Aido Data

DFoly

June 30, 2016

We want to predict whether or not there will be fraud based on the variables in the Data.

Initial Data Exploration

```
Data <- read.csv("C:/Users/dfoley/Dropbox/Aido/Data_Presentation/Fraud_data.csv")

summary(Data)
```

```
##       FRAUD             PPSN      SCHEME_CODE LOCATION_CODE
##  Min.   :0.000   0001135J:   1   F1:1113     D1     :  28
##  1st Qu.:0.000   0003334Z:   1   G1:  99     S1     :  26
##  Median :1.000   0006622M:   1   H1: 494     A3     :  24
##  Mean   :0.509   0009177V:   1   H2: 294     G4     :  24
##  3rd Qu.:1.000   0024727R:   1               V3     :  23
##  Max.   :1.000   0028683D:   1               E3     :  22
##                  (Other) :1994               (Other):1853
##         PAY_TYPE           REG_DATE          START_DATE        BIRTH_DATE
##  Basic      :1102   07/09/2008:   4   10/04/2001:   4   02/02/1995:   3
##  Exceptional: 253   15/07/2009:   4   12/05/2012:   4   12/04/1977:   3
##  Supplement : 472   23/04/2012:   4   22/03/2011:   4   15/01/1986:   3
##  Urgent     : 173   01/06/2007:   3   22/05/2012:   4   24/06/1973:   3
##                     02/02/2008:   3   01/11/2012:   3   25/03/1983:   3
##                     02/03/2012:   3   02/05/2009:   3   28/06/1982:   3
##                     (Other)   :1979   (Other)   :1978   (Other)   :1982
##    AGE_YEARS     OCC_CODE   MEANS_FROM_EMP      DELAY
##  Min.   :18.00   M   :779   Min.   :  100   Min.   : 0.00
##  1st Qu.:29.00   P   : 98   1st Qu.:10700   1st Qu.:14.00
##  Median :37.00   S   :435   Median :21150   Median :21.00
##  Mean   :38.67   SS  :687   Mean   :21595   Mean   :21.86
##  3rd Qu.:47.00   NA's:  1   3rd Qu.:31400   3rd Qu.:29.00
##  Max.   :79.00              Max.   :57800   Max.   :60.00
##
##   WEEKLY_RATE     PAID_TO_DATE        SEX       MARITAL_STATUS
##  Min.   : 50.0   Min.   :  4160   Female: 974   D:666
##  1st Qu.:120.0   1st Qu.: 29160   Male  :1026   M:446
##  Median :160.0   Median : 49390                 S:888
##  Mean   :154.3   Mean   : 59593
##  3rd Qu.:190.0   3rd Qu.: 81600
##  Max.   :220.0   Max.   :220290
##  NA's   :1       NA's   :1
##      NO_DEP        FUEL_AMOUNT      APPOINTMENT_STATUS  BOOK_NUMBER
##  Min.   : 0.00   Min.   : 40.00   Due       :746      Min.   :10017
##  1st Qu.: 2.00   1st Qu.: 70.00   Missed    :391      1st Qu.:19951
##  Median : 3.00   Median : 80.00   Up-to-Date:863      Median :29610
```

```
##  Mean   : 3.36   Mean   : 78.06                      Mean   :29721  
##  3rd Qu.: 4.00   3rd Qu.: 90.00                      3rd Qu.:39568  
##  Max.   :10.00   Max.   :140.00                      Max.   :49995  
##                                                                     
##     CARER_ID     PHOTO_VERIFIED  OPEN_CREDIT_LINES   DRIVING_STATUS
##  Min.   :    0   Min.   :0.000   Min.   : 0.000    Active   :1402  
##  1st Qu.:    0   1st Qu.:0.000   1st Qu.: 4.000    None     : 518  
##  Median :    0   Median :1.000   Median : 5.000    Suspended:  80  
##  Mean   : 1717   Mean   :0.697   Mean   : 5.082                    
##  3rd Qu.:    0   3rd Qu.:1.000   3rd Qu.: 6.000                    
##  Max.   :49966   Max.   :1.000   Max.   :11.000                    
##                                                                    
##   FREE_TRAVEL     EMPLOYER_NO                    IBAN      MED_ASSESSMENT
##  Min.   :0.000   Min.   :   18833   AAAA080879195849:   1   Min.   :1.00  
##  1st Qu.:0.000   1st Qu.:25438583   AAAA294430715538:   1   1st Qu.:1.00  
##  Median :0.000   Median :50508358   AAAA504389685081:   1   Median :2.00  
##  Mean   :0.219   Mean   :50085185   AAAB021425204220:   1   Mean   :2.31  
##  3rd Qu.:0.000   3rd Qu.:74127544   AAAB109040882998:   1   3rd Qu.:3.00  
##  Max.   :1.000   Max.   :99989585   AAAB190156605237:   1   Max.   :4.00  
##                                     (Other)         :1994                 
##  MED_CONDITION_SATISFIED MORT_OUTSTANDING PAY_LOCATION_CODE   PREF_LANG   
##  Min.   :0.0000          Min.   :      0   D1     :  28      Min.   :1.00  
##  1st Qu.:0.0000          1st Qu.:      0   V3     :  26      1st Qu.:1.00  
##  Median :0.0000          Median :      0   S1     :  25      Median :1.00  
##  Mean   :0.1605          Mean   : 58189   A3     :  23      Mean   :1.03  
##  3rd Qu.:0.0000          3rd Qu.:137688   E3     :  23      3rd Qu.:1.00  
##  Max.   :1.0000          Max.   :256947   Q2     :  23      Max.   :2.00  
##                                            (Other):1852                   
##     PHONE_NO        PRIORITY_CLAIM 
##  Min.   :3.54e+11   Min.   :0.000  
##  1st Qu.:3.54e+11   1st Qu.:0.000  
##  Median :3.54e+11   Median :0.000  
##  Mean   :3.54e+11   Mean   :0.096  
##  3rd Qu.:3.54e+11   3rd Qu.:0.000  
##  Max.   :3.54e+11   Max.   :1.000  
## 
```

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.1.3
```

Lets start off with an inital exploration of the data to see what variables may impact fraud

```r
cor(Data[,unlist(lapply(Data, is.numeric))], Data$FRAUD)
```

```
## Warning in cor(Data[, unlist(lapply(Data, is.numeric))], Data$FRAUD): the
## standard deviation is zero
```

```
##                                [,1]
## FRAUD                   1.000000000
## AGE_YEARS              -0.022102019
```

