Logistic Regression: Audit Analysis

Jonathan Velez

January 24, 2017

Introduction

The "audit.csv" dataset is an artificially constructed data set that contains the characteristics of n = 2000 individual tax returns. This analysis explores the data by preparing useful graphs and tables, generating predictive models, and evaluating the resulting models. Here the objective is to predict the binary (TARGET_Adjusted) and continuous (RISK_Adjustment) target variables.

The data set includes the following variables:
* ID: Unique identifier for each person.
* Age: Age of person.
* Employment: Type of employment.
* Education: Highest level of education.
* Marital: Current marital status.
* Occupation: Type of occupation.
* Income: Amount of income declared.
* Gender: Gender of person.
* Deductions: Total amount of expenses that a person claims in their financial statement.
* Hours: Average hours worked on a weekly basis.
* TARGET_Adjusted: The binary target variable for classification modeling, indicating nonproductive and productive audits (0 and 1, respectively). Productive audits are those that result in an adjustment being made to a client's financial statement.
* RISK_Adjustment: The continuous target variable; this variable records the monetary amount of any adjustment to the person's financial claims as a result of a productive audit. This variable is a measure of the size of the risk associated with the person.

Data Exploration

Preparation

```
# Load required packages
library("ggplot2")   ## data visualization
library("e1071")     ## skewness
library("knitr")     ## summary table
library("car")       ## scatter plot matrix
library("reshape2")  ## percentage table
library("plyr")      ## percentage table
library("ROCR")
library("leaps")

# Load data
data.file = "http://www.yurulin.com/class/spring2017_datamining/data/audit.csv"
df = read.csv(data.file, header = TRUE, sep = ',')
df = df[,-1] # remove the ID column
df$TARGET_Adjusted = factor(df$TARGET_Adjusted, levels=c("0", "1"))

# Identify any missing values and handle missing data appropriately
summary(df)
```

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

```
##
##
```

```r
# There are 100 NA's in Employment and 101 NA's in Occupation,
# but the interpretation of these missing values are unknown.
# Additionally, there are very few instances of Unemployed and
# Volunteer for Employment, and very few instances of Home and
# Military for Occupation. These records are removed.
summary(df$Employment)
```

```
## Consultant    Private  PSFederal    PSLocal    PSState    SelfEmp
##        148       1411         69        119         72         79
## Unemployed  Volunteer       NA's
##          1          1        100
```

```r
summary(df$Occupation)
```

```
##      Cleaner     Clerical    Executive      Farming         Home
##           91          232          289           58            5
##    Machinist     Military Professional   Protective       Repair
##          139            1          247           40          225
##        Sales      Service      Support    Transport         NA's
##          206          210           49          107          101
```

```r
df = na.omit(df) ## remove incomplete rows with NA's
df = df[-which(df$Employment == "Volunteer"),]
df$Employment = droplevels(df$Employment) ## drop unused levels
df = df[-which(df$Occupation == "Home"),]
df = df[-which(df$Occupation == "Military"),]
df$Occupation = droplevels(df$Occupation) ## drop unused levels

# Recode Education categories
df$Education = as.character(df$Education)
df$Education[df$Education=="Doctorate"] = "PostGraduate"
df$Education[df$Education=="Professional"] = "PostGraduate"
df$Education[df$Education=="Master"] = "PostGraduate"
df$Education[df$Education=="Associate"] = "SomeCol/2Yr"
df$Education[df$Education=="College"] = "SomeCol/2Yr"
df$Education[df$Education=="Vocational"] = "SomeCol/2Yr"
df$Education[df$Education=="Yr12"] = "LessThanHS"
df$Education[df$Education=="Yr11"] = "LessThanHS"
df$Education[df$Education=="Yr10"] = "LessThanHS"
df$Education[df$Education=="Yr9"] = "LessThanHS"
df$Education[df$Education=="Yr7t8"] = "LessThanHS"
df$Education[df$Education=="Yr5t6"] = "LessThanHS"
df$Education[df$Education=="Yr1t4"] = "LessThanHS"
df$Education[df$Education=="Preschool"] = "LessThanHS"
df$Education = as.factor(df$Education)
df$Education = factor(
  df$Education,
  levels = c("LessThanHS", "HSgrad", "SomeCol/2Yr", "Bachelor", "PostGraduate")
  )

dim(df) ## 1892 records and 11 features used in analysis
```

```
## [1] 1892   11
```

Response Variables

The dataset contains 1892 instances and 11 features after the initial preparation. 447 out of these 1892 instances were targeted for adjustment, resulting in a baseline probability of 23.62% for being targeted for adjustment. The density distribution of RISK_Adjustment where the target instance was adjusted reveals a multimodal distribution with positive non-zero skewness. The skewness of the distribution is greater than +5, indicating that the distribution is extremely skewed in the positive direction.

```
# Data summary of TARGET_Adjusted
summary(df$TARGET_Adjusted)
```

```
##    0    1
## 1445  447
```

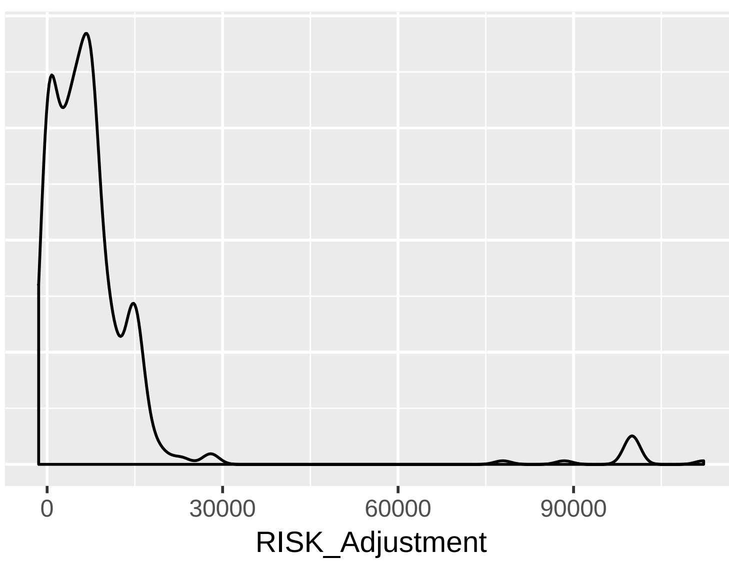
```
## Baseline probability of being targeted for adjustment
length(df$TARGET_Adjusted[df$TARGET_Adjusted=="1"])/length(df$TARGET_Adjusted)
```

```
## [1] 0.2362579
```

```
c(summary(df$RISK_Adjustment[df$TARGET_Adjusted=="1"]),
                    SD=round(sd(df$RISK_Adjustment[df$TARGET_Adjusted=="1"]), 2))
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.       SD
##  -1453.0   2283.0   5848.0   8604.0   9371.0 112200.0  15208.9
```

```
# Explore the density distribution of RISK_Adjustment where
# the target intance resulted in an adjustment
no.y = theme(axis.title.y=element_blank(), ## remove clutter on y axis
axis.text.y=element_blank(),
axis.ticks.y=element_blank())
ggplot(df[df$TARGET_Adjusted=="1",], aes(x=RISK_Adjustment)) + geom_density() + no.y
```




```
shapiro.test(df$RISK_Adjustment[df$TARGET_Adjusted=="1"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$RISK_Adjustment[df$TARGET_Adjusted == "1"]
## W = 0.4256, p-value < 2.2e-16
```

```r
skewness(df$RISK_Adjustment[df$TARGET_Adjusted=="1"])
```

```
## [1] 5.109743
```

Quantitative Predictor Variables

A table describing the central tendency and spread of each quantitative predictor variable is included below. Each quantitative predictor is explored with respect to TARGET_Adjusted and evaluated for any correlation with RISK_Adjustment.

The density distribution of Age reveals a bimodal distribution that is moderately skewed in the positive direction. The distribution of Age of adjusted instances is centered at around 44, whereas the distribution of Age of instances not adjusted is centered at around 36. The ANOVA table for Age by TARGET_Adjusted demonstrates an F-statistic of 120.9 with a p-value less than 2e-16, and clearly indicates a rejection of the null hypothesis of equal means for instances that were adjusted and not adjusted.

The density distribution of Income reveals a unimodal distribution that is highly skewed in the positive direction. The distribution of Income of adjusted instances is centered at around $60000, whereas the distribution of Income of instances not adjusted is centered at around $92000. The ANOVA table for Income by TARGET_Adjusted demonstrates an F-statistic of 76.01 with a p-value less than 2e-16, and clearly indicates a rejection of the null hypothesis of equal means for instances that were adjusted and not adjusted.

The density distribution of Deductions reveals an extremely skewed, multimodal distribution where most instances are zero. The distribution of Deductions of adjusted instances is centered at around $33, whereas the distribution of Deductions of instances not adjusted is centered at around $184. The ANOVA table for Deductions by TARGET_Adjusted demonstrates an F-statistic of 68.7 with a p-value less than 2e-16, and clearly indicates a rejection of the null hypothesis of equal means for instances that were adjusted and not adjusted.

The large amount of zero instances of Deductions suggests the need for exploration of the distribution of non-zero instances. Plotting the density distribution of non-zero Deductions instances reveals a bimodal distribution that is approximately normal. The distribution of non-zero Deductions of adjusted instances is centered at around $1205, whereas the distribution of non-zero Deductions of instances not adjusted is centered at around $2004. The ANOVA table for Deductions by TARGET_Adjusted demonstrates an F-statistic of 156.2 with a p-value less than 2e-16, and clearly indicates a rejection of the null hypothesis of equal means for instances that were adjusted and not adjusted. The results observed from exploring the distributions of all instances of Deductions and non-zero instances of Deductions suggests a possible need for including an additional qualitative predictor indicating whether or not the instance has a claimed deduction.

The density distribution of Hours reveals a multimodal distribution that is somewhat skewed in the positive direction. The distribution of Hours of adjusted instances is centered at around 39, whereas the distribution of Hours of instances not adjusted is centered at around 45. The ANOVA table for Hours by TARGET_Adjusted demonstrates an F-statistic of 89.02 with a p-value less than 2e-16, and clearly indicates a rejection of the null hypothesis of equal means for instances that were adjusted and not adjusted.

Exploring the correlations between quantitative variables for instances that resulted in adjustment indicates no correlation between any of the variables and RISK_Adjustment. There are, however, slight negative correlations between Income and Age, Hours and Age, and Hours and Income. The scatter plot matrix of these variables does not indicate any clear trends between variables.

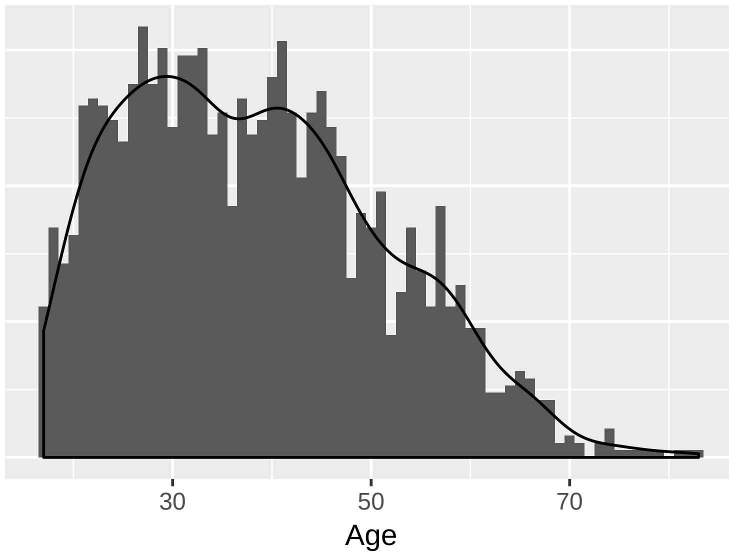
```r
Age = c(summary(df$Age), round(sd(df$Age), 2))
Income = c(summary(df$Income), round(sd(df$Income), 2))
Deductions = c(summary(df$Deductions), round(sd(df$Deductions), 2))
Hours = c(summary(df$Hours), round(sd(df$Hours), 2))
result = rbind(Age, Income, Deductions, Hours)
result = as.data.frame(result)
colnames(result)[7] = c("SD")
kable(result, caption = paste("Table 1:",
                              "Summary of numeric predictor variables",
                              "(Note: RISK_Adjustment description is based on",
                              "only the instances where the target was adjusted"))
```

Table 1: Table 1: Summary of numeric predictor variables (Note: RISK_Adjustment description is based on only the instances where the target was adjusted

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	SD
Age	17.0	28	37	38.31	47	83	13.01
Income	609.7	33980	59430	84320.00	113300	481300	69763.55
Deductions	0.0	0	0	68.91	0	2824	341.35
Hours	1.0	40	40	40.59	45	99	11.66

```
rm(list=c("Age", "Income", "Deductions", "Hours", "result"))
```

```r
# Explore the density distribution of Age
ggplot(df, aes(x=Age)) +
  geom_histogram(aes(y = ..density..), binwidth=1) +
  geom_density() + no.y
```



