Preprocesamiento.R

nicolasacevedo

2019-12-08

library(readr)  
library(Hmisc)

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##   
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':  
##   
## format.pval, units

library(dummies)

## dummies-1.5.6 provided by Decision Patterns

library(corrplot)

## corrplot 0.84 loaded

library(cluster)  
  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:Hmisc':  
##   
## src, summarize

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

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

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:dplyr':  
##   
## combine

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

library(pROC)

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

library(gbm)

## Loaded gbm 2.1.5

library(caret)

##   
## Attaching package: 'caret'

## The following object is masked from 'package:survival':  
##   
## cluster

library(e1071)

##   
## Attaching package: 'e1071'

## The following object is masked from 'package:Hmisc':  
##   
## impute

library(mlr)

## Loading required package: ParamHelpers

##   
## Attaching package: 'mlr'

## The following object is masked from 'package:e1071':  
##   
## impute

## The following object is masked from 'package:caret':  
##   
## train

## The following object is masked from 'package:Hmisc':  
##   
## impute

library(C50)  
  
library(leaps)  
library(pls)

##   
## Attaching package: 'pls'

## The following object is masked from 'package:mlr':  
##   
## crossval

## The following object is masked from 'package:caret':  
##   
## R2

## The following object is masked from 'package:corrplot':  
##   
## corrplot

## The following object is masked from 'package:stats':  
##   
## loadings

library(matrixStats)

##   
## Attaching package: 'matrixStats'

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

# Obtención ----  
Train <- read\_delim("Train.csv", ";", escape\_double = FALSE, trim\_ws = TRUE)

## Parsed with column specification:  
## cols(  
## Id = col\_double(),  
## Administrative = col\_double(),  
## Administrative\_Duration = col\_double(),  
## Informational = col\_double(),  
## Informational\_Duration = col\_double(),  
## ProductRelated = col\_double(),  
## ProductRelated\_Duration = col\_double(),  
## BounceRates = col\_double(),  
## ExitRates = col\_double(),  
## PageValues = col\_double(),  
## SpecialDay = col\_double(),  
## Month = col\_character(),  
## OperatingSystems = col\_double(),  
## Browser = col\_double(),  
## Region = col\_double(),  
## TrafficType = col\_double(),  
## VisitorType = col\_character(),  
## Weekend = col\_logical(),  
## Revenue = col\_double()  
## )

Test <- read\_delim("Test.csv", ";", escape\_double = FALSE, trim\_ws = TRUE)

## Parsed with column specification:  
## cols(  
## Id = col\_double(),  
## Administrative = col\_double(),  
## Administrative\_Duration = col\_double(),  
## Informational = col\_double(),  
## Informational\_Duration = col\_double(),  
## ProductRelated = col\_double(),  
## ProductRelated\_Duration = col\_double(),  
## BounceRates = col\_double(),  
## ExitRates = col\_double(),  
## PageValues = col\_double(),  
## SpecialDay = col\_double(),  
## Month = col\_character(),  
## OperatingSystems = col\_double(),  
## Browser = col\_double(),  
## Region = col\_double(),  
## TrafficType = col\_double(),  
## VisitorType = col\_character(),  
## Weekend = col\_logical(),  
## Revenue = col\_logical()  
## )

Train = as.data.frame(Train)  
Train$Id = NULL  
  
Test = as.data.frame(Test)  
Test$Id = NULL  
  
prep = rbind(Train,Test)  
index = c(901:1065)  
  
  
# Organizar datos ----   
  
names\_factor = c("Browser","TrafficType", "Region","VisitorType","Month", "OperatingSystems", "Revenue", "Weekend")  
prep = Train  
prep\_fin = Train  
prep\_fin[names\_factor] <- lapply(prep\_fin[,names\_factor], as.factor)   
prep\_fin = as.data.frame(prep\_fin)  
  
Train[names\_factor] <- lapply(prep\_fin[,names\_factor], as.factor)   
Train = as.data.frame(Train)  
  
  
Test[names\_factor] <- lapply(prep\_fin[,names\_factor], as.factor)

## Warning in `[<-.data.frame`(`\*tmp\*`, names\_factor, value = list(Browser =  
## structure(c(2L, : replacement element 1 has 900 rows to replace 165 rows

## Warning in `[<-.data.frame`(`\*tmp\*`, names\_factor, value = list(Browser =  
## structure(c(2L, : replacement element 2 has 900 rows to replace 165 rows

## Warning in `[<-.data.frame`(`\*tmp\*`, names\_factor, value = list(Browser =  
## structure(c(2L, : replacement element 3 has 900 rows to replace 165 rows

## Warning in `[<-.data.frame`(`\*tmp\*`, names\_factor, value = list(Browser =  
## structure(c(2L, : replacement element 4 has 900 rows to replace 165 rows

## Warning in `[<-.data.frame`(`\*tmp\*`, names\_factor, value = list(Browser =  
## structure(c(2L, : replacement element 5 has 900 rows to replace 165 rows

## Warning in `[<-.data.frame`(`\*tmp\*`, names\_factor, value = list(Browser =  
## structure(c(2L, : replacement element 6 has 900 rows to replace 165 rows

## Warning in `[<-.data.frame`(`\*tmp\*`, names\_factor, value = list(Browser =  
## structure(c(2L, : replacement element 7 has 900 rows to replace 165 rows

## Warning in `[<-.data.frame`(`\*tmp\*`, names\_factor, value = list(Browser =  
## structure(c(2L, : replacement element 8 has 900 rows to replace 165 rows

Test = as.data.frame(Test)  
  
  
# Exploración de los datos pre filtrado ----  
# http://www.rdatamining.com/docs/data-exploration-and-visualization-with-r  
  
# number of rows  
nrow(prep)

## [1] 900

# number of columns  
ncol(prep)

## [1] 18

# dimensionality  
dim(prep)

## [1] 900 18

# column names  
names(prep)

## [1] "Administrative" "Administrative\_Duration"  
## [3] "Informational" "Informational\_Duration"   
## [5] "ProductRelated" "ProductRelated\_Duration"  
## [7] "BounceRates" "ExitRates"   
## [9] "PageValues" "SpecialDay"   
## [11] "Month" "OperatingSystems"   
## [13] "Browser" "Region"   
## [15] "TrafficType" "VisitorType"   
## [17] "Weekend" "Revenue"

summary(prep)

