Untitled

#Title: "MSDS 6372 Group Project 2: Bank Project - Predicting if a customer will subscribe to a term deposit."  
#Group Members: Nicole Norelli,Nneanna Okpara,Sowmya Mani  
#Date: March 25 2021  
  
#Introduction: This Project is about the Bank Market Analysis to predict if a customer will subscribe to a term deposit  
  
#The data set used for this analysis consists of Bank Full Data:  
#Citation:  
#This dataset is public available for research. The details are described in [Moro et al., 2011].   
#[Moro et al., 2011] S. Moro, R. Laureano and P. Cortez. Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology.   
#In P. Novais et al. (Eds.), Proceedings of the European Simulation and Modelling Conference - ESM'2011, pp. 117-121, Guimarães, Portugal, October, 2011. EUROSIS.  
#Available at: [pdf] http://hdl.handle.net/1822/14838  
# [bib] http://www3.dsi.uminho.pt/pcortez/bib/2011-esm-1.txt  
#Bank Full.csv and Bank.csv  
  
##### Bank client data #####  
#1 - age (numeric)  
#2 - job : type of job (categorical:"admin.","unknown","unemployed","management","housemaid","entrepreneur","student",  
# "blue-collar","self-employed","retired","technician","services")   
#3 - marital : marital status (categorical: "married","divorced","single"; note: "divorced" means divorced or widowed)  
#4 - education (categorical: "unknown","secondary","primary","tertiary")  
#5 - default: has credit in default? (binary: "yes","no")  
#6 - balance: average yearly balance, in euros (numeric)   
#7 - housing: has housing loan? (binary: "yes","no")  
#8 - loan: has personal loan? (binary: "yes","no")  
##### related with the last contact of the current campaign #####  
#9 - contact: contact communication type (categorical: "unknown","telephone","cellular")   
#10 - day: last contact day of the month (numeric)  
#11 - month: last contact month of year (categorical: "jan", "feb", "mar", ..., "nov", "dec")  
#12 - duration: last contact duration, in seconds (numeric)  
##### other attributes #####  
#13 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)  
#14 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric, -1 means client was not previously contacted)  
#15 - previous: number of contacts performed before this campaign and for this client (numeric)  
#16 - poutcome: outcome of the previous marketing campaign (categorical: "unknown","other","failure","success")  
  
##### Output variable (desired target) #####  
#17 - y - has the client subscribed a term deposit? (binary: "yes","no")  
  
#plot(model.name)

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

#Libraries loaded for the ANalysis  
library(XML)

## Warning: package 'XML' was built under R version 4.0.3

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.0.3

##   
## 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(RCurl)

## Warning: package 'RCurl' was built under R version 4.0.3

library(httr)

## Warning: package 'httr' was built under R version 4.0.3

library(jsonlite)

## Warning: package 'jsonlite' was built under R version 4.0.3

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.0.3

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3 v purrr 0.3.4  
## v tibble 3.0.4 v stringr 1.4.0  
## v tidyr 1.1.2 v forcats 0.5.0  
## v readr 1.4.0

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

## Warning: package 'tibble' was built under R version 4.0.3

## Warning: package 'tidyr' was built under R version 4.0.3

## Warning: package 'readr' was built under R version 4.0.3

## Warning: package 'purrr' was built under R version 4.0.3

## Warning: package 'stringr' was built under R version 4.0.3

## Warning: package 'forcats' was built under R version 4.0.3

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x tidyr::complete() masks RCurl::complete()  
## x dplyr::filter() masks stats::filter()  
## x purrr::flatten() masks jsonlite::flatten()  
## x dplyr::lag() masks stats::lag()

library(naniar)

## Warning: package 'naniar' was built under R version 4.0.3

library(GGally)

## Warning: package 'GGally' was built under R version 4.0.3

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

library(ggplot2)  
library(class)  
library(caret)

## Warning: package 'caret' was built under R version 4.0.3

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

## The following object is masked from 'package:httr':  
##   
## progress

library(knnp)

## Warning: package 'knnp' was built under R version 4.0.3

##   
## Attaching package: 'knnp'

## The following object is masked from 'package:class':  
##   
## knn

library(e1071)

## Warning: package 'e1071' was built under R version 4.0.3

library(ggplot2)  
library(maps)

## Warning: package 'maps' was built under R version 4.0.3

##   
## Attaching package: 'maps'

## The following object is masked from 'package:purrr':  
##   
## map

library(dplyr)  
library(mapproj)

## Warning: package 'mapproj' was built under R version 4.0.3

library(ggplot2)  
library(dplyr)  
library(ggcorrplot)

## Warning: package 'ggcorrplot' was built under R version 4.0.3

library(GGally)  
library(viridis)

## Warning: package 'viridis' was built under R version 4.0.3

## Loading required package: viridisLite

library(gplots)

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

##   
## Attaching package: 'gplots'

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

library(leaps)

## Warning: package 'leaps' was built under R version 4.0.3

library(matrixStats)

##   
## Attaching package: 'matrixStats'

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

library(ResourceSelection)

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

## ResourceSelection 0.3-5 2019-07-22

library(MASS)

## Warning: package 'MASS' was built under R version 4.0.3

##   
## Attaching package: 'MASS'

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

library(glmnet)

## Warning: package 'glmnet' was built under R version 4.0.3

## Loading required package: Matrix

## Warning: package 'Matrix' was built under R version 4.0.3

##   
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':  
##   
## expand, pack, unpack

## Loaded glmnet 4.1

library(ROCR)

## Warning: package 'ROCR' 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

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

#Import the Bank Full Data  
Bank\_Full<-read.csv('C:/Sowmya/SMU/03\_Applied Stats/Group Project 2/bank-full.csv' ,sep=";")  
  
#Quick Peek at the SUmmary data of the available dataset  
summary(Bank\_Full)