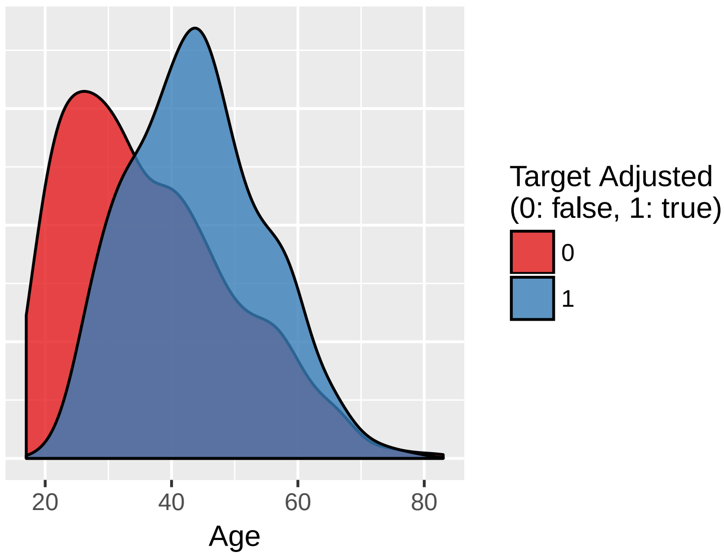

```r
shapiro.test(df$Age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$Age
## W = 0.96987, p-value < 2.2e-16
```

```r
skewness(df$Age)
```

```
## [1] 0.4686083
```

```r
# Conditional density plot of Age by TARGET_Adjusted
ggplot(df, aes(x=Age, fill=TARGET_Adjusted)) +
  geom_density(alpha = 0.8) +
  guides(fill=guide_legend(title="Target Adjusted\n(0: false, 1: true)")) +
  scale_fill_brewer(palette="Set1") + no.y
```




```r
# Analysis of differences in Age by TARGET_Adjusted
aggregate(Age~TARGET_Adjusted, data=df, mean)
```

```
##   TARGET_Adjusted      Age
## 1               0 36.53633
## 2               1 44.04251
```

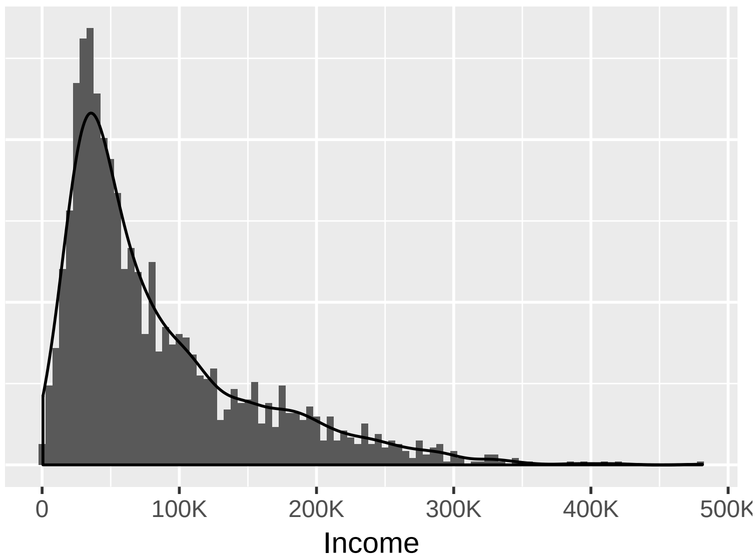
```r
aggregate(Age~TARGET_Adjusted, data=df, median)
```

```
##   TARGET_Adjusted Age
## 1               0  34
## 2               1  44
```

```r
summary(aov(Age~TARGET_Adjusted, data=df))
```

```
##                   Df Sum Sq Mean Sq F value Pr(>F)
## TARGET_Adjusted    1  19235   19235   120.9 <2e-16 ***
## Residuals       1890 300648     159
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Explore the density distribution of Income
ggplot(df, aes(x=Income)) +
  geom_histogram(aes(y = ..density..), binwidth=5000) +
  geom_density() + no.y +
  scale_x_continuous(labels=c("0", "100K", "200K", "300K", "400K", "500K"))
```




```r
shapiro.test(df$Income)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$Income
## W = 0.84764, p-value < 2.2e-16
```

```r
skewness(df$Income)
```

```
## [1] 1.506076
```

```r
# Conditional density plot of Income by TARGET_Adjusted
ggplot(df, aes(x=Income, fill=TARGET_Adjusted)) +
  geom_density(alpha = 0.8) +
  guides(fill=guide_legend(title="Target Adjusted\n(0: false, 1: true)")) +
  scale_fill_brewer(palette="Set1") + no.y +
  scale_x_continuous(labels=c("0", "100K", "200K", "300K", "400K", "500K"))
```


```r
# Analysis of differences in Income by TARGET_Adjusted
aggregate(Income~TARGET_Adjusted, data=df, mean)
```

```
##   TARGET_Adjusted   Income
## 1               0 91947.40
## 2               1 59662.88
```

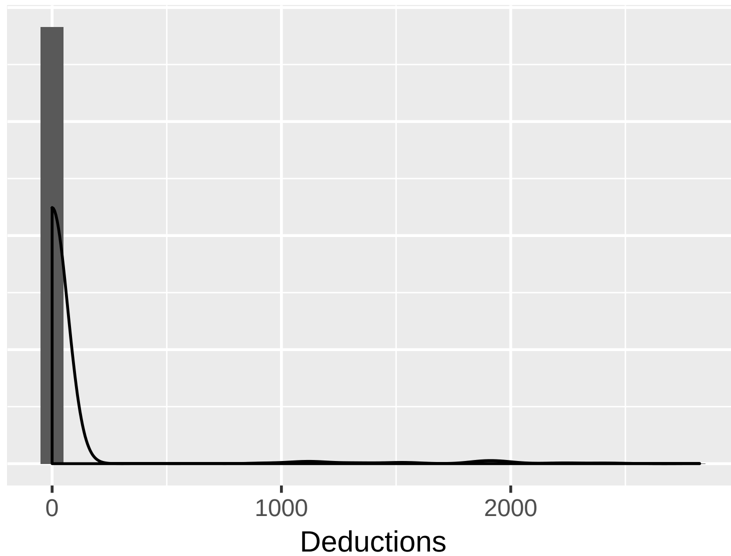
```r
aggregate(Income~TARGET_Adjusted, data=df, median)
```

```
##   TARGET_Adjusted   Income
## 1               0 70465.25
## 2               1 39979.20
```

```r
summary(aov(Income~TARGET_Adjusted, data=df))
```

```
##                   Df    Sum Sq   Mean Sq F value Pr(>F)
## TARGET_Adjusted    1 3.558e+11 3.558e+11   76.01 <2e-16 ***
## Residuals       1890 8.848e+12 4.681e+09
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Explore the density distribution of Deductions
ggplot(df, aes(x=Deductions)) +
  geom_histogram(aes(y = ..density..), binwidth=100) +
  geom_density() + no.y
```



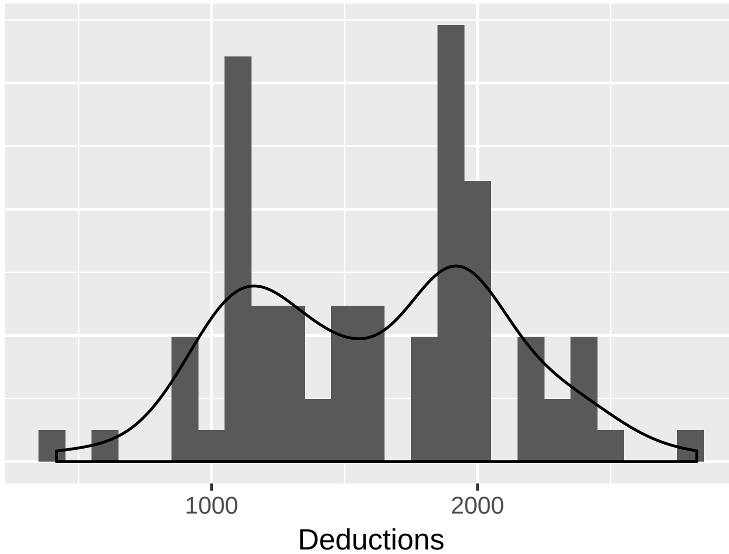

```r
shapiro.test(df$Deductions)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$Deductions
## W = 0.2025, p-value < 2.2e-16
```

```r
skewness(df$Deductions)
```

```
## [1] 5.127985
```

```r
## Density distribution where Deduction is not 0
ggplot(df[df$Deductions!=0,], aes(x=Deductions)) +
  geom_histogram(aes(y = ..density..), binwidth=100) +
  geom_density() + no.y
```



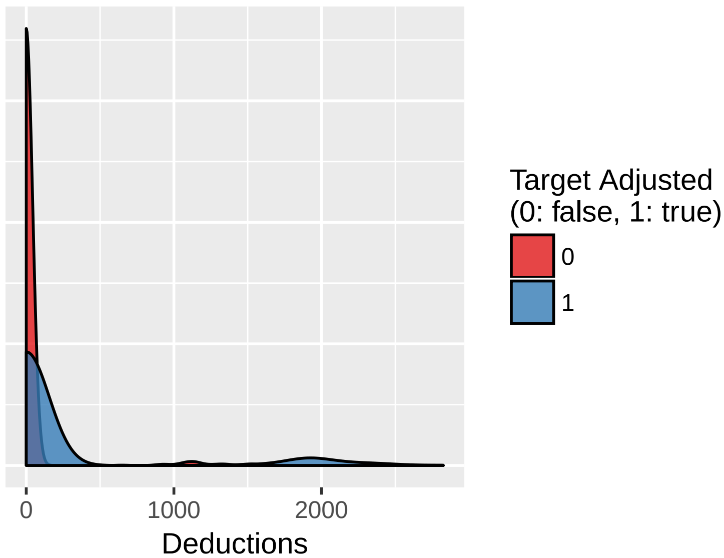

```r
shapiro.test(df$Deductions[df$Deductions!=0])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$Deductions[df$Deductions != 0]
## W = 0.96048, p-value = 0.01354
```

```r
skewness(df$Deductions[df$Deductions!=0])
```

```
## [1] 0.03286779
```

```r
# Conditional density plot of Deductions by TARGET_Adjusted
ggplot(df, aes(x=Deductions, fill=TARGET_Adjusted)) +
  geom_density(alpha = 0.8) +
  guides(fill=guide_legend(title="Target Adjusted\n(0: false, 1: true)")) +
  scale_fill_brewer(palette="Set1") + no.y
```




```r
# Conditional density plot of non-zero Deductions by TARGET_Adjusted
ggplot(df[df$Deductions!=0,], aes(x=Deductions, fill=TARGET_Adjusted)) +
  geom_density(alpha = 0.8) +
  guides(fill=guide_legend(title="Target Adjusted\n(0: false, 1: true)")) +
  scale_fill_brewer(palette="Set1") + no.y
```

```r
# Analysis of differences in Deductions by TARGET_Adjusted
aggregate(Deductions~TARGET_Adjusted, data=df, mean)
```

```
##   TARGET_Adjusted Deductions
## 1               0   33.36794
## 2               1  183.82103
```

```r
aggregate(Deductions~TARGET_Adjusted, data=df, median)
```

```
##   TARGET_Adjusted Deductions
## 1               0          0
## 2               1          0
```

```r
summary(aov(Deductions~TARGET_Adjusted, data=df))
```

```
##                   Df    Sum Sq Mean Sq F value Pr(>F)    
## TARGET_Adjusted    1   7727811 7727811    68.7 <2e-16 ***
## Residuals       1890 212612854  112494                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Analysis of differences in non-zero Deductions by TARGET_Adjusted
aggregate(Deductions~TARGET_Adjusted, data=df[df$Deductions!=0,], mean)
```

```
##   TARGET_Adjusted Deductions
## 1               0   1205.417
## 2               1   2004.098
```

```r
aggregate(Deductions~TARGET_Adjusted, data=df[df$Deductions!=0,], median)
```

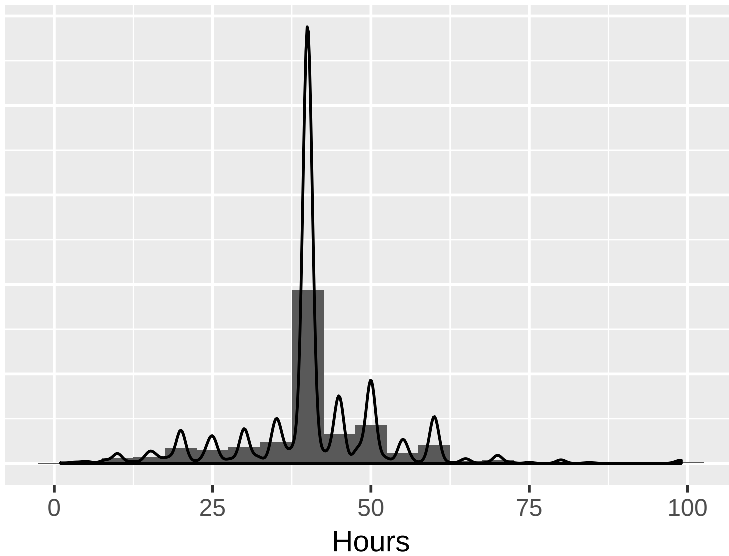
```
##   TARGET_Adjusted Deductions
## 1               0   1153.667
## 2               1   1902.000
```

```r
summary(aov(Deductions~TARGET_Adjusted, data=df[df$Deductions!=0,]))
```

```
##                 Df   Sum Sq  Mean Sq F value Pr(>F)    
## TARGET_Adjusted  1 12915327 12915327   156.2 <2e-16 ***
## Residuals       79  6532089    82685                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Create variable for whether deduction claimed
df$ClaimedDeduction = ifelse(df$Deductions==0, "NoDeduction", "Deduction")
df$ClaimedDeduction = factor(df$ClaimedDeduction, levels=c("NoDeduction", "Deduction"))
```

```r
# Explore the density distribution of Hours
ggplot(df, aes(x=Hours)) +
  geom_histogram(aes(y = ..density..), binwidth=5) +
  geom_density() + no.y
```