## Administrative Administrative\_Duration Informational   
## Min. : 0.000 Min. : -1.00 Min. : 0.0000   
## 1st Qu.: 0.000 1st Qu.: 0.00 1st Qu.: 0.0000   
## Median : 0.000 Median : 0.00 Median : 0.0000   
## Mean : 1.702 Mean : 61.46 Mean : 0.3767   
## 3rd Qu.: 3.000 3rd Qu.: 58.12 3rd Qu.: 0.0000   
## Max. :19.000 Max. :1521.00 Max. :16.0000   
## Informational\_Duration ProductRelated ProductRelated\_Duration  
## Min. : -1.00 Min. : 0.00 Min. : -1.0   
## 1st Qu.: 0.00 1st Qu.: 5.00 1st Qu.: 102.9   
## Median : 0.00 Median : 12.00 Median : 385.8   
## Mean : 27.43 Mean : 18.68 Mean : 770.9   
## 3rd Qu.: 0.00 3rd Qu.: 24.00 3rd Qu.: 969.1   
## Max. :1467.00 Max. :287.00 Max. :12983.8   
## BounceRates ExitRates PageValues SpecialDay   
## Min. :0.00000 Min. :0.00080 Min. : 0.000 Min. :0.00000   
## 1st Qu.:0.00000 1st Qu.:0.01362 1st Qu.: 0.000 1st Qu.:0.00000   
## Median :0.00000 Median :0.02857 Median : 0.000 Median :0.00000   
## Mean :0.02580 Mean :0.04943 Mean : 3.131 Mean :0.03956   
## 3rd Qu.:0.01571 3rd Qu.:0.05679 3rd Qu.: 0.000 3rd Qu.:0.00000   
## Max. :0.20000 Max. :0.20000 Max. :153.443 Max. :1.00000   
## Month OperatingSystems Browser Region   
## Length:900 Min. :1.000 Min. : 1.000 Min. :1.000   
## Class :character 1st Qu.:2.000 1st Qu.: 2.000 1st Qu.:1.000   
## Mode :character Median :2.000 Median : 2.000 Median :2.000   
## Mean :2.071 Mean : 2.289 Mean :2.951   
## 3rd Qu.:2.250 3rd Qu.: 2.000 3rd Qu.:4.000   
## Max. :7.000 Max. :10.000 Max. :9.000   
## TrafficType VisitorType Weekend Revenue   
## Min. : 1.000 Length:900 Mode :logical Min. :0.00   
## 1st Qu.: 1.000 Class :character FALSE:681 1st Qu.:0.00   
## Median : 2.000 Mode :character TRUE :219 Median :0.00   
## Mean : 3.179 Mean :0.08   
## 3rd Qu.: 3.000 3rd Qu.:0.00   
## Max. :15.000 Max. :1.00

# describe check all columns  
describe(prep[, 1])

## prep[, 1]   
## n missing distinct Info Mean Gmd .05 .10   
## 900 0 20 0.815 1.702 2.512 0 0   
## .25 .50 .75 .90 .95   
## 0 0 3 5 8   
##   
## lowest : 0 1 2 3 4, highest: 15 16 17 18 19  
##   
## Value 0 1 2 3 4 5 6 7 8 9 10  
## Frequency 511 74 78 70 53 30 25 12 14 11 4  
## Proportion 0.568 0.082 0.087 0.078 0.059 0.033 0.028 0.013 0.016 0.012 0.004  
##   
## Value 11 12 13 14 15 16 17 18 19  
## Frequency 5 3 3 1 1 2 1 1 1  
## Proportion 0.006 0.003 0.003 0.001 0.001 0.002 0.001 0.001 0.001

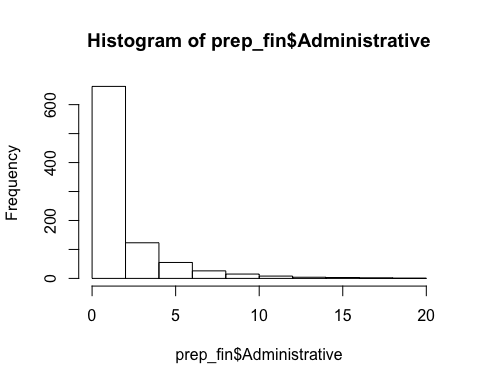
summary(prep\_fin)

## Administrative Administrative\_Duration Informational   
## Min. : 0.000 Min. : -1.00 Min. : 0.0000   
## 1st Qu.: 0.000 1st Qu.: 0.00 1st Qu.: 0.0000   
## Median : 0.000 Median : 0.00 Median : 0.0000   
## Mean : 1.702 Mean : 61.46 Mean : 0.3767   
## 3rd Qu.: 3.000 3rd Qu.: 58.12 3rd Qu.: 0.0000   
## Max. :19.000 Max. :1521.00 Max. :16.0000   
##   
## Informational\_Duration ProductRelated ProductRelated\_Duration  
## Min. : -1.00 Min. : 0.00 Min. : -1.0   
## 1st Qu.: 0.00 1st Qu.: 5.00 1st Qu.: 102.9   
## Median : 0.00 Median : 12.00 Median : 385.8   
## Mean : 27.43 Mean : 18.68 Mean : 770.9   
## 3rd Qu.: 0.00 3rd Qu.: 24.00 3rd Qu.: 969.1   
## Max. :1467.00 Max. :287.00 Max. :12983.8   
##   
## BounceRates ExitRates PageValues SpecialDay   
## Min. :0.00000 Min. :0.00080 Min. : 0.000 Min. :0.00000   
## 1st Qu.:0.00000 1st Qu.:0.01362 1st Qu.: 0.000 1st Qu.:0.00000   
## Median :0.00000 Median :0.02857 Median : 0.000 Median :0.00000   
## Mean :0.02580 Mean :0.04943 Mean : 3.131 Mean :0.03956   
## 3rd Qu.:0.01571 3rd Qu.:0.05679 3rd Qu.: 0.000 3rd Qu.:0.00000   
## Max. :0.20000 Max. :0.20000 Max. :153.443 Max. :1.00000   
##   
## Month OperatingSystems Browser Region TrafficType   
## Feb:160 1:205 2 :540 1 :368 1 :280   
## Mar:740 2:470 1 :204 3 :172 2 :228   
## 3:184 4 : 70 2 : 95 3 :221   
## 4: 40 5 : 33 4 : 89 10 : 44   
## 7: 1 6 : 22 7 : 53 8 : 35   
## 3 : 11 6 : 51 4 : 20   
## (Other): 20 (Other): 72 (Other): 72   
## VisitorType Weekend Revenue  
## New\_Visitor : 87 FALSE:681 0:828   
## Returning\_Visitor:813 TRUE :219 1: 72   
##   
##   
##   
##   
##

