loss=0.2351485  P(node) =0.2244444
##     class counts:   309    95
##    probabilities: 0.765 0.235
##   left son=12 (397 obs) right son=13 (7 obs)
##   Primary splits:
##       Deductions < 1708     to the left,  improve=8.334377, (0 missing)
##       Education  splits as  LRRRLRLRLLLLLLLL, improve=6.324153, (0 missing)
##       Age        < 42.5     to the left,  improve=2.880322, (0 missing)
##       Hours      < 63.5     to the left,  improve=2.729904, (0 missing)
##       Employment splits as  RLRLRL-LR, improve=2.427843, (0 missing)
##
## Node number 7: 430 observations,    complexity param=0.03985507
##   predicted class=1  expected loss=0.3860465  P(node) =0.2388889
##     class counts:   166   264
##    probabilities: 0.386 0.614
##   left son=14 (202 obs) right son=15 (228 obs)
##   Primary splits:
##       Education  splits as  RRLRLR-RLLLLLLLL, improve=16.826480, (0 missing)
##       Age        < 31.5     to the left,  improve= 9.172073, (0 missing)
##       Deductions < 1299.833 to the left,  improve= 8.927086, (0 missing)
##       Hours      < 37.5     to the left,  improve= 8.492604, (0 missing)
##       Occupation splits as  -LR----RL-L-L--, improve= 6.860421, (0 missing)
##   Surrogate splits:
##       Occupation splits as  -LR----RL-L-L--, agree=0.644, adj=0.243, (0 split)
##       Age        < 30.5     to the left,  agree=0.572, adj=0.089, (0 split)
##       Employment splits as  LRLRRL---, agree=0.570, adj=0.084, (0 split)
##       Hours      < 39.5     to the left,  agree=0.560, adj=0.064, (0 split)
##       Income     < 50839.86 to the right, agree=0.549, adj=0.040, (0 split)
##
## Node number 12: 397 observations
##   predicted class=0  expected loss=0.2216625  P(node) =0.2205556
##     class counts:   309    88
##    probabilities: 0.778 0.222
##
## Node number 13: 7 observations
##   predicted class=1  expected loss=0  P(node) =0.003888889
##     class counts:     0     7
##    probabilities: 0.000 1.000
##
## Node number 14: 202 observations,    complexity param=0.03985507
##   predicted class=0  expected loss=0.4653465  P(node) =0.1122222
##     class counts:   108    94
##    probabilities: 0.535 0.465
##   left son=28 (69 obs) right son=29 (133 obs)
##   Primary splits:
##       Age        < 37.5     to the left,  improve=8.763299, (0 missing)
##       Deductions < 1299.833 to the left,  improve=4.762274, (0 missing)
##       Employment splits as  LLRLRR---, improve=4.598519, (0 missing)
##       Education  splits as  --R-R---RLRRLLLL, improve=3.607635, (0 missing)
##       Occupation splits as  -RR----LL-L-R--, improve=3.273916, (0 missing)
##   Surrogate splits:
##       Education splits as  --R-R---RRLRRRRR, agree=0.668, adj=0.029, (0 split)
##
```