## age job marital education   
## Min. :18.00 Length:45211 Length:45211 Length:45211   
## 1st Qu.:33.00 Class :character Class :character Class :character   
## Median :39.00 Mode :character Mode :character Mode :character   
## Mean :40.94   
## 3rd Qu.:48.00   
## Max. :95.00   
## default balance housing loan   
## Length:45211 Min. : -8019 Length:45211 Length:45211   
## Class :character 1st Qu.: 72 Class :character Class :character   
## Mode :character Median : 448 Mode :character Mode :character   
## Mean : 1362   
## 3rd Qu.: 1428   
## Max. :102127   
## contact day month duration   
## Length:45211 Min. : 1.00 Length:45211 Min. : 0.0   
## Class :character 1st Qu.: 8.00 Class :character 1st Qu.: 103.0   
## Mode :character Median :16.00 Mode :character Median : 180.0   
## Mean :15.81 Mean : 258.2   
## 3rd Qu.:21.00 3rd Qu.: 319.0   
## Max. :31.00 Max. :4918.0   
## campaign pdays previous poutcome   
## Min. : 1.000 Min. : -1.0 Min. : 0.0000 Length:45211   
## 1st Qu.: 1.000 1st Qu.: -1.0 1st Qu.: 0.0000 Class :character   
## Median : 2.000 Median : -1.0 Median : 0.0000 Mode :character   
## Mean : 2.764 Mean : 40.2 Mean : 0.5803   
## 3rd Qu.: 3.000 3rd Qu.: -1.0 3rd Qu.: 0.0000   
## Max. :63.000 Max. :871.0 Max. :275.0000   
## y   
## Length:45211   
## Class :character   
## Mode :character   
##   
##   
##

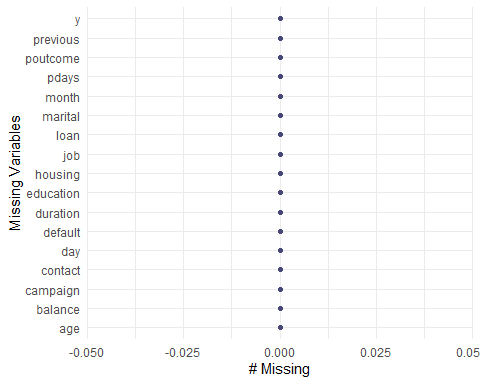
str(Bank\_Full)

## 'data.frame': 45211 obs. of 17 variables:  
## $ age : int 58 44 33 47 33 35 28 42 58 43 ...  
## $ job : chr "management" "technician" "entrepreneur" "blue-collar" ...  
## $ marital : chr "married" "single" "married" "married" ...  
## $ education: chr "tertiary" "secondary" "secondary" "unknown" ...  
## $ default : chr "no" "no" "no" "no" ...  
## $ balance : int 2143 29 2 1506 1 231 447 2 121 593 ...  
## $ housing : chr "yes" "yes" "yes" "yes" ...  
## $ loan : chr "no" "no" "yes" "no" ...  
## $ contact : chr "unknown" "unknown" "unknown" "unknown" ...  
## $ day : int 5 5 5 5 5 5 5 5 5 5 ...  
## $ month : chr "may" "may" "may" "may" ...  
## $ duration : int 261 151 76 92 198 139 217 380 50 55 ...  
## $ campaign : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ pdays : int -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...  
## $ previous : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ poutcome : chr "unknown" "unknown" "unknown" "unknown" ...  
## $ y : chr "no" "no" "no" "no" ...

#Checking for Missing Data  
sapply(Bank\_Full,function(x) sum(is.na(x)))

## age job marital education default balance housing loan   
## 0 0 0 0 0 0 0 0   
## contact day month duration campaign pdays previous poutcome   
## 0 0 0 0 0 0 0 0   
## y   
## 0

gg\_miss\_var(Bank\_Full)+xlab("Missing Variables")



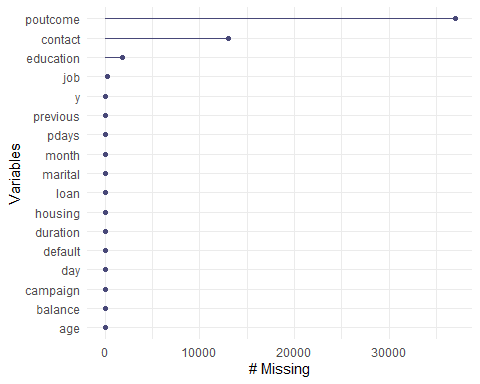
#The Bank dataset has 45,211 observations with 17 variables providing more information on the Bank Clients.There is no missing data in the data set  
  
# missing values using other than NA (Find the Unknowns in the data)  
sapply(Bank\_Full, function(x) sum(x %in% 'unknown'))

## age job marital education default balance housing loan   
## 0 288 0 1857 0 0 0 0   
## contact day month duration campaign pdays previous poutcome   
## 13020 0 0 0 0 0 0 36959   
## y   
## 0

#We find 288 unknows in the job category, 1,857 in education, 13,020 in contact, 36,959 in poutcome  
  
Bank\_Full\_NA = Bank\_Full %>%  
 dplyr::na\_if('unknown')  
sapply(Bank\_Full\_NA, function(x) sum(is.na(x)))

## age job marital education default balance housing loan   
## 0 288 0 1857 0 0 0 0   
## contact day month duration campaign pdays previous poutcome   
## 13020 0 0 0 0 0 0 36959   
## y   
## 0

#Change unknown to NA   
gg\_miss\_var(Bank\_Full\_NA)



# This is interesting:   
# Contact (contact communication type) is missing in a block from May 5 to partway through July 4, so it's not random  
# Similar issue with poutcome (outcome of previous marketing campaign). Values start to fill in around late October. Of course there's a lot more missing in this one.  
# Another thing to note: Because this data spans 2+ years (it seems to be in calendar order), I wonder if we should put a year variable in to keep track...  
summary(Bank\_Full\_NA)

