Fraud Detection

2023-11-23

```
# Loading the necessary libraries
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(ggplot2)
library(corrplot)

## corrplot 0.92 loaded

library(randomForest)

## randomForest 4.7-1.1

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

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

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

# Loading the dataset
data_fraud <- read.csv("C:/Users/HP/Documents/enron.csv")
```

```r
# Checking for missing values
missing_values <- colSums(is.na(data_fraud))
print(missing_values)
```

```
##                       Name                     salary                  to_messa
ges
##                          0                         51
60
##         deferral_payments            total_payments              loan_advan
ces
##                        107                         21
142
##                      bonus             email_address restricted_stock_defer
red
##                         64                          0
128
##           deferred_income         total_stock_value                   expen
ses
##                         97                         20
51
##   from_poi_to_this_person   exercised_stock_options              from_messa
ges
##                         60                         44
60
##                      other   from_this_person_to_poi
poi
##                         53                         60
0
##       long_term_incentive   shared_receipt_with_poi          restricted_st
ock
##                         80                         60
36
##             director_fees
##                        129
```

```r
# Impute missing values
# List of variables with missing values and their counts
variables_with_missing <- c("salary", "to_messages", "deferral_payments", "to
tal_payments",
                            "loan_advances", "bonus", "restricted_stock_defe
rred",
                            "deferred_income", "total_stock_value", "expense
s",
                            "from_poi_to_this_person", "exercised_stock_opti
ons",
                            "from_messages", "other", "from_this_person_to_p
oi",
                            "long_term_incentive", "shared_receipt_with_poi"
,
                            "restricted_stock", "director_fees")
```

```r
# Replace missing values with the mean for each variable
for (variable in variables_with_missing) {
  if (sum(is.na(data_fraud[[variable]])) > 0) {
    mean_value <- mean(data_fraud[[variable]], na.rm = TRUE)
    data_fraud[[variable]][is.na(data_fraud[[variable]])] <- mean_value
  }
}

# Verify that missing values have been replaced
missing_values_after <- colSums(is.na(data_fraud))
print(missing_values_after)
```

```
##                      Name                    salary                to_messa
ges
##                         0                         0
0
##         deferral_payments            total_payments              loan_advan
ces
##                         0                         0
0
##                     bonus             email_address restricted_stock_defer
red
##                         0                         0
0
##           deferred_income         total_stock_value                   expen
ses
##                         0                         0
0
##   from_poi_to_this_person   exercised_stock_options              from_messa
ges
##                         0                         0
0
##                     other   from_this_person_to_poi
poi
##                         0                         0
0
##       long_term_incentive   shared_receipt_with_poi          restricted_st
ock
##                         0                         0
0
##             director_fees
##                         0
```

```r
# Summary Statistics
summary_stats <- summary(data_fraud)

# Display the summary statistics
print(summary_stats)
```