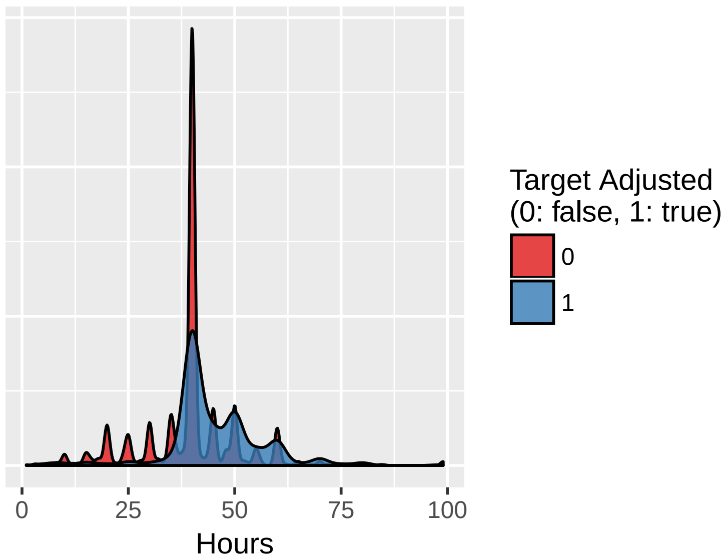

```r
shapiro.test(df$Hours)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$Hours
## W = 0.8882, p-value < 2.2e-16
```

```r
skewness(df$Hours)
```

```
## [1] 0.2805636
```

```r
# Conditional density plot of Income by TARGET_Adjusted
ggplot(df, aes(x=Hours, fill=TARGET_Adjusted)) +
  geom_density(alpha = 0.8) +
  guides(fill=guide_legend(title="Target Adjusted\n(0: false, 1: true)")) +
  scale_fill_brewer(palette="Set1") + no.y
```




```r
# Analysis of differences in Income by TARGET_Adjusted
aggregate(Hours~TARGET_Adjusted, data=df, mean)
```

```
##   TARGET_Adjusted    Hours
## 1               0 39.21522
## 2               1 45.03579
```

```r
aggregate(Hours~TARGET_Adjusted, data=df, median)
```

```
##   TARGET_Adjusted Hours
## 1               0    40
## 2               1    40
```

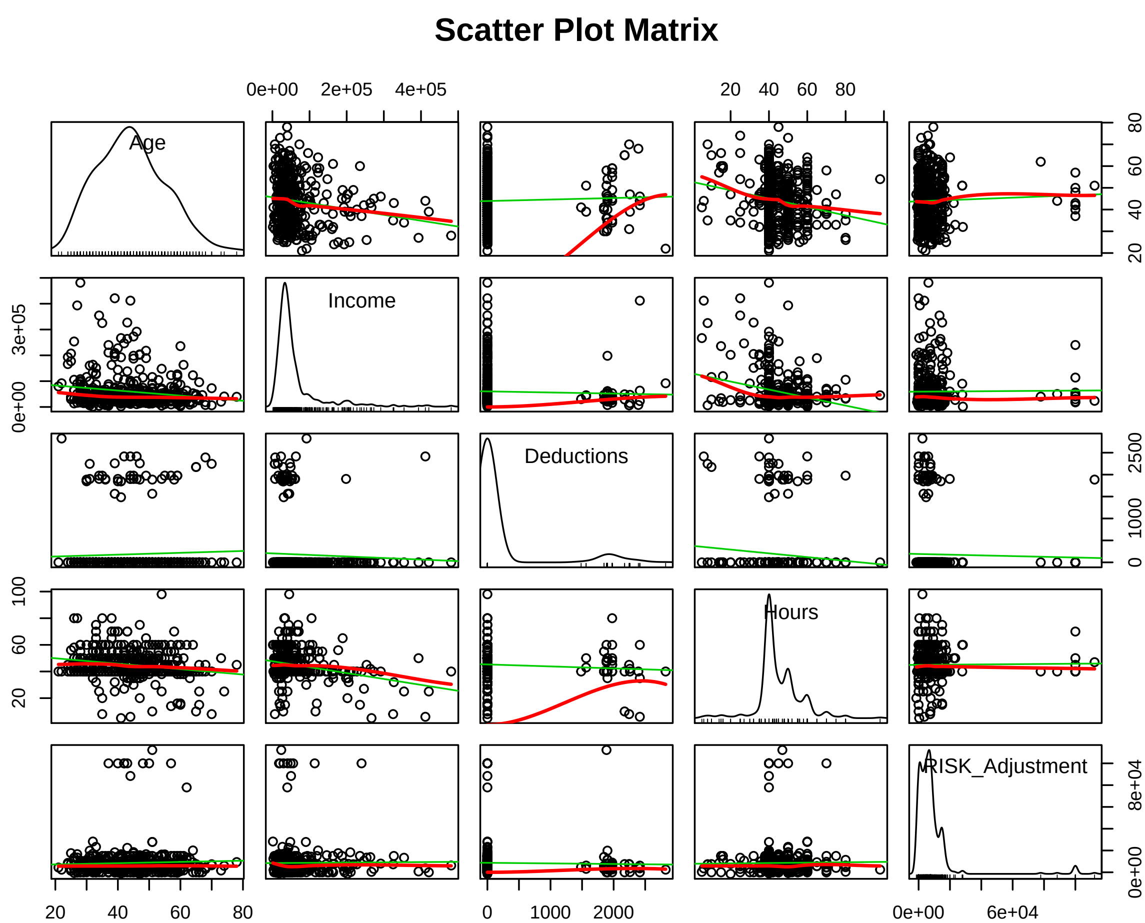
```r
summary(aov(Hours~TARGET_Adjusted, data=df))
```

```
##                   Df Sum Sq Mean Sq F value Pr(>F)
## TARGET_Adjusted    1  11566   11566   89.02 <2e-16 ***
## Residuals       1890 245549     130
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Correlations for cases where instance was adjusted
df.numeric = df[df$TARGET_Adjusted=="1", sapply(df, is.numeric)]
cor(df.numeric)
```

```
##                       Age       Income  Deductions       Hours
## Age             1.00000000 -0.163454388  0.03804272 -0.19805834
## Income         -0.16345439  1.000000000 -0.03878489 -0.26221274
## Deductions      0.03804272 -0.038784893  1.00000000 -0.07987956
## Hours          -0.19805834 -0.262212736 -0.07987956  1.00000000
## RISK_Adjustment  0.04001768  0.009529812 -0.02084898  0.01214487
##                RISK_Adjustment
## Age                0.040017677
## Income             0.009529812
## Deductions        -0.020848982
## Hours              0.012144871
## RISK_Adjustment    1.000000000
```

```r
suppressWarnings(
  scatterplotMatrix(df.numeric, spread=F, lty.smooth=2, main="Scatter Plot Matrix")
)
```



```r
rm(df.numeric)
```

Qualitative Predictor Variables

Several tables describing the counts and percentages of each qualitative predictor variable are included below. The percentage of each qualitative predictor is explored with respect to TARGET_Adjusted. The distribution of RISK_Adjustment is conditioned on each variable, and the means of the various categories are tested for significant differences.

The bar graph of TARGET_Adjusted faceted on Gender clearly indicates that a large proportion of males are adjusted, and relatively few females are adjusted. The distribution of RISK_Adjustment for adjusted instances is centered at around $10900 for females and at around $8200 for males (both with medians of approximately $5900). The ANOVA table for RISK_Adjustment by Gender demonstrates an F-statistic of 1.77 with a p-value of 0.184, and indicates a failure to reject the null hypothesis of equal means for adjusted instances of males and females.

The summary table describing the proportion of each Marital category that was targeted for adjustment suggests that it may be beneficial to simply recode Marital as a boolean variable describing whether the instance is married with the spouse present. 44.13% of married instances were targeted for adjustment, compared to only about 7.68% on average for all other cases. The bar graph of TARGET_Adjusted faceted on Marital clearly indicates that a large proportion of married instances are adjusted, and relatively few are adjusted in all other cases. The distribution of RISK_Adjustment for adjusted instances is centered at around $8670 for married instances and at around $8190 for all other cases (both with medians of about $5800). The ANOVA table for RISK_Adjustment by Marital demonstrates an F-statistic of 0.051 with a p-value of 0.822, and clearly indicates a failure to reject the null hypothesis of equal means for adjusted instances of married cases and all other cases.

The bar graph of TARGET_Adjusted faceted on Education clearly indicates that the proportion of instances that are adjusted increases with the level of education. The distributions of RISK_Adjustment for adjusted instances for the various Education categories are centered between approximately $7770 and $13275. The ANOVA table for RISK_Adjustment by Education demonstrates an F-statistic of 1.025 with a p-value of 0.394, and clearly indicates a failure to reject the null hypothesis of equal means for adjusted instances of the various Education categories.

The bar graph of TARGET_Adjusted faceted on Employment clearly indicates that a larger proportion of self-employed instances are adjusted compared to all other categories. Additionally, more instances are employed in the private sector than all other categories combined. The distributions of RISK_Adjustment for adjusted instances for the various Employment categories are centered between approximately $7060 and $16550. The ANOVA table for RISK_Adjustment by Employment demonstrates an F-statistic of 3.452 with a p-value of 0.00452, and indicates a potential rejection of the null hypothesis of equal means for adjusted instances of the various Employment categories (at the $p<0.01$ significance level). Pairwise t-testing indicates that it may be likely that there is a difference in means between instances of private sector employment and local public sector employment.

The bar graph of TARGET_Adjusted faceted on Occupation indicates that a very large proportion of executive, professional, and protective instances are adjusted, whereas low proportions of any other occupation are adjusted. The distributions of RISK_Adjustment for adjusted instances for the various Occupation categories are centered between approximately $2025 and $7250. The ANOVA table for RISK_Adjustment by Occupation demonstrates an F-statistic of 0.712 with a p-value of 0.727, and clearly indicates a failure to reject the null hypothesis of equal means for adjusted instances of the various Occupation categories.

The bar graph of TARGET_Adjusted faceted on ClaimedDeduction indicates that half of the deduction instances are adjusted, whereas a relatively low proportion of no deduction instances are adjusted. The distribution of RISK_Adjustment for adjusted instances is centered at around $7840 for instances with deductions and at around $8680 for instances with no deductions. The ANOVA table for RISK_Adjustment by Gender demonstrates an F-statistic of 0.114 with a p-value of 0.73, and clearly indicates a failure to reject the null hypothesis of equal means for adjusted instances of deductions and no deductions.

```r
# Gender summary table
tbl = melt(table(df[,c(11,7)]))
tbl = cbind(tbl,
            Percentage=ddply(tbl, .(Gender), summarize,
                             percentage=round(value/sum(value)*100,2))$percentage)
tbl = data.frame(tbl)
colnames(tbl)[3] = c("Count")
kable(tbl[,c(2,1,3,4)], caption="Table 2: Counts and Percentages of adjustments for Gender")
```

Table 2: Table 2: Counts and Percentages of adjustments for Gender

Gender	TARGET_Adjusted	Count	Percentage
Female	0	521	88.76
Female	1	66	11.24
Male	0	924	70.80
Male	1	381	29.20

```r
# Marital summary table
tbl = melt(table(df[,c(11,4)]))
tbl = cbind(tbl,
            Percentage=ddply(tbl, .(Marital), summarize,
                             percentage=round(value/sum(value)*100,2))$percentage)
tbl = data.frame(tbl)
colnames(tbl)[3] = c("Count")
kable(tbl[,c(2,1,3,4)], caption="Table 3: Counts and Percentages of adjustments for Marital")
```

Table 3: Table 3: Counts and Percentages of adjustments for Marital

Marital	TARGET_Adjusted	Count	Percentage
Absent	0	600	95.39
Absent	1	29	4.61
Divorced	0	235	92.16
Divorced	1	20	7.84
Married	0	490	55.87
Married	1	387	44.13
Married-spouse-absent	0	19	90.48
Married-spouse-absent	1	2	9.52
Unmarried	0	58	92.06
Unmarried	1	5	7.94
Widowed	0	43	91.49
Widowed	1	4	8.51

```r
# Education summary table
tbl = melt(table(df[,c(11,3)]))
tbl = cbind(tbl,
            Percentage=ddply(tbl, .(Education), summarize,
                             percentage=round(value/sum(value)*100,2))$percentage)
tbl = data.frame(tbl)
colnames(tbl)[3] = c("Count")
kable(tbl[,c(2,1,3,4)], caption="Table 4: Counts and Percentages of adjustments for Education")
```