## age job marital education   
## Min. :18.00 Length:45211 Length:45211 Length:45211   
## 1st Qu.:33.00 Class :character Class :character Class :character   
## Median :39.00 Mode :character Mode :character Mode :character   
## Mean :40.94   
## 3rd Qu.:48.00   
## Max. :95.00   
## default balance housing loan   
## Length:45211 Min. : -8019 Length:45211 Length:45211   
## Class :character 1st Qu.: 72 Class :character Class :character   
## Mode :character Median : 448 Mode :character Mode :character   
## Mean : 1362   
## 3rd Qu.: 1428   
## Max. :102127   
## contact day month duration   
## Length:45211 Min. : 1.00 Length:45211 Min. : 0.0   
## Class :character 1st Qu.: 8.00 Class :character 1st Qu.: 103.0   
## Mode :character Median :16.00 Mode :character Median : 180.0   
## Mean :15.81 Mean : 258.2   
## 3rd Qu.:21.00 3rd Qu.: 319.0   
## Max. :31.00 Max. :4918.0   
## campaign pdays previous poutcome   
## Min. : 1.000 Min. : -1.0 Min. : 0.0000 Length:45211   
## 1st Qu.: 1.000 1st Qu.: -1.0 1st Qu.: 0.0000 Class :character   
## Median : 2.000 Median : -1.0 Median : 0.0000 Mode :character   
## Mean : 2.764 Mean : 40.2 Mean : 0.5803   
## 3rd Qu.: 3.000 3rd Qu.: -1.0 3rd Qu.: 0.0000   
## Max. :63.000 Max. :871.0 Max. :275.0000   
## y   
## Length:45211   
## Class :character   
## Mode :character   
##   
##   
##

# Note: pdays: -1 means not previously contacted. We might need to do something with that

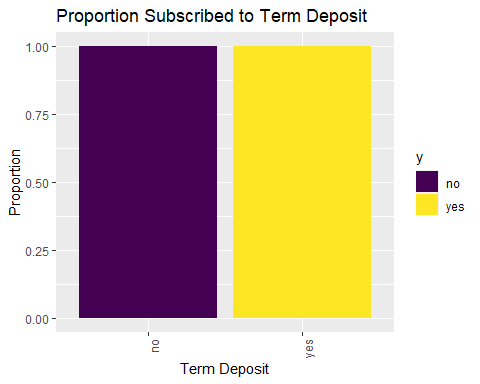
#The data seem to be sequentially ordered, adding an ID number in order  
Bank\_Full = Bank\_Full %>%  
 mutate(id = row\_number())  
# Adding a new variable to include year which could add more insights to the data  
Bank\_Full = Bank\_Full %>%  
 mutate(year = case\_when(id>=1 & id<=27729 ~ '2008',  
 id>=27730 & id<=42591 ~ '2009',  
 id>=42592 ~ '2010'))  
Bank\_Full$year = factor(Bank\_Full$year)  
  
# Let's reorder month so it makes sense on graphs  
Bank\_Full$month = factor(Bank\_Full$month, levels=c("jan","feb","mar","apr","may","jun","jul","aug","sep","oct","nov","dec"))  
  
# A look at categorical  
ggpairs(Bank\_Full,columns=c(2:5,7:9,11,16:17),aes(colour=y)) # that's a little busy, let's break it down



#Analyzing the data with plots  
str(Bank\_Full)

## 'data.frame': 45211 obs. of 19 variables:  
## $ age : int 58 44 33 47 33 35 28 42 58 43 ...  
## $ job : chr "management" "technician" "entrepreneur" "blue-collar" ...  
## $ marital : chr "married" "single" "married" "married" ...  
## $ education: chr "tertiary" "secondary" "secondary" "unknown" ...  
## $ default : chr "no" "no" "no" "no" ...  
## $ balance : int 2143 29 2 1506 1 231 447 2 121 593 ...  
## $ housing : chr "yes" "yes" "yes" "yes" ...  
## $ loan : chr "no" "no" "yes" "no" ...  
## $ contact : chr "unknown" "unknown" "unknown" "unknown" ...  
## $ day : int 5 5 5 5 5 5 5 5 5 5 ...  
## $ month : Factor w/ 12 levels "jan","feb","mar",..: 5 5 5 5 5 5 5 5 5 5 ...  
## $ duration : int 261 151 76 92 198 139 217 380 50 55 ...  
## $ campaign : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ pdays : int -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...  
## $ previous : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ poutcome : chr "unknown" "unknown" "unknown" "unknown" ...  
## $ y : chr "no" "no" "no" "no" ...  
## $ id : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ year : Factor w/ 3 levels "2008","2009",..: 1 1 1 1 1 1 1 1 1 1 ...

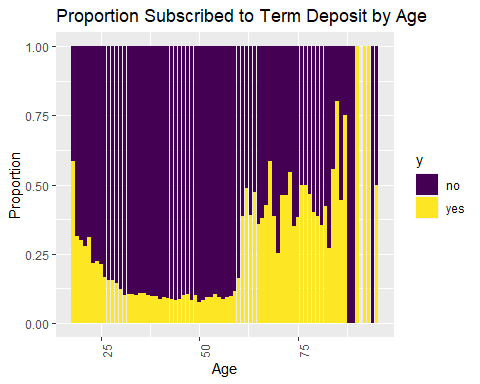
#attach(Bank\_Full)  
#Response  
Bank\_Full %>%   
 ggplot(aes(x=y, fill=y)) +   
 geom\_bar(position="fill") +   
 scale\_fill\_viridis\_d() +  
 ylab("Proportion") +xlab("Term Deposit")+  
 ggtitle("Proportion Subscribed to Term Deposit")+ theme(axis.text.x = element\_text(angle=90, hjust=1))



#The count of Yes and No on term deposit look equal  
  
#Bank\_Full %>% count(y)  
#No s count 39,922; Yes count 5,289   
  
# Age vs response  
t(aggregate(age~y,data=Bank\_Full,summary))

## [,1] [,2]   
## y "no" "yes"   
## age.Min. "18.00000" "18.00000"  
## age.1st Qu. "33.00000" "31.00000"  
## age.Median "39.00000" "38.00000"  
## age.Mean "40.83899" "41.67007"  
## age.3rd Qu. "48.00000" "50.00000"  
## age.Max. "95.00000" "95.00000"

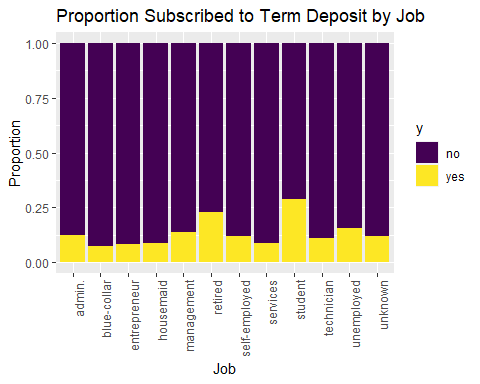
Bank\_Full %>%   
 ggplot(aes(x=age, fill=y)) +   
 geom\_bar(position="fill") +   
 scale\_fill\_viridis\_d() +  
 ylab("Proportion") +xlab("Age")+  
 ggtitle("Proportion Subscribed to Term Deposit by Age")+ theme(axis.text.x = element\_text(angle=90, hjust=1))



#The older the age the higher proportions of Yes to a term deposit  
#Age and term deposit seems to be correlated.  
  