```
## MEANS_FROM_EMP          -0.691814409
## DELAY                    0.557881319
## WEEKLY_RATE                       NA
## PAID_TO_DATE                      NA
## NO_DEP                   0.398714291
## FUEL_AMOUNT              0.213654006
## BOOK_NUMBER              0.035143006
## CARER_ID                -0.018820769
## PHOTO_VERIFIED           0.003164441
## OPEN_CREDIT_LINES        0.012542585
## FREE_TRAVEL             -0.011951586
## EMPLOYER_NO             -0.023686280
## MED_ASSESSMENT           0.149375417
## MED_CONDITION_SATISFIED -0.259908686
## MORT_OUTSTANDING        -0.001932028
## PREF_LANG               -0.006094250
## PHONE_NO                          NA
## PRIORITY_CLAIM          -0.053397883
```

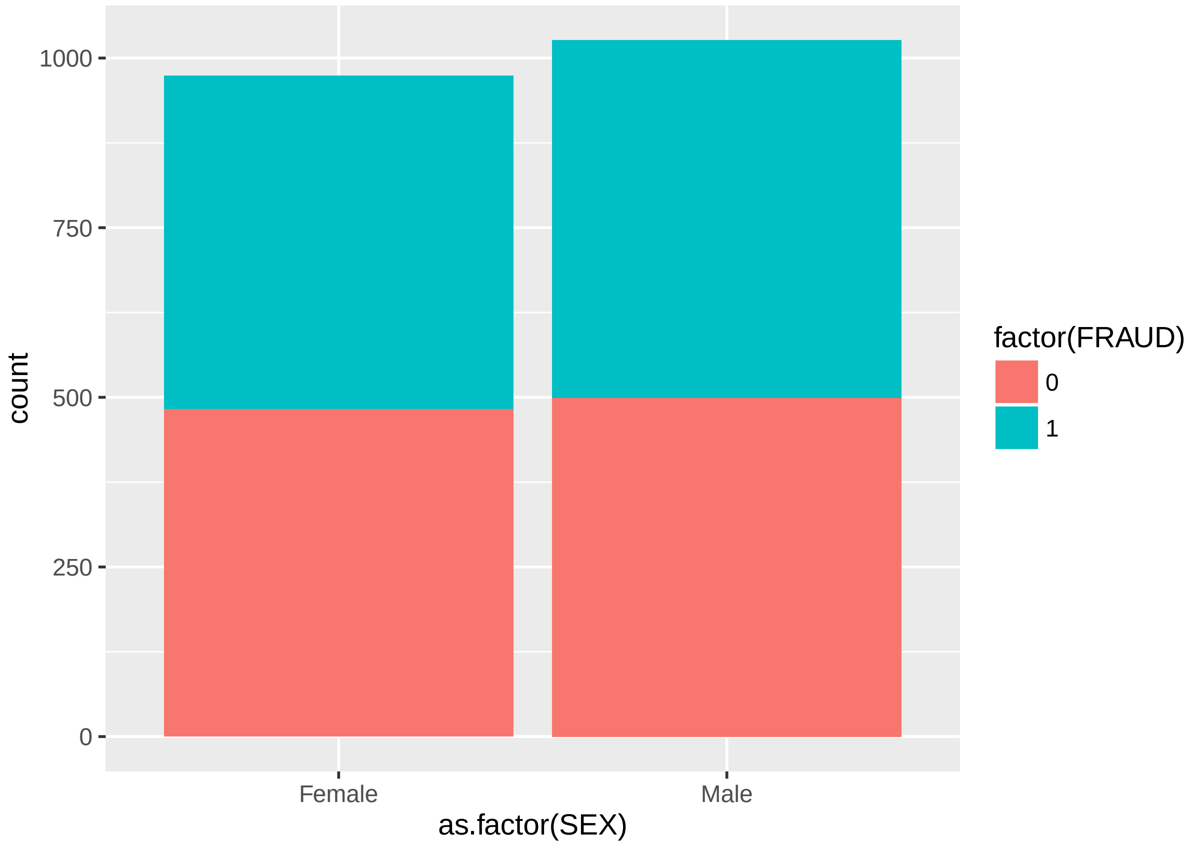
A few variables look to be correlated with FRAUD
In particular:
MEANS_FROM_EMP
DELAY
NO_DEP
FUEL_AMOUNT
MED_ASSESSMENT
MED_CONDITION_SATISFIED

No other variables appear to have any strong correlation

```
table(Data$FRAUD, Data$SEX)
```

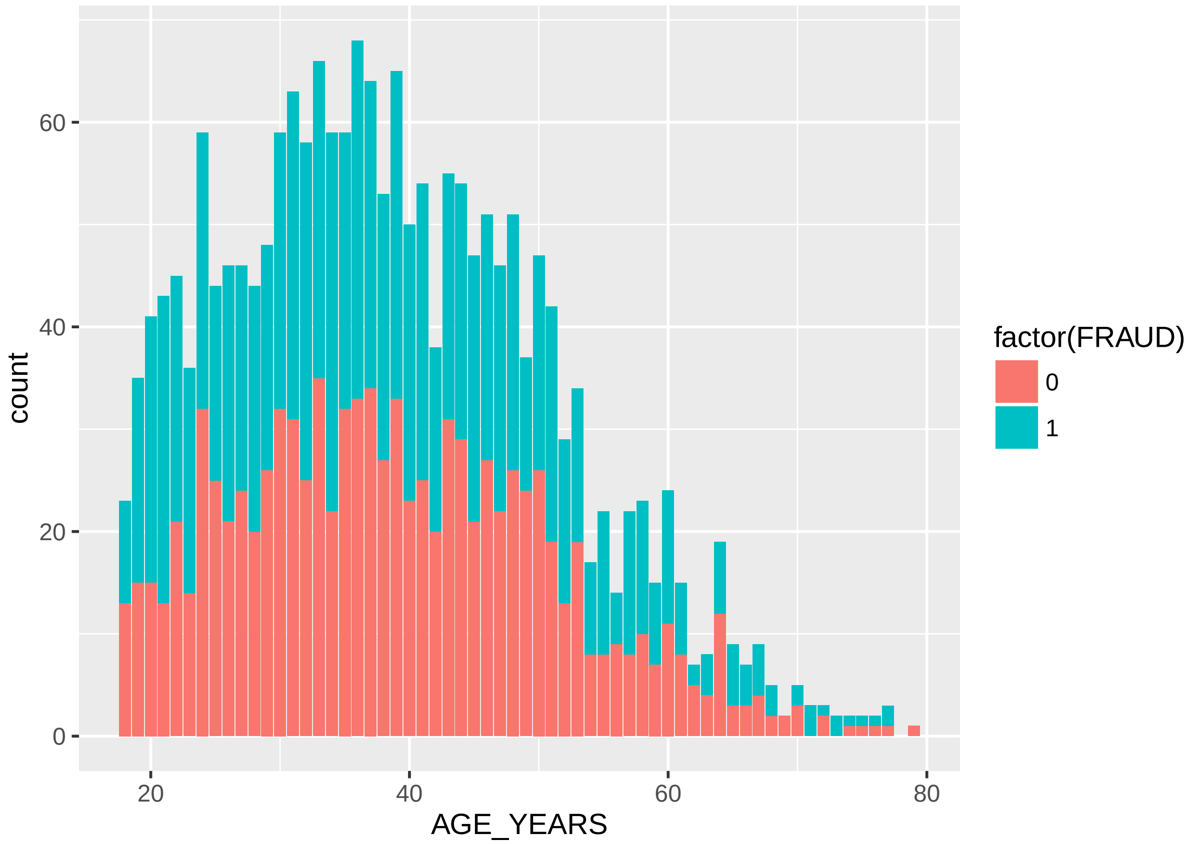
```
##
##     Female Male
##   0    483  499
##   1    491  527
```