Table 4: Table 4: Counts and Percentages of adjustments for Education

Education	TARGET_Adjusted	Count	Percentage
LessThanHS	0	211	95.48
LessThanHS	1	10	4.52
HSgrad	0	533	84.47
HSgrad	1	98	15.53
SomeCol/2Yr	0	451	80.11
SomeCol/2Yr	1	112	19.89
Bachelor	0	192	57.83
Bachelor	1	140	42.17
PostGraduate	0	58	40.00
PostGraduate	1	87	60.00

```
# Employment summary table
tbl = melt(table(df[,c(11,2)]))
tbl = cbind(tbl,
            Percentage=ddply(tbl, .(Employment), summarize,
                             percentage=round(value/sum(value)*100,2))$percentage)
tbl = data.frame(tbl)
colnames(tbl)[3] = c("Count")
kable(tbl[,c(2,1,3,4)], caption="Table 5: Counts and Percentages of adjustments for Employment")
```

Table 5: Table 5: Counts and Percentages of adjustments for Employment

Employment	TARGET_Adjusted	Count	Percentage
Consultant	0	108	72.97
Consultant	1	40	27.03
Private	0	1107	78.73
Private	1	299	21.27
PSFederal	0	49	72.06
PSFederal	1	19	27.94
PSLocal	0	88	73.95
PSLocal	1	31	26.05
PSState	0	49	68.06
PSState	1	23	31.94
SelfEmp	0	44	55.70
SelfEmp	1	35	44.30

```
# Occupation summary table
tbl = melt(table(df[,c(11,5)]))
tbl = cbind(tbl,
            Percentage=ddply(tbl, .(Occupation), summarize,
                             percentage=round(value/sum(value)*100,2))$percentage)
tbl = data.frame(tbl)
colnames(tbl)[3] = c("Count")
kable(tbl[,c(2,1,3,4)], caption="Table 6: Counts and Percentages of adjustments for Occupation")
```

Table 6: Table 6: Counts and Percentages of adjustments for Occupation

Occupation	TARGET_Adjusted	Count	Percentage
Cleaner	0	85	93.41
Cleaner	1	6	6.59
Clerical	0	198	85.34
Clerical	1	34	14.66
Executive	0	154	53.29
Executive	1	135	46.71
Farming	0	51	89.47
Farming	1	6	10.53
Machinist	0	121	87.05
Machinist	1	18	12.95
Professional	0	145	58.70
Professional	1	102	41.30
Protective	0	25	62.50
Protective	1	15	37.50
Repair	0	177	78.67
Repair	1	48	21.33
Sales	0	159	77.18
Sales	1	47	22.82
Service	0	203	96.67
Service	1	7	3.33
Support	0	35	71.43
Support	1	14	28.57
Transport	0	92	85.98
Transport	1	15	14.02

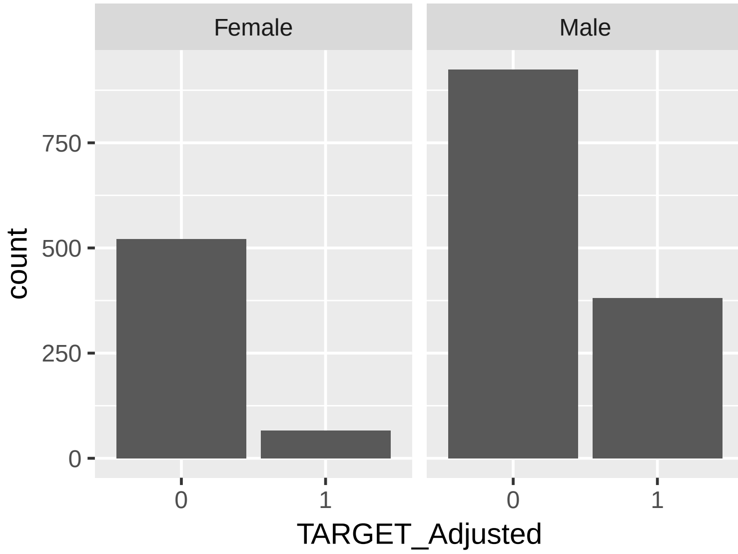
```r
# ClaimedDeduction summary table
tbl = melt(table(df[,c(11,12)]))
tbl = cbind(tbl,
            Percentage=ddply(tbl, .(ClaimedDeduction), summarize,
                             percentage=round(value/sum(value)*100,2))$percentage)
tbl = data.frame(tbl)
colnames(tbl)[3] = c("Count")
kable(tbl[,c(2,1,3,4)], caption="Table 7: Counts and Percentages of adjustments for ClaimedDeduction")
```

Table 7: Table 7: Counts and Percentages of adjustments for ClaimedDeduction

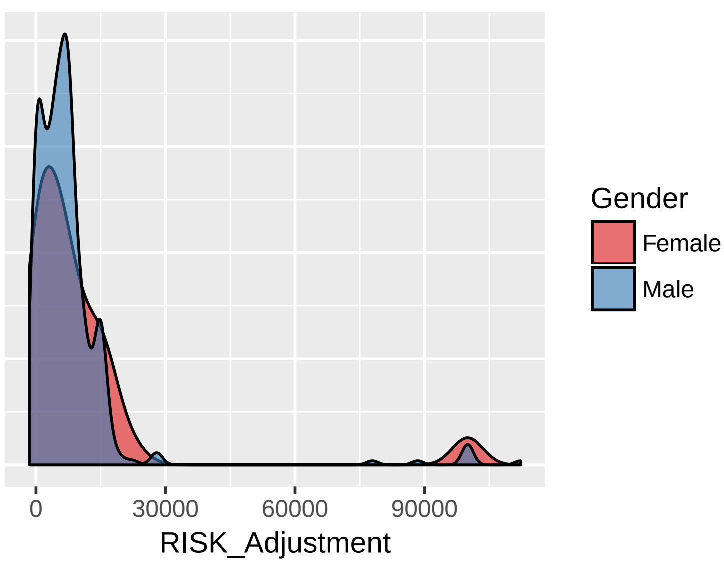
ClaimedDeduction	TARGET_Adjusted	Count	Percentage
NoDeduction	0	1405	77.58
NoDeduction	1	406	22.42
Deduction	0	40	49.38
Deduction	1	41	50.62

```r
rm(tbl)
```

```r
# TARGET_Adjusted counts faceted on Gender
ggplot(data=df, aes(x=TARGET_Adjusted)) + geom_bar() +
  facet_wrap(~Gender, nrow=1)
```




```r
# Conditional density plot of RISK_Adjustment by Gender
ggplot(df[df$TARGET_Adjusted=="1",], aes(x=RISK_Adjustment, fill=Gender)) +
  geom_density(alpha = 0.6) +
  scale_fill_brewer(palette="Set1") + no.y
```



```r
# Analysis of differences in RISK_Adjustment by Gender
aggregate(RISK_Adjustment~Gender, data=df[df$TARGET_Adjusted=="1",], mean)
```

```
##   Gender RISK_Adjustment
## 1 Female       10901.152
## 2   Male        8206.003
```

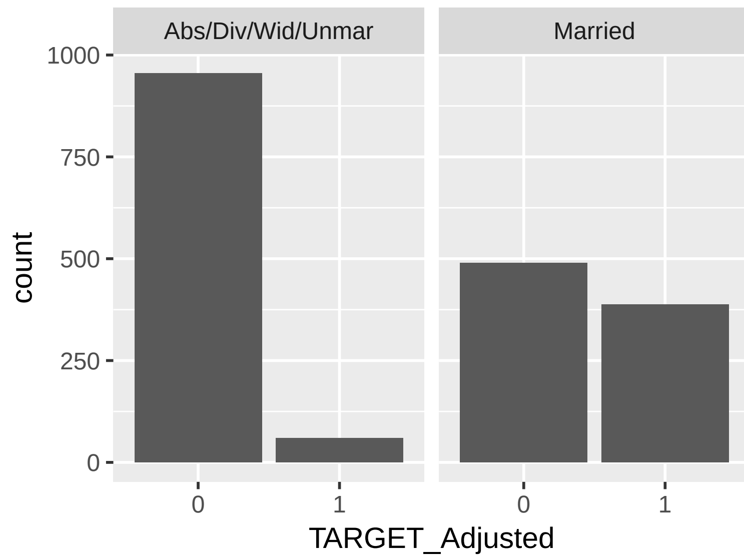
```r
aggregate(RISK_Adjustment~Gender, data=df[df$TARGET_Adjusted=="1",], median)
```

```
##   Gender RISK_Adjustment
## 1 Female            5932
## 2   Male            5848
```

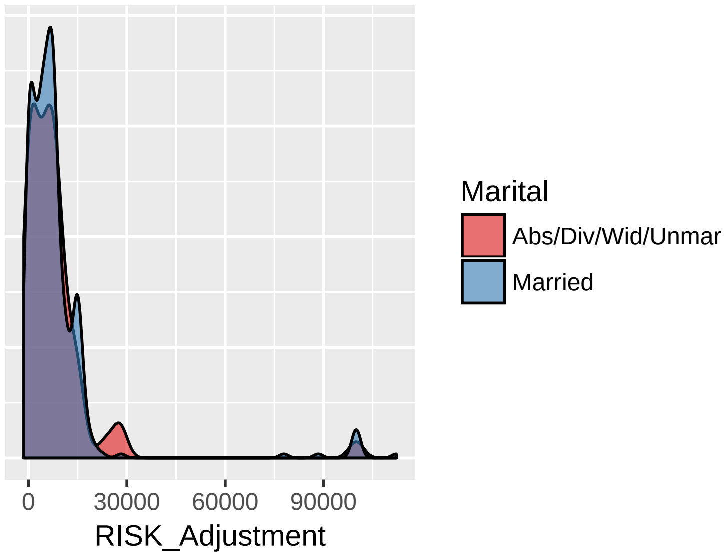
```r
summary(aov(RISK_Adjustment~Gender, data=df[df$TARGET_Adjusted=="1",]))
```

```
##              Df    Sum Sq   Mean Sq F value Pr(>F)
## Gender        1 4.086e+08 408626862    1.77  0.184
## Residuals   445 1.028e+11 230912091
```

```r
# Recode marital indicating whether married with a spouse present
df$Marital = as.character(df$Marital)
df$Marital <- ifelse(
  df$Marital=="Married",
  c("Married"),
  c("Abs/Div/Wid/Unmar"))
df$Marital = as.factor(df$Marital)
df$Marital = factor(df$Marital,
  levels = c("Abs/Div/Wid/Unmar", "Married")
  )

# TARGET_Adjusted counts faceted on Marital
ggplot(data=df, aes(x=TARGET_Adjusted)) + geom_bar() +
  facet_wrap(~Marital, nrow=1)
```



```r
# Conditional density plot of RISK_Adjustment by Marital
ggplot(df[df$TARGET_Adjusted=="1",], aes(x=RISK_Adjustment, fill=Marital)) +
  geom_density(alpha = 0.6) +
  scale_fill_brewer(palette="Set1") + no.y
```




```r
# Analysis of differences in RISK_Adjustment by Gender
aggregate(RISK_Adjustment~Marital, data=df[df$TARGET_Adjusted=="1",], mean)
```

```
##              Marital RISK_Adjustment
## 1 Abs/Div/Wid/Unmar        8191.983
## 2            Married        8667.814
```

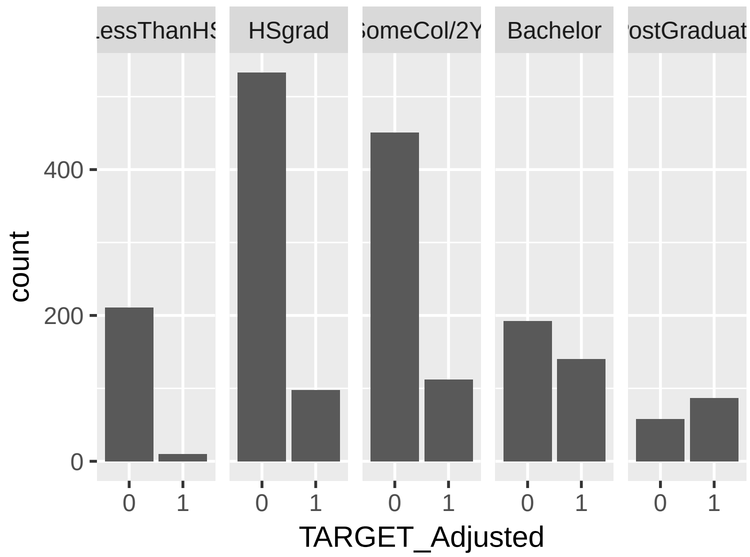
```r
aggregate(RISK_Adjustment~Marital, data=df[df$TARGET_Adjusted=="1",], median)
```

```
##              Marital RISK_Adjustment
## 1 Abs/Div/Wid/Unmar            5767
## 2            Married            5848
```

