Marketing_prediction.R

sudhanshmehta

2020-04-27

```
# loading all the libraries

library(ISLR)
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(lattice)
library(glmnet)

## Loading required package: Matrix

## Loaded glmnet 3.0-2

library(tree)
library(MASS)

##
## Attaching package: 'MASS'

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

library(ROSE)

## Loaded ROSE 0.0-3

library(rpart)


#######################################  PART 3
#######################################
```

```r
bank=read.table("bank.csv",sep=";",header=TRUE)

head(bank)
```

```
##   age         job marital education default balance housing loan  contact
day
## 1  30  unemployed married   primary      no    1787      no   no cellular
19
## 2  33    services married secondary      no    4789     yes  yes cellular
11
## 3  35  management  single  tertiary      no    1350     yes   no cellular
16
## 4  30  management married  tertiary      no    1476     yes  yes  unknown
3
## 5  59 blue-collar married secondary      no       0     yes   no  unknown
5
## 6  35  management  single  tertiary      no     747      no   no cellular
23
##   month duration campaign pdays previous poutcome  y
## 1   oct       79        1    -1        0  unknown no
## 2   may      220        1   339        4  failure no
## 3   apr      185        1   330        1  failure no
## 4   jun      199        4    -1        0  unknown no
## 5   may      226        1    -1        0  unknown no
## 6   feb      141        2   176        3  failure no
```

```r
############################
## Data Preparation ########
############################

any(is.na(bank))
```

```
## [1] FALSE
```

```r
# There are no missing values in the data set.

colnames(bank)
```

```
##  [1] "age"       "job"       "marital"   "education" "default"   "balance"
##  [7] "housing"   "loan"      "contact"   "day"       "month"
"duration"
## [13] "campaign"  "pdays"     "previous"  "poutcome"  "y"
```

```r
glimpse(bank)
```

```
## Observations: 4,521
## Variables: 17
```

```
## Registered S3 method overwritten by 'cli':
##   method     from
##   print.tree tree
```

```
## $ age       <int> 30, 33, 35, 30, 59, 35, 36, 39, 41, 43, 39, 43, 36, 20,
31,…
## $ job       <fct> unemployed, services, management, management, blue-
collar, …
## $ marital   <fct> married, married, single, married, married, single,
married…
## $ education <fct> primary, secondary, tertiary, tertiary, secondary,
tertiary…
## $ default   <fct> no, no, no, no, no, no, no, no, no, no, no, no, no, no,
no,…
## $ balance   <int> 1787, 4789, 1350, 1476, 0, 747, 307, 147, 221, -88,
9374, 2…
## $ housing   <fct> no, yes, yes, yes, yes, no, yes, yes, yes, yes, yes,
yes, n…
## $ loan      <fct> no, yes, no, yes, no, no, no, no, no, yes, no, no, no,
no, …
## $ contact   <fct> cellular, cellular, cellular, unknown, unknown,
cellular, c…
## $ day       <int> 19, 11, 16, 3, 5, 23, 14, 6, 14, 17, 20, 17, 13, 30, 29,
29…
## $ month     <fct> oct, may, apr, jun, may, feb, may, may, may, apr, may,
apr,…
## $ duration  <int> 79, 220, 185, 199, 226, 141, 341, 151, 57, 313, 273,
113, 3…
## $ campaign  <int> 1, 1, 1, 4, 1, 2, 1, 2, 2, 1, 1, 2, 2, 1, 1, 2, 5, 1, 1,
1,…
## $ pdays     <int> -1, 339, 330, -1, -1, 176, 330, -1, -1, 147, -1, -1, -1,
-1…
## $ previous  <int> 0, 4, 1, 0, 0, 3, 2, 0, 0, 2, 0, 0, 0, 0, 1, 0, 0, 2, 0,
1,…
## $ poutcome  <fct> unknown, failure, failure, unknown, unknown, failure,
other…
## $ y         <fct> no, no, no, no, no, no, no, no, no, no, no, no, no, no, yes,
no…
```

```r
summary(bank)
```

```
##       age                 job          marital         education
default
##  Min.   :19.00   management :969   divorced: 528   primary  : 678   no
:4445
##  1st Qu.:33.00   blue-collar:946   married :2797   secondary:2306   yes:
76
##  Median :39.00   technician :768   single  :1196   tertiary :1350
##  Mean   :41.17   admin.     :478                   unknown  : 187
##  3rd Qu.:49.00   services   :417
##  Max.   :87.00   retired    :230
##                  (Other)    :713
##     balance      housing     loan           contact          day
##  Min.   :-3313   no :1962   no :3830   cellular :2896   Min.   : 1.00
```

```
##  1st Qu.:   69   yes:2559   yes: 691   telephone: 301   1st Qu.: 9.00
##  Median :  444                         unknown  :1324   Median :16.00
##  Mean   : 1423                                          Mean   :15.92
##  3rd Qu.: 1480                                          3rd Qu.:21.00
##  Max.   :71188                                          Max.   :31.00
##
##      month         duration       campaign          pdays
##  may    :1398   Min.   :   4   Min.   : 1.000   Min.   : -1.00
##  jul    : 706   1st Qu.: 104   1st Qu.: 1.000   1st Qu.: -1.00
##  aug    : 633   Median : 185   Median : 2.000   Median : -1.00
##  jun    : 531   Mean   : 264   Mean   : 2.794   Mean   : 39.77
##  nov    : 389   3rd Qu.: 329   3rd Qu.: 3.000   3rd Qu.: -1.00
##  apr    : 293   Max.   :3025   Max.   :50.000   Max.   :871.00
##  (Other): 571
##     previous          poutcome      y
##  Min.   : 0.0000   failure: 490   no :4000
##  1st Qu.: 0.0000   other  : 197   yes: 521
##  Median : 0.0000   success: 129
##  Mean   : 0.5426   unknown:3705
##  3rd Qu.: 0.0000
##  Max.   :25.0000
##
```

