Part3.R

Cheng Jun

2021-03-26

library(caTools)

## Warning: package 'caTools' was built under R version 4.0.4

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.0.4

library(randomForest)

## Warning: package 'randomForest' was built under R version 4.0.4

## 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

library(caret)

## Warning: package 'caret' was built under R version 4.0.4

## Loading required package: lattice

## Warning: package 'lattice' was built under R version 4.0.4

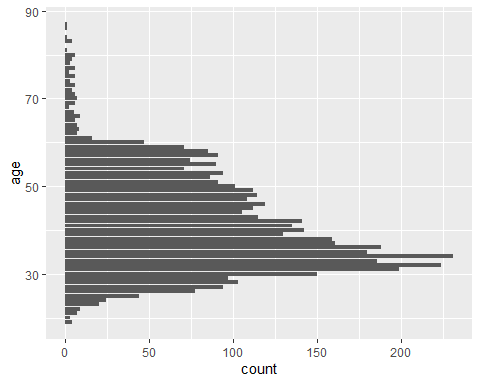
setwd("C:/Users/Cheng Jun/Desktop/SIM/Year 2/Machine Learning/Coursework")  
  
data <- read.table("bank.csv",header=1,sep = ';')  
sum(is.na(data))

## [1] 0

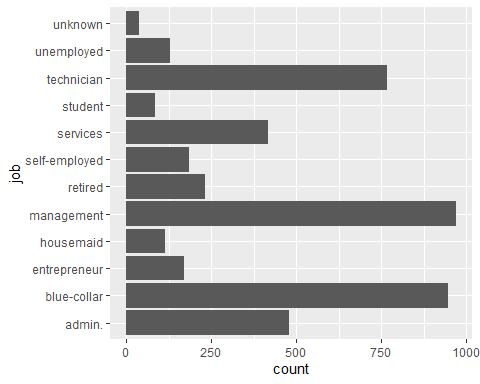
columns = names(data)  
columns

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

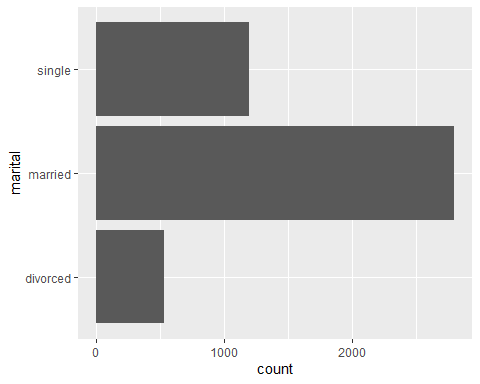
#Illustration of data  
#plot age row  
ggplot(data) + geom\_bar(aes(y = age))



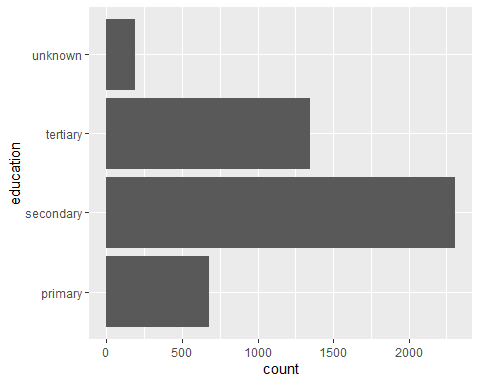
#plot job row  
ggplot(data) + geom\_bar(aes(y = job))



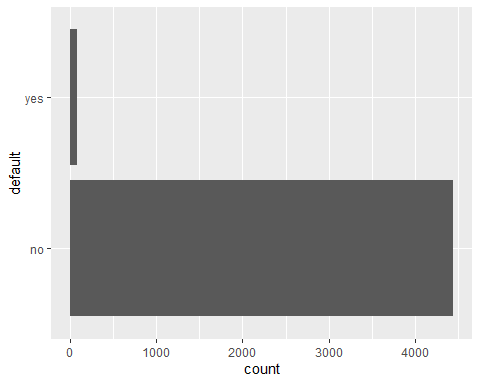
#plot marital row  
ggplot(data) + geom\_bar(aes(y = marital))



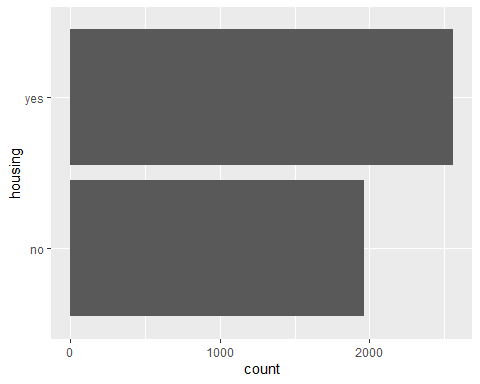
#plot education row  
ggplot(data) + geom\_bar(aes(y = education))