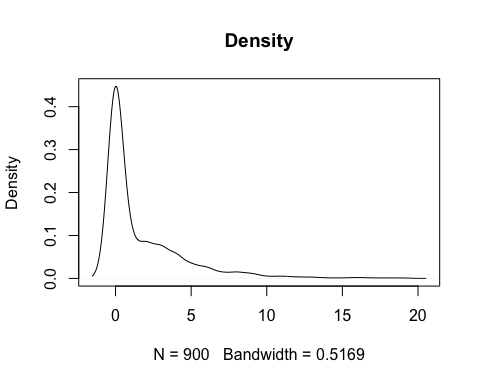
str(prep\_fin)

## 'data.frame': 900 obs. of 18 variables:  
## $ Administrative : num 0 0 1 0 2 0 0 1 0 1 ...  
## $ Administrative\_Duration: num 0 0 0 0 118 0 0 202 0 39 ...  
## $ Informational : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ Informational\_Duration : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ ProductRelated : num 4 6 21 1 4 6 14 2 1 15 ...  
## $ ProductRelated\_Duration: num 70.5 203.2 405.7 0 42 ...  
## $ BounceRates : num 0.05 0 0.0107 0.2 0 ...  
## $ ExitRates : num 0.1 0.0511 0.047 0.2 0.0667 ...  
## $ PageValues : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ SpecialDay : num 0 0 0 0 0 0.4 0 0 0.2 0 ...  
## $ Month : Factor w/ 2 levels "Feb","Mar": 2 2 1 2 2 1 2 2 1 2 ...  
## $ OperatingSystems : Factor w/ 5 levels "1","2","3","4",..: 3 2 1 3 1 1 1 2 2 4 ...  
## $ Browser : Factor w/ 10 levels "1","2","3","4",..: 2 2 1 2 2 1 1 2 4 1 ...  
## $ Region : Factor w/ 9 levels "1","2","3","4",..: 4 7 1 3 7 3 1 8 1 1 ...  
## $ TrafficType : Factor w/ 15 levels "1","2","3","4",..: 3 2 4 11 3 3 3 6 3 10 ...  
## $ VisitorType : Factor w/ 2 levels "New\_Visitor",..: 2 2 2 2 2 2 2 1 2 2 ...  
## $ Weekend : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 2 1 1 1 ...  
## $ Revenue : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. Id = col\_double(),  
## .. Administrative = col\_double(),  
## .. Administrative\_Duration = col\_double(),  
## .. Informational = col\_double(),  
## .. Informational\_Duration = col\_double(),  
## .. ProductRelated = col\_double(),  
## .. ProductRelated\_Duration = col\_double(),  
## .. BounceRates = col\_double(),  
## .. ExitRates = col\_double(),  
## .. PageValues = col\_double(),  
## .. SpecialDay = col\_double(),  
## .. Month = col\_character(),  
## .. OperatingSystems = col\_double(),  
## .. Browser = col\_double(),  
## .. Region = col\_double(),  
## .. TrafficType = col\_double(),  
## .. VisitorType = col\_character(),  
## .. Weekend = col\_logical(),  
## .. Revenue = col\_double()  
## .. )

# Histogramas  
hist(prep\_fin$Administrative)



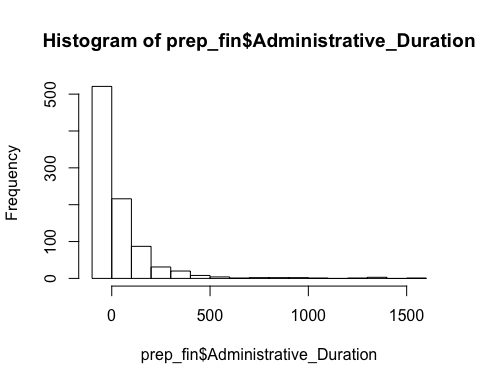
prep\_fin$Administrative %>% density() %>% plot(main = 'Density')



table(prep\_fin$Administrative)

##   
## 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19   
## 511 74 78 70 53 30 25 12 14 11 4 5 3 3 1 1 2 1 1 1

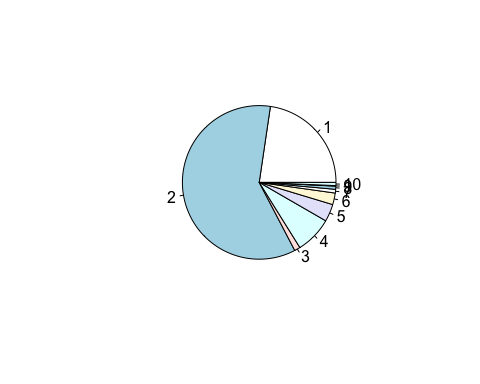
hist(prep\_fin$Administrative\_Duration)



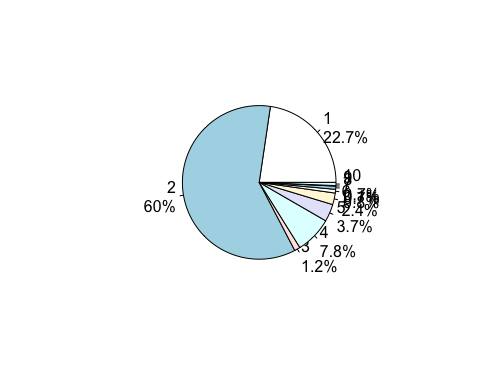
library(magrittr)  
  