```
ggplot(Data, aes(x = as.factor(SEX), fill=factor(FRAUD)))+geom_bar()
```



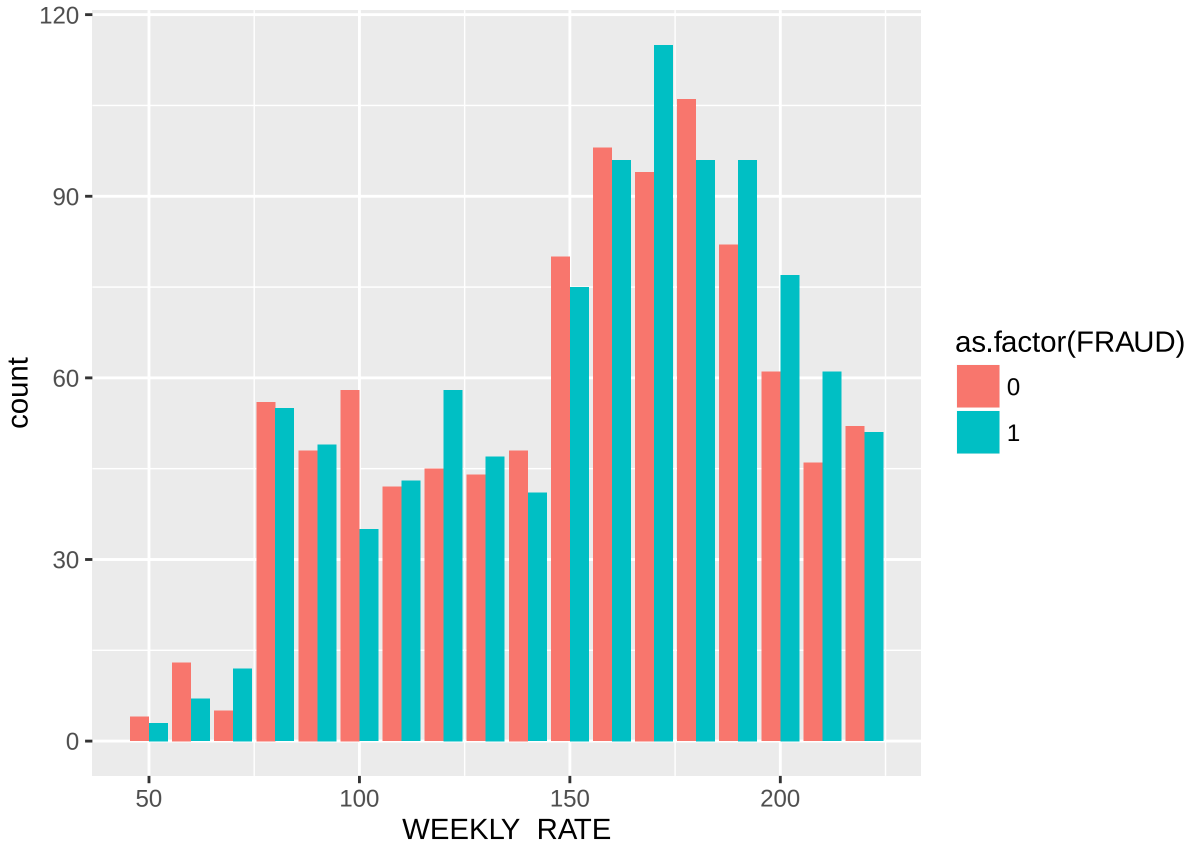
About 49% of Males and Females commit fraud

```
ggplot(Data, aes(x = AGE_YEARS, fill = factor(FRAUD))) + geom_bar()
```



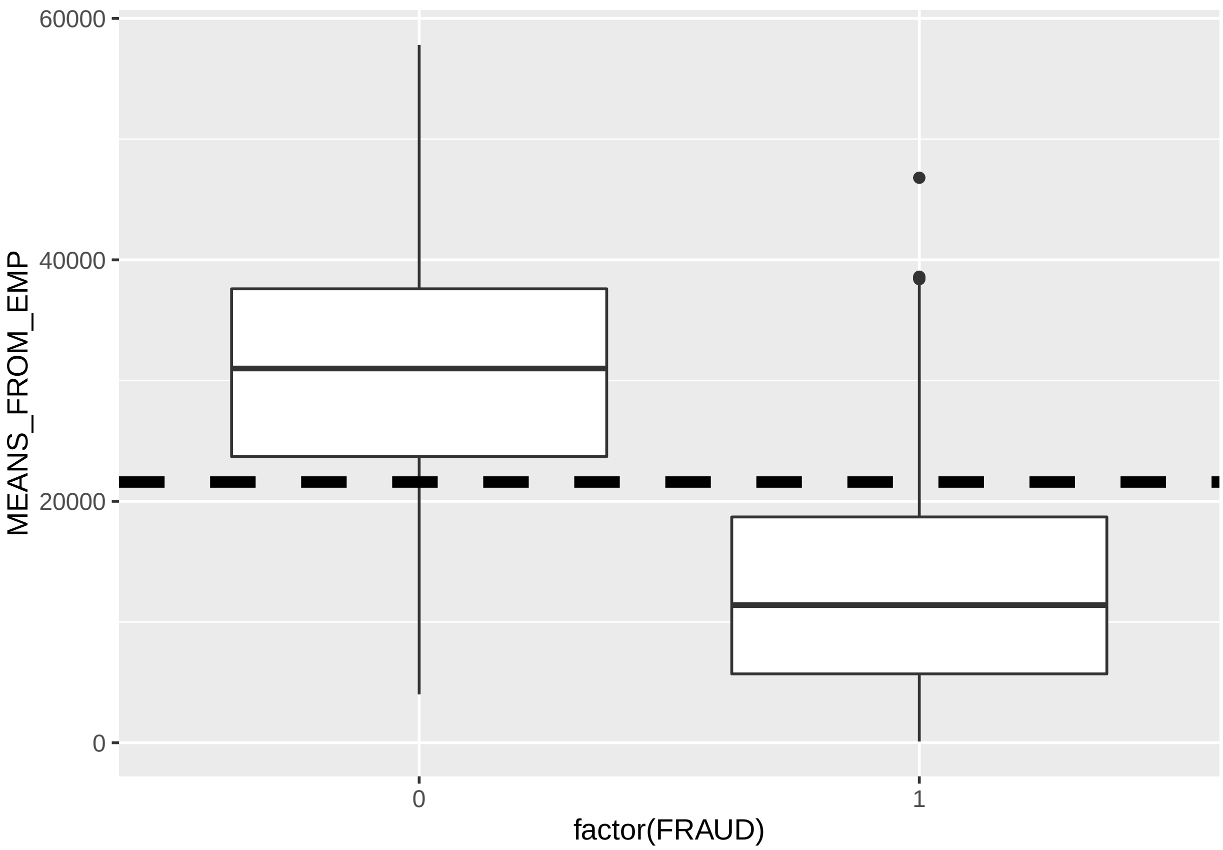