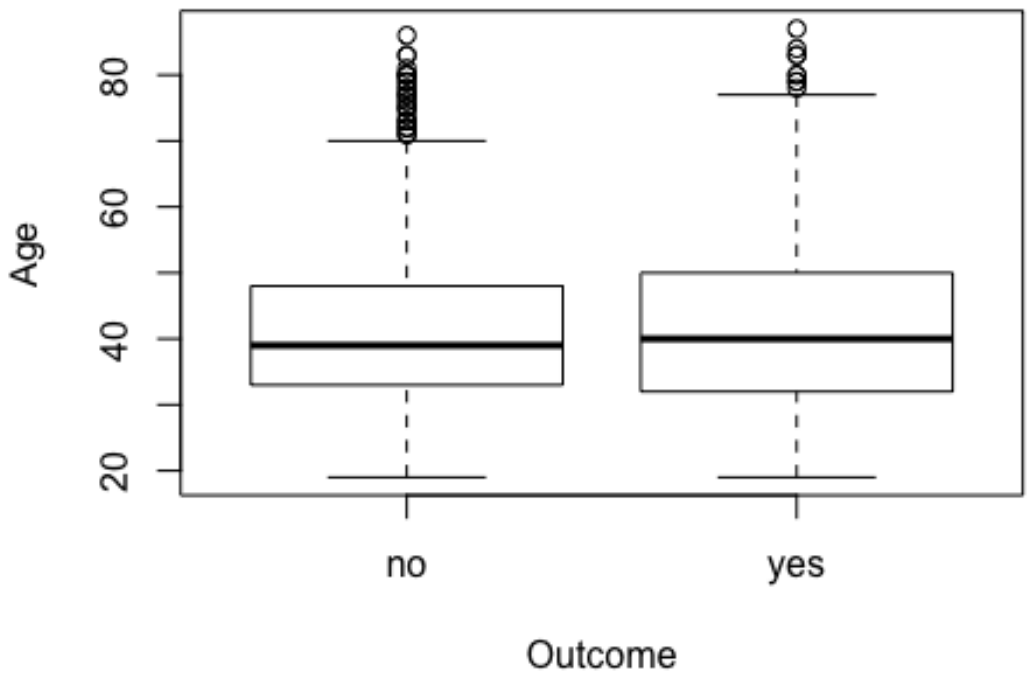
```
# The following variables need to be removed from the dataset as they are not
useful
# for analysis purpose :
# pdays, previous, poutcome, durartion

col_drop = c("pdays", "previous", "poutcome", "duration")
cleaned_df = bank[,! (names(bank) %in% col_drop)]

summary(cleaned_df)
```

```
##       age                 job          marital         education
default
##  Min.   :19.00   management :969   divorced: 528   primary  : 678   no
:4445
##  1st Qu.:33.00   blue-collar:946   married :2797   secondary:2306   yes:
76
##  Median :39.00   technician :768   single  :1196   tertiary :1350
##  Mean   :41.17   admin.     :478                   unknown  : 187
##  3rd Qu.:49.00   services   :417
##  Max.   :87.00   retired    :230
##                  (Other)    :713
##     balance      housing     loan           contact          day
##  Min.   :-3313   no :1962   no :3830   cellular :2896   Min.   : 1.00
##  1st Qu.:   69   yes:2559   yes: 691   telephone: 301   1st Qu.: 9.00
##  Median :  444                         unknown  :1324   Median :16.00
##  Mean   : 1423                                          Mean   :15.92
##  3rd Qu.: 1480                                          3rd Qu.:21.00
```

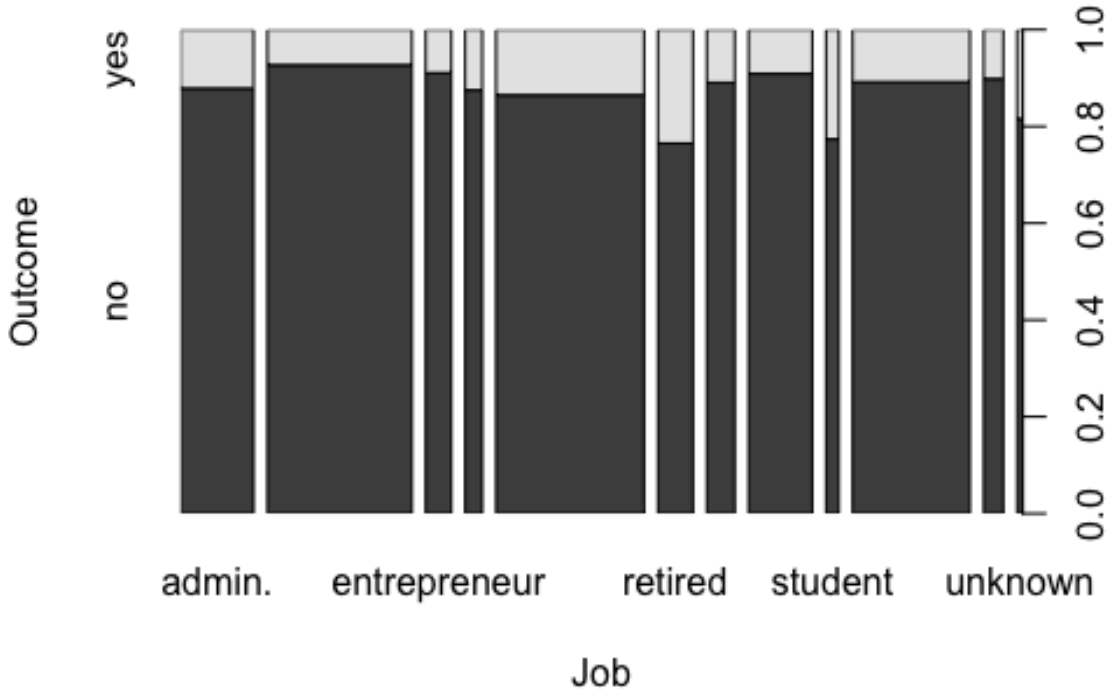
```
##  Max.   :71188                                                Max.   :31.00
##
##      month         campaign        y
##  may    :1398   Min.   : 1.000   no :4000
##  jul    : 706   1st Qu.: 1.000   yes: 521
##  aug    : 633   Median : 2.000
##  jun    : 531   Mean   : 2.794
##  nov    : 389   3rd Qu.: 3.000
##  apr    : 293   Max.   :50.000
##  (Other): 571
```

```
########################################## EDA
####################################
```