```
## Node number 15: 228 observations,    complexity param=0.01207729
##   predicted class=1  expected loss=0.254386  P(node) =0.1266667
##     class counts:    58   170
##    probabilities: 0.254 0.746
##   left son=30 (17 obs) right son=31 (211 obs)
##   Primary splits:
##       Hours      < 26       to the left,  improve=5.664911, (0 missing)
##       Occupation splits as  -LR----LR-L-L--, improve=4.872180, (0 missing)
##       Employment splits as  LRRRLR---, improve=3.113450, (0 missing)
##       Deductions < 1481.333 to the left,  improve=2.837382, (0 missing)
##       Age        < 30.5     to the left,  improve=2.778934, (0 missing)
##   Surrogate splits:
##       Income < 272652.9 to the right, agree=0.934, adj=0.118, (0 split)
##       Age    < 65.5     to the right, agree=0.930, adj=0.059, (0 split)
##
## Node number 28: 69 observations,    complexity param=0.01207729
##   predicted class=0  expected loss=0.2608696  P(node) =0.03833333
##     class counts:    51    18
##    probabilities: 0.739 0.261
##   left son=56 (62 obs) right son=57 (7 obs)
##   Primary splits:
##       Hours      < 55.5     to the left,  improve=5.5395710, (0 missing)
##       Employment splits as  LLRLLR---, improve=2.3996790, (0 missing)
##       Income     < 63321.59 to the left,  improve=2.1196670, (0 missing)
##       Education  splits as  --R-L---R-R---LL, improve=1.3063310, (0 missing)
##       Age        < 26.5     to the right, improve=0.7982381, (0 missing)
##   Surrogate splits:
##       Employment splits as  LLLLLR---, agree=0.928, adj=0.286, (0 split)
##
## Node number 29: 133 observations,    complexity param=0.02657005
##   predicted class=1  expected loss=0.4285714  P(node) =0.07388889
##     class counts:    57    76
##    probabilities: 0.429 0.571
##   left son=58 (15 obs) right son=59 (118 obs)
##   Primary splits:
##       Age        < 62       to the right, improve=6.489750, (0 missing)
##       Education  splits as  --R-R---RL-LLLLL, improve=3.392857, (0 missing)
##       Deductions < 1299.833 to the left,  improve=3.126857, (0 missing)
##       Occupation splits as  -RR----LR-L-R--, improve=2.914331, (0 missing)
##       Hours      < 32.5     to the left,  improve=2.897243, (0 missing)
##   Surrogate splits:
##       Education splits as  --R-R---RR-RRLRR, agree=0.895, adj=0.067, (0 split)
##
## Node number 30: 17 observations
##   predicted class=0  expected loss=0.3529412  P(node) =0.009444444
##     class counts:    11     6
##    probabilities: 0.647 0.353
##
## Node number 31: 211 observations
##   predicted class=1  expected loss=0.2227488  P(node) =0.1172222
##     class counts:    47   164
##    probabilities: 0.223 0.777
##
## Node number 56: 62 observations
```

```
##   predicted class=0  expected loss=0.1935484  P(node) =0.03444444
##     class counts:    50    12
##    probabilities: 0.806 0.194
##
## Node number 57: 7 observations
##   predicted class=1  expected loss=0.1428571  P(node) =0.003888889
##     class counts:     1     6
##    probabilities: 0.143 0.857
##
## Node number 58: 15 observations
##   predicted class=0  expected loss=0.1333333  P(node) =0.008333333
##     class counts:    13     2
##    probabilities: 0.867 0.133
##
## Node number 59: 118 observations,    complexity param=0.01086957
##   predicted class=1  expected loss=0.3728814  P(node) =0.06555556
##     class counts:    44    74
##    probabilities: 0.373 0.627
##   left son=118 (89 obs) right son=119 (29 obs)
##   Primary splits:
##       Employment splits as  LLRRRR---, improve=4.244945, (0 missing)
##       Deductions < 1299.833 to the left,  improve=2.069324, (0 missing)
##       Education  splits as  --R-L---RL-LL-LL, improve=1.957705, (0 missing)
##       Income     < 29715.21 to the left,  improve=1.811441, (0 missing)
##       Occupation splits as  -RR----LR-L-R--, improve=1.543584, (0 missing)
##   Surrogate splits:
##       Income < 4346.63  to the right, agree=0.78, adj=0.103, (0 split)
##       Hours  < 61       to the left,  agree=0.78, adj=0.103, (0 split)
##
## Node number 118: 89 observations,    complexity param=0.01086957
##   predicted class=1  expected loss=0.4494382  P(node) =0.04944444
##     class counts:    40    49
##    probabilities: 0.449 0.551
##   left son=236 (25 obs) right son=237 (64 obs)
##   Primary splits:
##       Income     < 29742.27 to the left,  improve=3.6961940, (0 missing)
##       Occupation splits as  -RL----LR-L-R--, improve=1.3388550, (0 missing)
##       Education  splits as  --R-L---RL-LL-LL, improve=1.2813070, (0 missing)
##       Gender     splits as  RL, improve=0.7315877, (0 missing)
##       Hours      < 55.5     to the right, improve=0.5107666, (0 missing)
##   Surrogate splits:
##       Education  splits as  --R-R---LR-RR-RL, agree=0.753, adj=0.12, (0 split)
##       Occupation splits as  -RR----RL-R-R--, agree=0.730, adj=0.04, (0 split)
##
## Node number 119: 29 observations
##   predicted class=1  expected loss=0.137931  P(node) =0.01611111
##     class counts:     4    25
##    probabilities: 0.138 0.862
##
## Node number 236: 25 observations
##   predicted class=0  expected loss=0.32  P(node) =0.01388889
##     class counts:    17     8
##    probabilities: 0.680 0.320
##
```