```
ggplot(Data, aes(x = WEEKLY_RATE, fill = as.factor(FRAUD))) +
  geom_bar(stat = 'count', position = 'dodge')
```

```
## Warning: Removed 1 rows containing non-finite values (stat_count).
```



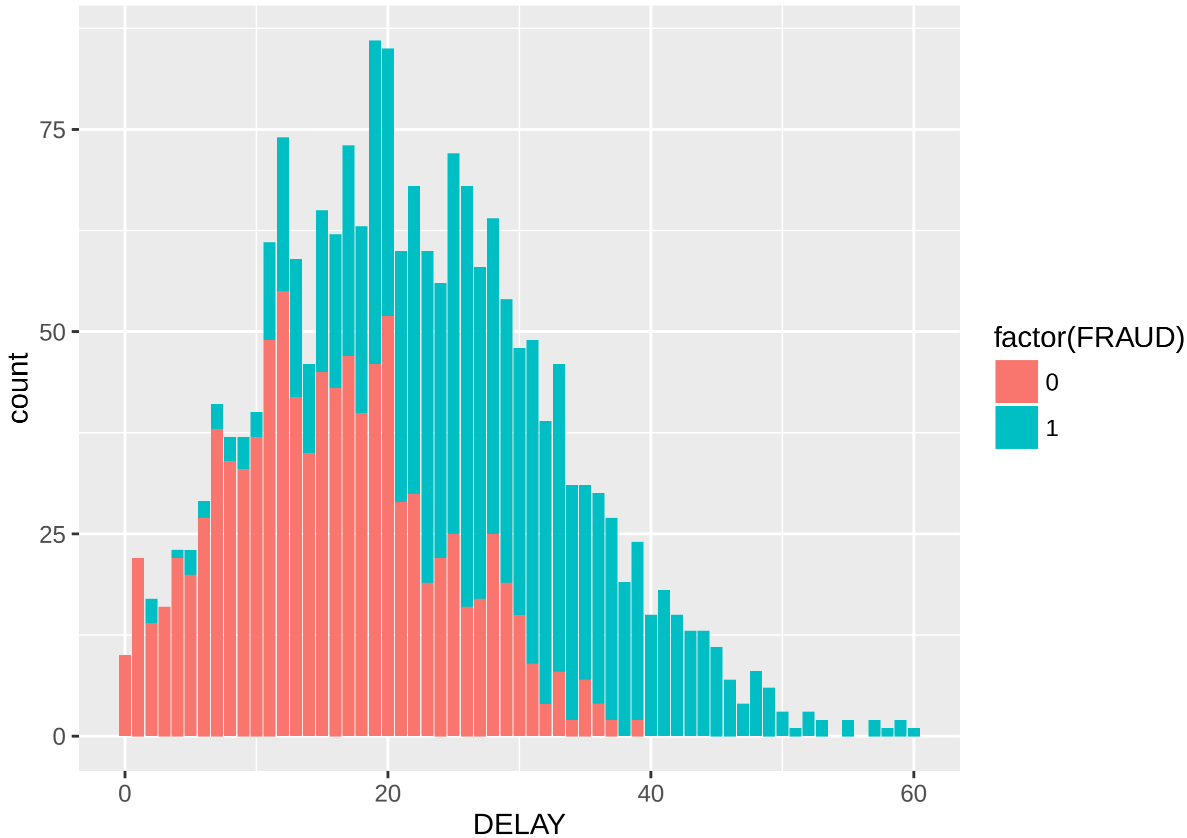
The level of fraud doesnt seem to be particularly
more prevalent in any age group or at any WEEKLY_RATE.

```
ggplot(Data, aes(x = factor(FRAUD), y =  MEANS_FROM_EMP)) +
  geom_boxplot() + geom_hline(aes(yintercept = mean(MEANS_FROM_EMP)),
                              linetype = 'dashed', lwd = 2)
```



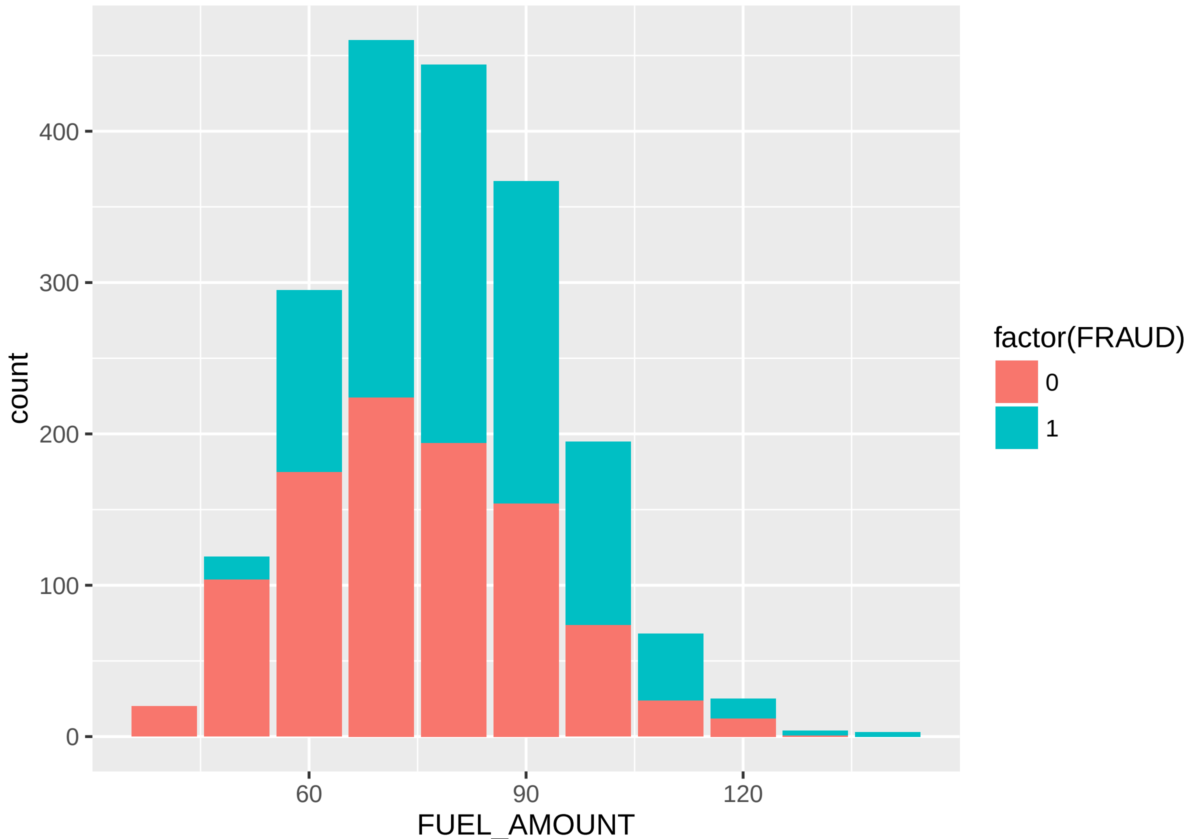
clearly more likely to commit fraud if below the mean value
of MEAN_FROM_EMP