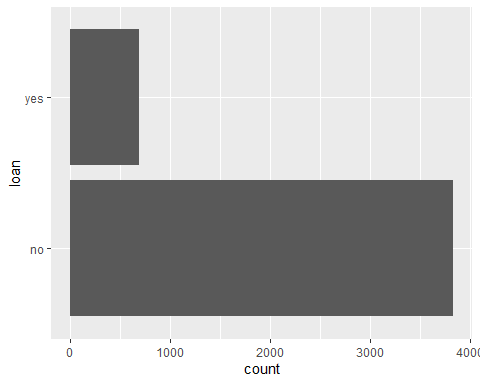
#plot default row  
ggplot(data) + geom\_bar(aes(y = default))



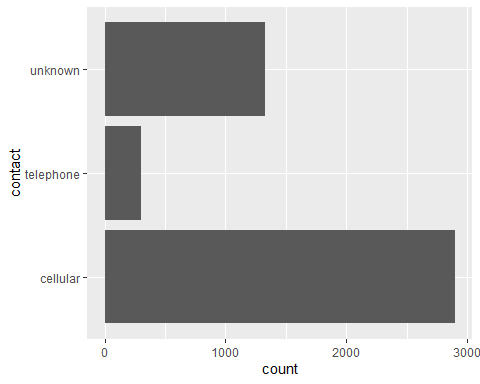
#As default has mostly 'no' value if we place it in the dataframe it might result in overfitting due to it not being equally distributed.   
data = data[,-c(5)]   
#plot housing row  
ggplot(data) + geom\_bar(aes(y = housing))



#plot loan row  
ggplot(data) + geom\_bar(aes(y = loan))



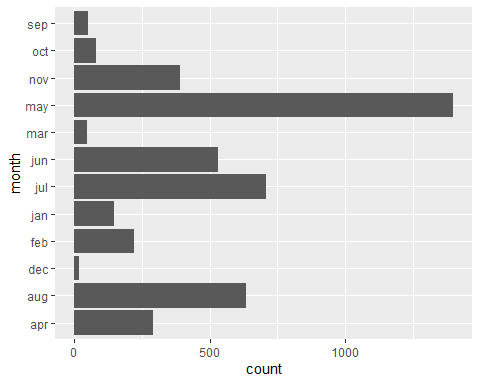
#plot contact row  
ggplot(data) + geom\_bar(aes(y = contact))



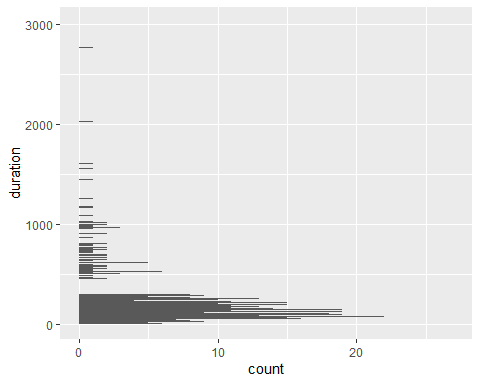
#plot day row  
ggplot(data) + geom\_bar(aes(y =day))



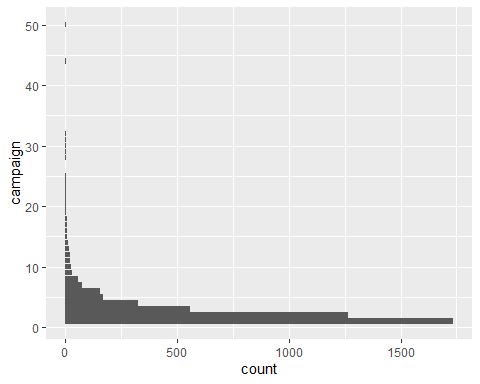
#plot month row  
ggplot(data) + geom\_bar(aes(y = month))



#plot duration row  
ggplot(data) + geom\_bar(aes(y =duration))



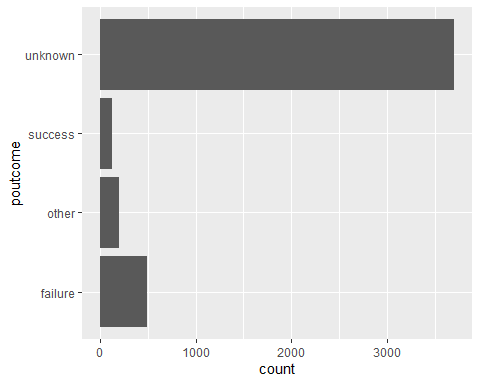
#plot campaign row  
ggplot(data) + geom\_bar(aes(y =campaign))