  
# Categóricas  
# Browser  
table(prep\_fin$Browser)

##   
## 1 2 3 4 5 6 7 8 9 10   
## 204 540 11 70 33 22 7 6 1 6

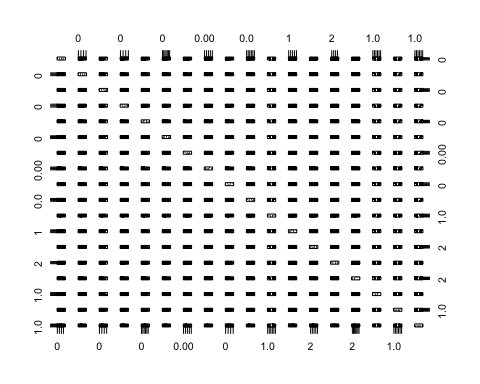
library(dplyr)  
prep\_fin$Browser %>% table() %>% pie()



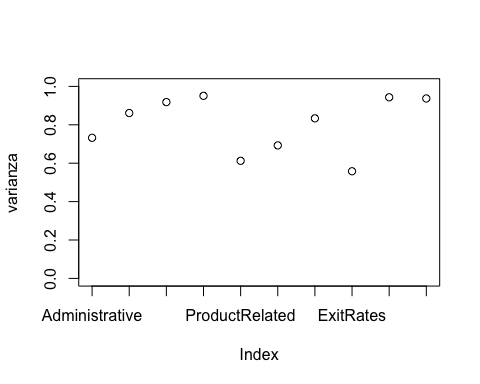
# add percentages  
tab <- prep\_fin$Browser %>% table()  
precentages <- tab %>% prop.table() %>% round(3) \* 100  
txt <- paste0(names(tab), '\n', precentages, '%')  
pie(tab, labels=txt)



# Multiple   
pairs(prep\_fin)



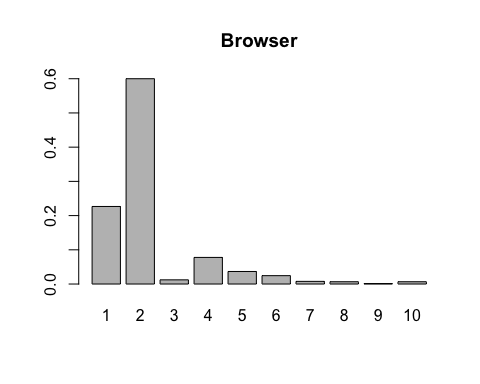
# Filtrado  
# Varianza Continuas ----  
x = prep\_fin  
x[names\_factor] = NULL  
x$Revenue = NULL  
  
x = scale(x, center = FALSE, scale = TRUE)  
  
varianza = var(x)  
varianza = diag(varianza)  
  
plot(varianza,xaxt="n", ylim = c(0,1))  
axis(1,at=1:10,labels=names(varianza))



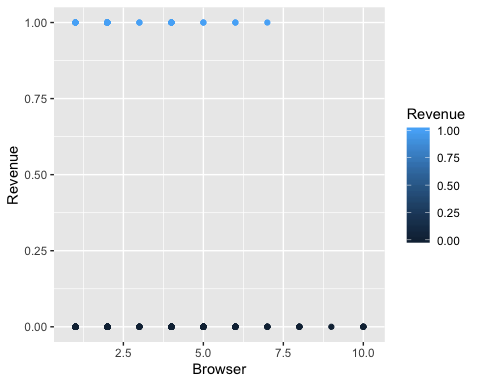
min(varianza) # No se quita ninguna

## [1] 0.5580471

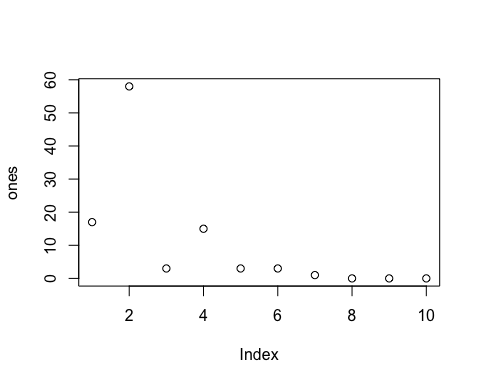
# Varianza Categoricas ----  
x = prep\_fin[names\_factor]  
x$Revenue = NULL  
  
barplot(prop.table(table(x[,1])), main = names\_factor[1])



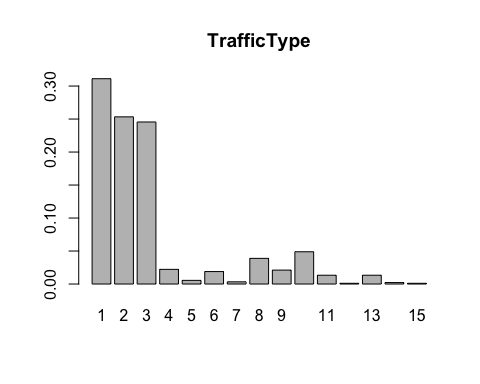
df = as.data.frame(cbind(x[,1],ifelse(Train$Revenue ==1,1,0)))  
colnames(df) = c(names\_factor[1],'Revenue')  
ggplot(df, aes(x = Browser, y = Revenue, color = Revenue)) +   
 geom\_point(position = position\_dodge(width = 0.4))