```
ggplot(Data, aes(x =  DELAY, fill = factor(FRAUD))) + geom_bar()
```



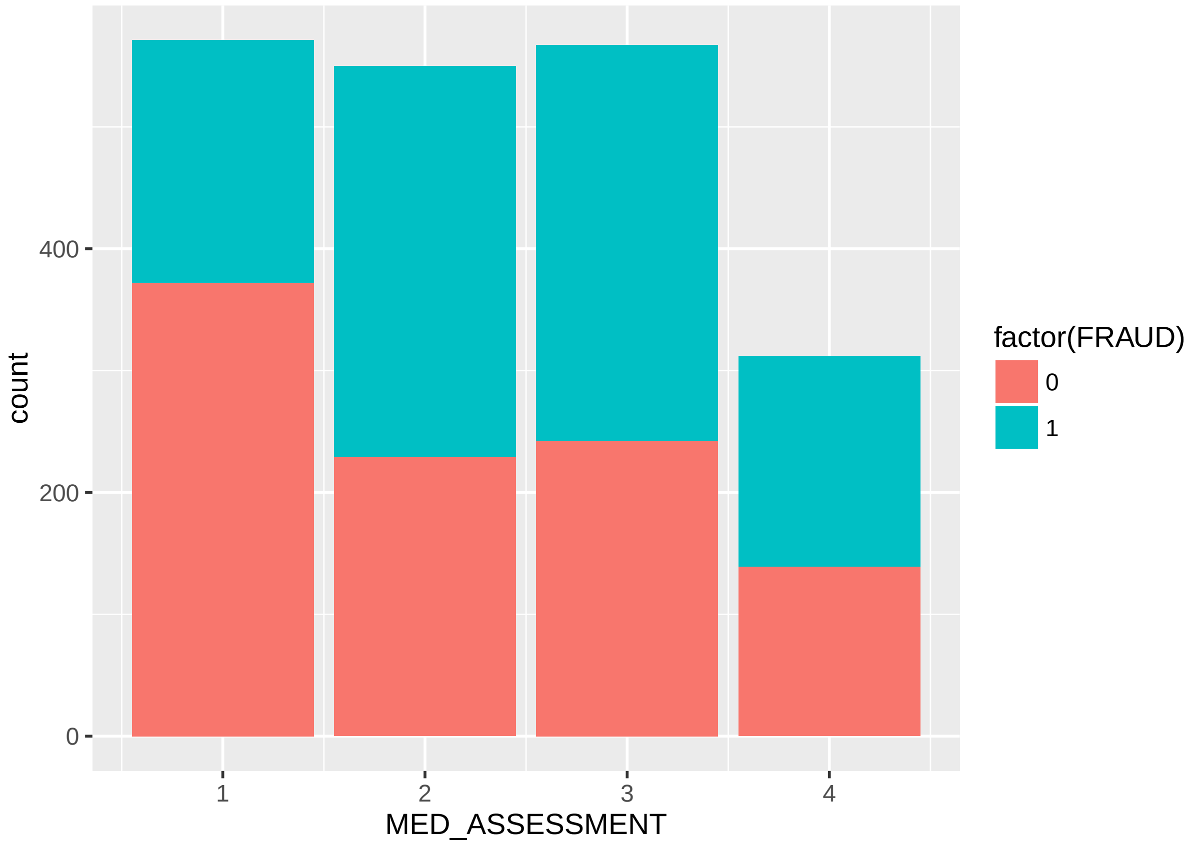
As delay increases there are increasingly higher proportion of FRAUDS

```
ggplot(Data, aes(x = FUEL_AMOUNT, fill = factor(FRAUD))) + geom_bar()
```



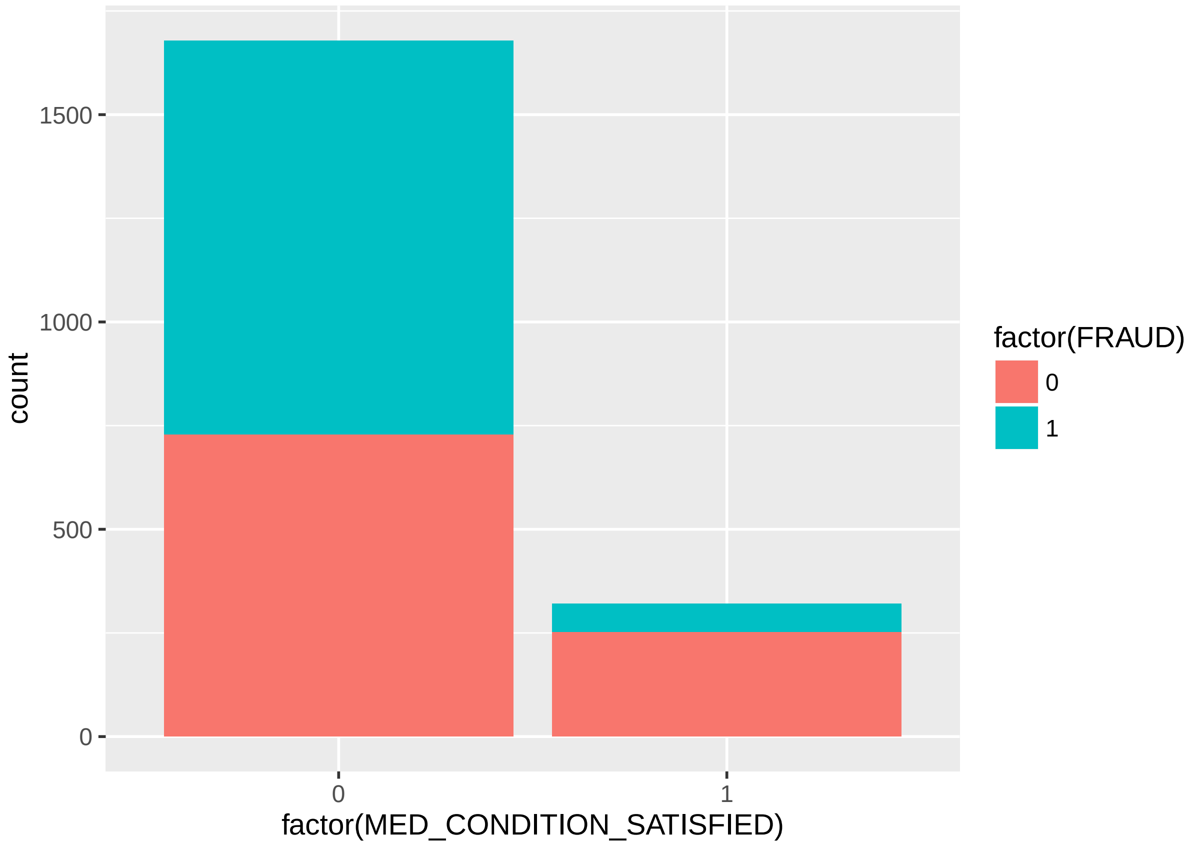
There is also perhaps some relevant information we can use here especially at higher fuel levels.