```
## Node number 237: 64 observations
##   predicted class=1  expected loss=0.359375  P(node) =0.03555556
##     class counts:    23    41
##    probabilities: 0.359 0.641
```

```r
auditPred <- predict(auditTree, audit2[-train, ], type = "class")
dtt = table(auditPred, audit2[-train, ]$TARGET_Adjusted)
confusionMatrix(auditPred, audit2[-train, ]$TARGET_Adjusted)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 137  22
##          1  14  27
##
##                Accuracy : 0.82
##                  95% CI : (0.7596, 0.8706)
##     No Information Rate : 0.755
##     P-Value [Acc > NIR] : 0.01752
##
##                   Kappa : 0.4851
##  Mcnemar's Test P-Value : 0.24335
##
##             Sensitivity : 0.9073
##             Specificity : 0.5510
##          Pos Pred Value : 0.8616
##          Neg Pred Value : 0.6585
##              Prevalence : 0.7550
##          Detection Rate : 0.6850
##    Detection Prevalence : 0.7950
##       Balanced Accuracy : 0.7292
##
##        'Positive' Class : 0
##
```

```r
derror = (dtt[1, 2] + dtt[2, 1])/200
derror
```

```
## [1] 0.18
```

```r
dprecision = dtt[1, 1]/(dtt[1, 1] + dtt[1, 2])
dprecision
```

```
## [1] 0.8616352
```

```r
drecall = dtt[1, 1]/(dtt[1, 1] + dtt[2, 1])
drecall
```

```
## [1] 0.9072848
```

```r
df1_score = (2 * dprecision * drecall)/(dprecision + drecall)
df1_score
```

```
## [1] 0.883871
```

Though we didn't use 10-fold cross valiadation on the Naive Bayesian and Desicion Tree model. But from the general accuracy result, we can see, the Naive Bayesian and Desicion Tree model all didn't show better predicitons than logistic regression. Therefore, the best model here is logistic regression by dropping Marital and Income variable.

**–b.For the best model, compute the odds ratio and interpret the effect of each predictors.**

Since result from 10-fold cross valiadation is not stable. I got the best result by dropping Marital and Income, but it may be different from the output in the PDF file.

```
library(aod)
library(Rcpp)

audit3 = audit2[c(-4, -6)]
Xdel = model.matrix(TARGET_Adjusted ~ ., data = audit3)[, -1]
xtrain = Xdel
ytrain = audit3$TARGET_Adjusted
m2 = glm(TARGET_Adjusted ~ ., family = binomial, data = data.frame(TARGET_Adjusted = ytrain,
    xtrain))