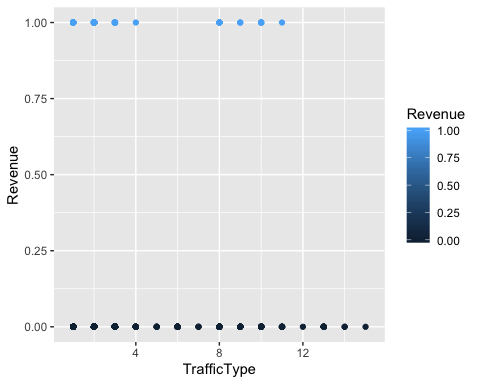
t = table(df$Revenue,df$Browser)  
prop = round(prop.table(t,1)\*100,digits=0)  
ones = prop[2,]  
plot(ones)



barplot(prop.table(table(x[,2])), main = names\_factor[2])



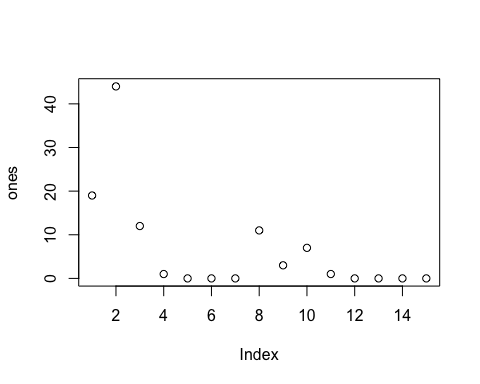
df = as.data.frame(cbind(x[,2],ifelse(Train$Revenue ==1,1,0)))  
colnames(df) = c(names\_factor[2],'Revenue')  
ggplot(df, aes(x = TrafficType, y = Revenue, color = Revenue)) +   
 geom\_point(position = position\_dodge(width = 0.4))



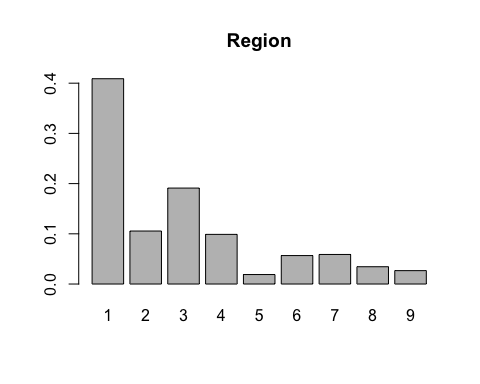
t = table(df$Revenue,df$TrafficType)  
  
prop = round(prop.table(t,1)\*100,digits=0)  
prop

##   
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15  
## 0 32 24 26 2 1 2 0 3 2 5 1 0 1 0 0  
## 1 19 44 12 1 0 0 0 11 3 7 1 0 0 0 0

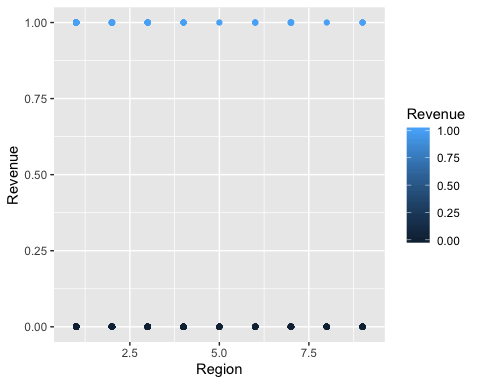
ones = prop[2,]  
plot(ones)



barplot(prop.table(table(x[,3])), main = names\_factor[3])



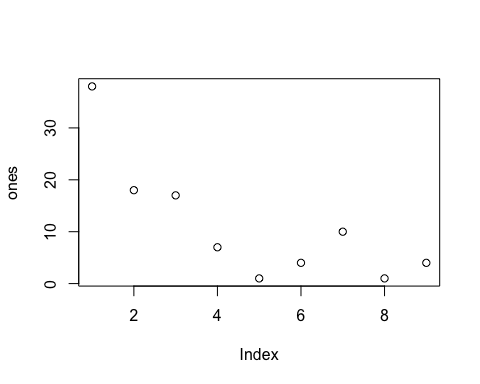
df = as.data.frame(cbind(x[,3],ifelse(Train$Revenue ==1,1,0)))  
colnames(df) = c(names\_factor[3],'Revenue')  
ggplot(df, aes(x = Region, y = Revenue, color = Revenue)) +   
 geom\_point(position = position\_dodge(width = 0.4))



t = table(df$Revenue,df$Region)  
  
prop = round(prop.table(t,1)\*100,digits=0)  
prop

##   
## 1 2 3 4 5 6 7 8 9  
## 0 41 10 19 10 2 6 6 4 3  
## 1 38 18 17 7 1 4 10 1 4

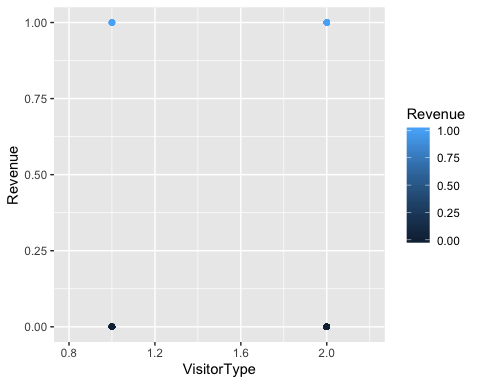
ones = prop[2,]  
plot(ones)



barplot(prop.table(table(x[,4])), main = names\_factor[4])



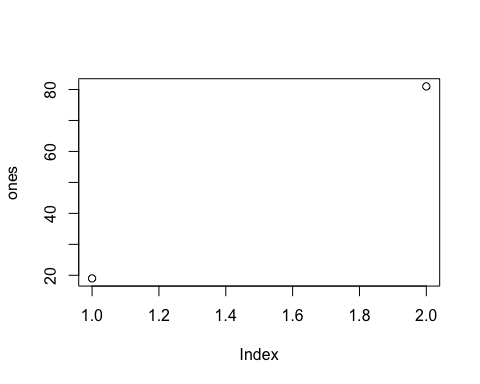
df = as.data.frame(cbind(x[,4],ifelse(Train$Revenue ==1,1,0)))  
colnames(df) = c(names\_factor[4],'Revenue')  
ggplot(df, aes(x = VisitorType, y = Revenue, color = Revenue)) +   
 geom\_point(position = position\_dodge(width = 0.4))