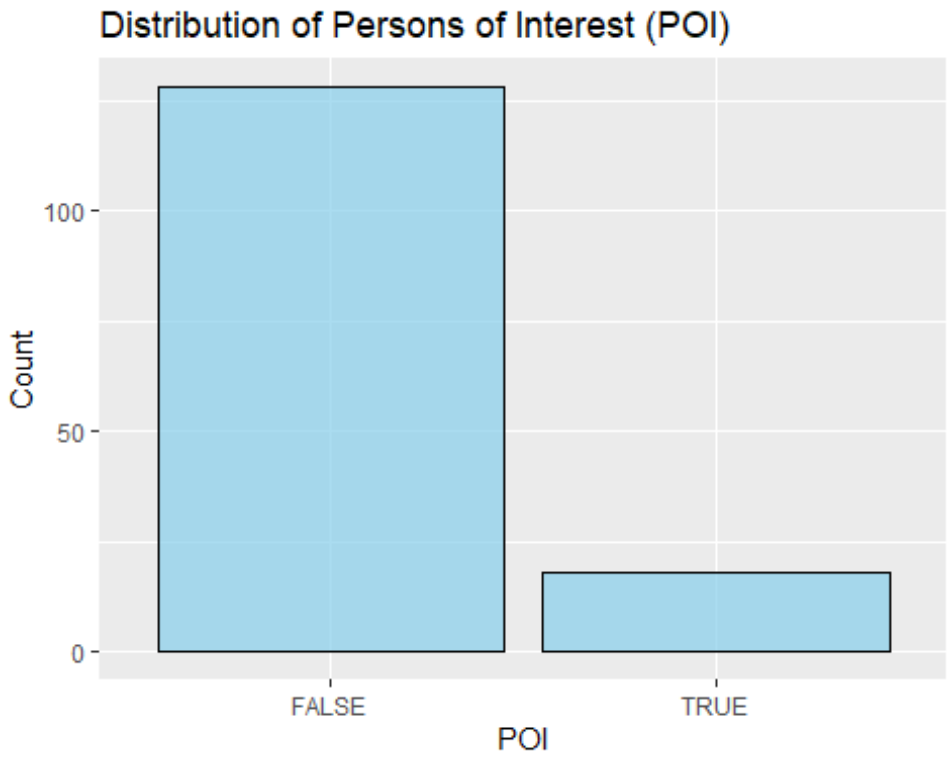
```
##      Name               salary          to_messages      deferral_payments
##  Length:146         Min.   :     477   Min.   :   57.0   Min.   : -102500
##  Class :character   1st Qu.:  239800   1st Qu.:  904.2   1st Qu.: 1642674
##  Mode  :character   Median :  323310   Median : 2073.9   Median : 1642674
##                     Mean   :  562194   Mean   : 2073.9   Mean   : 1642674
##                     3rd Qu.:  562194   3rd Qu.: 2073.9   3rd Qu.: 1642674
##                     Max.   :26704229   Max.   :15149.0   Max.   :32083396
##  total_payments      loan_advances          bonus          email_address
##  Min.   :      148   Min.   :  400000   Min.   :   70000   Length:146
##  1st Qu.:   554468   1st Qu.:41962500   1st Qu.:  712500   Class :characte
r
##  Median :  1354124   Median :41962500   Median : 2374235   Mode  :characte
r
##  Mean   :  5081526   Mean   :41962500   Mean   : 2374235
##  3rd Qu.:  3625040   3rd Qu.:41962500   3rd Qu.: 2374235
##  Max.   :309886585   Max.   :83925000   Max.   :97343619
##  restricted_stock_deferred deferred_income     total_stock_value
##  Min.   :-7576788          Min.   :-27992891   Min.   :   -44093
##  1st Qu.:  166411          1st Qu.: -1140475   1st Qu.:   612908
##  Median :  166411          Median : -1140475   Median :  1710454
##  Mean   :  166411          Mean   : -1140475   Mean   :  6773957
##  3rd Qu.:  166411          3rd Qu.:  -700896   3rd Qu.:  6331709
##  Max.   :15456290          Max.   :     -833   Max.   :434509511
##     expenses       from_poi_to_this_person exercised_stock_options
##  Min.   :    148   Min.   :  0.00          Min.   :     3285
##  1st Qu.:  33848   1st Qu.: 25.75          1st Qu.:   757865
##  Median :  82474   Median : 64.90          Median :  2250522
##  Mean   : 108729   Mean   : 64.90          Mean   :  5987054
##  3rd Qu.: 108729   3rd Qu.: 64.90          3rd Qu.:  5987054
##  Max.   :5235198   Max.   :528.00          Max.   :311764000
##  from_messages         other          from_this_person_to_poi    poi
##  Min.   :   12.0   Min.   :       2   Min.   :  0.00          Mode :logica
l
##  1st Qu.:   36.0   1st Qu.:    2686   1st Qu.:  6.00          FALSE:128
##  Median :  608.8   Median :  401306   Median : 41.23          TRUE :18
##  Mean   :  608.8   Mean   :  919065   Mean   : 41.23
##  3rd Qu.:  608.8   3rd Qu.:  919065   3rd Qu.: 41.23
##  Max.   :14368.0   Max.   :42667589   Max.   :609.00
##  long_term_incentive shared_receipt_with_poi restricted_stock
##  Min.   :   69223    Min.   :   2.0          Min.   : -2604490
##  1st Qu.:  544169    1st Qu.: 591.5          1st Qu.:   360528
##  Median : 1470361    Median :1176.5          Median :   775992
##  Mean   : 1470361    Mean   :1176.5          Mean   :  2321741
##  3rd Qu.: 1470361    3rd Qu.:1176.5          3rd Qu.:  2321741
##  Max.   :48521928    Max.   :5521.0          Max.   :130322299
##  director_fees
##  Min.   :   3285
##  1st Qu.: 166805
##  Median : 166805
##  Mean   : 166805
```

```
##  3rd Qu.: 166805
##  Max.   :1398517
```