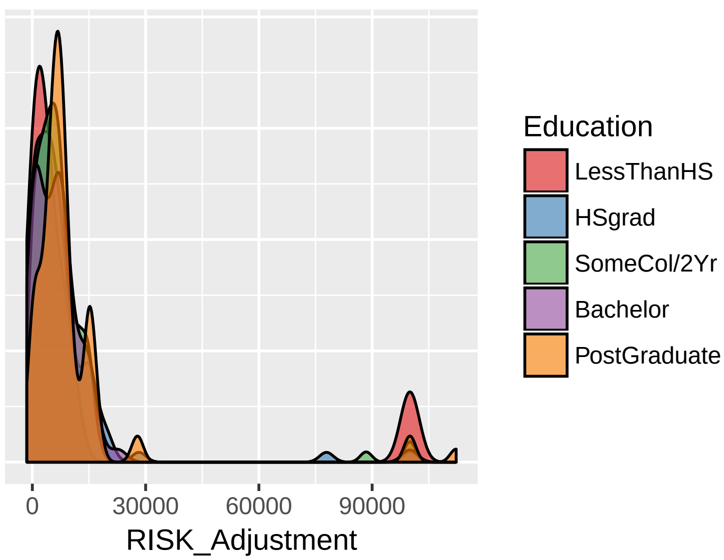
```r
summary(aov(RISK_Adjustment~Marital, data=df[df$TARGET_Adjusted=="1",]))
```

```
##              Df    Sum Sq   Mean Sq F value Pr(>F)
## Marital       1 1.176e+07  11761412   0.051  0.822
## Residuals   445 1.032e+11 231803923
```

```
# TARGET_Adjusted counts faceted on Education
ggplot(data=df, aes(x=TARGET_Adjusted)) + geom_bar() +
  facet_wrap(~Education, nrow=1)
```




```
# Conditional density plot of RISK_Adjustment by Education
ggplot(df[df$TARGET_Adjusted=="1",], aes(x=RISK_Adjustment, fill=Education)) +
  geom_density(alpha = 0.6) +
  scale_fill_brewer(palette="Set1") + no.y
```



```r
# Analysis of differences in RISK_Adjustment by Education
aggregate(RISK_Adjustment~Education, data=df[df$TARGET_Adjusted=="1",], mean)
```

```
##      Education RISK_Adjustment
## 1   LessThanHS       13273.800
## 2       HSgrad        7770.806
## 3  SomeCol/2Yr        8460.259
## 4     Bachelor        7496.157
## 5 PostGraduate       10973.276
```

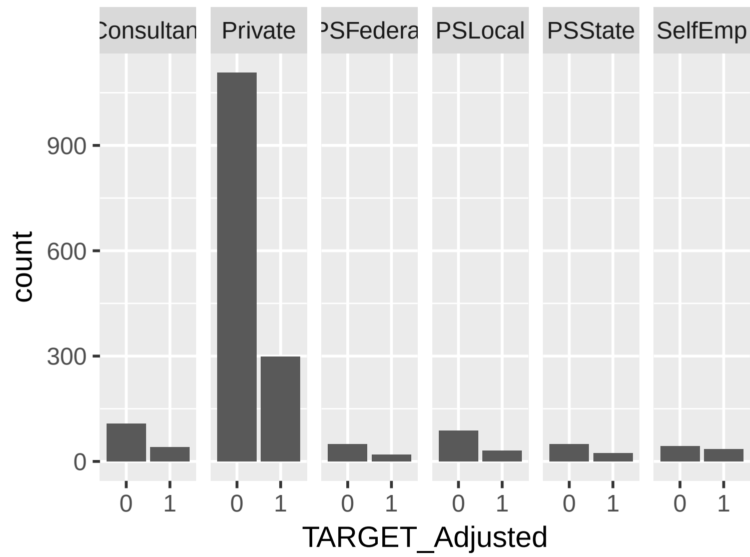
```r
aggregate(RISK_Adjustment~Education, data=df[df$TARGET_Adjusted=="1",], median)
```

```
##      Education RISK_Adjustment
## 1   LessThanHS          3204.0
## 2       HSgrad          4847.0
## 3  SomeCol/2Yr          5577.0
## 4     Bachelor          6047.5
## 5 PostGraduate          7168.0
```

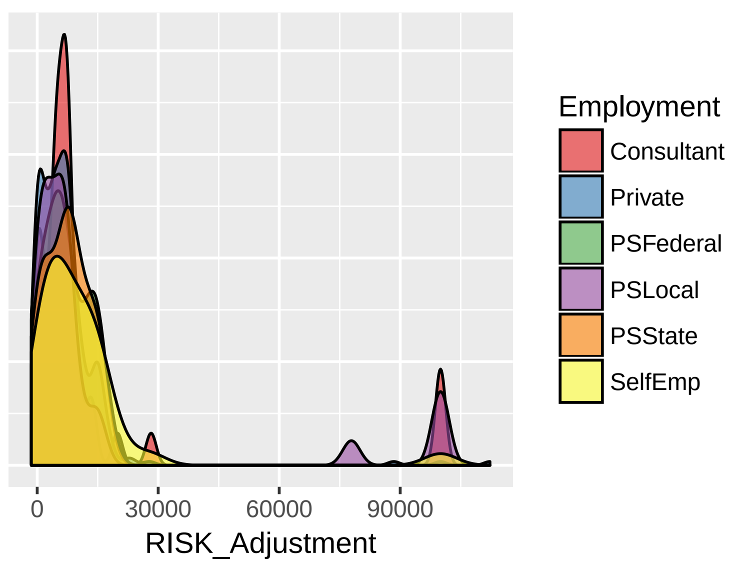
```r
summary(aov(RISK_Adjustment~Education, data=df[df$TARGET_Adjusted=="1",]))
```

```
##              Df    Sum Sq   Mean Sq F value Pr(>F)
## Education     4 9.486e+08 237153281   1.025  0.394
## Residuals   442 1.022e+11 231257679
```

```r
# TARGET_Adjusted counts faceted on Employment
ggplot(data=df, aes(x=TARGET_Adjusted)) + geom_bar() +
  facet_wrap(~Employment, nrow=1)
```




```r
# Conditional density plot of RISK_Adjustment by Employment
ggplot(df[df$TARGET_Adjusted=="1",], aes(x=RISK_Adjustment, fill=Employment)) +
  geom_density(alpha = 0.6) +
  scale_fill_brewer(palette="Set1") + no.y
```



```r
# Analysis of differences in RISK_Adjustment by Employment
aggregate(RISK_Adjustment~Employment, data=df[df$TARGET_Adjusted=="1",], mean)
```

```
##   Employment RISK_Adjustment
## 1 Consultant       13441.700
## 2    Private        7058.258
## 3  PSFederal        6997.000
## 4    PSLocal       16547.645
## 5    PSState        7335.609
## 6    SelfEmp       10949.629
```

```r
aggregate(RISK_Adjustment~Employment, data=df[df$TARGET_Adjusted=="1",], median)
```

```
##   Employment RISK_Adjustment
## 1 Consultant          6284.5
## 2    Private          5554.0
## 3  PSFederal          6545.0
## 4    PSLocal          6109.0
## 5    PSState          7671.0
## 6    SelfEmp          7978.0
```

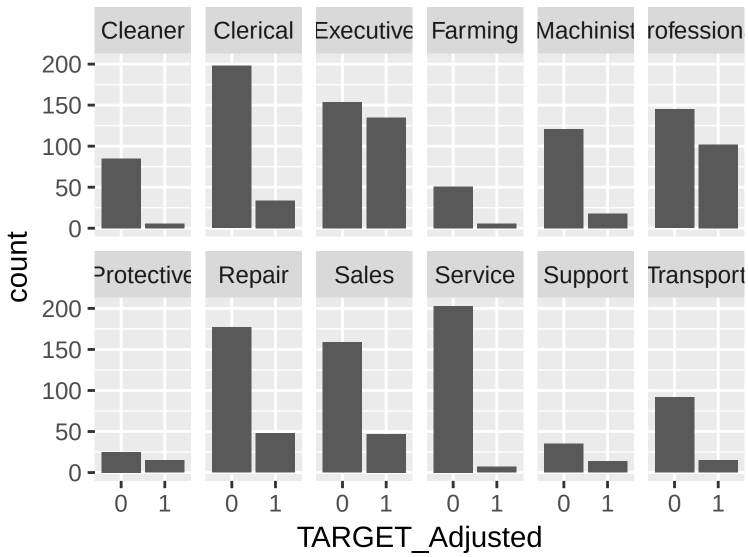
```r
summary(aov(RISK_Adjustment~Employment, data=df[df$TARGET_Adjusted=="1",]))
```

```
##              Df    Sum Sq   Mean Sq F value  Pr(>F)
## Employment    5 3.885e+09 777065020   3.452 0.00452 **
## Residuals   441 9.928e+10 225122862
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

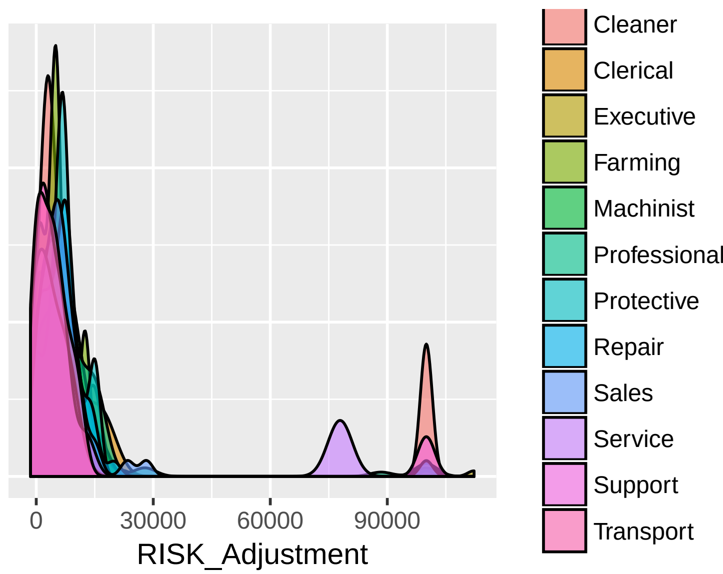
```r
pairwise.t.test(df$RISK_Adjustment[df$TARGET_Adjusted=="1"],
                df$Employment[df$TARGET_Adjusted=="1"])
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  df$RISK_Adjustment[df$TARGET_Adjusted == "1"] and df$Employment[df$TARGET_Adjusted == "1"]
##
##           Consultant Private PSFederal PSLocal PSState
## Private   0.166      -       -         -       -
## PSFederal 1.000      1.000   -         -       -
## PSLocal   1.000      0.013   0.353     -       -
## PSState   1.000      1.000   1.000     0.340   -
## SelfEmp   1.000      1.000   1.000     1.000   1.000
##
## P value adjustment method: holm
```

```r
# TARGET_Adjusted counts faceted on Occupation
ggplot(data=df, aes(x=TARGET_Adjusted)) + geom_bar() +
  facet_wrap(~Occupation, nrow=2)
```




```r
# Conditional density plot of RISK_Adjustment by Occupation
ggplot(df[df$TARGET_Adjusted=="1",], aes(x=RISK_Adjustment, fill=Occupation)) +
  geom_density(alpha = 0.6) + no.y
```



```r
# Analysis of differences in RISK_Adjustment by Occupation
aggregate(RISK_Adjustment~Occupation, data=df[df$TARGET_Adjusted=="1",], mean)
```

```
##      Occupation RISK_Adjustment
## 1       Cleaner       19435.167
## 2      Clerical        9304.676
## 3     Executive        7771.807
## 4       Farming        4863.833
## 5     Machinist        6159.778
## 6  Professional        9965.971
## 7    Protective        6888.867
## 8        Repair        8274.938
## 9         Sales        8499.745
## 10      Service       13841.429
## 11      Support       10771.357
## 12    Transport        3966.933
```

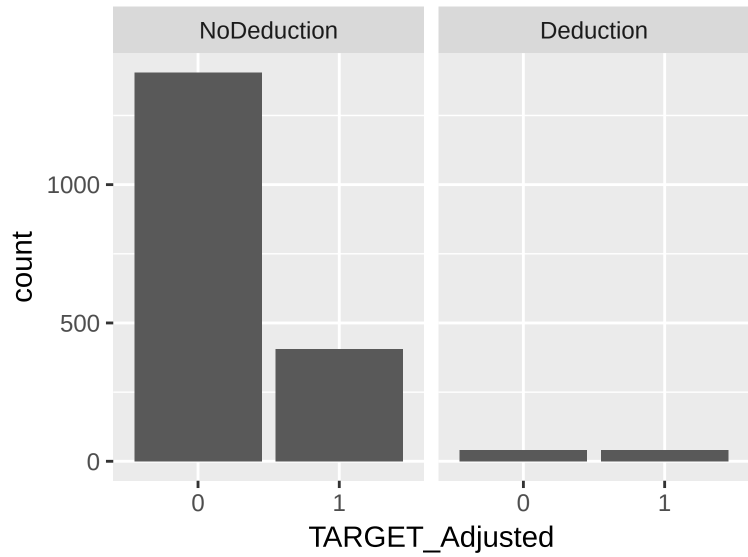
```r
aggregate(RISK_Adjustment~Occupation, data=df[df$TARGET_Adjusted=="1",], median)
```