# Job vs response  
prop.table(table(Bank\_Full$y,Bank\_Full$job),2)

##   
## admin. blue-collar entrepreneur housemaid management retired  
## no 0.87797331 0.92725031 0.91728312 0.91209677 0.86244449 0.77208481  
## yes 0.12202669 0.07274969 0.08271688 0.08790323 0.13755551 0.22791519  
##   
## self-employed services student technician unemployed unknown  
## no 0.88157061 0.91116996 0.71321962 0.88943004 0.84497314 0.88194444  
## yes 0.11842939 0.08883004 0.28678038 0.11056996 0.15502686 0.11805556

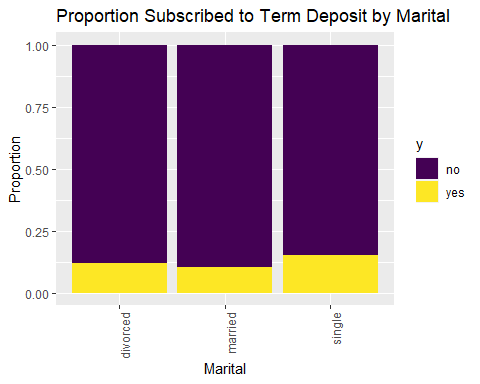
Bank\_Full %>%   
 ggplot(aes(x=job, fill=y)) +   
 geom\_bar(position="fill") +   
 scale\_fill\_viridis\_d() +  
 ylab("Proportion") +xlab("Job")+  
 ggtitle("Proportion Subscribed to Term Deposit by Job")+ theme(axis.text.x = element\_text(angle=90, hjust=1))



#Looks like retired and students are on the higher proportions of Yes to a term deposit  
  
# Marital vs response  
prop.table(table(Bank\_Full$y,Bank\_Full$marital),2)

##   
## divorced married single  
## no 0.8805454 0.8987653 0.8505082  
## yes 0.1194546 0.1012347 0.1494918

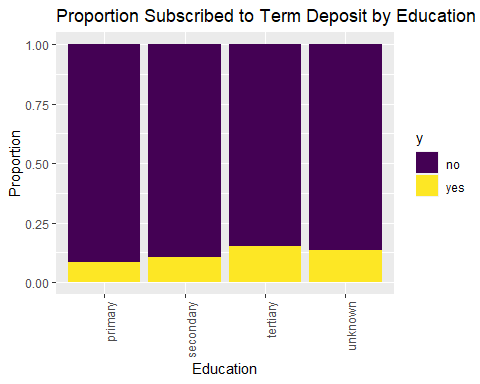
Bank\_Full %>%   
 ggplot(aes(x=marital, fill=y)) +   
 geom\_bar(position="fill") +   
 scale\_fill\_viridis\_d() +  
 ylab("Proportion") + xlab("Marital")+  
 ggtitle("Proportion Subscribed to Term Deposit by Marital")+ theme(axis.text.x = element\_text(angle=90, hjust=1))



# Slightly higher for singles  
  
# Education vs response  
prop.table(table(Bank\_Full$y,Bank\_Full$education),2)

##   
## primary secondary tertiary unknown  
## no 0.91373522 0.89440565 0.84993610 0.86429725  
## yes 0.08626478 0.10559435 0.15006390 0.13570275

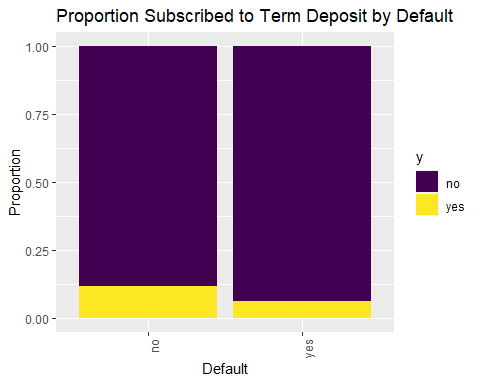
Bank\_Full %>%   
 ggplot(aes(x=education, fill=y)) +   
 geom\_bar(position="fill") +   
 scale\_fill\_viridis\_d() +  
 ylab("Proportion") +xlab("Education")+  
 ggtitle("Proportion Subscribed to Term Deposit by Education")+ theme(axis.text.x = element\_text(angle=90, hjust=1))



# Looks like it increases slightly as education increases  
  
# default vs response  
prop.table(table(Bank\_Full$y,Bank\_Full$default),2)

##   
## no yes  
## no 0.88203892 0.93619632  
## yes 0.11796108 0.06380368

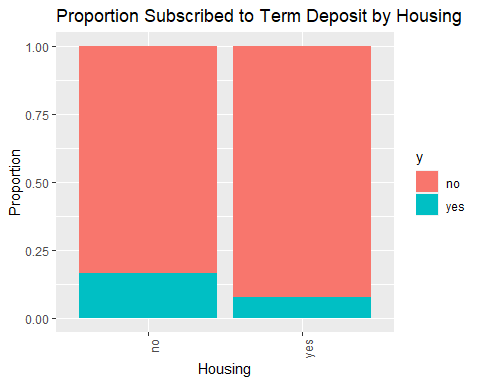
Bank\_Full %>%   
 ggplot(aes(x=default, fill=y)) +   
 geom\_bar(position="fill") +   
 scale\_fill\_viridis\_d() +  
 ylab("Proportion") +xlab("Default")+  
 ggtitle("Proportion Subscribed to Term Deposit by Default")+ theme(axis.text.x = element\_text(angle=90, hjust=1))