##DATA VISUALIZATION

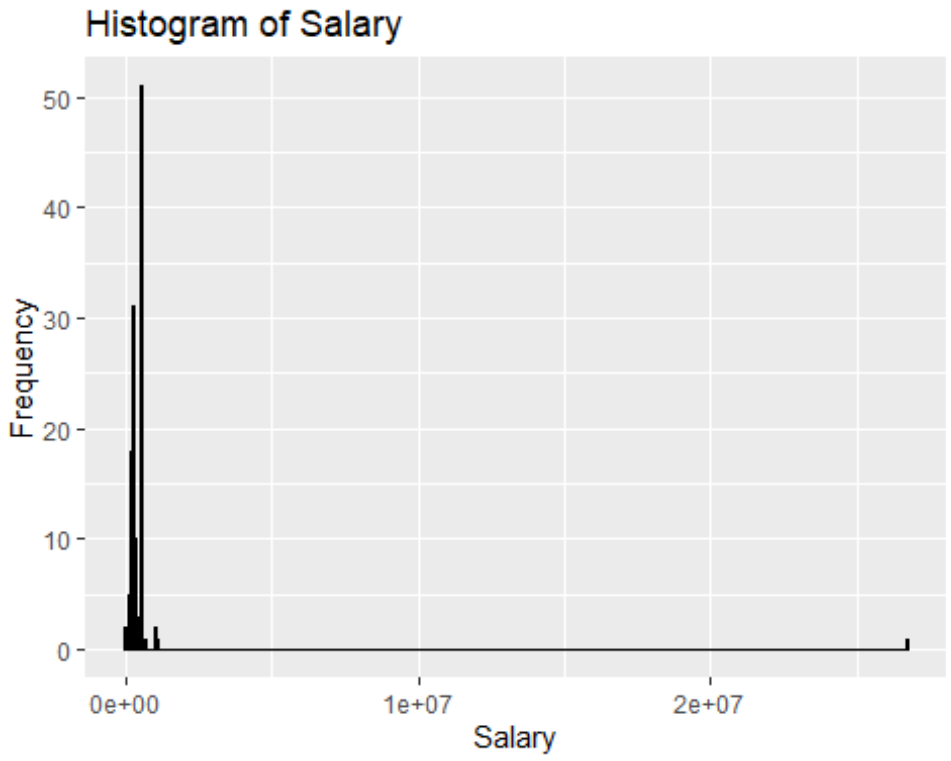
```
# POI

ggplot(data_fraud, aes(x = factor(poi))) +
  geom_bar(fill = "skyblue", color = "black", alpha = 0.7) +
  labs(title = "Distribution of Persons of Interest (POI)", x = "POI", y = "C
ount")
```