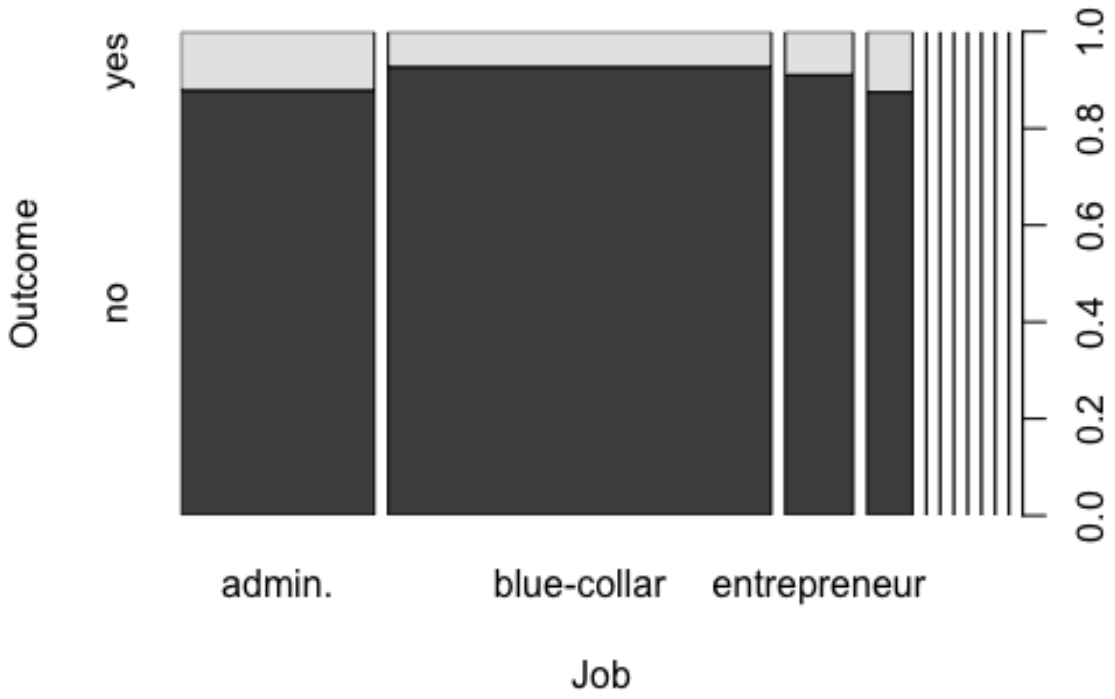
```
plot(cleaned_df$y, cleaned_df$age, xlab = "Outcome", ylab = "Age")
```



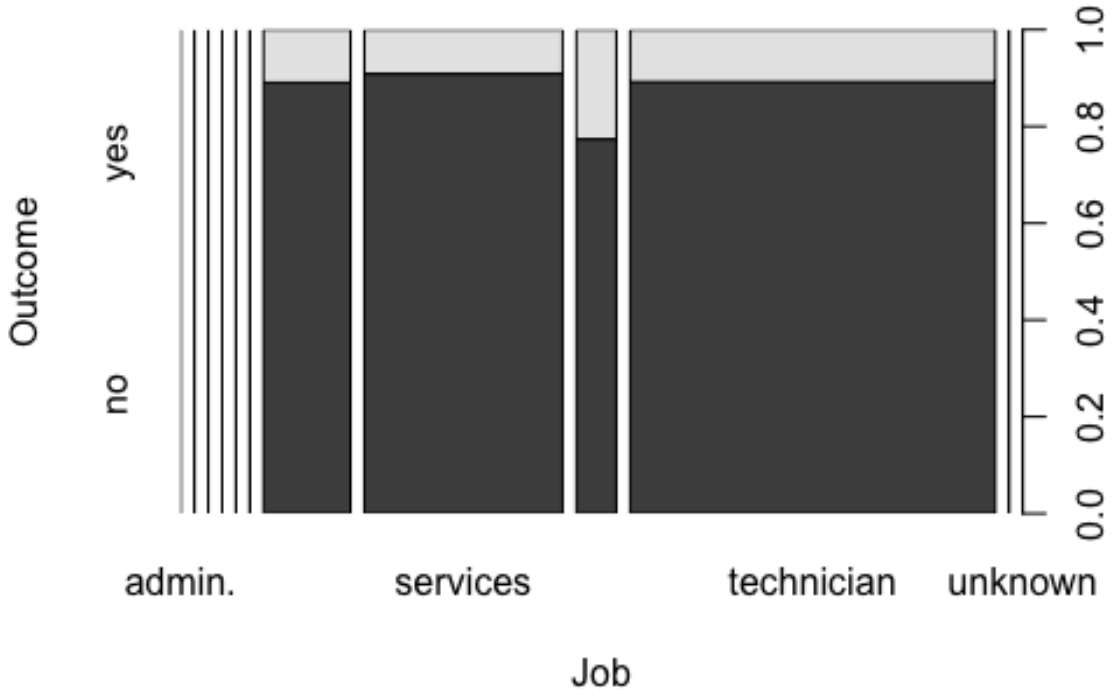

```
spineplot(y~job, data = cleaned_df, xlab = "Job", ylab = "Outcome")
```



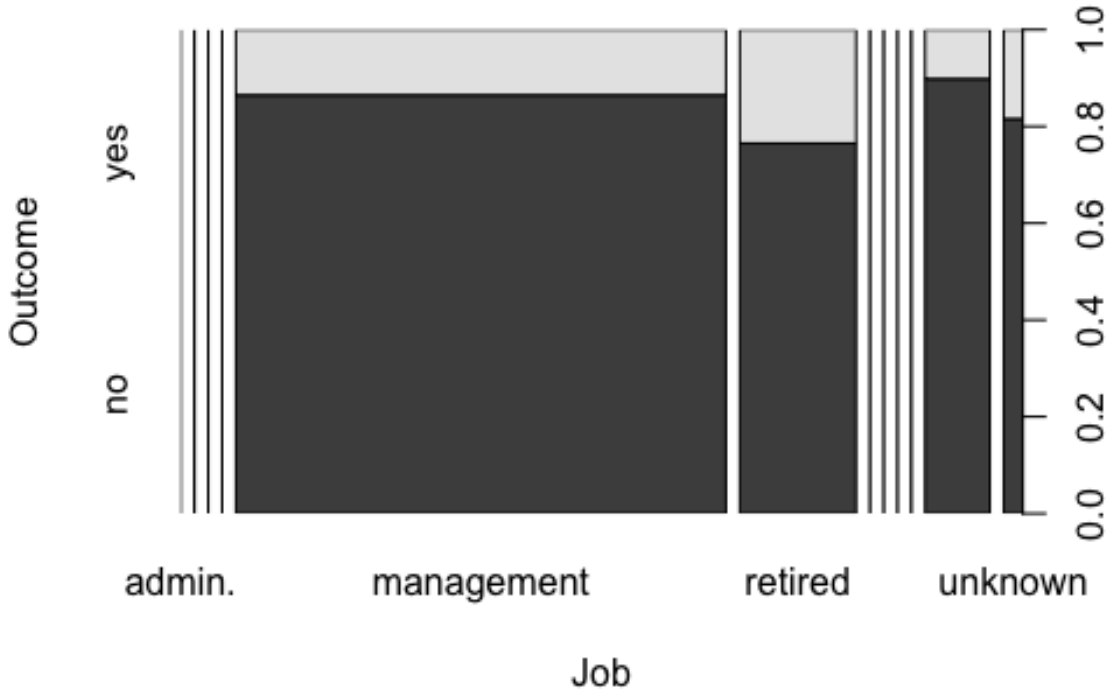

```
job_cat1 = c("admin.","blue-collar","entrepreneur","housemaid")
job_cat2 = c("self-employed","services","student","technician")
job_cat3 = c("management","retired","unemployed","unknown")
spineplot(y~job, data = cleaned_df[(cleaned_df$job %in% job_cat1),], xlab =
"Job", ylab = "Outcome")
```