summary(m2)
```

```
##
## Call:
## glm(formula = TARGET_Adjusted ~ ., family = binomial, data = data.frame(TARGET_Adjusted = ytrain,
##     xtrain))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1801  -0.2824  -0.1636  -0.0002   4.6377
##
## Coefficients: (1 not defined because of singularities)
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -6.966e+00  1.479e+00  -4.710 2.48e-06 ***
## Age                     3.230e-02  9.623e-03   3.357 0.000789 ***
## EmploymentPrivate       5.148e-01  4.742e-01   1.086 0.277666
## EmploymentPSFederal     5.251e-02  7.576e-01   0.069 0.944744
## EmploymentPSLocal       2.520e-01  6.607e-01   0.381 0.702915
## EmploymentPSState       1.316e-01  7.207e-01   0.183 0.855110
## EmploymentSelfEmp      -1.722e-01  7.845e-01  -0.219 0.826302
## EmploymentUnemployed   -1.127e+01  3.956e+03  -0.003 0.997727
## EmploymentVolunteer    -1.457e+01  3.956e+03  -0.004 0.997061
## EmploymentNewEmploy     9.166e-01  1.341e+00   0.683 0.494296
## EducationBachelor       5.046e-01  6.495e-01   0.777 0.437199
## EducationCollege       -8.442e-01  6.980e-01  -1.209 0.226497
## EducationDoctorate      5.151e-01  1.042e+00   0.494 0.621192
## EducationHSgrad        -5.105e-01  6.617e-01  -0.772 0.440347
## EducationMaster        -2.059e-01  8.191e-01  -0.251 0.801509
## EducationPreschool     -1.380e+01  1.591e+03  -0.009 0.993079
## EducationProfessional   1.368e+00  1.016e+00   1.347 0.177999
## EducationVocational    -4.747e-01  8.607e-01  -0.552 0.581273
## EducationYr10          -3.042e+00  2.003e+00  -1.518 0.128952
## EducationYr11          -7.500e-02  9.747e-01  -0.077 0.938662
## EducationYr12          -5.045e+00  1.508e+01  -0.334 0.738033
## EducationYr1t4         -1.456e+01  1.579e+03  -0.009 0.992645
## EducationYr5t6         -1.397e+00  1.504e+00  -0.929 0.353118
```

```
## EducationYr7t8         -1.460e+01  6.487e+02  -0.023 0.982042
## EducationYr9           -1.147e+01  1.666e+02  -0.069 0.945114
## OccupationClerical      1.107e+00  1.124e+00   0.985 0.324711
## OccupationExecutive     2.062e+00  1.066e+00   1.934 0.053127 .
## OccupationFarming       2.492e-01  1.454e+00   0.171 0.863911
## OccupationHome         -1.203e+01  1.732e+03  -0.007 0.994458
## OccupationMachinist     2.765e-01  1.251e+00   0.221 0.825043
## OccupationMilitary     -1.211e+01  3.956e+03  -0.003 0.997557
## OccupationProfessional  1.784e+00  1.090e+00   1.636 0.101749
## OccupationProtective    1.573e+00  1.312e+00   1.199 0.230552
## OccupationRepair        9.836e-01  1.091e+00   0.901 0.367501
## OccupationSales         8.851e-01  1.130e+00   0.783 0.433499
## OccupationService       6.796e-02  1.246e+00   0.055 0.956493
## OccupationSupport       1.546e+00  1.210e+00   1.278 0.201424
## OccupationTransport     1.211e+00  1.131e+00   1.071 0.284226
## OccupationNewOccupy            NA         NA      NA       NA
## GenderMale              9.125e-01  3.172e-01   2.877 0.004016 **
## Deductions              9.679e-04  3.000e-04   3.226 0.001255 **
## Hours                   1.619e-02  1.035e-02   1.564 0.117732
## RISK_Adjustment         4.640e-03  7.254e-04   6.396 1.60e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2164.3  on 1999  degrees of freedom
## Residual deviance:  524.4  on 1958  degrees of freedom
## AIC: 608.4
##
## Number of Fisher Scoring iterations: 16
```

We can use the confint function to obtain confidence intervals for the coefficient estimates. This is important because the wald.test function refers to the coefficients by their order in the model. Generate odds ratios and 95% CI