# Higher for those who have less credit, but there are very few defaults relatively, so it might not be that useful  
  
# housing vs response  
prop.table(table(Bank\_Full$y,Bank\_Full$housing),2)

##   
## no yes  
## no 0.8329764 0.9230004  
## yes 0.1670236 0.0769996

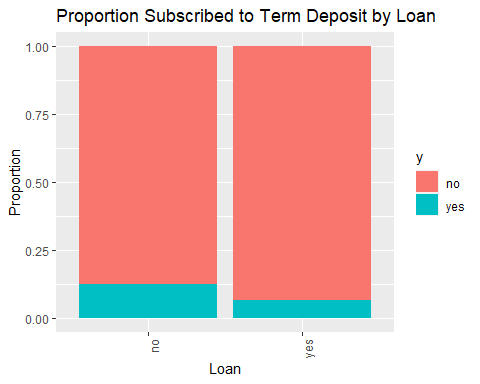
Bank\_Full %>%   
 ggplot(aes(x=housing, fill=y)) +   
 geom\_bar(position="fill") +   
 # scale\_fill\_viridis\_d() +  
 ylab("Proportion") +xlab("Housing")+  
 ggtitle("Proportion Subscribed to Term Deposit by Housing")+ theme(axis.text.x = element\_text(angle=90, hjust=1))



# Higher for clients with no housing loans, this looks like a good variable   
  
# loan vs response  
prop.table(table(Bank\_Full$y,Bank\_Full$loan),2)

##   
## no yes  
## no 0.87344273 0.93318609  
## yes 0.12655727 0.06681391

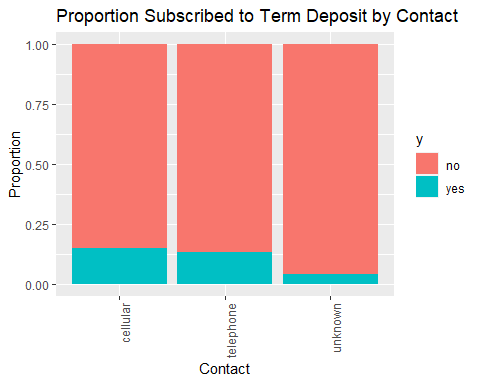
Bank\_Full %>%   
 ggplot(aes(x=loan, fill=y)) +   
 geom\_bar(position="fill") +   
 # scale\_fill\_viridis\_d() +  
 ylab("Proportion") +xlab("Loan")+  
 ggtitle("Proportion Subscribed to Term Deposit by Loan")+ theme(axis.text.x = element\_text(angle=90, hjust=1))



# Higher for clients with no loans, although most people are in the no category for loan anyway  
  
# contact vs response  
prop.table(table(Bank\_Full$y,Bank\_Full$contact),2)

##   
## cellular telephone unknown  
## no 0.85081100 0.86579491 0.95929339  
## yes 0.14918900 0.13420509 0.04070661

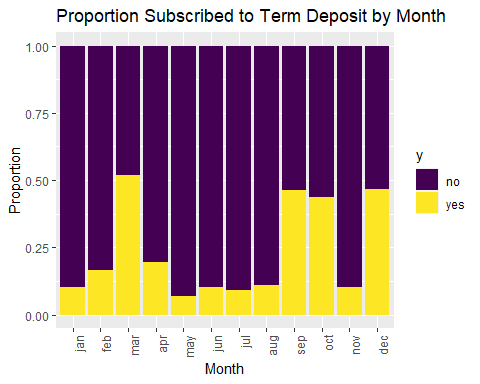
Bank\_Full %>%   
 ggplot(aes(x=contact, fill=y)) +   
 geom\_bar(position="fill") +   
 # scale\_fill\_viridis\_d() +  
 ylab("Proportion") +xlab("Contact")+  
 ggtitle("Proportion Subscribed to Term Deposit by Contact")+ theme(axis.text.x = element\_text(angle=90, hjust=1))



# This doesn't seem like it's going to be a good variable anyway, so maybe all the NAs won't be an issue.The clients contacted through cellular seemed to have opened a term deposit.  
  
# month vs response  
prop.table(table(Bank\_Full$y,Bank\_Full$month),2)

##   
## jan feb mar apr may jun  
## no 0.89878831 0.83352208 0.48008386 0.80320600 0.93280546 0.89777195  
## yes 0.10121169 0.16647792 0.51991614 0.19679400 0.06719454 0.10222805  
##   
## jul aug sep oct nov dec  
## no 0.90906454 0.88986714 0.53540587 0.56233062 0.89848866 0.53271028  
## yes 0.09093546 0.11013286 0.46459413 0.43766938 0.10151134 0.46728972

Bank\_Full %>%   
 ggplot(aes(x=month, fill=y)) +   
 geom\_bar(position="fill") +   
 scale\_fill\_viridis\_d() +  
 ylab("Proportion") +xlab("Month")+  
 ggtitle("Proportion Subscribed to Term Deposit by Month")+ theme(axis.text.x = element\_text(angle=90, hjust=1))



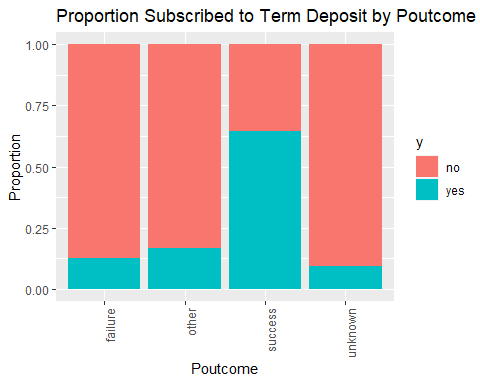
# This is interesting. Higher proportions in Dec, Mar, Oct, Sept; however, those seem to be the months with less data...  
summary(Bank\_Full$month)

## jan feb mar apr may jun jul aug sep oct nov dec   
## 1403 2649 477 2932 13766 5341 6895 6247 579 738 3970 214

# Dec only has 214, Mar has 477, Oct has 738, Sept has 579. All the other months have 1403-13766  
# also see breakdown by year  
  