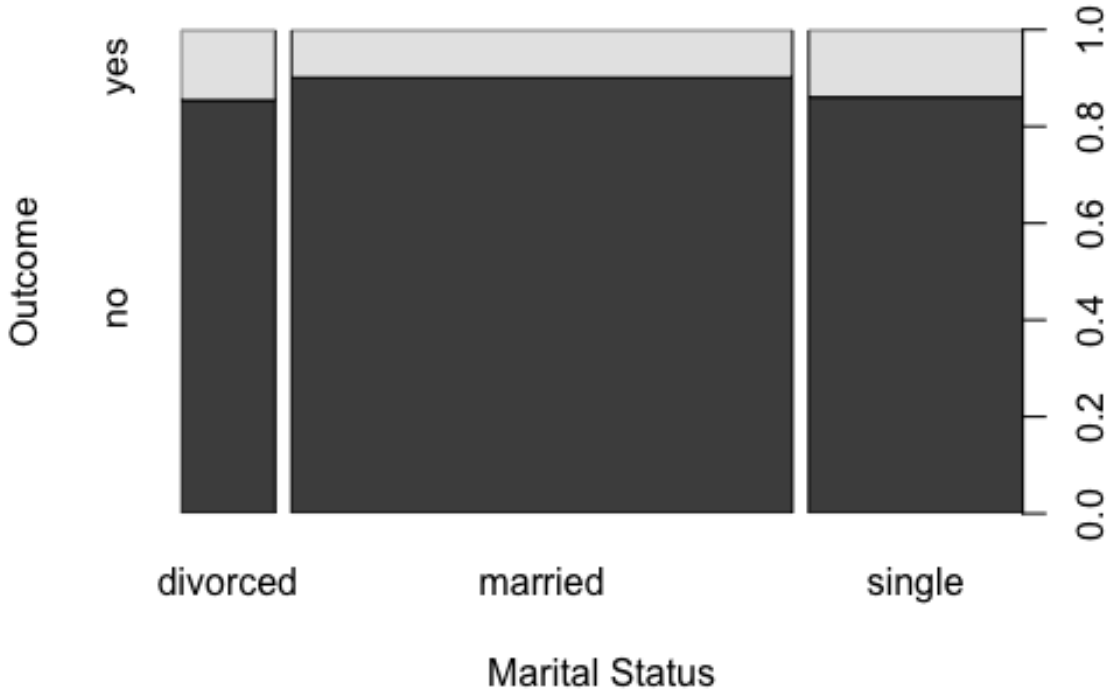

```r
spineplot(y~job, data = cleaned_df[(cleaned_df$job %in% job_cat2),], xlab =
"Job", ylab = "Outcome")
```



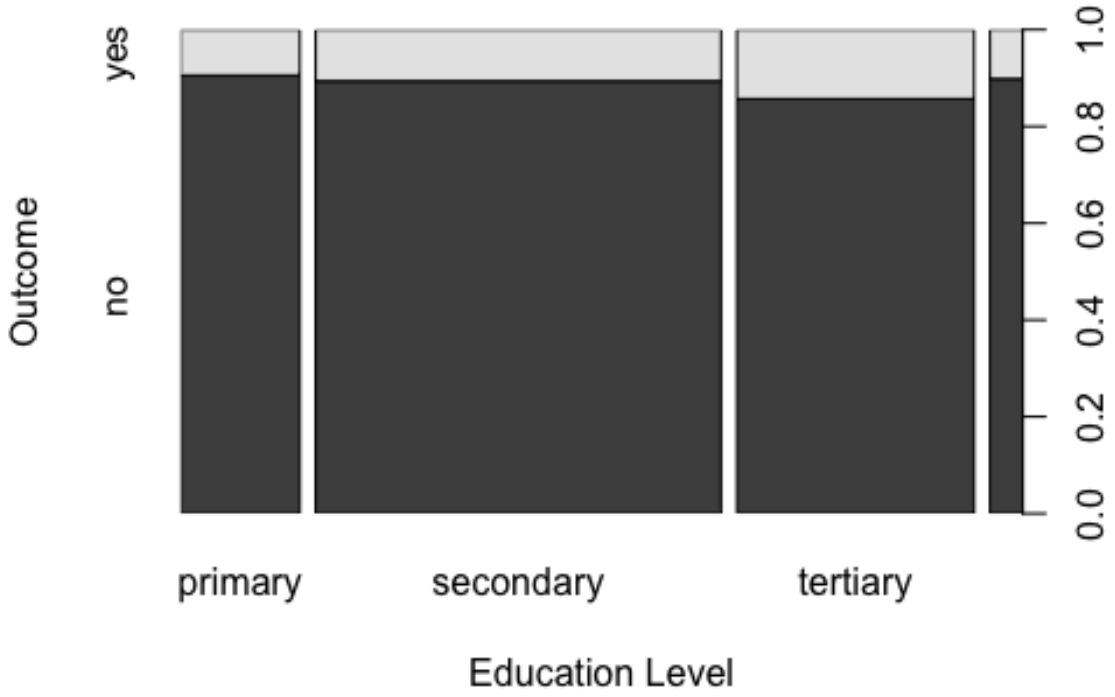

```r
spineplot(y~job, data = cleaned_df[(cleaned_df$job %in% job_cat3),], xlab =
"Job", ylab = "Outcome")
```




```
# martital status
table(cleaned_df$marital,cleaned_df$y)

##
##               no  yes
##   divorced  451   77
##   married  2520  277
##   single   1029  167

spineplot(y~marital, data = cleaned_df, xlab = "Marital Status", ylab =
"Outcome")
```