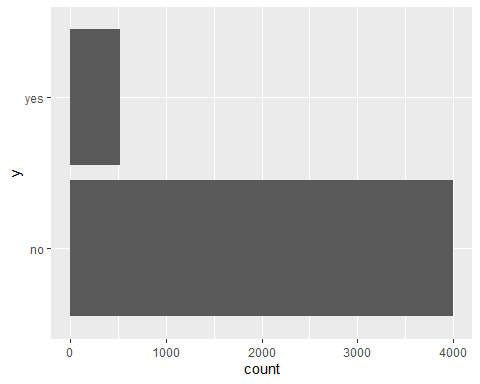
#for pdays row  
ggplot(data, aes(x = pdays)) + geom\_histogram(fill="grey", position="dodge",binwidth=100)



data$pdays[data$pdays==-1] = mean(data$pdays)  
#plot poutcome row  
ggplot(data) + geom\_bar(aes(y = poutcome))



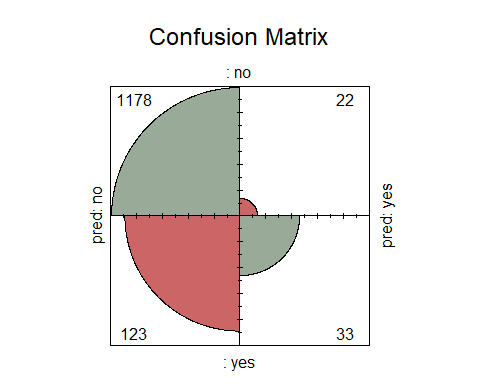
#As poutcome also mostly has unknown only, so we remove it to prevent overfitting from occuring  
data = data[,-c(15)]  
#plot y row   
ggplot(data) + geom\_bar(aes(y =y))



#for balance row  
data$balance[data$balance==0] = mean(data$balance)  
  
data <- transform(  
 data,  
 age = age,  
 job = as.integer(factor(job)),  
 marital = as.integer(factor(marital)),  
 education=as.integer(factor(education)),  
 balance = balance,  
 housing = as.integer(factor(housing)),  
 loan = as.integer(factor(loan)),  
 contact = as.integer(factor(contact)),  
 day = day,  
 month = as.integer(factor(month)),  
 duration = duration,  
 campaign = campaign,  
 pdays = pdays  
)  
sapply(data, class)

## age job marital education balance housing   
## "integer" "integer" "integer" "integer" "numeric" "integer"   
## loan contact day month duration campaign   
## "integer" "integer" "integer" "integer" "integer" "integer"   
## pdays previous y   
## "numeric" "integer" "character"

#train test split  
split = sample.split(data$y,SplitRatio=0.7)  
training = subset(data,split==TRUE)  
test = subset(data,split==FALSE)  
  
#scaling the data  
training[,c(1:14)] = scale(training[,c(1:14)])  
  
#random forest  
rf <- randomForest(  
 as.factor(y) ~ .,  
 data=training,  
)  
  
test[,c(1:14)] = scale(test[,c(1:14)])  
  
pred = predict(rf, newdata=test[,c(1:14)])  
cm = table(test[,15], pred)  
#confusion matrix  
fourfoldplot(cm, color = c("#CC6666", "#99AA99"),conf.level = 0, margin =1, main = "Confusion Matrix")



res = confusionMatrix(cm)  
  
print(res)

## Confusion Matrix and Statistics  
##   
## pred  
## no yes  
## no 1178 22  
## yes 123 33  
##   
## Accuracy : 0.8931   
## 95% CI : (0.8754, 0.909)  
## No Information Rate : 0.9594   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.269   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.9055   
## Specificity : 0.6000   
## Pos Pred Value : 0.9817   
## Neg Pred Value : 0.2115   
## Prevalence : 0.9594   
## Detection Rate : 0.8687   
## Detection Prevalence : 0.8850   
## Balanced Accuracy : 0.7527   
##   
## 'Positive' Class : no   
##

Precisionvalue = res$byClass["Pos Pred Value"]  
print(Precisionvalue)

## Pos Pred Value   
## 0.9816667

Accuracy=res$overall["Accuracy"]  
print(Accuracy)

## Accuracy   
## 0.8930678