# outcome vs response  
prop.table(table(Bank\_Full$y,Bank\_Full$poutcome),2)

##   
## failure other success unknown  
## no 0.87390329 0.83315217 0.35274653 0.90838497  
## yes 0.12609671 0.16684783 0.64725347 0.09161503

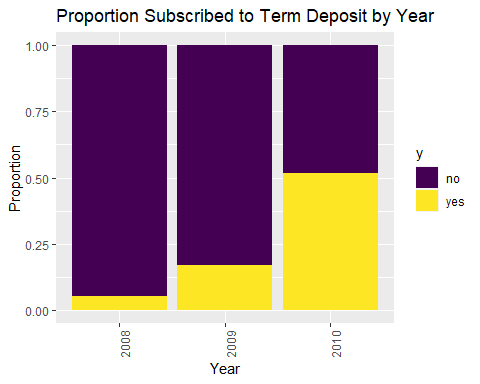
Bank\_Full %>%   
 ggplot(aes(x=poutcome, fill=y)) +   
 geom\_bar(position="fill") +   
 # scale\_fill\_viridis\_d() +  
 ylab("Proportion") +xlab("Poutcome")+  
 ggtitle("Proportion Subscribed to Term Deposit by Poutcome")+ theme(axis.text.x = element\_text(angle=90, hjust=1))



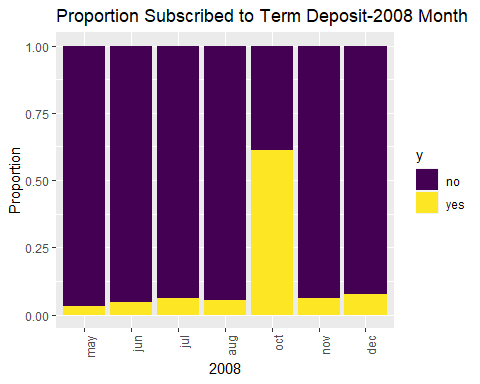
# Well, even though we have a lot of NAs, success is a really strong predictor of our outcome  
# We probably need to find a way to incorporate that.  
  
# year vs response  
prop.table(table(Bank\_Full$y,Bank\_Full$year),2)

##   
## 2008 2009 2010  
## no 0.94947528 0.82936348 0.48396947  
## yes 0.05052472 0.17063652 0.51603053

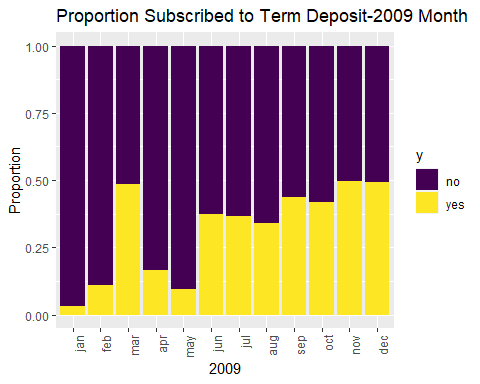
Bank\_Full %>%   
 ggplot(aes(x=year, fill=y)) +   
 geom\_bar(position="fill") +   
 scale\_fill\_viridis\_d() +  
 ylab("Proportion") +xlab("Year")+  
 ggtitle("Proportion Subscribed to Term Deposit by Year")+ theme(axis.text.x = element\_text(angle=90, hjust=1))



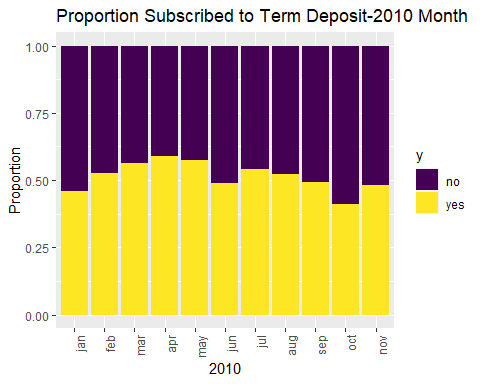
# Well this seems important.Slightly higher for 2010. There is a lot of missing values (unknowns) in 2008. This was also the financial crisis. We see lot of yes and no for 2010 are can be considered a great dataset for prediction. We will use 2008 and 2009 to understand the data but the data seems to be more clean and normal in 2010.  
  
# Break down months by year  
#Year=2008  
Bank\_Full %>%   
 filter(year == 2008) %>%  
 ggplot(aes(x=month, fill=y)) +   
 geom\_bar(position="fill") +   
 scale\_fill\_viridis\_d() +  
 ylab("Proportion") +xlab("2008")+  
 ggtitle("Proportion Subscribed to Term Deposit-2008 Month")+ theme(axis.text.x = element\_text(angle=90, hjust=1))



#Oct month shows higher on the Yes side  
  
#Year=2009  
Bank\_Full %>%   
 filter(year == 2009) %>%  
 ggplot(aes(x=month, fill=y)) +   
 geom\_bar(position="fill") +   
 scale\_fill\_viridis\_d() +  
 ylab("Proportion") +xlab("2009")+  
 ggtitle("Proportion Subscribed to Term Deposit-2009 Month")+ theme(axis.text.x = element\_text(angle=90, hjust=1))



#Higher from Oct until dec  
  
#Year=2010  
Bank\_Full %>%   
 filter(year == 2010) %>%  
 ggplot(aes(x=month, fill=y)) +   
 geom\_bar(position="fill") +   
 scale\_fill\_viridis\_d() +  
 ylab("Proportion") +xlab("2010")+  
 ggtitle("Proportion Subscribed to Term Deposit-2010 Month")+ theme(axis.text.x = element\_text(angle=90, hjust=1))

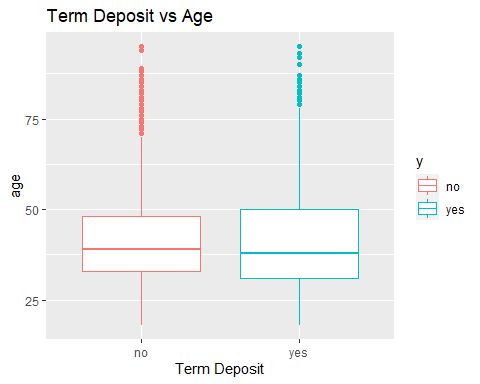


#Year 2010 seema to have had more cleints with term deposits than year 2008 and 2009. The yes and no are almost equal.The 2010 data looks better for prediction compared to 2008 and 2009.  
#The Yes on term deposit seems to be increasing with year.