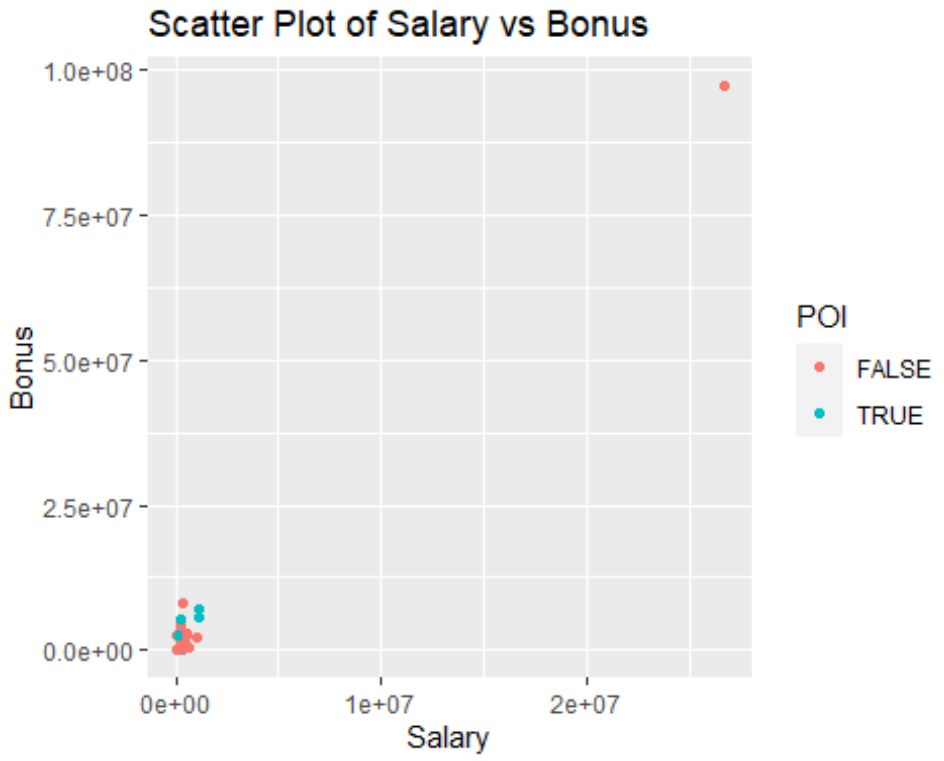

```
# Salary
# Convert 'salary' to numeric
data_fraud$salary <- as.numeric(data_fraud$salary)

# Create a histogram for 'salary'
ggplot(data_fraud, aes(x = salary)) +
  geom_histogram(binwidth = 50000, fill = "skyblue", color = "black", alpha =
0.7) +
  labs(title = "Histogram of Salary", x = "Salary", y = "Frequency")
```

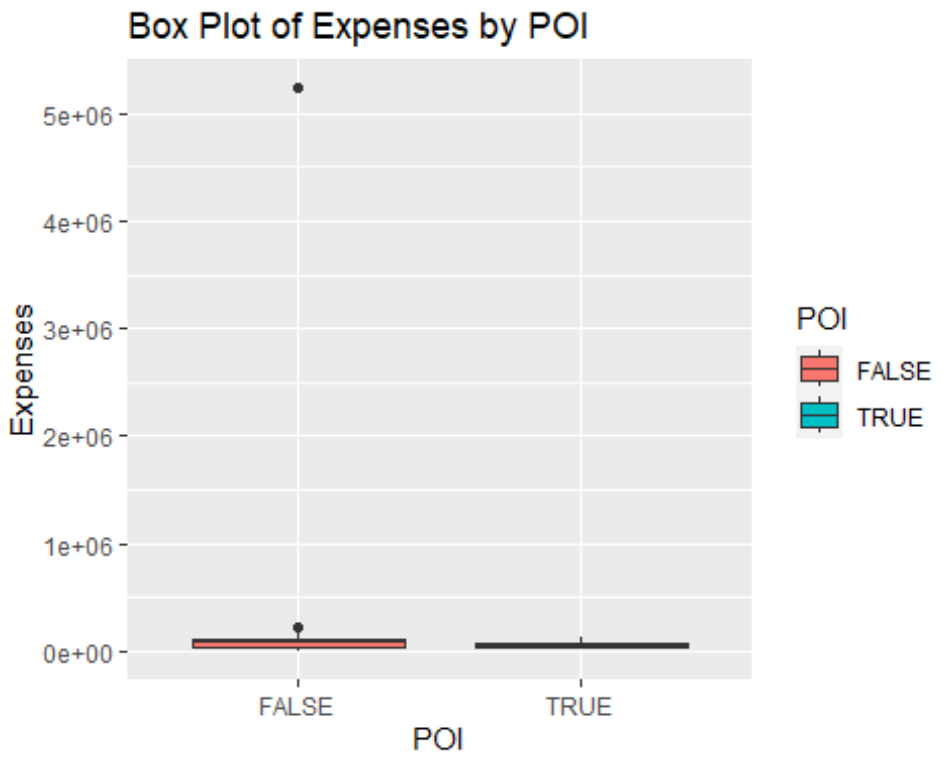


##Relationship between salary and bonus

```
ggplot(data_fraud, aes(x = salary, y = bonus, color = poi)) +
  geom_point() +
  labs(title = "Scatter Plot of Salary vs Bonus", x = "Salary", y = "Bonus",
color = "POI")
```