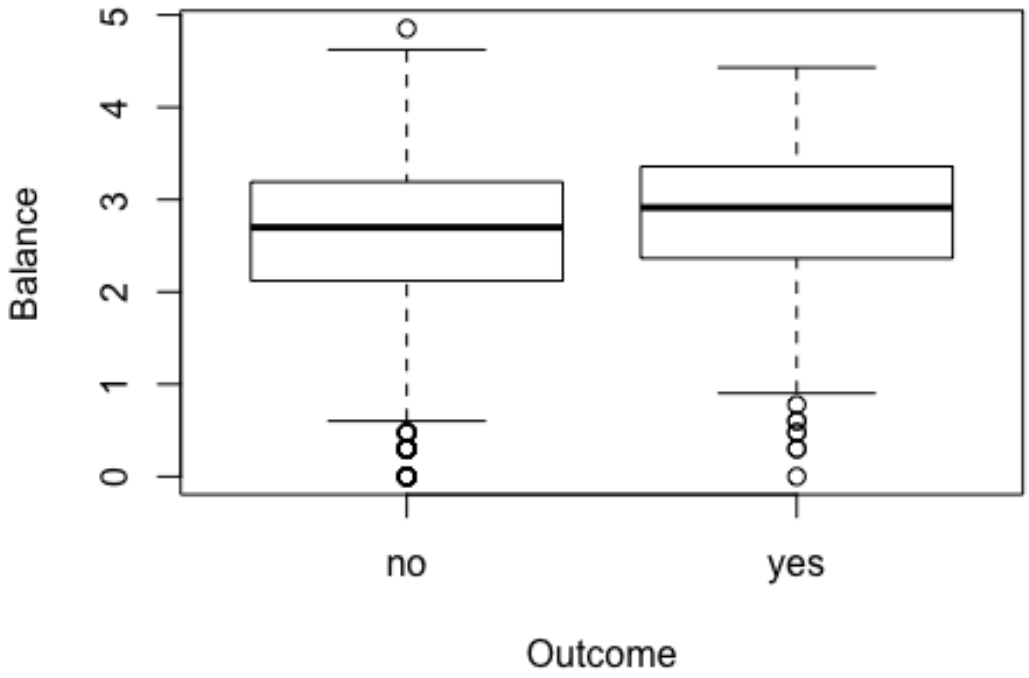



```
# education
table(cleaned_df$education,cleaned_df$y)

##
##                no  yes
##   primary     614   64
##   secondary  2061  245
##   tertiary   1157  193
##   unknown     168   19

spineplot(y~education, data = cleaned_df, xlab = "Education Level", ylab =
"Outcome")
```

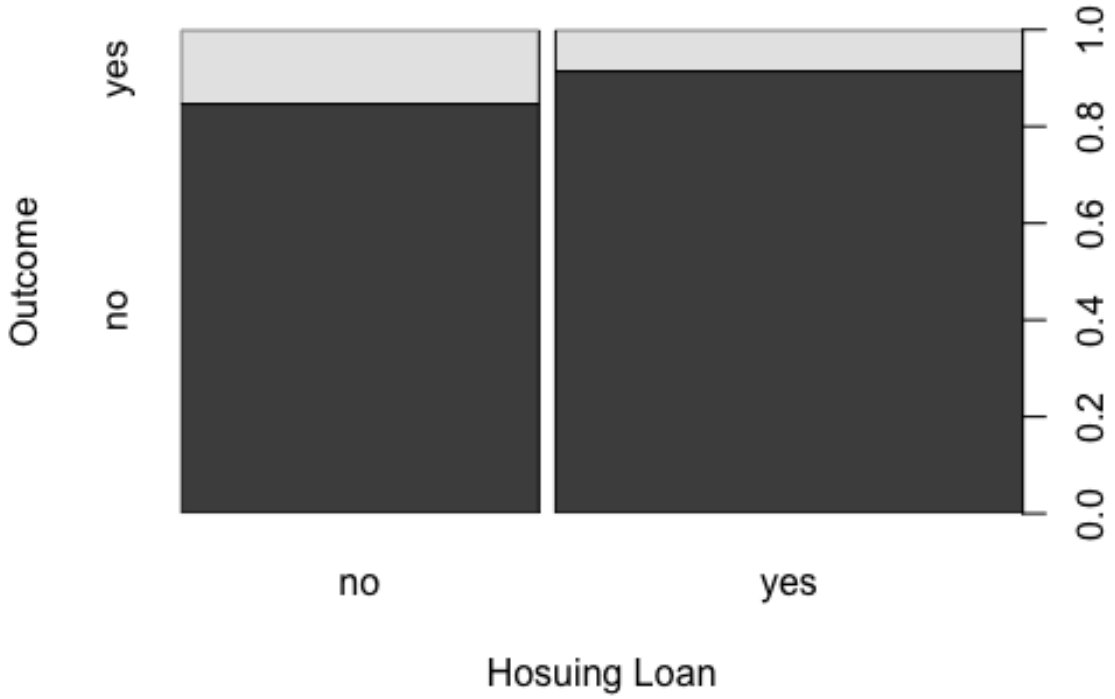



```r
# default
table(cleaned_df$default,cleaned_df$y)

##
##          no  yes
##   no  3933  512
##   yes   67    9

spineplot(y~default, data = cleaned_df, xlab = "Default", ylab = "Outcome")
```