```
##      Occupation RISK_Adjustment
## 1       Cleaner          3636.5
## 2      Clerical          5038.5
## 3     Executive          5672.0
## 4       Farming          4773.0
## 5     Machinist          5686.0
## 6  Professional          7258.0
## 7    Protective          6541.0
## 8        Repair          7298.0
## 9         Sales          5554.0
## 10      Service          2023.0
## 11      Support          3047.5
## 12    Transport          3760.0
```

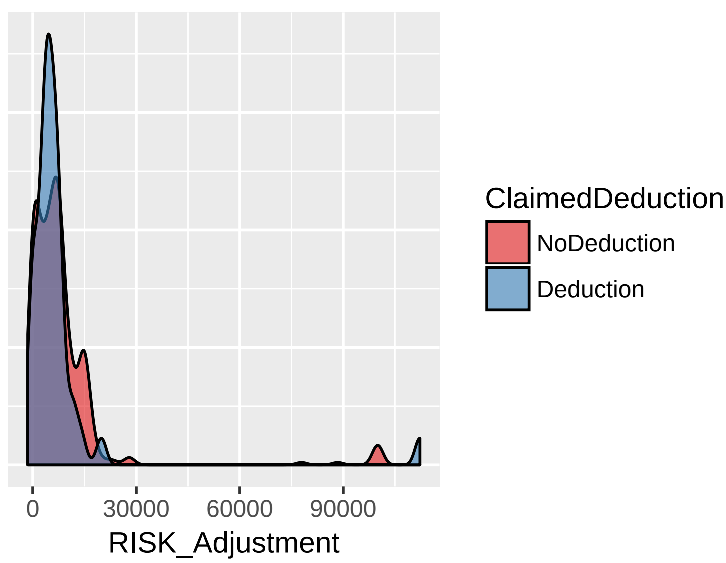
```r
summary(aov(RISK_Adjustment~Occupation, data=df[df$TARGET_Adjusted=="1",]))
```

```
##              Df    Sum Sq   Mean Sq F value Pr(>F)
## Occupation   11 1.825e+09 165899483   0.712  0.727
## Residuals   435 1.013e+11 232964628
```

```r
# TARGET_Adjusted counts faceted on Occupation
ggplot(data=df, aes(x=TARGET_Adjusted)) + geom_bar() +
  facet_wrap(~ClaimedDeduction, nrow=1)
```




```r
# Conditional density plot of RISK_Adjustment by Occupation
ggplot(df[df$TARGET_Adjusted=="1",], aes(x=RISK_Adjustment, fill=ClaimedDeduction)) +
  geom_density(alpha = 0.6) +
  scale_fill_brewer(palette="Set1") + no.y
```



```r
# Analysis of differences in RISK_Adjustment by Occupation
aggregate(RISK_Adjustment~ClaimedDeduction, data=df[df$TARGET_Adjusted=="1",], mean)
```

```
##   ClaimedDeduction RISK_Adjustment
## 1      NoDeduction        8681.264
## 2        Deduction        7838.293
```

```r
aggregate(RISK_Adjustment~ClaimedDeduction, data=df[df$TARGET_Adjusted=="1",], median)
```

```
##   ClaimedDeduction RISK_Adjustment
## 1      NoDeduction          5988.5
## 2        Deduction          4790.0
```

```r
summary(aov(RISK_Adjustment~ClaimedDeduction, data=df[df$TARGET_Adjusted=="1",]))
```

```
##                   Df    Sum Sq   Mean Sq F value Pr(>F)
## ClaimedDeduction   1 2.646e+07  26462294   0.114  0.736
## Residuals        445 1.031e+11 231770888
```

Logistic Regression Analysis of TARGET_Adjusted

Five different models are evaluated using various subsets of predictors. The model generated using all predictors (model A) produces the best performance in terms of F-score and AUC (0.603 and 0.885), but the model generated using only Age, Gender, Marital, Education, Occupation, and ClaimedDeduction (model C) produces comparable results (0.602 and 0.872) while offering a simpler model. Model C is retained as the best model for exploration of odds, and the lift curve and ROC curve are plotted for this model. Given the model summary, one can see that qualitative predictor variables with multiple levels are expanded into a set of binary variables. Among all predictors, Age, MaritalMarried, EducationSomeCol/2Yr, EducationBachelor, EducationPostGraduate, OccupationExecutive, and ClaimedDeductionDeduction are the most significant predictors.

The relationship between odds ratio and predictors can be discussed based on the estimated coefficients (which represent the log odds). The Intercept is the log odds in situation where all the properties are absent (i.e., Age = 0, GenderMale = 0, . . . , and ClaimedDeductionDeduction = 0). Its odds ratio is $e^{\beta_{Intercept}}$. Binary variables only have two values {0, 1}, so keeping other conditions constant, the change from 0 to 1 implies a multiplicative change in odds of success. For example, there are two persons who share the same attributes except that one is married (MaritalMarried = 1 with odds of success r) and the other one is unmarried (MaritalMarried = 0 with odds of success r'). We can claim that its odds ratio, on average, is $r/r' = e^{\beta_{MaritalMarried}}$. Quantitative predictor variables, such as Age, one unit increase would cause a change in the odds of success by $e^{\beta_{Age}}$. Therefore, its odds ratio is $e^{\beta_{Age}}$. The log odds and odds ratio for each predictor variable is included in the table below. The table indicates that post-graduate education and being married with a spouse present results in the largest change of the odds of success.

```r
# 10-fold cross validation of logistic regression model
# Note: Function assumes response to be a binary factor,
#       predictors required in the form to be  in the
#       form of "x1+x2", and if using a single predictor
#       it must be a factor with more than two levels
cv.logreg = function(formula, data) {
  ## extract response variable
  response = Reduce(paste, deparse(formula))
  response = strsplit(response, "~")[[1]][1]
  response = gsub(" ", "", response)
  ## create a matrix by expanding factors into a set of variables
  newdata = model.matrix(formula, data=data)[,-1]
  newdata = cbind(newdata, data[,response])
  colnames(newdata)[length(colnames(newdata))] = c(response)
  newdata = as.data.frame(newdata)
  newdata[,response] = as.factor(newdata[,response]-1) ## -1 to get 0/1 response
  ## split into folds
  n = length(data[,response])
  newdata = newdata[sample(n),] ## randomly shuffle rows
  folds = cut(seq(1:n), breaks=10, labels=FALSE) ## cut folds for cross val
  result = NULL
  formula = as.formula(paste(response,"~.", sep=""))

  for(i in 1:10){
    test = which(folds==i, arr.ind=TRUE) ## select indices for test data
    ## logistic regression
    model = glm(formula, family=binomial(link="logit"), data=newdata[-test,])
    ## predict using type="response" to return predicted probabilities
    prediction = predict(model,
                         newdata[test, -which(names(newdata)%in%c(response))],
                         type="response")
```

```
    temp = cbind(prediction, newdata[test, response])
    result = rbind(result, temp)
  }
  result[,2] = result[,2]-1 ## make the actuals 0/1
  return(result)
}

# Evaluates model performance in terms of accuracy, precision,
# recall, Fscore, and AUC. Input must be a data matrix where
# col 1 are predicted probabilities and col 2 are 0/1 observations.
evaluation = function(result, cutoff=0.5, conf.mat=FALSE) {
  yprobs = result[,1] ## extract predicted probabilities
  y = result[,2] ## extract ground truth results
  ## classified binary values
  ypreds = factor(floor(yprobs + (1-cutoff)), levels=c("0", "1"))
  confusion.matrix = table(y, ypreds) ## confusion matrix
  if(conf.mat) { print(confusion.matrix) }
  TP = confusion.matrix[2,2] ## if "1" is positive
  TN = confusion.matrix[1,1] ## if "0" is negative
  FP = confusion.matrix[1,2]
  FN = confusion.matrix[2,1]
  accuracy = (TP+TN)/length(y)
  precision = TP/(FP+TP)
  recall = TP/(FN+TP)
  Fscore = 2/(1/precision + 1/recall)
  ## calculate auc value
  suppressWarnings(require("pROC"))
  auc = auc(y,yprobs)
  eval = c(accuracy, precision, recall, Fscore, auc) ## combine all measures
  names(eval) = c("Acc", "Prec", "Rec", "FScr", "AUC")
  return(eval)
}

# Plots ROC and lift. Input must be a data matrix where
# col 1 are predicted probabilities and col 2 are 0/1 labels.
roc.and.lift = function(result) {
  result = as.data.frame(result)
  colnames(result) = c("yprobs", "y")
  n.test = dim(result)[1]

  par(mfrow=c(1,2)) ## set parameter to combine plots
  ## to plot lift curve
  rank.cb = as.data.frame(result[order(result$yprobs, decreasing = TRUE),]) ## rank probs
  colnames(rank.cb) = c('predicted','actual')
  base.rate = mean(result$y) ## baseline increase rate
  cat("baserate",base.rate,"\n")
  ax = dim(n.test) # x-axis
  ay.base = dim(n.test) # y-axis for baseline
  ay.pred = dim(n.test) # y-axis for predictions
  ax[1] = 1;
  ay.base[1] = base.rate
  ay.pred[1] = rank.cb$actual[1]
  for(i in 2:n.test) {
```

```r
    ax[i] = i;
    ay.base[i] = base.rate * i
    ay.pred[i] = ay.pred[i-1] + rank.cb$actual[i] # cumulative positive cases
  }
  plot(ax, ay.pred, xlab="Num. cases", ylab="Num. successes", main="Lift curve", type="l")
  points(ax, ay.base, type="l", col="red")

  ## to plot roc
  require("ROCR", quietly=TRUE)
  newdata = data.frame(predictions=result$yprobs, labels=result$y)
  result = prediction(newdata$predictions, newdata$labels)
  perf = performance(result,'sens','fpr')
  plot(perf, main="ROC curve")
  x = seq(0, 1, by=0.05)
  y = x
  points(x, y, type="l", col="red")
  par(mfrow=c(1,1)) ## reset parameter to default
}
```

```r
# Evaluation of five models
f.A = TARGET_Adjusted~Age+Gender+Marital+Education+Occupation+Employment+Hours+Income+ClaimedDeduction+
f.B = TARGET_Adjusted~Age+Gender+Marital+Education+Occupation+Income+ClaimedDeduction
f.C = TARGET_Adjusted~Age+Gender+Marital+Education+Occupation+ClaimedDeduction
f.D = TARGET_Adjusted~Age+Gender+Marital+Education+ClaimedDeduction
f.E = TARGET_Adjusted~Age+Gender+Marital+Education

set.seed(1337)
cv.A = cv.logreg(f.A, df)
cv.B = cv.logreg(f.B, df)
cv.C = cv.logreg(f.C, df)
cv.D = cv.logreg(f.D, df)
cv.E = cv.logreg(f.E, df)

eval.A = evaluation(cv.A)
eval.B = evaluation(cv.B)
eval.C = evaluation(cv.C)
eval.D = evaluation(cv.D)
eval.E = evaluation(cv.E)

evals = data.frame(eval.A, eval.B, eval.C, eval.D, eval.E)
colnames(evals) = c("Model A", "Model B", "Model C", "Model D", "Model E")
rownames(evals) = c("Accuracy", "Precision", "Recall", "F-score", "AUC")
kable(evals, caption = "Table 8: Evaluation of five models")
```

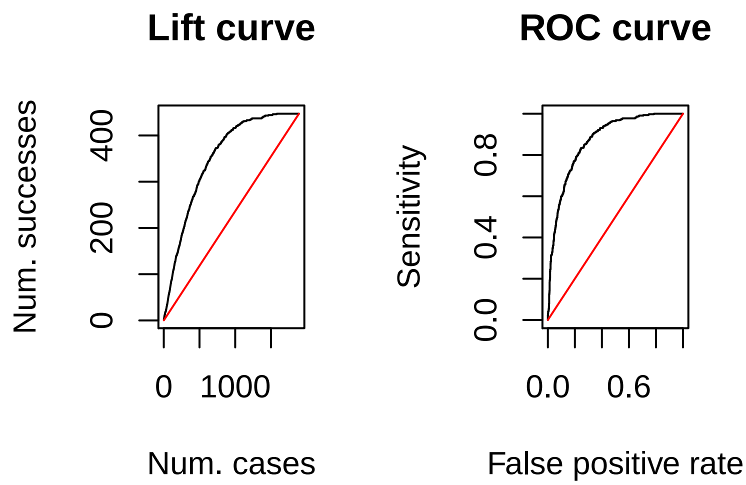
Table 8: Table 8: Evaluation of five models

	Model A	Model B	Model C	Model D	Model E
Accuracy	0.8324524	0.8303383	0.8303383	0.8224101	0.8165962
Precision	0.6846591	0.6750000	0.6750000	0.6784566	0.6644737
Recall	0.5391499	0.5436242	0.5436242	0.4720358	0.4519016
F-score	0.6032541	0.6022305	0.6022305	0.5567282	0.5379494
AUC	0.8849229	0.8725250	0.8717463	0.8665877	0.8612441

	Model A	Model B	Model C	Model D	Model E