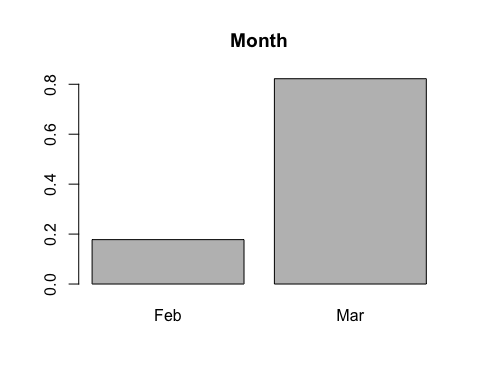
t = table(df[,2],df[,1])  
prop = round(prop.table(t,1)\*100,digits=0)  
prop

##   
## 1 2  
## 0 9 91  
## 1 19 81

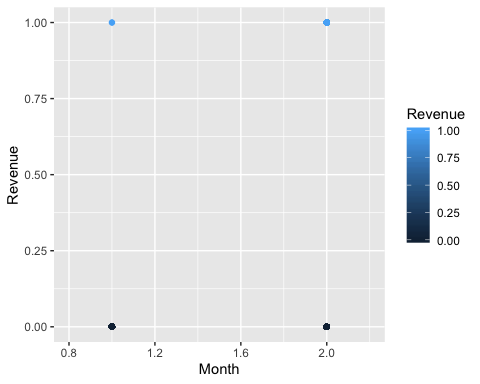
ones = prop[2,]  
plot(ones)



barplot(prop.table(table(x[,5])), main = names\_factor[5])



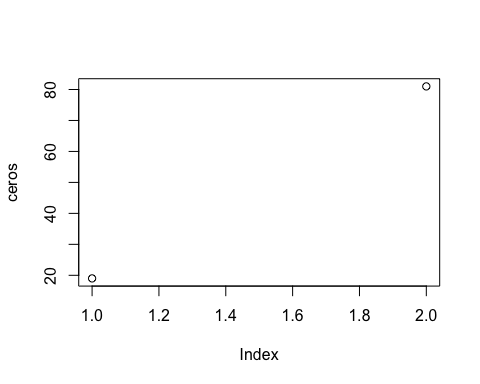
df = as.data.frame(cbind(x[,5],ifelse(Train$Revenue ==1,1,0)))  
colnames(df) = c(names\_factor[5],'Revenue')  
ggplot(df, aes(x = Month, y = Revenue, color = Revenue)) +   
 geom\_point(position = position\_dodge(width = 0.4))



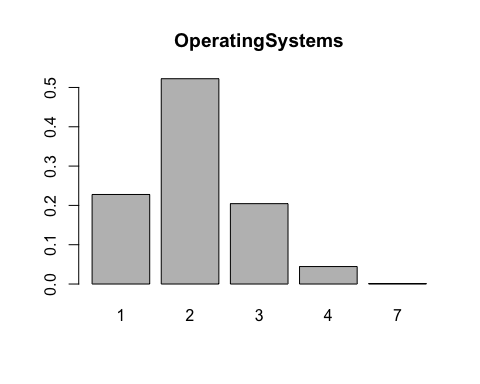
t = table(df[,2],df[,1])  
prop = round(prop.table(t,1)\*100,digits=0)  
prop

##   
## 1 2  
## 0 19 81  
## 1 3 97

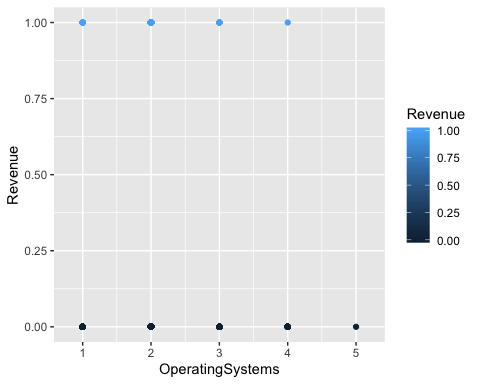
ceros = prop[1,]  
plot(ceros)



barplot(prop.table(table(x[,6])), main = names\_factor[6])



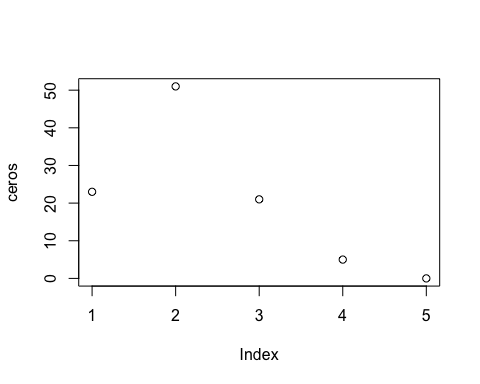
df = as.data.frame(cbind(x[,6],ifelse(Train$Revenue ==1,1,0)))  
colnames(df) = c(names\_factor[6],'Revenue')  
ggplot(df, aes(x = OperatingSystems, y = Revenue, color = Revenue)) +   
 geom\_point(position = position\_dodge(width = 0.4))



t = table(df[,2],df[,1])  
prop = round(prop.table(t,1)\*100,digits=0)  
prop

##   
## 1 2 3 4 5  
## 0 23 51 21 5 0  
## 1 21 68 10 1 0

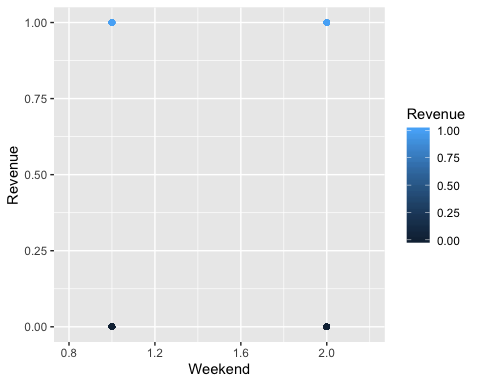
ceros = prop[1,]  
plot(ceros)



barplot(prop.table(table(x[,7])), main = names\_factor[8])