# continuous variables  
# Age vs Response  
t(aggregate(age~y,data=Bank\_Full,summary))

## [,1] [,2]   
## y "no" "yes"   
## age.Min. "18.00000" "18.00000"  
## age.1st Qu. "33.00000" "31.00000"  
## age.Median "39.00000" "38.00000"  
## age.Mean "40.83899" "41.67007"  
## age.3rd Qu. "48.00000" "50.00000"  
## age.Max. "95.00000" "95.00000"

Bank\_Full %>%  
 ggplot(aes(x=y, y=age, color=y)) +  
 geom\_boxplot() +  
 scale\_fill\_viridis\_d() +  
 ggtitle("Term Deposit vs Age") +  
 xlab("Term Deposit")



# clients between age 35 to 50 seems to be more likely to open a term deposit  
  
# Balance vs Response  
t(aggregate(balance~y,data=Bank\_Full,summary))

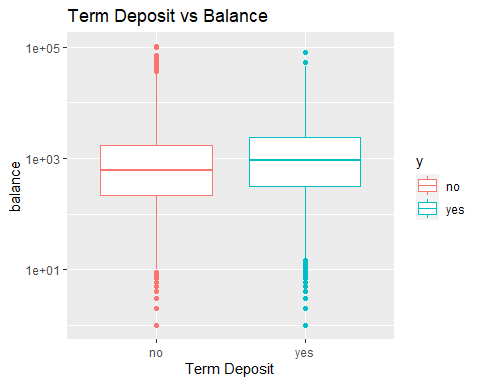
## [,1] [,2]   
## y "no" "yes"   
## balance.Min. " -8019.000" " -3058.000"  
## balance.1st Qu. " 58.000" " 210.000"  
## balance.Median " 417.000" " 733.000"  
## balance.Mean " 1303.715" " 1804.268"  
## balance.3rd Qu. " 1345.000" " 2159.000"  
## balance.Max. "102127.000" " 81204.000"

Bank\_Full %>%  
 ggplot(aes(x=y, y=balance, color=y)) +  
 geom\_boxplot() +  
 scale\_fill\_viridis\_d() +  
 ggtitle("Term Deposit vs Balance") +  
 xlab("Term Deposit")+scale\_y\_log10()

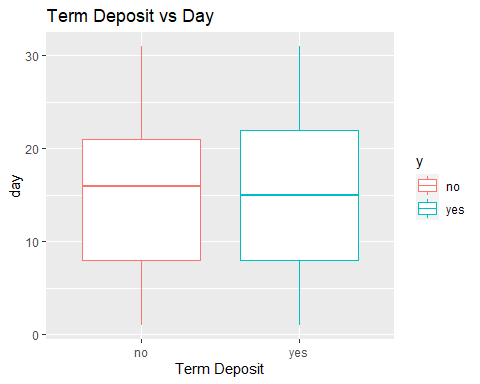
## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous y-axis

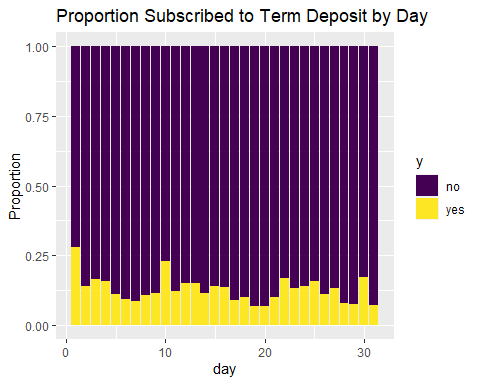
## Warning: Removed 7280 rows containing non-finite values (stat\_boxplot).



# looks like it could benefit from a transformation  
# The clients with more balance seems to be having a term deposit Yes. There seems to be a leverage point on Nos.The clients who opened a term deposit had a more balance than the ones who did not have a term deposit  
  
# Day vs Response (not sure this is the best way to look at this)  
Bank\_Full %>%  
 ggplot(aes(x=y, y=day, color=y)) +  
 geom\_boxplot() +  
 scale\_fill\_viridis\_d() +  
 ggtitle("Term Deposit vs Day") +  
 xlab("Term Deposit")



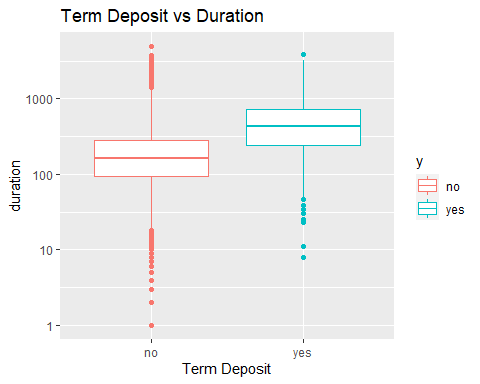
#Last Day contacted of the month seems to be almost same for those who had a opened term deposit or not.  
  
# Probably better to do this:  
Bank\_Full %>%   
 ggplot(aes(x=day, fill=y)) +   
 geom\_bar(position="fill") +   
 scale\_fill\_viridis\_d() +  
 ylab("Proportion") +  
 ggtitle("Proportion Subscribed to Term Deposit by Day")



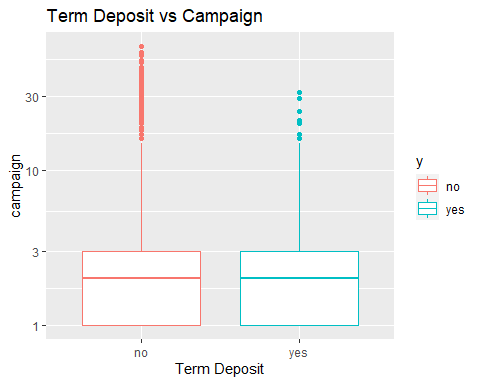
# maybe more likely on days 1, 10, 22, 30? Maybe something to do with paydays  
  