```r
# Model C selected as best model; plot roc and lift
roc.and.lift(cv.C)
```

```
## baserate 0.2362579
```




```r
rm(list=c("f.A", "f.B", "f.C", "f.D", "f.E",
          "cv.A", "cv.B", "cv.C", "cv.D", "cv.E",
          "eval.A", "eval.B", "eval.C", "eval.D", "eval.E", "evals"))
```

```r
# Generate logistic regression model
log.model = glm(TARGET_Adjusted~Age+Gender+Marital+Education+Occupation+ClaimedDeduction,
                family=binomial("logit"), data=df)
summary(log.model)
```

```
##
## Call:
## glm(formula = TARGET_Adjusted ~ Age + Gender + Marital + Education +
##     Occupation + ClaimedDeduction, family = binomial("logit"),
##     data = df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.11743  -0.56037  -0.26066  -0.07208   2.77880
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -6.352959   0.641885  -9.897  < 2e-16 ***
## Age                        0.021447   0.005873   3.652 0.000260 ***
## GenderMale                 0.191083   0.200013   0.955 0.339399
## MaritalMarried             2.537355   0.184806  13.730  < 2e-16 ***
## EducationHSgrad            1.086605   0.365699   2.971 0.002965 **
## EducationSomeCol/2Yr       1.440223   0.371265   3.879 0.000105 ***
## EducationBachelor          2.215625   0.386734   5.729 1.01e-08 ***
## EducationPostGraduate      2.908020   0.440671   6.599 4.14e-11 ***
## OccupationClerical         1.155429   0.516757   2.236 0.025357 *
## OccupationExecutive        1.601974   0.484981   3.303 0.000956 ***
## OccupationFarming          0.119753   0.663280   0.181 0.856723
## OccupationMachinist        0.431574   0.532588   0.810 0.417749
## OccupationProfessional     1.292186   0.508066   2.543 0.010980 *
## OccupationProtective       1.619299   0.614720   2.634 0.008433 **
## OccupationRepair           0.613728   0.491318   1.249 0.211611
## OccupationSales            1.006450   0.502038   2.005 0.044992 *
## OccupationService         -0.553209   0.615293  -0.899 0.368601
## OccupationSupport          1.102263   0.598374   1.842 0.065461 .
## OccupationTransport        0.259258   0.547165   0.474 0.635628
## ClaimedDeductionDeduction  1.213254   0.312084   3.888 0.000101 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2068.8  on 1891  degrees of freedom
## Residual deviance: 1345.3  on 1872  degrees of freedom
## AIC: 1385.3
##
## Number of Fisher Scoring iterations: 6
```

```r
odds.tbl = as.data.frame(cbind(log.model$coefficients, exp(log.model$coefficients)))
colnames(odds.tbl) <- c("Log odds", "Odds ratio")
kable(odds.tbl, caption = "Table 9: Log odds and odds ratio for each predictor variable")
```

Table 9: Table 9: Log odds and odds ratio for each predictor variable

	Log odds	Odds ratio
(Intercept)	-6.3529591	0.0017416
Age	0.0214470	1.0216787
GenderMale	0.1910828	1.2105597
MaritalMarried	2.5373554	12.6461827
EducationHSgrad	1.0866048	2.9641929
EducationSomeCol/2Yr	1.4402233	4.2216385
EducationBachelor	2.2156253	9.1671394
EducationPostGraduate	2.9080204	18.3204955
OccupationClerical	1.1554286	3.1753842
OccupationExecutive	1.6019741	4.9628200
OccupationFarming	0.1197534	1.1272189
OccupationMachinist	0.4315737	1.5396787
OccupationProfessional	1.2921860	3.6407366
OccupationProtective	1.6192989	5.0495489
OccupationRepair	0.6137284	1.8473060
OccupationSales	1.0064497	2.7358705
OccupationService	-0.5532090	0.5751014
OccupationSupport	1.1022631	3.0109725
OccupationTransport	0.2592576	1.2959676
ClaimedDeductionDeduction	1.2132543	3.3644156

```r
rm(odds.tbl)
```

Linear Regression Analysis of RISK_Adjustment

The linear regression analysis of RISK_Adjustment aims to predict the adjustment amount for cases that are adjusted. Only records that resulted in an adjustment are used in this analysis. The linear regression model of RISK_Adjustment against all predictor variables results in a poor model that describes only 7.87% of the variation in the data (adjusted r-squared of 0.0194), and cross validation results in a root mean squared error of $16065.87. Inspecting the residual plots demonstrates several violations of the assumptions of linear regression. The scatterplot of residuals against fitted values demonstrates a negative linear trend that is asymmetrically distributed, indicating that the model does not meet the assumption of linearity. The normal probability plot of the standardized residuals indicates the non-normality of their distribution, and clearly shows a violation of the assumption of normality. The scatterplot of scale against location presents a cluster of points following a curved line, indicating the violation of the assumption of homoscedasticity. These violations indicate a need to apply some sort of transformation to the data.

The response variable, RISK_Adjustment, is shifted (+2000) and a log 10 transformation is applied, but the resulting model still does not satisfy the assumption of normality of the distribution of residuals, and the model only describes 5.13% of the variation in the data (RMSE = 0.3791). Table 10 describes the increase in root mean squared error for the exclusion of each predictor variable in the model, and indicates that the most important variables in predicting RISK_Adjustment are Occupation followed by Income, and excluding Age, Gender, Deductions, or ClaimedDeduction actually reduces errors (however, while also decreasing the r-squared value). A polynomial regression model was also generated using degree = 3 (selected using cross validation of in-sample vs out-of-sample predictions over varying degrees), but this model only describes 2.93% of the variation in the data (RMSE = 0.3833) and does not significantly improve performance.

Predicting the adjustment amount has demonstrated itself to be a more difficult problem. The linear regression model of RISK_Adjustment against all predictors for cases in which the target was adjusted resulted in a model which describes only 7.87% of the variation in the data (adjusted r-squared of 0.0194), and cross validation results in a root mean squared error of $16065.87. However, further inspection indicates that this model does not satisfy the assumptions of linear regression. A log-shift transformation was applied to the data, but this did not improve the performance of the model (R-squared = 0.0513, RMSE = 0.3791). Occupation and Income result in the largest increase in RMSE when excluded from the model, and are considered to be the most important variables in predicting RISK_Adjustment. Polynomial regression additionally did not improve performance (R-squared = 0.0293, RMSE = 0.3833).

```r
## function for 10-fold cross validation of regression performance based on RMSE
cv.linreg = function(formula, data) {
  ## extract response variable
  response = Reduce(paste, deparse(formula))
  response = strsplit(response, "~")[[1]][1]
  response = gsub(" ", "", response)
  ## create a matrix by expanding factors into a set of variables
  newdata = model.matrix(formula, data=data)[,-1]
  newdata = cbind(newdata, data[,response])
  colnames(newdata)[length(colnames(newdata))] = c(response)
  newdata = as.data.frame(newdata)
  ## split into folds
  n = length(data[,response])
  newdata = newdata[sample(n),] ## randomly shuffle rows
  folds = cut(seq(1:n), breaks=10, labels=FALSE) ## cut folds for cross val
  result = NULL
  formula = as.formula(paste(response,"~.", sep=""))

  for(i in 1:10) {
    test = which(folds==i, arr.ind=TRUE) # select indices for test data
    test.data = newdata[test,]
```

```r
    train.data = newdata[-test,]
    model = lm(formula, data=newdata[-test,])
    predicted = predict(model, newdata=newdata[test,])
    observation = newdata[test, response]
    temp = predicted - observation
    result = c(result, temp)
  }
  rmse = sqrt(mean(result^2))
  return(rmse)
}

# Function for comparing in-sample and out-of-sample error of
# polynomial regression over various degrees
cross.val.poly.reg =
  function(data, response, poly.var, lin.var, deg=12, train.set=0.5) {
    ## measure performance in terms of RMSE
    rmse = function(y, p) { return(sqrt(mean((y - p)^2))) }
    performance = data.frame()
    ## split data into a training set and test set for cross-validation
    n = length(data[,response])
    train = sort(sample(1:n, round(train.set*n)))
    formula = as.formula(paste(response,"~poly(",poly.var,", degree=d)+",lin.var,sep=""))

    for (d in 1:deg) {
      poly.fit = lm(formula, data=data[train,])
      performance = rbind(performance,
                          data.frame(Degree=d, Error="Training",
                                     RMSE = rmse(data[train,response],
                                                 predict(poly.fit))
                                     )
                          )
      performance = rbind(performance,
                          data.frame(Degree=d, Error="Cross-Validation",
                                     RMSE = rmse(data[-train,response],
                                                 predict(poly.fit, newdata=data[-train,]))
                                     )
                          )
      }

    ## Plot the performance of polynomial regression models for each degree
    require("ggplot2")
    require("scales")
    ggplot(performance , aes(x=Degree, y=RMSE, linetype=Error)) +
      geom_point() + geom_line() + scale_y_continuous(labels=comma)
}
```

```r
# Perform regression on only records that resulted in adjustments
adj.df = df[df$TARGET_Adjusted=="1",]

# Remove the TARGET_Adjusted column
adj.df = adj.df[,-which(names(df)%in%c("TARGET_Adjusted"))]

par(mfrow=c(2,2))
set.seed(1337)
# Explore linear regression of RISK_Adjustment against all predictors
model = lm(RISK_Adjustment~., adj.df)
summary(model)
```

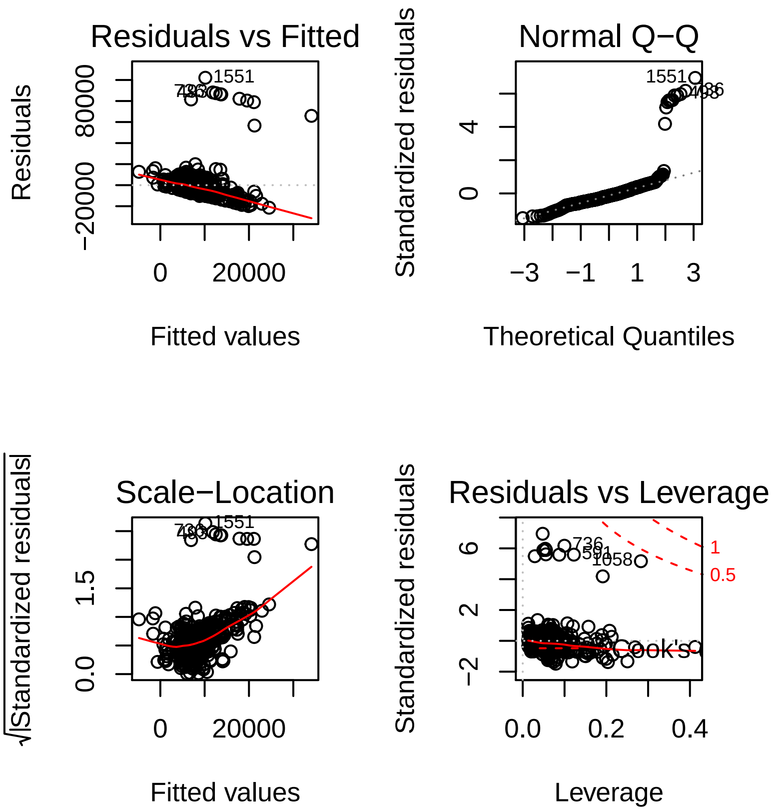
```
##
## Call:
## lm(formula = RISK_Adjustment ~ ., data = adj.df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -21444  -6254  -2280   2645 102092
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               3.145e+04  1.070e+04   2.939  0.00347 **
## Age                       8.733e+00  7.637e+01   0.114  0.90902
## EmploymentPrivate        -7.119e+03  2.644e+03  -2.692  0.00738 **
## EmploymentPSFederal      -7.912e+03  4.423e+03  -1.789  0.07437 .
## EmploymentPSLocal         3.942e+03  4.108e+03   0.960  0.33776
## EmploymentPSState        -7.793e+03  4.066e+03  -1.917  0.05598 .
## EmploymentSelfEmp        -3.353e+03  3.620e+03  -0.926  0.35490
## EducationHSgrad          -4.698e+03  5.165e+03  -0.910  0.36357
## EducationSomeCol/2Yr     -3.960e+03  5.206e+03  -0.761  0.44721
## EducationBachelor        -4.323e+03  5.323e+03  -0.812  0.41714
## EducationPostGraduate    -1.095e+03  5.503e+03  -0.199  0.84231
## MaritalMarried            1.539e+03  2.210e+03   0.696  0.48656
## OccupationClerical       -1.095e+04  6.998e+03  -1.565  0.11830
## OccupationExecutive      -1.299e+04  6.590e+03  -1.972  0.04930 *
## OccupationFarming        -1.918e+04  9.028e+03  -2.124  0.03425 *
## OccupationMachinist      -1.331e+04  7.390e+03  -1.801  0.07236 .
## OccupationProfessional   -1.194e+04  6.775e+03  -1.763  0.07868 .
## OccupationProtective     -2.053e+04  7.784e+03  -2.638  0.00866 **
## OccupationRepair         -1.194e+04  6.794e+03  -1.758  0.07950 .
## OccupationSales          -1.256e+04  6.894e+03  -1.822  0.06921 .
## OccupationService        -8.220e+03  8.572e+03  -0.959  0.33815
## OccupationSupport        -9.731e+03  7.633e+03  -1.275  0.20306
## OccupationTransport      -1.726e+04  7.471e+03  -2.311  0.02132 *
## Income                   -1.086e-02  1.650e-02  -0.658  0.51067
## GenderMale               -3.719e+03  3.017e+03  -1.233  0.21828
## Deductions               -1.129e+01  9.978e+00  -1.132  0.25843
## Hours                     2.114e+01  7.423e+01   0.285  0.77597
## ClaimedDeductionDeduction  2.222e+04  2.011e+04   1.105  0.26983
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15060 on 419 degrees of freedom
```