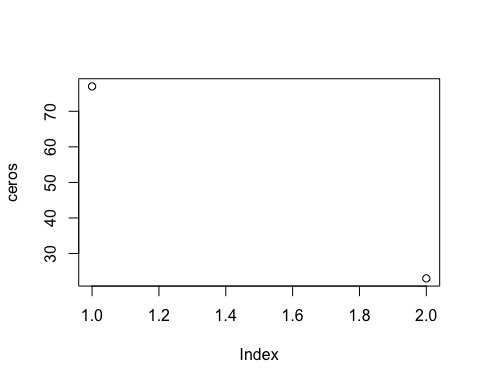
df = as.data.frame(cbind(x[,7],ifelse(Train$Revenue ==1,1,0)))  
colnames(df) = c(names\_factor[8],'Revenue')  
ggplot(df, aes(x = Weekend, y = Revenue, color = Revenue)) +   
 geom\_point(position = position\_dodge(width = 0.4))



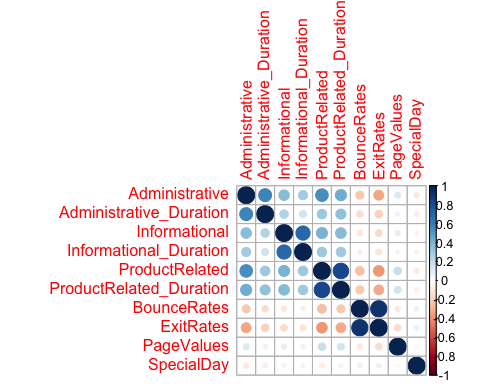
t = table(df[,2],df[,1])  
prop = round(prop.table(t,1)\*100,digits=0)  
prop

##   
## 1 2  
## 0 77 23  
## 1 60 40

ceros = prop[1,]  
plot(ceros)



# test\_fin = prep\_fin[901:1065,]  
# train\_fin = prep\_fin[1:900,]  
par(mfrow = c(1, 1))  
# Correlaciones para las variables no dummies  
vc = cor(prep\_fin[,1:10])  
corrplot::corrplot(vc,method="circle")



#DATA FINAL   
  
test\_xxs = prep\_fin[901:1065,]  
train\_xxs = prep\_fin[1:900,]  
test\_xxs[test\_xxs == "NaN"] = 0  
  
  
# Outliers ----  
library(DMwR)

## Loading required package: grid

library(tidyverse)

## ── Attaching packages ──────────────────────────────────────────────────────────────── tidyverse 1.3.0 ──

## ✔ tibble 2.1.3 ✔ stringr 1.4.0  
## ✔ tidyr 1.0.0 ✔ forcats 0.4.0  
## ✔ purrr 0.3.3

## ── Conflicts ─────────────────────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ randomForest::combine() masks dplyr::combine()  
## ✖ matrixStats::count() masks dplyr::count()  
## ✖ tidyr::extract() masks magrittr::extract()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ purrr::lift() masks caret::lift()  
## ✖ randomForest::margin() masks ggplot2::margin()  
## ✖ purrr::set\_names() masks magrittr::set\_names()  
## ✖ dplyr::src() masks Hmisc::src()  
## ✖ dplyr::summarize() masks Hmisc::summarize()

x = prep\_fin  
x[names\_factor] = NULL  
x = scale(x, center = FALSE, scale = TRUE)  
  
outlier.scores <- lofactor(x, k=5) ## CUÁNTOS VECINOS USAR??  
  
# pick top 5 as outliers  
outliers <- order(outlier.scores, decreasing=T)[1:round(0.01\*nrow(x))]  
labels <- 1:length(outlier.scores)  
labels[-outliers] <- "."  
biplot(prcomp(x), cex=.8, xlabs=labels, na.rm = TRUE)

## Warning in plot.window(...): "na.rm" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "na.rm" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "na.rm" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "na.rm" is not a  
## graphical parameter

## Warning in box(...): "na.rm" is not a graphical parameter

## Warning in title(...): "na.rm" is not a graphical parameter

## Warning in text.default(x, xlabs, cex = cex[1L], col = col[1L], ...): "na.rm" is  
## not a graphical parameter

## Warning in plot.window(...): "na.rm" is not a graphical parameter

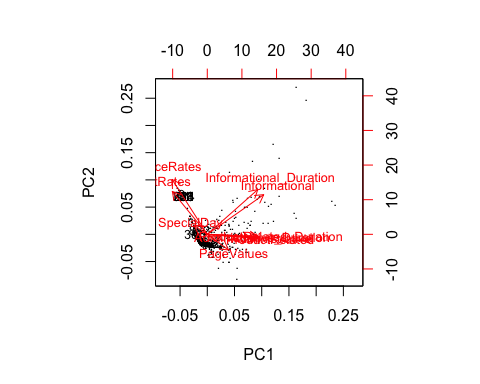
## Warning in plot.xy(xy, type, ...): "na.rm" is not a graphical parameter

## Warning in title(...): "na.rm" is not a graphical parameter

## Warning in axis(3, col = col[2L], ...): "na.rm" is not a graphical parameter

## Warning in axis(4, col = col[2L], ...): "na.rm" is not a graphical parameter

## Warning in text.default(y, labels = ylabs, cex = cex[2L], col = col[2L], :  
## "na.rm" is not a graphical parameter



biplot(prcomp(x), cex=.8, xlabs=labels, na.rm = TRUE, xlim = c(-0.09,0.15), ylim = c(-0.08,0.09))

## Warning in plot.window(...): "na.rm" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "na.rm" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "na.rm" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "na.rm" is not a  
## graphical parameter

## Warning in box(...): "na.rm" is not a graphical parameter

## Warning in title(...): "na.rm" is not a graphical parameter

## Warning in text.default(x, xlabs, cex = cex[1L], col = col[1L], ...): "na.rm" is  
## not a graphical parameter

## Warning in plot.window(...): "na.rm" is not a graphical parameter

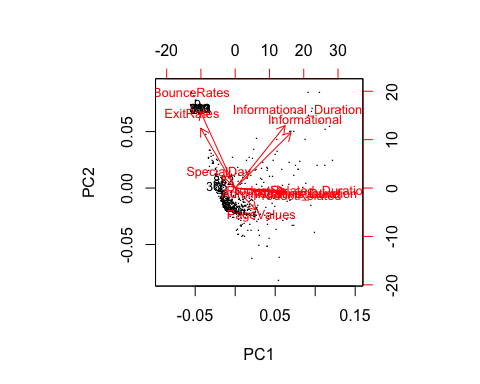
## Warning in plot.xy(xy, type, ...): "na.rm" is not a graphical parameter