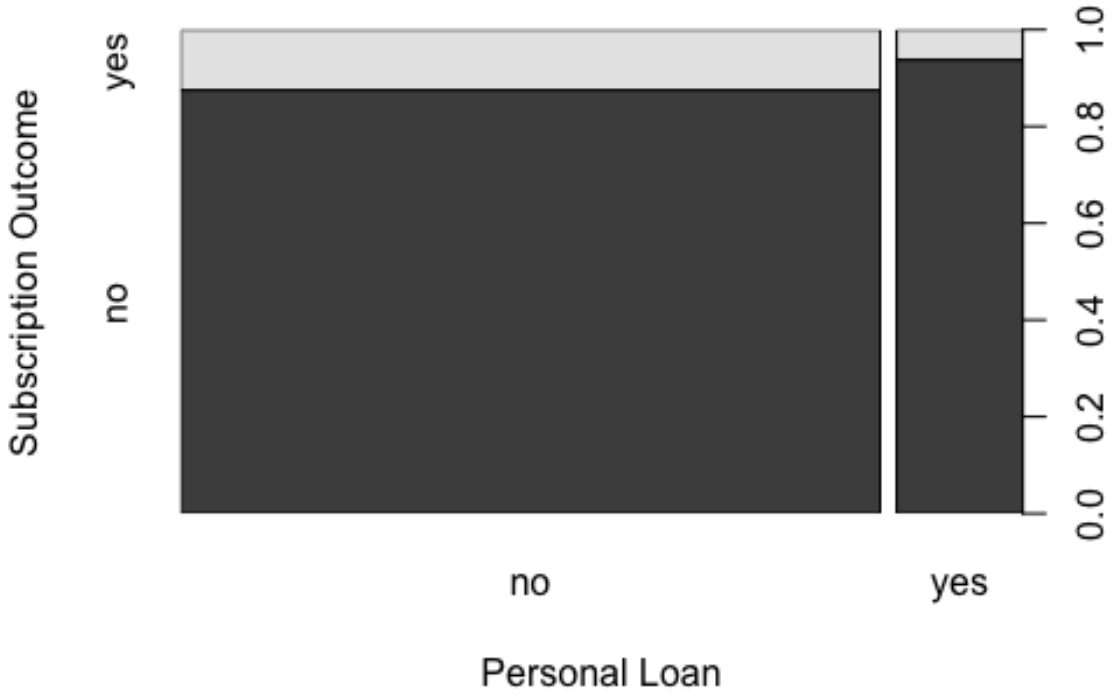

```
# bank balance
plot(cleaned_df$y, log10(cleaned_df$balance), xlab = "Outcome", ylab =
"Balance")

## Warning in is.factor(y): NaNs produced

## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out =
z$out[z$group
## == : Outlier (-Inf) in boxplot 1 is not drawn

## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out =
z$out[z$group
## == : Outlier (-Inf) in boxplot 2 is not drawn
```



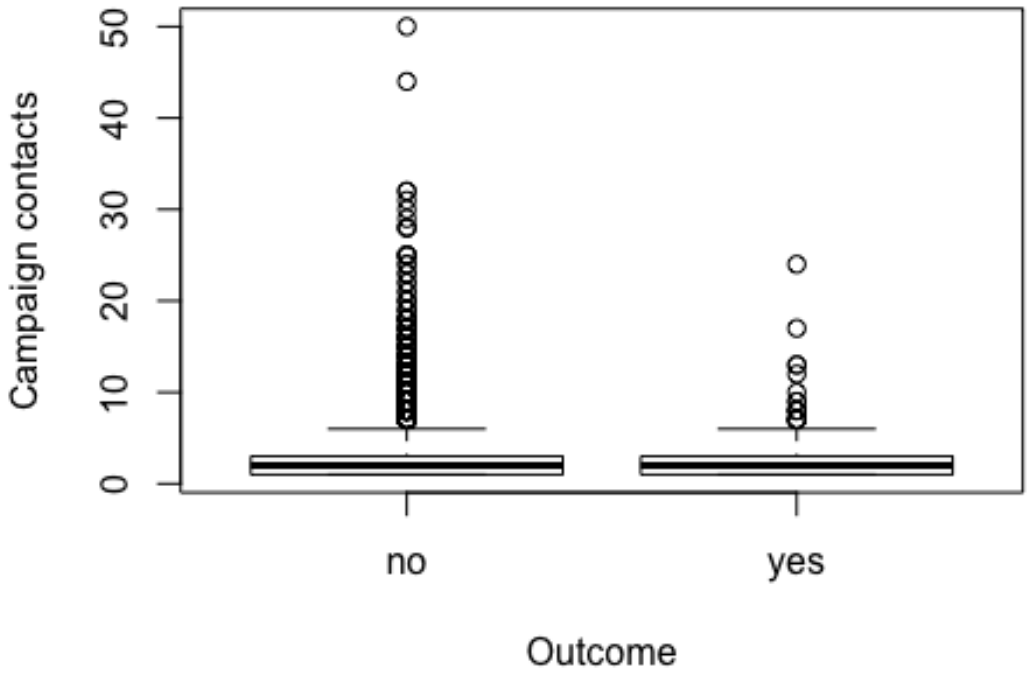

```
summary(cleaned_df[cleaned_df$y=="yes",]$balance)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   -1206     171     710    1572    2160   26965

summary(cleaned_df[cleaned_df$y=="no",]$balance)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -3313.0    61.0   419.5  1403.2  1407.0 71188.0

# housing loan
summary(cleaned_df$housing)

##   no  yes
## 1962 2559

table(cleaned_df$housing,cleaned_df$y)

##
##         no  yes
##   no  1661  301
##   yes 2339  220

spineplot(y~housing, data = cleaned_df, xlab = "Hosuing Loan", ylab =
"Outcome")
```




```
# personal loan
summary(cleaned_df$loan)

##   no  yes
## 3830  691

table(cleaned_df$loan,cleaned_df$y)

##
##         no  yes
##   no  3352  478
##   yes  648   43

spineplot(y~loan, data = cleaned_df, xlab = "Personal Loan", ylab =
"Subscription Outcome")
```




```
# number of contacts performed
summary(cleaned_df$campaign)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   2.000   2.794   3.000  50.000

plot(cleaned_df$y, cleaned_df$campaign, xlab = "Outcome", ylab = "Campaign
contacts")
```




```
summary(cleaned_df[cleaned_df$y=="yes",]$campaign)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   2.000   2.267   3.000  24.000

summary(cleaned_df[cleaned_df$y=="no",]$campaign)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   2.000   2.862   3.000  50.000

############################
## Train / Test Split ######
############################

set.seed(-1)
train = sample(1:nrow(cleaned_df), 3164)

############################
## Modeling ###############
############################


# Modifying Training dataset - imbalanced dataset
```

```r
modified_training_data <- ROSE(y~., data = cleaned_df[train,], seed = 1)$data
table(modified_training_data$y)
```

```
##
##   no  yes
## 1642 1522
```

```r
# Logistic Regression 2
lg_fit <- glm(y~., data = modified_training_data, family = binomial)
lg_prob = predict(lg_fit, newdata = cleaned_df[-train,], type="response")
lg_pred = ifelse(lg_prob>0.5, "yes","no")
actual = cleaned_df[-train,]$y
mean(lg_pred==actual)
```

```
## [1] 0.6978629
```