```
## Multiple R-squared:  0.07871,    Adjusted R-squared:  0.01935
## F-statistic: 1.326 on 27 and 419 DF,  p-value: 0.1299
```

```r
cv.linreg(RISK_Adjustment~., adj.df)
```

```
## [1] 16065.87
```

```r
plot(model)
```



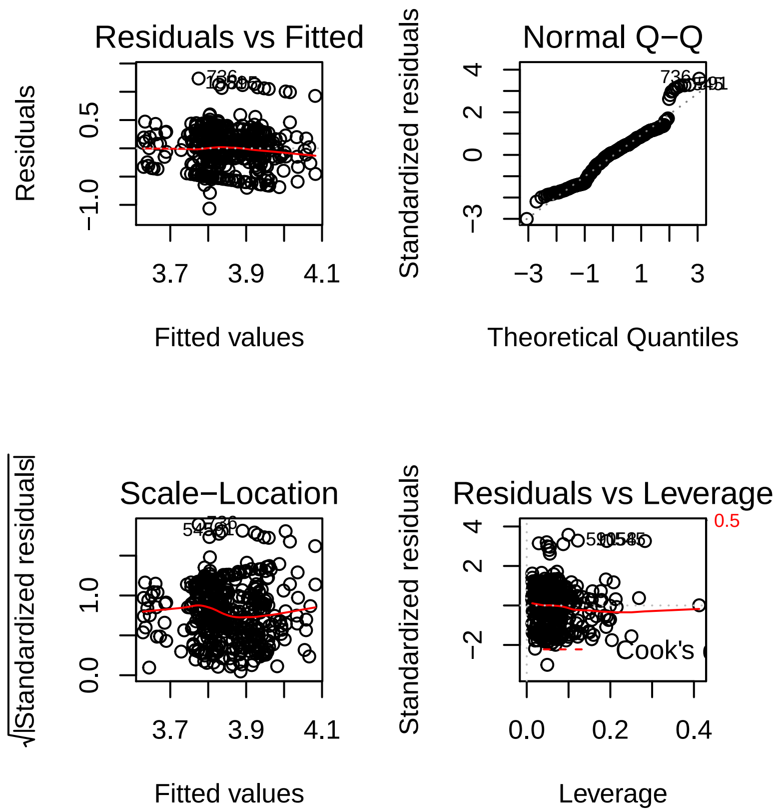
```r
# Apply log-shift transformation to response variable
adj.df$RISK_Adjustment = log10(adj.df$RISK_Adjustment + 2000)

par(mfrow=c(2,2))
set.seed(1337)
# Explore linear regression of RISK_Adjustment against all predictors after transformations
model = lm(RISK_Adjustment~., adj.df)
summary(model)
```

```
##
## Call:
## lm(formula = RISK_Adjustment ~ ., data = adj.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.06521 -0.24223  0.03561  0.21590  1.23394
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                3.967e+00  2.582e-01  15.361   <2e-16 ***
## Age                        3.471e-04  1.843e-03   0.188   0.8507
## EmploymentPrivate         -1.180e-01  6.381e-02  -1.849   0.0651 .
## EmploymentPSFederal       -1.016e-01  1.067e-01  -0.952   0.3416
## EmploymentPSLocal         -4.012e-02  9.913e-02  -0.405   0.6859
## EmploymentPSState         -9.841e-02  9.813e-02  -1.003   0.3165
## EmploymentSelfEmp         -1.384e-02  8.736e-02  -0.158   0.8742
## EducationHSgrad           -7.796e-03  1.247e-01  -0.063   0.9502
## EducationSomeCol/2Yr       3.934e-02  1.256e-01   0.313   0.7543
## EducationBachelor          1.577e-02  1.285e-01   0.123   0.9024
## EducationPostGraduate      1.366e-01  1.328e-01   1.029   0.3043
## MaritalMarried             4.269e-02  5.334e-02   0.800   0.4239
## OccupationClerical        -1.177e-01  1.689e-01  -0.697   0.4862
## OccupationExecutive       -1.514e-01  1.590e-01  -0.952   0.3418
## OccupationFarming         -2.730e-01  2.179e-01  -1.253   0.2108
## OccupationMachinist       -1.241e-01  1.783e-01  -0.696   0.4869
## OccupationProfessional    -1.373e-01  1.635e-01  -0.840   0.4017
## OccupationProtective      -1.412e-01  1.878e-01  -0.752   0.4528
## OccupationRepair          -1.041e-01  1.640e-01  -0.635   0.5260
## OccupationSales           -1.353e-01  1.664e-01  -0.813   0.4165
## OccupationService         -1.672e-01  2.069e-01  -0.808   0.4194
## OccupationSupport         -1.982e-01  1.842e-01  -1.076   0.2825
## OccupationTransport       -2.887e-01  1.803e-01  -1.601   0.1101
## Income                    -2.457e-08  3.982e-07  -0.062   0.9508
## GenderMale                -3.092e-02  7.280e-02  -0.425   0.6712
## Deductions                -2.863e-04  2.408e-04  -1.189   0.2352
## Hours                      1.295e-03  1.791e-03   0.723   0.4701
## ClaimedDeductionDeduction  5.385e-01  4.852e-01   1.110   0.2677
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3635 on 419 degrees of freedom
## Multiple R-squared:  0.05129,    Adjusted R-squared:  -0.009842
## F-statistic: 0.839 on 27 and 419 DF,  p-value: 0.7003
```

```r
rmse = cv.linreg(RISK_Adjustment~., adj.df)
rmse
```

```
## [1] 0.3791465
```

```r
plot(model)
```



```r
# Explore variable importance
result = NULL
## Age
predictor = cv.linreg(RISK_Adjustment~Employment+Education+Marital+Occupation+
                        Income+Gender+Deductions+Hours+ClaimedDeduction, adj.df)
predictor = rmse - predictor
result = rbind(result, Age=predictor)

## Employment
predictor = cv.linreg(RISK_Adjustment~Age+Education+Marital+Occupation+
                        Income+Gender+Deductions+Hours+ClaimedDeduction, adj.df)
predictor = rmse - predictor
result = rbind(result, Employment=predictor)

## Marital
predictor = cv.linreg(RISK_Adjustment~Age+Employment+Education+Occupation+
                        Income+Gender+Deductions+Hours+ClaimedDeduction, adj.df)
predictor = rmse - predictor
result = rbind(result, Marital=predictor)

## Occupation
predictor = cv.linreg(RISK_Adjustment~Age+Employment+Education+Marital+
                        Income+Gender+Deductions+Hours+ClaimedDeduction, adj.df)
predictor = rmse - predictor
result = rbind(result, Occupation=predictor)

## Income
predictor = cv.linreg(RISK_Adjustment~Age+Employment+Education+Marital+Occupation+
                        Gender+Deductions+Hours+ClaimedDeduction, adj.df)
predictor = rmse - predictor
result = rbind(result, Income=predictor)

## Gender
predictor = cv.linreg(RISK_Adjustment~Age+Employment+Education+Marital+Occupation+
                        Income+Deductions+Hours+ClaimedDeduction, adj.df)
predictor = rmse - predictor
result = rbind(result, Gender=predictor)

## Deductions
predictor = cv.linreg(RISK_Adjustment~Age+Employment+Education+Marital+Occupation+
                        Income+Gender+Hours+ClaimedDeduction, adj.df)
predictor = rmse - predictor
result = rbind(result, Deductions=predictor)

## Hours
predictor = cv.linreg(RISK_Adjustment~Age+Employment+Education+Marital+Occupation+
                        Income+Gender+Deductions+ClaimedDeduction, adj.df)
predictor = rmse - predictor
result = rbind(result, Hours=predictor)

## ClaimedDeduction
predictor = cv.linreg(RISK_Adjustment~Age+Employment+Education+Marital+Occupation+
                        Income+Gender+Deductions+Hours, adj.df)
```

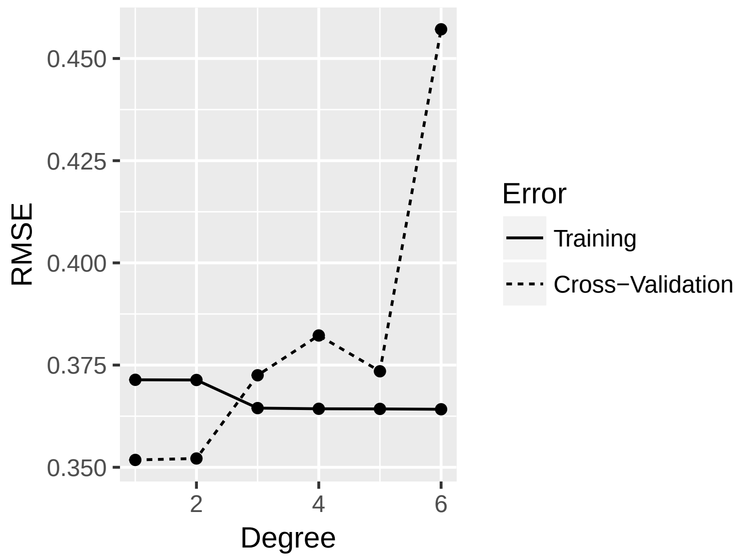
```r
predictor = rmse - predictor
result = rbind(result, ClaimedDeduction=predictor)

result = as.data.frame(result)
colnames(result)[1] = c("RMSE Increase")
kable(result, caption = "Table 10: Increase in RMSE given exclusion of predictor")
```

Table 10: Table 10: Increase in RMSE given exclusion of predictor

	RMSE Increase
Age	-0.0049392
Employment	0.0030656
Marital	0.0004087
Occupation	0.0055869
Income	0.0043646
Gender	-0.0004521
Deductions	-0.0040813
Hours	0.0030744
ClaimedDeduction	-0.0030624

```r
rm(list=c("rmse", "predictor", "result"))
```

```r
set.seed(1337)
# Cross validation of in-sample and out-of-sample performance over varying degrees of income
cross.val.poly.reg(adj.df, "RISK_Adjustment", "Income",
                   "Education+Marital+Hours+ClaimedDeduction", deg=6)
```




```r
model = lm(RISK_Adjustment~poly(Income, degree=3)+Education+Marital+Hours+ClaimedDeduction,
           adj.df)
summary(model)
```

```
##
## Call:
## lm(formula = RISK_Adjustment ~ poly(Income, degree = 3) + Education +
##     Marital + Hours + ClaimedDeduction, data = adj.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.08846 -0.21897  0.02584  0.21246  1.21402
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                3.697200   0.144137  25.651   <2e-16 ***
## poly(Income, degree = 3)1  0.129236   0.380128   0.340    0.734
## poly(Income, degree = 3)2 -0.034682   0.383214  -0.091    0.928
## poly(Income, degree = 3)3 -0.602614   0.375113  -1.606    0.109
## EducationHSgrad            0.015809   0.120478   0.131    0.896
## EducationSomeCol/2Yr       0.054274   0.119363   0.455    0.650
## EducationBachelor          0.019613   0.118882   0.165    0.869
## EducationPostGraduate      0.147224   0.121598   1.211    0.227
## MaritalMarried             0.017284   0.054964   0.314    0.753
## Hours                      0.001987   0.001660   1.197    0.232
## ClaimedDeductionDeduction -0.033442   0.059737  -0.560    0.576
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3604 on 436 degrees of freedom
```

```
## Multiple R-squared:  0.0293, Adjusted R-squared:  0.007038
## F-statistic: 1.316 on 10 and 436 DF,  p-value: 0.219
```

```r
rmse = cv.linreg(RISK_Adjustment~., adj.df)
rmse
```

```
## [1] 0.3832941
```

Results

The exploratory analysis indicates that there are significant differences in Age, Income, Deductions, and Hours between cases that were adjusted and not adjusted. However, there were no correlations between any of the quantitative predictors and the adjustment amount. Exploration of the qualitative predictor variables indicated that cases where the individual is male, married with a spouse present, has a post-graduate education, has a particular occupation (executive, professional, protective), or claimed a deduction were more likely to have been adjusted. However, there were no significant differences in the adjustment amounts between the categories of any of the qualitative predictors.

Cross validation of the logistic regression model of TARGET_Adjusted against all predictors demonstrates an F-score of 0.603 and an AUC of 0.885. The logistic regression model including only Age, Gender, Marital, Education, Occupation, and ClaimedDeduction as predictors demonstrates comparable performance with an F-score of 0.602 and an AUC of 0.872. Observation of the odds ratio of each variable demonstrates that having a post-graduate education or being married with a spouse present results in the largest change of the odds of success.