## Warning in title(...): "na.rm" is not a graphical parameter

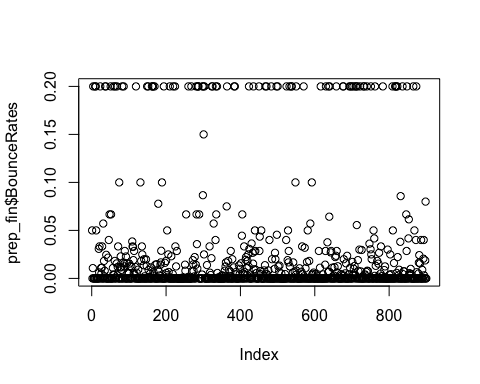
## Warning in axis(3, col = col[2L], ...): "na.rm" is not a graphical parameter

## Warning in axis(4, col = col[2L], ...): "na.rm" is not a graphical parameter

## Warning in text.default(y, labels = ylabs, cex = cex[2L], col = col[2L], :  
## "na.rm" is not a graphical parameter



plot(prep\_fin$BounceRates)



# who are outliers  
print(outliers)

## [1] 9 11 224 323 751 783 816 882 365

print(Train[outliers,])

## Administrative Administrative\_Duration Informational Informational\_Duration  
## 9 0 0 0 0  
## 11 0 0 0 0  
## 224 1 -1 0 -1  
## 323 0 0 0 0  
## 751 0 0 0 0  
## 783 0 0 0 0  
## 816 0 0 0 0  
## 882 0 0 1 0  
## 365 0 0 0 0  
## ProductRelated ProductRelated\_Duration BounceRates ExitRates PageValues  
## 9 1 0 0.2 0.2 0  
## 11 3 0 0.2 0.2 0  
## 224 1 -1 0.2 0.2 0  
## 323 3 0 0.2 0.2 0  
## 751 3 0 0.2 0.2 0  
## 783 4 0 0.2 0.2 0  
## 816 3 0 0.2 0.2 0  
## 882 1 13 0.0 0.1 0  
## 365 2 33 0.0 0.1 0  
## SpecialDay Month OperatingSystems Browser Region TrafficType  
## 9 0.2 Feb 2 4 1 3  
## 11 0.0 Mar 2 2 4 1  
## 224 0.0 Feb 1 2 1 5  
## 323 0.0 Mar 3 2 6 3  
## 751 0.0 Mar 1 8 2 1  
## 783 0.0 Mar 1 2 1 1  
## 816 0.0 Mar 1 1 3 1  
## 882 0.0 Mar 3 2 3 1  
## 365 0.2 Feb 1 1 1 3  
## VisitorType Weekend Revenue  
## 9 Returning\_Visitor FALSE 0  
## 11 Returning\_Visitor FALSE 0  
## 224 Returning\_Visitor TRUE 0  
## 323 Returning\_Visitor FALSE 0  
## 751 Returning\_Visitor FALSE 0  
## 783 Returning\_Visitor FALSE 0  
## 816 Returning\_Visitor FALSE 0  
## 882 Returning\_Visitor FALSE 0  
## 365 Returning\_Visitor FALSE 0

Train = Train[-outliers,]  
  
  
# Desbalance  
table(Train$Revenue)

##   
## 0 1   
## 819 72

Train$Revenue = as.factor(Train$Revenue)  
## now using SMOTE to create a more "balanced problem"  
newData <- SMOTE(Revenue ~ ., Train, perc.over = 1200, perc.under = 100)  
# perc.over: Qu? porcentaje se quiere hacer m?s de la chiquita  
# perc.under: Qu? porcentaje se quiere hacer menos de la grande  
  
################################### ver si sin round predice mejor !!!!!! ################  
  
  
## Checking visually the created data  
par(mfrow = c(1, 2))  
plot(Train[, 1], Train[, 2], pch = 19 + as.integer(Train[, 3]),  
 main = "Original Data")

## Warning in plot.xy(xy, type, ...): unimplemented pch value '26'

## Warning in plot.xy(xy, type, ...): unimplemented pch value '31'

plot(newData[, 1], newData[, 2], pch = 19 + as.integer(newData[,3]),  
 main = "SMOTE'd Data")

## Warning in plot.xy(xy, type, ...): unimplemented pch value '26'

## Warning in plot.xy(xy, type, ...): unimplemented pch value '26'  
  
## Warning in plot.xy(xy, type, ...): unimplemented pch value '26'

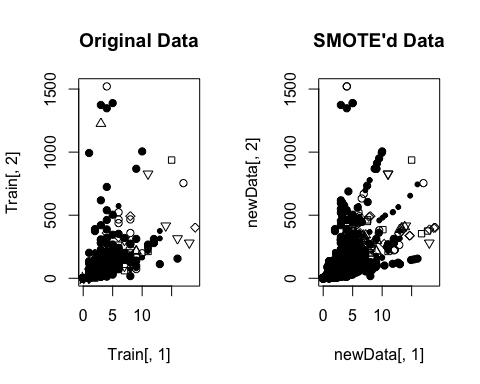
## Warning in plot.xy(xy, type, ...): unimplemented pch value '31'

## Warning in plot.xy(xy, type, ...): unimplemented pch value '30'

## Warning in plot.xy(xy, type, ...): unimplemented pch value '29'

## Warning in plot.xy(xy, type, ...): unimplemented pch value '27'

## Warning in plot.xy(xy, type, ...): unimplemented pch value '30'  
  
## Warning in plot.xy(xy, type, ...): unimplemented pch value '30'



table(newData$Revenue)

##   
## 0 1   
## 864 936

Train\_final = newData