```r
exp(cbind(OR = coef(m2), confint(m2)))
```

```
##                                  OR         2.5 %        97.5 %
## (Intercept)            9.431707e-04  2.989624e-05  1.292917e-02
## Age                    1.032826e+00  1.013450e+00  1.052539e+00
## EmploymentPrivate      1.673334e+00  7.067693e-01  4.658751e+00
## EmploymentPSFederal    1.053912e+00  2.153969e-01  4.505271e+00
## EmploymentPSLocal      1.286593e+00  3.437726e-01  4.780178e+00
## EmploymentPSState      1.140645e+00  2.570003e-01  4.621596e+00
## EmploymentSelfEmp      8.418425e-01  1.594278e-01  3.711391e+00
## EmploymentUnemployed   1.277387e-05            NA           Inf
## EmploymentVolunteer    4.685239e-07            NA           Inf
## EmploymentNewEmploy    2.500658e+00  2.048447e-01  6.240808e+01
## EducationBachelor      1.656292e+00  5.242108e-01  7.101457e+00
## EducationCollege       4.299107e-01  1.194203e-01  1.968886e+00
## EducationDoctorate     1.673785e+00  1.809317e-01  1.273757e+01
## EducationHSgrad        6.001654e-01  1.842791e-01  2.616457e+00
## EducationMaster        8.138947e-01  1.644263e-01  4.441354e+00
## EducationPreschool     1.017263e-06 5.699097e-282  2.606163e-64
## EducationProfessional  3.928666e+00  5.019180e-01  2.973473e+01
```

```
## EducationVocational    6.220819e-01  1.063627e-01  3.540281e+00
## EducationYr10          4.775474e-02  5.448176e-04  1.078402e+00
## EducationYr11          9.277408e-01  1.112952e-01  6.187943e+00
## EducationYr12          6.444357e-03  9.352632e-08  7.092363e+00
## EducationYr1t4         4.758139e-07 1.442299e-284 7.919345e-148
## EducationYr5t6         2.474596e-01  6.955075e-03  3.328538e+00
## EducationYr7t8         4.553503e-07 1.100746e-118  3.802728e-18
## EducationYr9           1.043904e-05  2.468268e-06  4.254166e-05
## OccupationClerical     3.026400e+00  4.648905e-01  5.991238e+01
## OccupationExecutive    7.861467e+00  1.456759e+00  1.469517e+02
## OccupationFarming      1.282967e+00  5.599020e-02  3.444776e+01
## OccupationHome         5.948440e-06 9.074081e-307  1.516759e-89
## OccupationMachinist    1.318458e+00  1.244379e-01  2.930648e+01
## OccupationMilitary     5.497328e-06            NA           Inf
## OccupationProfessional 5.955081e+00  1.028837e+00  1.140908e+02
## OccupationProtective   4.822796e+00  3.916845e-01  1.149916e+02
## OccupationRepair       2.674038e+00  4.530311e-01  5.118989e+01
## OccupationSales        2.423141e+00  3.628109e-01  4.819542e+01
## OccupationService      1.070319e+00  1.031900e-01  2.368971e+01
## OccupationSupport      4.693044e+00  5.416262e-01  1.010671e+02
## OccupationTransport    3.356795e+00  4.975995e-01  6.676087e+01
## OccupationNewOccupy              NA            NA            NA
## GenderMale             2.490637e+00  1.365051e+00  4.765850e+00
## Deductions             1.000968e+00  1.000356e+00  1.001544e+00
## Hours                  1.016324e+00  9.958765e-01  1.037189e+00
## RISK_Adjustment        1.004650e+00  1.003460e+00  1.006395e+00
```

The transformation from odds to log of odds is the log transformation. Usually, the greater the odds, the greater the log of odds and vice versa. If the OR is $>1$ the control is better than the intervention.If the OR is $<1$ the intervention is better than the control.

In this model we can see, most the predictors's OR is larger than 1, that means the control is better than the intervention.

Take Age as an exmaple, the OR of Age is 9.431707e-04; 95% confidence interval [CI], 2.989624e-05 to 1.292917e-02. The odds of the other predictors and levels were 90.57% less than in the Age with the true population effect between 97.1% and 98.7%. This result was statistically significant.

The other predictors are the same. EmploymentPrivate is 1.032826e+00, EmploymentPSFederal is 1.053912e+00,EmploymentPSLocal is 1.286593e+00,EmploymentPSState is 1.140645e+00, EmploymentSelfEmp is 8.418425e-01, EmploymentUnemployed is 1.277387e-05,EmploymentVolunteer is 4.685239e-07. Except SelfEmp and Volunteer, they other levels in Employment all have lots of effects on TARGET_Adjustment.EducationCollege is 4.299107e-01,

EducationHSgrad is 6.001654e-01, EducationMaster is 8.138947e-01, EducationVocational is 6.220819e-01,EducationYr11 is 9.277408e-01. Most of the levels in Education have large OR, which means Education generally has little effect on model.

OccupationExecutive is 7.861467e+00. OccupationHome is 5.948440e-06. OccupationMilitary is 5.497328e-06, OccupationProfessional is 5.955081e+00, OccupationProtective is 4.822796e+00. For most of the levels in Occupation,the odds ratio is larger than 1. We can think Occupation didn't have much effect onthe model.

GenderMale is 2.490637e+00. Deductions is 1.000968e+00.Hours is 1.016324e+00. We can think all this predictors have much effects on the best model. Since these predictors have smaller odds ratio and within the appropriate confident intervevals.

We can also use varImp to test the importance of variables.