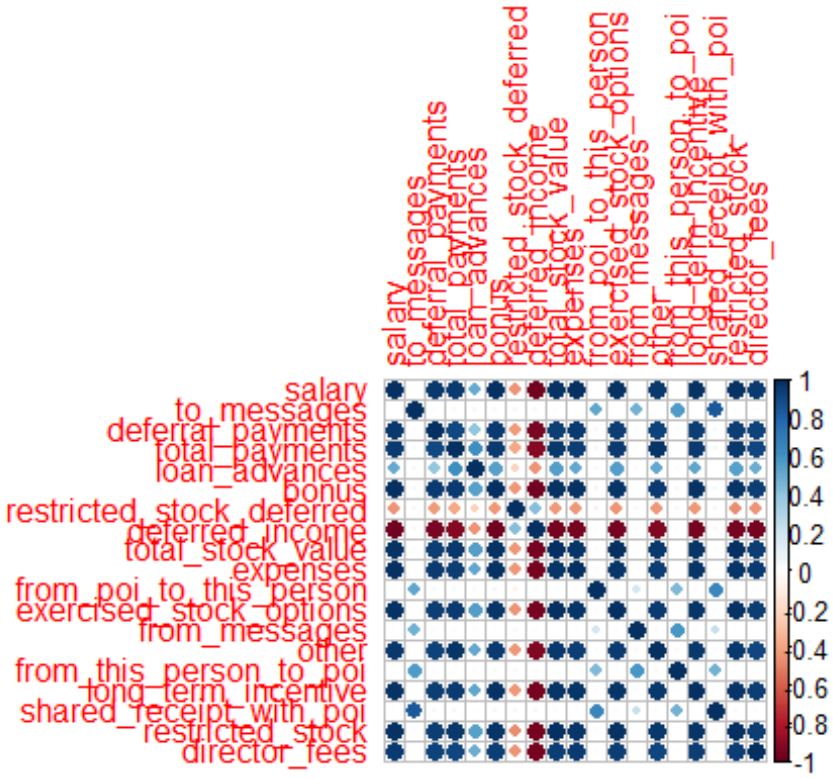



```r
ggplot(data_fraud, aes(x = factor(poi), y = expenses, fill = factor(poi))) +
  geom_boxplot() +
  labs(title = "Box Plot of Expenses by POI", x = "POI", y = "Expenses", fill
= "POI")
```




```r
# Assuming you have calculated the correlation matrix
correlation_matrix <- cor(data_fraud[, -c(1, 8, 18)], use = "complete.obs")

# Create a half correlation matrix plot with larger boxes and font size
corrplot(correlation_matrix, method = "circle")
```



##Dropping 'Names' and 'Emails' since they are not relevant

```
# Drop variables 1 and 8
data_fraud <- data_fraud[, -c(1, 8)]
```