```
ggplot(Data, aes(x = MED_ASSESSMENT, fill = factor(FRAUD))) + geom_bar()
```



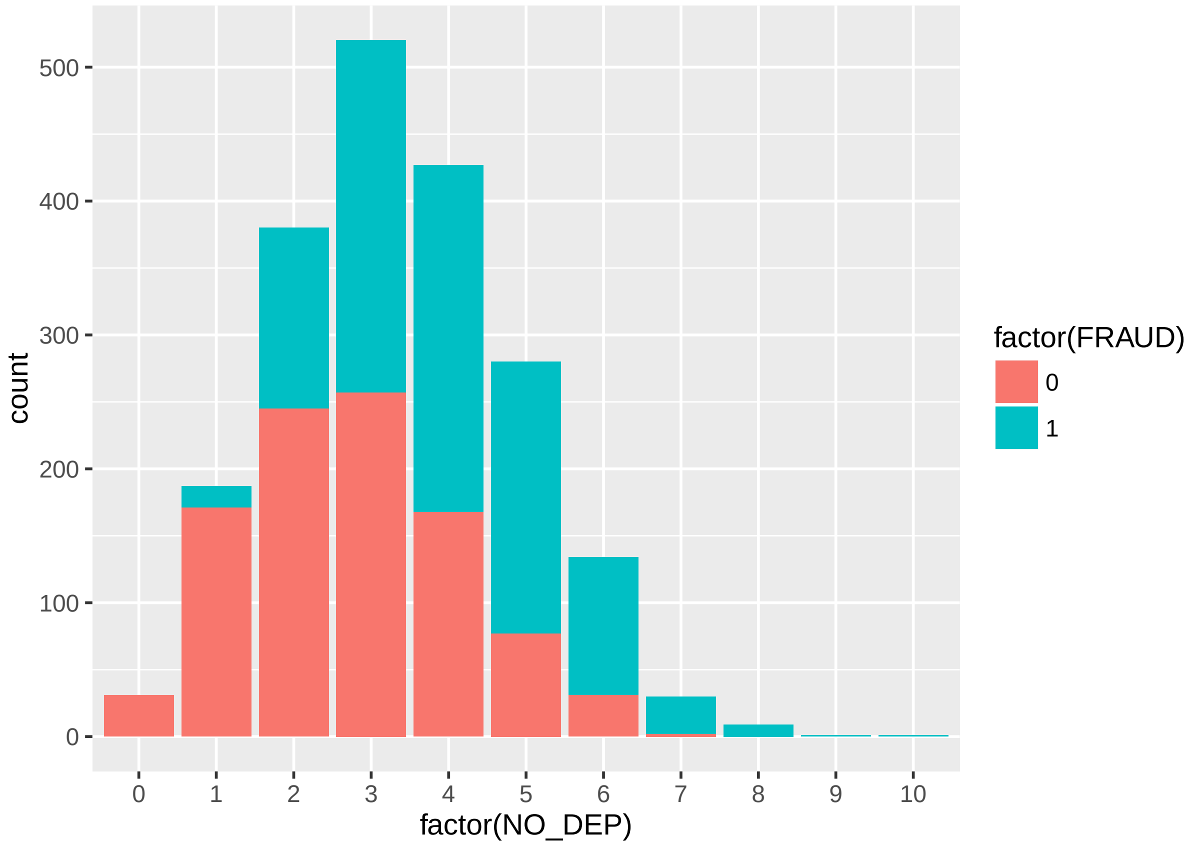
If you score one in medical assesment there is a lower chance you will commit FRAUD.

```
ggplot(Data, aes(x = factor(MED_CONDITION_SATISFIED), fill = factor(FRAUD))) + geom_bar()
```



Assuming 1 is having satisfied medical condition, you are less likely to commit fraud.

```
ggplot(Data, aes(x =  factor(NO_DEP), fill = factor(FRAUD))) + geom_bar()
```



Data Cleaning

Check pattern of missing values using Amelia package

```r
library(Amelia)
```

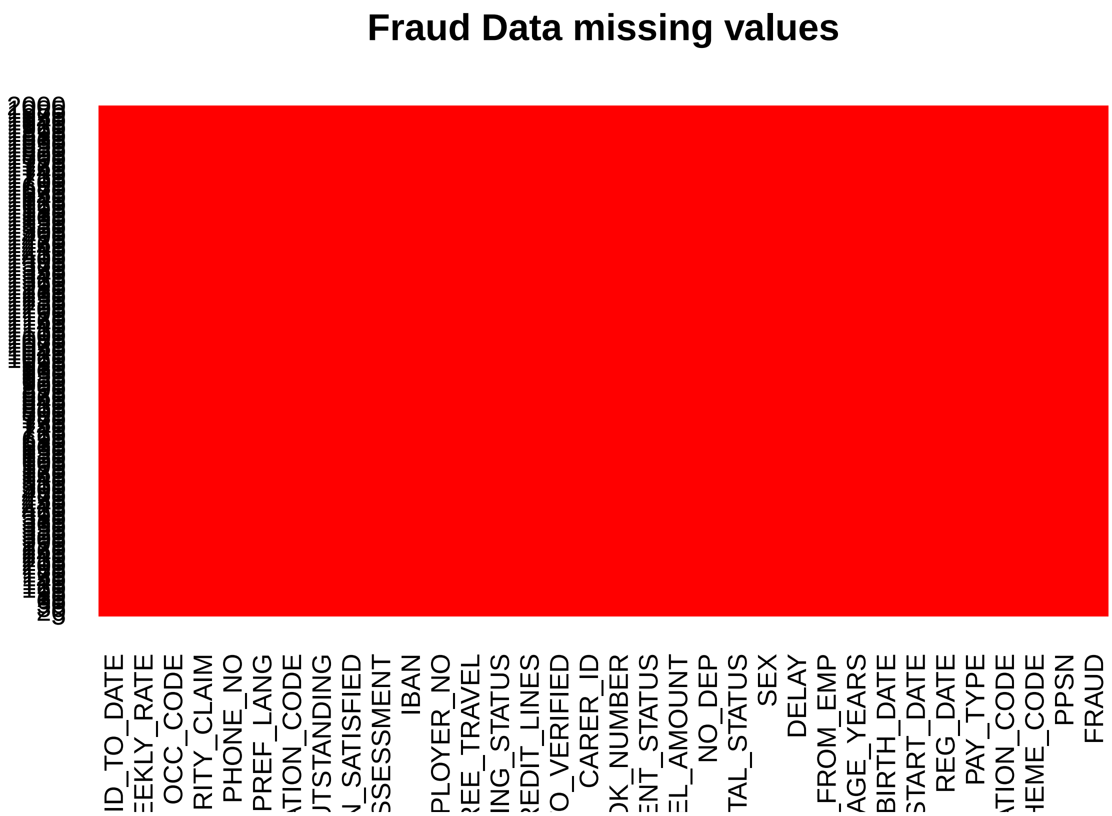
```
## Warning: package 'Amelia' was built under R version 3.1.3
```