```r
varImp(m2)
```

```
##                           Overall
## Age                   3.356511969
## EmploymentPrivate     1.085576918
## EmploymentPSFederal   0.069308362
## EmploymentPSLocal     0.381388127
## EmploymentPSState     0.182602133
## EmploymentSelfEmp     0.219446707
## EmploymentUnemployed  0.002848229
## EmploymentVolunteer   0.003683775
## EmploymentNewEmploy   0.683492190
## EducationBachelor     0.776931989
## EducationCollege      1.209432885
## EducationDoctorate    0.494161101
## EducationHSgrad       0.771607938
## EducationMaster       0.251394422
## EducationPreschool    0.008673934
## EducationProfessional 1.346942239
## EducationVocational   0.551526574
## EducationYr10         1.518248947
## EducationYr11         0.076951496
## EducationYr12         0.334458866
## EducationYr1t4        0.009218706
## EducationYr5t6        0.928557808
## EducationYr7t8        0.022508546
## EducationYr9          0.068843482
## OccupationClerical    0.984823237
## OccupationExecutive   1.933885211
## OccupationFarming     0.171397163
## OccupationHome        0.006945596
## OccupationMachinist   0.221063931
## OccupationMilitary    0.003061349
## OccupationProfessional 1.636432115
## OccupationProtective  1.198937111
## OccupationRepair      0.901163925
## OccupationSales       0.783218211
## OccupationService     0.054555582
## OccupationSupport     1.277505576
## OccupationTransport   1.070874025
## GenderMale            2.876895893
## Deductions            3.226101123
## Hours                 1.564365979
## RISK_Adjustment       6.395741030
```

The list shows the importance of each predictors. The higher their overall score, the more important they are to the model. Excluding the RISK_Adjusted, the most important one is Age, then is Deductions, the third one is GernderMale. The results are consistent with the OR results. Hours, OccupationTransport, OccupationSport, OccupationProtective, OccupationProfessional,OccupationExecutive,EducationYr10, EducationProfessional, EducationCollege and EmploymentPrivate all have middle effects on the model. And the other predictors and levels, they have little effects on model and are the least important.

**–c.Apply linear and non-linear regression analysis to predict RISK__Adjustment. Evaluate the models through cross-validation and on holdout samples.**

**–Use all predictors in a standard linear regression model to predict the response variable. Report the model performance using R2, adjusted R2 and RMSE. Interpret the regression result.**

```
audit3 = audit2[c(-11)]  #drop TARGET_Adju
fit1 = lm(RISK_Adjustment ~ ., data = audit3)
summary(fit1)
```