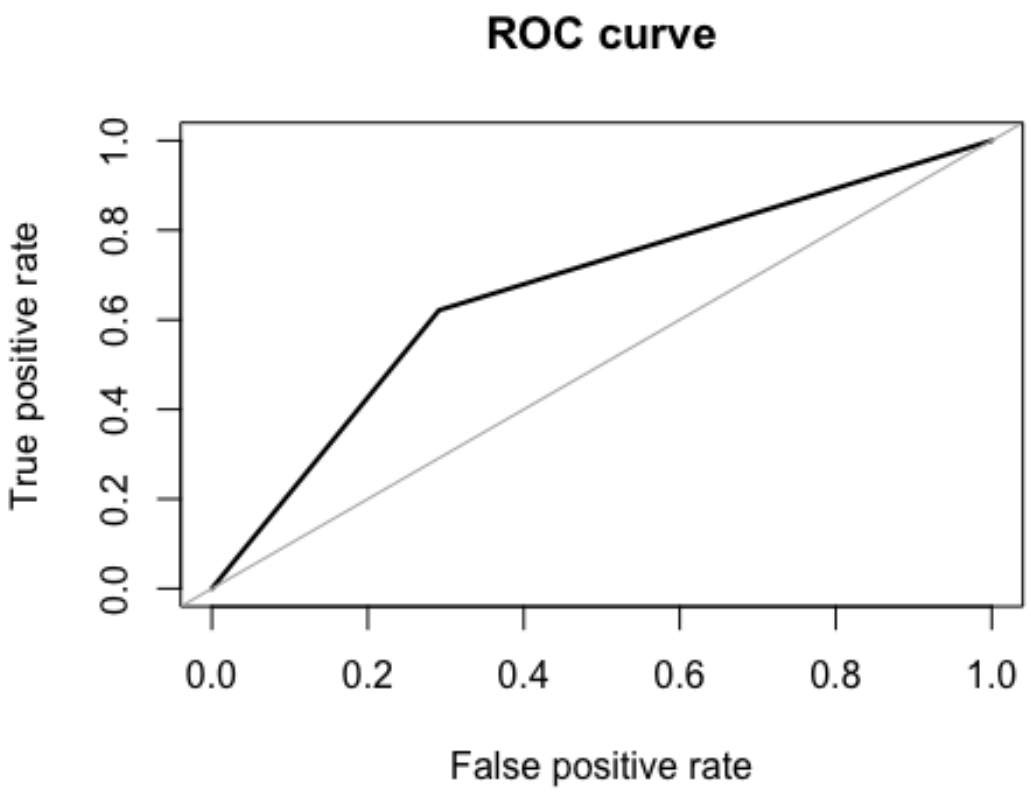
```r
confusion_matrix1 <- table(lg_pred, actual)
confusion_matrix1
```

```
##        actual
## lg_pred  no yes
##     no  842  64
##     yes 346 105
```

```r
cat("Accuracy of Logistic Regression : ",((confusion_matrix1[1,"no"] +
confusion_matrix1[2,"yes"])/1357),"\n")
```

```
## Accuracy of Logistic Regression :  0.6978629
```

```r
roc_1 = roc.curve(cleaned_df[-train,]$y, lg_pred, plotit = TRUE)
```




```
roc_1
```

```
## Area under the curve (AUC): 0.665
```

```
# Classification Tree

tree_fit2 <- rpart(y~., method = "class", data = modified_training_data,
control = rpart.control(maxdepth = 20, cp=0.0026281))
#summary(tree_fit2)
printcp(tree_fit2)
```

```
##
## Classification tree:
## rpart(formula = y ~ ., data = modified_training_data, method = "class",
##     control = rpart.control(maxdepth = 20, cp = 0.0026281))
##
## Variables actually used in tree construction:
##  [1] age       balance   campaign  contact   day       education housing
##  [8] job       loan      marital   month
##
## Root node error: 1522/3164 = 0.48104
##
## n= 3164
##
```

```
##           CP nsplit rel error  xerror     xstd
## 1  0.1294350      0   1.00000 1.00000 0.018465
## 2  0.0998686      1   0.87057 0.90604 0.018326
## 3  0.0167543      2   0.77070 0.77070 0.017851
## 4  0.0067893      6   0.69120 0.72142 0.017593
## 5  0.0056943     13   0.63666 0.73062 0.017644
## 6  0.0054753     17   0.61235 0.71156 0.017535
## 7  0.0052562     20   0.59593 0.71419 0.017551
## 8  0.0045992     23   0.58016 0.70565 0.017500
## 9  0.0041612     27   0.56176 0.68003 0.017339
## 10 0.0036137     33   0.53679 0.67280 0.017291
## 11 0.0035480     39   0.50986 0.65769 0.017187
## 12 0.0035042     46   0.47766 0.65769 0.017187
## 13 0.0032852     50   0.46058 0.65703 0.017183
## 14 0.0029566     55   0.44415 0.64389 0.017089
## 15 0.0026281     57   0.43824 0.62286 0.016930
## 16 0.0026281     60   0.43035 0.61235 0.016847
```

```
plot(tree_fit2, uniform = TRUE)
text(tree_fit2, all=TRUE, cex=0.75, splits=TRUE, use.n=TRUE, xpd = TRUE)
```


```
library(maptree)

## Loading required package: cluster
```