```
## Loading required package: Rcpp
```

```
## Warning: package 'Rcpp' was built under R version 3.1.3
```

```
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.4, built: 2015-12-05)
## ## Copyright (C) 2005-2016 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```r
missmap(Data, main="Fraud Data missing values",
        col=c("blue", "red"), legend=FALSE)
```



It doesnt look like there is any data missing from the variables we
are going to be working with
NO_DEP and DELAY have zero values so we should inspect them to see
if they are errors or not

```
summary(Data$MEANS_FROM_EMP)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     100   10700   21150   21590   31400   57800
```

```
Data$MEANS_FROM_EMP <- scale(Data$MEANS_FROM_EMP)
```

They dont seem to be errors, however all values of zero are not frauds
We have also normalised the MEANS_FROM_EMP for any algorithms we use

Try Logistic Regression

Split into training and test sets (90/10)

```
Data$MED_ASSESSMENT <- as.factor(Data$MED_ASSESSMENT)
Data$MED_CONDITION_SATISFIED <- as.factor(Data$MED_CONDITION_SATISFIED)
Data$NO_DEP <- as.factor(Data$NO_DEP)
```

```r
Data$MEANS_NEW <- ifelse(Data$MEANS_FROM_EMP <= mean(Data$MEANS_FROM_EMP),1,0)
brks <- c(0,20,40,60)
Data$DELAY_RANGE <- cut(Data$DELAY, breaks = brks, include.lowest = T)
summary(Data$DELAY_RANGE )
```

```
##  [0,20] (20,40] (40,60]
##     969     919     112
```

```r
brks2 <- c(0,60,100,140)
Data$FUEL_NEW <- cut(Data$FUEL_AMOUNT, breaks = brks2, include.lowest = T)
summary(Data$FUEL_NEW)
```

```
##    [0,60]  (60,100] (100,140]
##       434      1466       100
```

```r
Data$NO_DEP2 <- ifelse(as.numeric(Data$NO_DEP) > 4,1,0)
Data$NO_DEP2 <- as.factor(Data$NO_DEP2)


Data$gp <- runif(dim(Data)[1])
trainingSet <- subset(Data, Data$gp > 0.1)
testSet <- subset(Data,Data$gp <= 0.1)
```

```r
reg1 <- glm(FRAUD ~ MEANS_FROM_EMP + DELAY + NO_DEP +
              FUEL_AMOUNT + MED_ASSESSMENT + MED_CONDITION_SATISFIED,
            data = trainingSet, family =  binomial(link = 'logit'))
summary(reg1)
```

```
##
## Call:
## glm(formula = FRAUD ~ MEANS_FROM_EMP + DELAY + NO_DEP + FUEL_AMOUNT +
##     MED_ASSESSMENT + MED_CONDITION_SATISFIED, family = binomial(link = "logit"),
##     data = trainingSet)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.96907  -0.23256   0.02181   0.25794   3.09717
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -16.98086  598.48538  -0.028   0.9774
## MEANS_FROM_EMP           -2.38553    0.13989 -17.053  < 2e-16 ***
## DELAY                     0.15008    0.01163  12.906  < 2e-16 ***
## NO_DEP1                  15.43118  598.48534   0.026   0.9794
## NO_DEP2                  18.24755  598.48523   0.030   0.9757
## NO_DEP3                  19.62547  598.48529   0.033   0.9738
## NO_DEP4                  20.67963  598.48539   0.035   0.9724
## NO_DEP5                  21.98400  598.48551   0.037   0.9707
## NO_DEP6                  22.97988  598.48570   0.038   0.9694
## NO_DEP7                  24.32543  598.48664   0.041   0.9676
## NO_DEP8                  38.59980 1333.36553   0.029   0.9769
```

```
## NO_DEP9                    45.02335 4001.19338   0.011   0.9910
## NO_DEP10                   37.56084 4001.19339   0.009   0.9925
## FUEL_AMOUNT                -0.08028    0.01149  -6.990 2.75e-12 ***
## MED_ASSESSMENT2             0.35933    0.26482   1.357   0.1748
## MED_ASSESSMENT3             0.60503    0.26378   2.294   0.0218 *
## MED_ASSESSMENT4             0.43600    0.30252   1.441   0.1495
## MED_CONDITION_SATISFIED1   -1.31575    0.30904  -4.257 2.07e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2488.80  on 1795  degrees of freedom
## Residual deviance:  835.23  on 1778  degrees of freedom
## AIC: 871.23
##
## Number of Fisher Scoring iterations: 16
```

There seems to be significance in all but one variable

Test Fit

```r
fit1 <- predict(reg1, newdata = testSet, type = 'response')
fit1 <- ifelse(fit1 > 0.5,1,0)
misClasificError1 <- mean(fit1 != testSet$FRAUD)
print(paste('Accuracy',1-misClasificError1))
```

```
## [1] "Accuracy 0.901960784313726"
```

Logistic Part 2

Try and increase performance by using logit on redfined variables

```r
reg2 <- glm(FRAUD ~ MEANS_NEW + DELAY_RANGE + NO_DEP2 +
              FUEL_NEW + MED_ASSESSMENT + MED_CONDITION_SATISFIED,
            data = trainingSet, family =  binomial(link = 'logit'))
summary(reg2)
```

```
##
## Call:
## glm(formula = FRAUD ~ MEANS_NEW + DELAY_RANGE + NO_DEP2 + FUEL_NEW +
##     MED_ASSESSMENT + MED_CONDITION_SATISFIED, family = binomial(link = "logit"),
##     data = trainingSet)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.66448  -0.39811   0.00008   0.46829   2.71299
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -3.6546     0.2713 -13.473  < 2e-16 ***
```

```
## MEANS_NEW                  3.0138     0.1530  19.699  < 2e-16 ***
## DELAY_RANGE(20,40]         1.9520     0.1507  12.949  < 2e-16 ***
## DELAY_RANGE(40,60]        18.1385   337.3166   0.054  0.95712
## NO_DEP21                   1.0501     0.1703   6.166 7.00e-10 ***
## FUEL_NEW(60,100]           0.4862     0.2010   2.419  0.01557 *
## FUEL_NEW(100,140]          0.2435     0.3873   0.629  0.52963
## MED_ASSESSMENT2            0.5724     0.2187   2.617  0.00886 **
## MED_ASSESSMENT3            0.6731     0.2194   3.067  0.00216 **
## MED_ASSESSMENT4            0.3578     0.2517   1.422  0.15510
## MED_CONDITION_SATISFIED1  -1.1811     0.2519  -4.689 2.75e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2488.8  on 1795  degrees of freedom
## Residual deviance: 1224.4  on 1785  degrees of freedom
## AIC: 1246.4
##
## Number of Fisher Scoring iterations: 16
```

Test Fit