```
##
## Call:
## lm(formula = RISK_Adjustment ~ ., data = audit3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -14702  -2576   -590    609 104027
##
## Coefficients: (1 not defined because of singularities)
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   6.101e+02  1.856e+03   0.329   0.7425
## Age                           3.448e+01  1.729e+01   1.994   0.0463 *
## EmploymentPrivate            -1.426e+03  7.251e+02  -1.966   0.0494 *
## EmploymentPSFederal          -1.898e+03  1.215e+03  -1.562   0.1184
## EmploymentPSLocal             9.483e+02  1.057e+03   0.897   0.3699
## EmploymentPSState            -1.700e+03  1.199e+03  -1.418   0.1562
## EmploymentSelfEmp            -1.552e+02  1.141e+03  -0.136   0.8919
## EmploymentUnemployed         -4.297e+02  8.175e+03  -0.053   0.9581
## EmploymentVolunteer          -5.082e+03  8.190e+03  -0.621   0.5349
## EmploymentNewEmploy          -1.184e+03  1.380e+03  -0.858   0.3908
## EducationBachelor            -6.205e+02  1.082e+03  -0.574   0.5664
## EducationCollege             -1.017e+03  1.056e+03  -0.963   0.3354
## EducationDoctorate            9.414e+02  1.899e+03   0.496   0.6202
## EducationHSgrad              -1.613e+03  1.040e+03  -1.551   0.1212
## EducationMaster               9.971e+02  1.311e+03   0.760   0.4471
## EducationPreschool           -1.974e+03  3.494e+03  -0.565   0.5720
## EducationProfessional         8.213e+03  1.966e+03   4.179 3.06e-05 ***
## EducationVocational          -1.702e+03  1.316e+03  -1.294   0.1959
## EducationYr10                 5.945e+01  1.467e+03   0.041   0.9677
## EducationYr11                -1.530e+03  1.390e+03  -1.101   0.2711
## EducationYr12                -1.480e+03  2.205e+03  -0.671   0.5021
## EducationYr1t4               -3.874e+03  3.453e+03  -1.122   0.2621
## EducationYr5t6               -2.717e+03  1.983e+03  -1.371   0.1706
## EducationYr7t8               -2.619e+03  1.716e+03  -1.526   0.1272
## EducationYr9                 -2.592e+03  1.854e+03  -1.398   0.1622
## MaritalDivorced              -7.208e+02  6.498e+02  -1.109   0.2675
## MaritalMarried                2.724e+03  5.177e+02   5.260 1.59e-07 ***
## MaritalMarried-spouse-absent -1.029e+03  1.786e+03  -0.576   0.5645
## MaritalUnmarried             -2.250e+02  1.055e+03  -0.213   0.8311
## MaritalWidowed               -1.011e+03  1.231e+03  -0.821   0.4115
## OccupationClerical            5.628e+01  1.044e+03   0.054   0.9570
## OccupationExecutive           2.740e+02  1.029e+03   0.266   0.7901
## OccupationFarming            -1.941e+03  1.405e+03  -1.381   0.1675
## OccupationHome                2.800e+02  3.732e+03   0.075   0.9402
```

```
## OccupationMachinist          -1.090e+03  1.097e+03  -0.994   0.3202
## OccupationMilitary            1.100e+02  8.148e+03   0.014   0.9892
## OccupationProfessional        5.630e+01  1.101e+03   0.051   0.9592
## OccupationProtective         -9.032e+02  1.610e+03  -0.561   0.5749
## OccupationRepair             -8.450e+02  1.018e+03  -0.830   0.4068
## OccupationSales              -1.897e+02  1.042e+03  -0.182   0.8555
## OccupationService            -5.454e+02  1.027e+03  -0.531   0.5953
## OccupationSupport             9.031e+02  1.465e+03   0.617   0.5375
## OccupationTransport          -2.120e+03  1.166e+03  -1.818   0.0691 .
## OccupationNewOccupy                  NA         NA      NA       NA
## Income                        2.860e-03  3.078e-03   0.929   0.3528
## GenderMale                   -1.108e+02  4.981e+02  -0.223   0.8239
## Deductions                    1.005e+00  5.359e-01   1.875   0.0609 .
## Hours                         3.033e+01  1.641e+01   1.848   0.0648 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8032 on 1953 degrees of freedom
## Multiple R-squared:  0.09416,    Adjusted R-squared:  0.07283
## F-statistic: 4.413 on 46 and 1953 DF,  p-value: < 2.2e-16
```