##MODEL GENERATION

```
# Splitting the data into training and testing sets
# Converting poi into binary
data_fraud$poi <- ifelse(data_fraud$poi == "TRUE", 1, 0)
set.seed(123)  # For reproducibility
train_index <- createDataPartition(data_fraud$poi, p = 0.8, list = FALSE)
train_data <- data_fraud[train_index, ]
test_data <- data_fraud[-train_index, ]

# Logistics regression model

# Define the logistic regression model with regularization
model <- glm(poi ~., data = train_data, family = "binomial")

# Make predictions on the test set
predictions <- predict(model, newdata = test_data, type = "response")

# Convert probabilities to binary predictions (1 or 0) based on a threshold
threshold <- 0.5
binary_predictions <- ifelse(predictions > threshold, 1, 0)
```

```r
# Evaluate the model
confusion_matrix <- confusionMatrix(table(test_data$poi, binary_predictions))
print(confusion_matrix)
```

```
## Confusion Matrix and Statistics
##
##    binary_predictions
##      0  1
##   0 26  1
##   1  1  1
##
##                Accuracy : 0.931
##                  95% CI : (0.7723, 0.9915)
##     No Information Rate : 0.931
##     P-Value [Acc > NIR] : 0.6768
##
##                   Kappa : 0.463
##
##  Mcnemar's Test P-Value : 1.0000
##
##             Sensitivity : 0.9630
##             Specificity : 0.5000
##          Pos Pred Value : 0.9630
##          Neg Pred Value : 0.5000
##              Prevalence : 0.9310
##          Detection Rate : 0.8966
##    Detection Prevalence : 0.9310
##       Balanced Accuracy : 0.7315
##
##        'Positive' Class : 0
##
```

```r
# Check model summary for coefficients and statistics
summary(model)
```

```
##
## Call:
## glm(formula = poi ~ ., family = "binomial", data = train_data)
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -1.412e+01  2.025e+01  -0.697   0.4856
## salary                     8.109e-07  3.121e-06   0.260   0.7950
## to_messages               -1.181e-04  1.037e-03  -0.114   0.9093
## deferral_payments         -2.602e-08  6.756e-07  -0.039   0.9693
## total_payments            -1.828e-07  3.238e-07  -0.565   0.5723
## loan_advances              1.010e-07  4.534e-07   0.223   0.8237
## bonus                     -4.155e-07  5.053e-07  -0.822   0.4109
## restricted_stock_deferred  2.550e-07  6.796e-07   0.375   0.7075
## deferred_income            2.564e-07  6.916e-07   0.371   0.7108
```

```
## total_stock_value         -1.404e-07  1.633e-07  -0.860   0.3898
## expenses                  -9.605e-06  1.088e-05  -0.883   0.3775
## from_poi_to_this_person   -1.415e-03  6.075e-03  -0.233   0.8158
## exercised_stock_options    3.454e-07  1.621e-07   2.131   0.0331 *
## from_messages             -5.714e-03  2.977e-03  -1.919   0.0549 .
## other                     -3.542e-07  8.517e-07  -0.416   0.6775
## from_this_person_to_poi    4.165e-02  2.295e-02   1.815   0.0695 .
## long_term_incentive        1.412e-07  7.162e-07   0.197   0.8437
## shared_receipt_with_poi    1.030e-03  1.423e-03   0.724   0.4693
## restricted_stock           2.400e-07  3.576e-07   0.671   0.5021
## director_fees              4.764e-05  6.224e-05   0.765   0.4440
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 93.372  on 116  degrees of freedom
## Residual deviance: 55.090  on  97  degrees of freedom
## AIC: 95.09
##
## Number of Fisher Scoring iterations: 8
```

##RANDOM FOREST MODEL

```
# setting 'poi' as a factor variable
train_data$poi <- as.factor(train_data$poi)
# Define the Random Forest model
mod <- randomForest(poi ~ ., data = train_data, ntree = 100, mtry = 8)
print(mod)

##
## Call:
##  randomForest(formula = poi ~ ., data = train_data, ntree = 100,      mtry
= 8)
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 8
##
##         OOB estimate of  error rate: 16.24%
## Confusion matrix:
##    0 1 class.error
## 0 98 3  0.02970297
## 1 16 0  1.00000000

importance(mod)

##                           MeanDecreaseGini
## salary                          0.94717893
## to_messages                     0.68157689
## deferral_payments               1.21349155
## total_payments                  0.95955629
```

```
## loan_advances                    0.06833333
## bonus                            2.03563003
## restricted_stock_deferred        0.02619048
## deferred_income                  3.72563548
## total_stock_value                1.91692871
## expenses                         2.67329801
## from_poi_to_this_person          0.97881783
## exercised_stock_options          2.15416966
## from_messages                    0.69387412
## other                            2.94189645
## from_this_person_to_poi          1.74969842
## long_term_incentive              1.56915024
## shared_receipt_with_poi          1.61073412
## restricted_stock                 0.92832053
## director_fees                    0.01714286
```

```r
# plotting variable importance
res3 <- data.frame(varImp(mod)) #  variable importance
res3$impProcent <- round(res3$Overall/sum(res3$Overall)*100)
res3$var<- rownames(res3)

res3 %>% ggplot(aes(x=impProcent, y=reorder(var, impProcent))) +
  geom_bar(stat="identity")
```