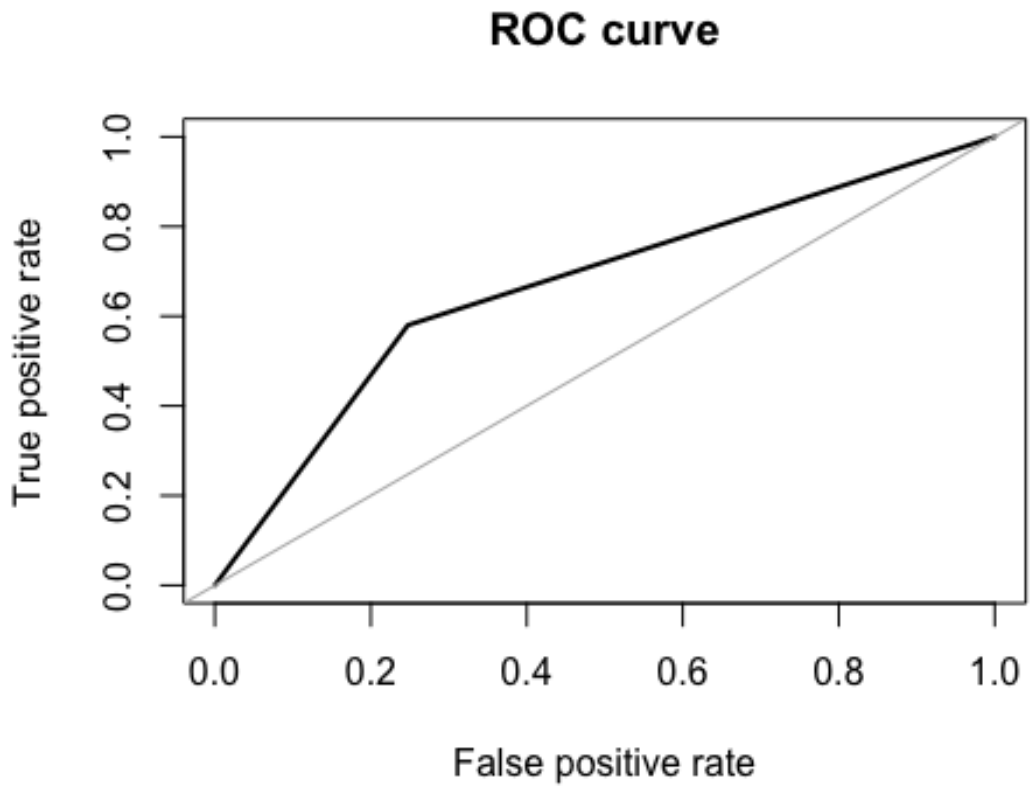
```
tree_pred_2 = predict(tree_fit2, cleaned_df[-train,], type="class")
confusion_matrix2 <- table(tree_pred_2, actual = cleaned_df[-train,]$y)
confusion_matrix2
```

```
##            actual
## tree_pred_2  no yes
##         no  894  71
##         yes 294  98
```

```
cat("Accuracy of CT  : ",((confusion_matrix2[1,"no"] +
confusion_matrix2[2,"yes"])/1357),"\n" )
```

```
## Accuracy of CT  :  0.7310243
```

```
roc_2 =roc.curve(cleaned_df[-train,]$y, tree_pred_2, plotit = TRUE)
```



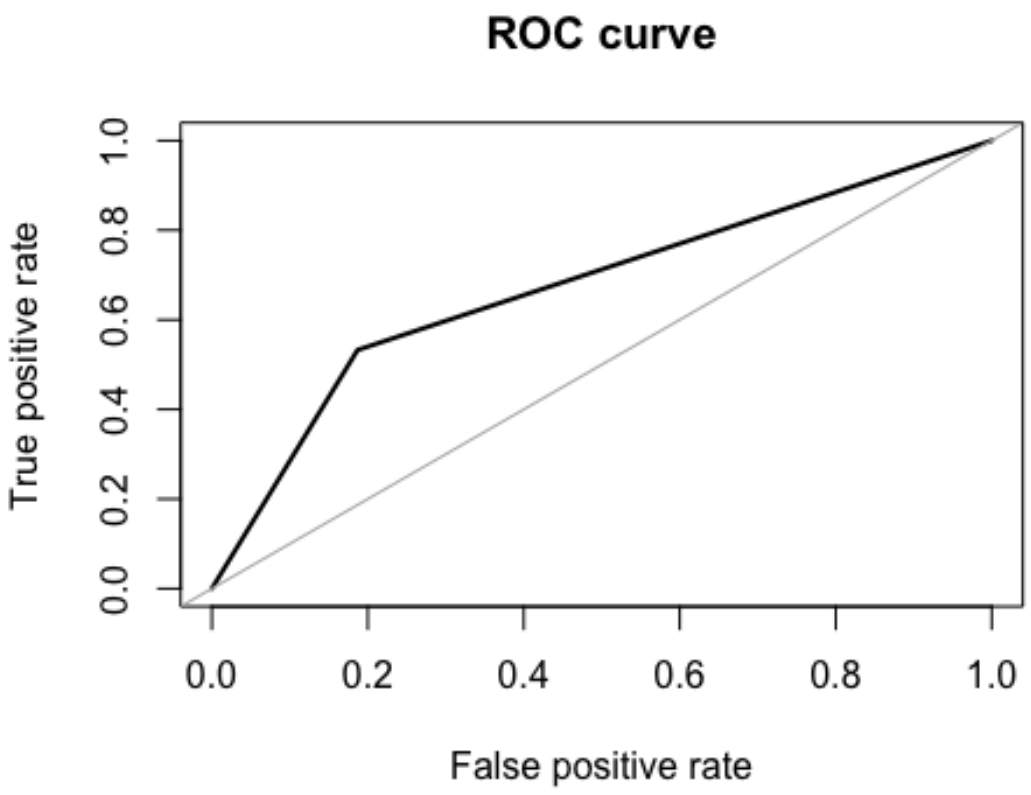

```
roc_2
```

```
## Area under the curve (AUC): 0.666
```

```
# Random Forests

library(randomForest)
```

```
## randomForest 4.6-14
```

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

##
## Attaching package: 'randomForest'

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

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

set.seed(0)
rf_fit <- randomForest(y~., data = modified_training_data, ntree = 500)
rf_fit

##
## Call:
##  randomForest(formula = y ~ ., data = modified_training_data,      ntree =
500)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 16.43%
## Confusion matrix:
##       no  yes class.error
## no  1358  284   0.1729598
## yes  236 1286   0.1550591

confusion_matrix3 <- table( predicted = predict(rf_fit, newdata =
cleaned_df[-train,], type = "class"),
                            actual = cleaned_df[-train,]$y)
confusion_matrix3

##          actual
## predicted  no yes
##       no  966  79
##       yes 222  90

cat("Accuracy of RF  : ",((confusion_matrix3[1,"no"] +
confusion_matrix3[2,"yes"])/1357),"\n" )

## Accuracy of RF  :  0.7781872

roc_3 = roc.curve(cleaned_df[-train,]$y, predict(rf_fit, newdata =
cleaned_df[-train,], type = "class"), plotit = TRUE)
```




```
roc_3
```

```
## Area under the curve (AUC): 0.673
```

```
#################################  RESULTS
##########################################
```

```
cat("\n\n Model Performance : \n\n")
```

```
##
##
##  Model Performance :
```

```
cat("AUC of Logistic Regression : ", roc_1$auc,"\n")
```

```
## AUC of Logistic Regression :  0.665028
```

```
cat("AUC of Classification Tree : ", roc_2$auc,"\n")
```

```
## AUC of Classification Tree :  0.6662035
```

```
cat("AUC of RF : ", roc_3$auc,"\n")
```

```
## AUC of RF :  0.6728378
```