```r
model.mse = mean(residuals(fit1)^2)
rmse = sqrt(model.mse)
rmse
```

```
## [1] 7937.427
```

Multiple R-squared = 0.09416, which means the model accounts for 9.416% of the variance in RISK_Adjustment.

Adjust R-squared = 0.07283, means 7.283% of variability in the response RISK_Adjustment is explained by the model with penalty for the number of estimated coefficients.

Adjust R-square is more realistic because it accounts for the number of variables in the model.

p-value: The coefficient is significantly different from zero at the $p < 0.001$ level. Therefore, the coefficients for EducationPrefessional, MaritalMarried are significant with p-values less than 0.001. Whereas, the coefficients for the rest levels of variales are not significant.

Explain each predictor, for example Age, the coefficient is 3.448e+01, means an increase of 1 percent in Salary can cause 3.448e+01 increase in RISK_Adjustment.

RMSE is 7937.427, used to measure differences between value predicted by a model of an estimator and value actually observed. It will be used to compare different models later.

**–Use different combination of predictors in standard linear and non-linear regression models to predict the response variable. Evaluate which model performs better using out-of-sample RMSE.**

```r
audit3 = audit2[c(-2, -5)]
```

I excluded the employment and Occupation two variables because these two factor got some problem. Some of the levels in the factor are too few observations in it. Thus, they would be taken into new levels in the factor and can't be analyzed. Therefore, I excluded these two variables so that we can do leave one out on linear and non-linear regression.

Define leave-one-out function and evaluated by RMSE.

```
leave.one.out <- function(formula, data) {
    n = length(audit3$RISK_Adjustment)
    error = dim(n)
    for (k in 1:n) {
        id = c(1:n)
        id.train = id[id != k]
        fit = lm(formula, data = audit3[id.train, ])
        predicted = predict(fit, newdata = audit3[-id.train, ])
        observation = audit3$RISK_Adjustment[-id.train]
        error[k] = predicted - observation
    }
    rmse = sqrt(mean(error^2))
    return(rmse)
}
```

Linear Regression

```
formulaA = RISK_Adjustment ~ Age + Education + Marital + Income + Gender + Hours +
    Deductions
leave.one.out(formulaA, audit3)
```

```
## [1] 8106.716
```

```
formulaB = RISK_Adjustment ~ Age + Education + Marital + Hours
leave.one.out(formulaB, audit3)
```

```
## [1] 8100.306
```

Non-linear Regression

```
formulaC = RISK_Adjustment ~ poly(Age, degree = 2) + poly(Hours, degree = 3) +
    Income + Deductions
leave.one.out(formulaC, audit3)
```

```
## [1] 8236.966
```

```
formulaD = RISK_Adjustment ~ poly(Age, degree = 2) + poly(Hours, degree = 4) +
    Income
leave.one.out(formulaD, audit3)
```

```
## [1] 8238.219
```

The best model should have the lowest RSME. The second model in linear regression shows the lowest RMSE and therefore, that is the best model.

**–From the best model, identify the most important predictor in the model, and explain how you determine the importance of the predictors.**

We can find the most important predictor in the model calculating the RSME after dropping that predictorw. If the RSME gets very large, we suppose that this predictor is very important to the model.

Drop Age

```
leave.one.out(RISK_Adjustment ~ Education + Marital + Hours, data = audit3)
```

```
## [1] 8109.512
```

Drop Education

```r
leave.one.out(RISK_Adjustment ~ Age + Marital + Hours, data = audit3)
```

```
## [1] 8148.792
```

Drop Marital

```r
leave.one.out(RISK_Adjustment ~ Age + Education + Hours, data = audit3)
```

```
## [1] 8200.658
```

Drop Hours

```r
leave.one.out(RISK_Adjustment ~ Age + Education + Marital, data = audit3)
```

```
## [1] 8104.054
```

We can see that by dropping Marital, we get the largest RMSE. Thus, Marital is the most important predictor in the best model.