```
fit <- predict(reg2, newdata = testSet, type = 'response')
fit <- ifelse(fit> 0.5,1,0)

misClasificError <- mean(fit != testSet$FRAUD)
print(paste('Accuracy',1-misClasificError))
```

```
## [1] "Accuracy 0.838235294117647"
```

Random Forest

Next we wil try a randomforest wichi tend to perform well with this kind of problem

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.1.3
```

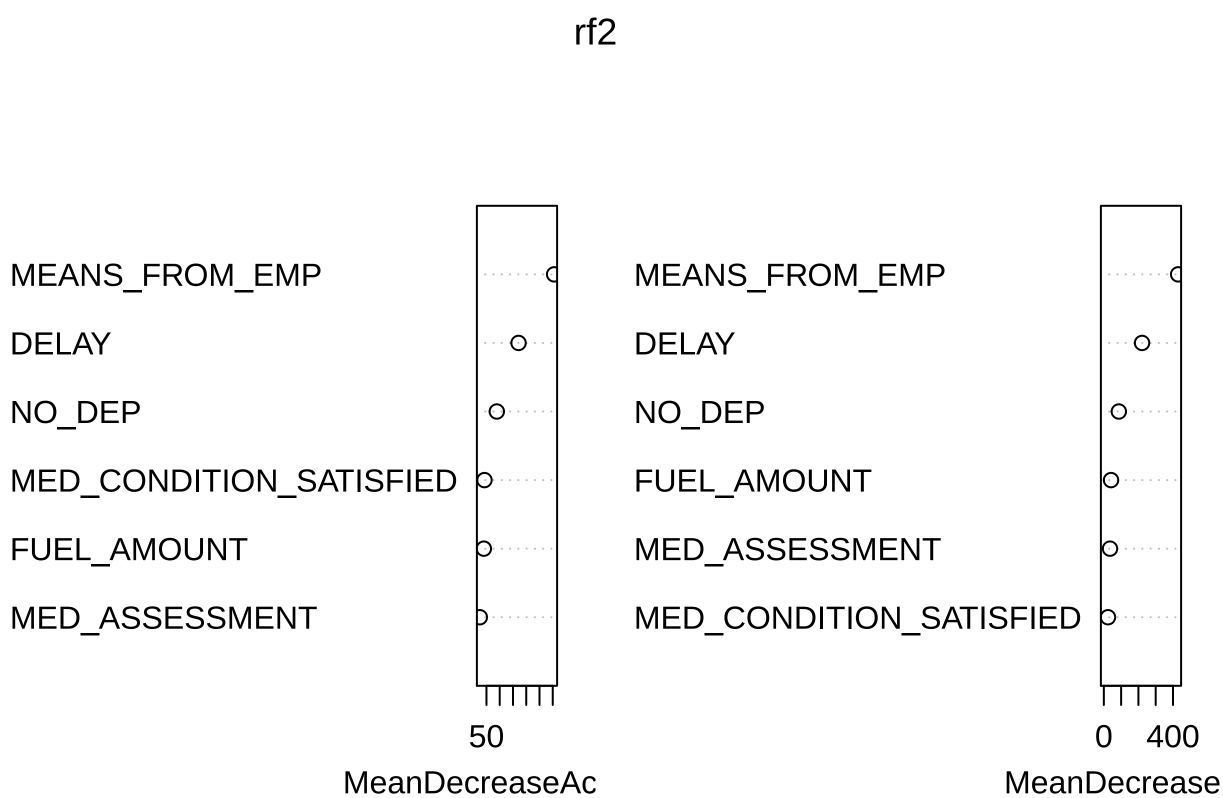
```
## randomForest 4.6-12
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
set.seed(111)
rf2 <- randomForest(as.factor(FRAUD) ~ MEANS_FROM_EMP + DELAY + NO_DEP +
              FUEL_AMOUNT + MED_ASSESSMENT + MED_CONDITION_SATISFIED,
                 data = trainingSet,
                   importance = TRUE,
                   ntree = 2000)
varImpPlot(rf2)
```




```r
fit3 <- predict(rf2, newdata = testSet, type = 'response')
misClasificError3 <- mean(fit3 != testSet$FRAUD)
print(paste('Accuracy',1-misClasificError3))
```

```
## [1] "Accuracy 0.901960784313726"
```

Accuracy was reduced slightly

Finally we perform a conditional random forest from the party package

```r
library(party)
```

```
## Warning: package 'party' was built under R version 3.1.3
```

```
## Loading required package: grid
```

```
## Loading required package: mvtnorm

## Warning: package 'mvtnorm' was built under R version 3.1.3

## Loading required package: modeltools

## Warning: package 'modeltools' was built under R version 3.1.3

## Loading required package: stats4

## Loading required package: strucchange

## Warning: package 'strucchange' was built under R version 3.1.3

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 3.1.3

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

## Warning: package 'sandwich' was built under R version 3.1.3
```

```
set.seed(333)
rf1 <- cforest(as.factor(FRAUD) ~ MEANS_FROM_EMP + DELAY + NO_DEP +
              FUEL_AMOUNT + MED_ASSESSMENT + MED_CONDITION_SATISFIED,
                  data = trainingSet,
               controls=cforest_unbiased(ntree=2000, mtry=3))



fit4 <- predict(rf1, newdata = testSet, type = 'response')
misClasificError4 <- mean(fit4 != testSet$FRAUD)
print(paste('Accuracy',1-misClasificError4))
```

```
## [1] "Accuracy 0.887254901960784"
```

It looks as though our original Logistic Regression is actually the most accurate on the tests set. We can perform Cross validation to confirm.

```
formula = as.factor(FRAUD) ~ MEANS_NEW + DELAY_RANGE +
                  NO_DEP2 + FUEL_NEW + MED_ASSESSMENT +
                  MED_CONDITION_SATISFIED
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.1.3

## Loading required package: lattice
```

```
train_control <- trainControl(method = 'cv', number = 10)
modelLog = train(formula, data=trainingSet, method="glm", family=binomial,
                 trControl=train_control)
print(modelLog)
```

```
## Generalized Linear Model
##
## 1796 samples
##   38 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1616, 1616, 1616, 1617, 1617, 1616, ...
## Resampling results
##
##   Accuracy   Kappa      Accuracy SD  Kappa SD
##   0.8529857  0.7056808  0.02322506   0.04658199
##
##
```

Try on RandomForest

```
train_control2 <- trainControl(method = 'cv', number = 10)
modelLog2 = train(formula, data=trainingSet, method="rf",
                 trControl=train_control)
print(modelLog2)
```

```
## Random Forest
##
## 1796 samples
##   38 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1617, 1616, 1617, 1616, 1616, 1617, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa      Accuracy SD  Kappa SD
##    2    0.8457604  0.6910251  0.02439651   0.04908086
##    6    0.8418560  0.6836242  0.02988816   0.05984692
##   10    0.8412973  0.6825154  0.03136126   0.06276969
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 2.
```