# Duration vs Response  
Bank\_Full %>%  
 ggplot(aes(x=y, y=duration, color=y)) +  
 geom\_boxplot() +  
 scale\_fill\_viridis\_d() +  
 ggtitle("Term Deposit vs Duration") +  
 xlab("Term Deposit")+scale\_y\_log10()

## Warning: Transformation introduced infinite values in continuous y-axis

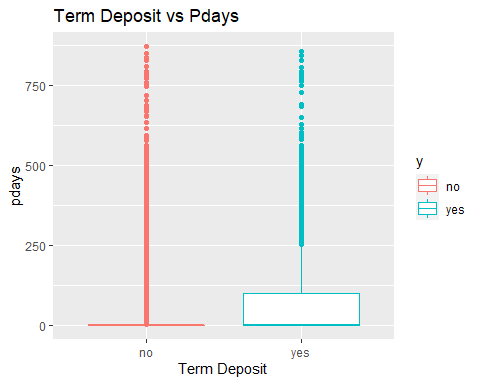
## Warning: Removed 3 rows containing non-finite values (stat\_boxplot).



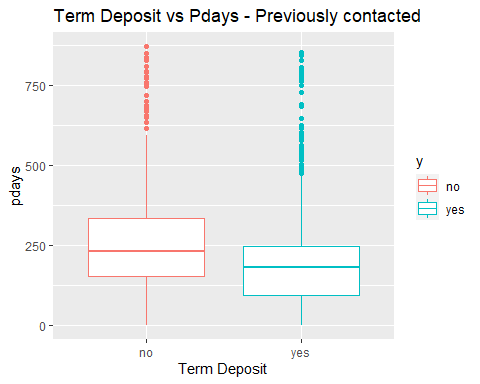
# Definitely a difference there - yes tends to have a longer duration  
# might want to look at that outlier in the high 4000s of duration  
  
# Campaign vs Response  
Bank\_Full %>%  
 ggplot(aes(x=y, y=campaign, color=y)) +  
 geom\_boxplot() +  
 scale\_fill\_viridis\_d() +  
 ggtitle("Term Deposit vs Campaign") +  
 xlab("Term Deposit")+scale\_y\_log10()



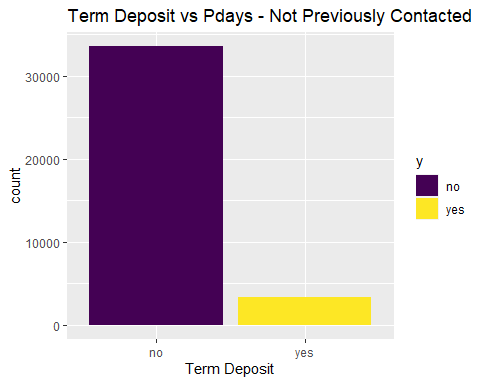
# The number of contacts performed during this campaign days is almost the same as for having term deposit and not having a term deposit   
# There might be a cut-off above which there are only No's (like >35 or 40)  
  
# Pdays vs Response (NOTE the meaning of -1)  
Bank\_Full %>%  
 ggplot(aes(x=y, y=pdays, color=y)) +  
 geom\_boxplot() +  
 scale\_fill\_viridis\_d() +  
 ggtitle("Term Deposit vs Pdays") +  
 xlab("Term Deposit")



# let's see what this looks like when we take out the -1:  
#The number of days that passed by after the client was last contacted from a previous campaign had more yes than nos.  
Bank\_Full %>%  
 filter(pdays != -1) %>%  
 ggplot(aes(x=y, y=pdays, color=y)) +  
 geom\_boxplot() +  
 scale\_fill\_viridis\_d() +  
 ggtitle("Term Deposit vs Pdays - Previously contacted") +  
 xlab("Term Deposit")



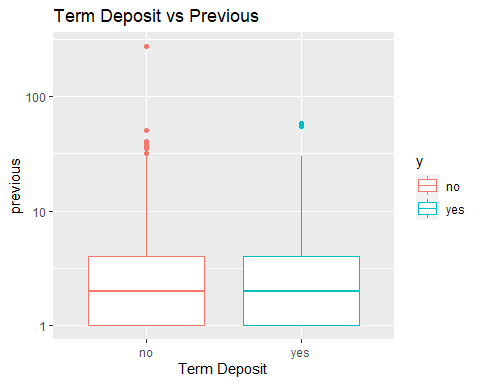
# So for those previously contacted: less days are associated with more yes in response  
# let's look at those not previously contacted:  
  
Bank\_Full %>%  
 filter(pdays == -1) %>%  
 ggplot(aes(x=y, fill=y)) +  
 geom\_bar() +  
 scale\_fill\_viridis\_d() +  
 ggtitle("Term Deposit vs Pdays - Not Previously Contacted") +  
 xlab("Term Deposit")



# look at count for not previously contacted. They are more than for no s than the yes s.  
  
#Bank\_Full %>% filter(pdays == -1) %>% count(y)  
# no:33570, yes:3384 (1/9 are yes)  
# look at count for previously contacted  
  
#Bank\_Full %>% filter(pdays != -1) %>% count(y)  
# No:6352, yes:1905 (1/3 are yes)  
# that's a big difference  
  
# Previous vs Response  
Bank\_Full %>%  
 ggplot(aes(x=y, y=previous, color=y)) +  
 geom\_boxplot() +  
 scale\_fill\_viridis\_d() +  
 ggtitle("Term Deposit vs Previous") +  
 xlab("Term Deposit")+scale\_y\_log10()

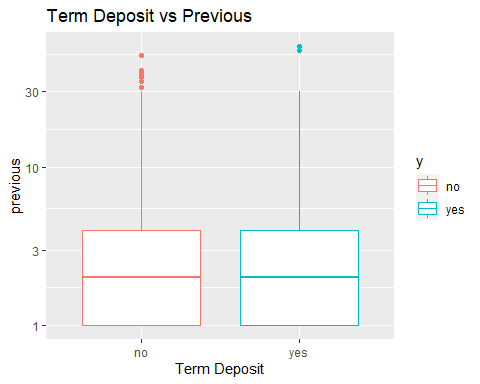
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 36954 rows containing non-finite values (stat\_boxplot).



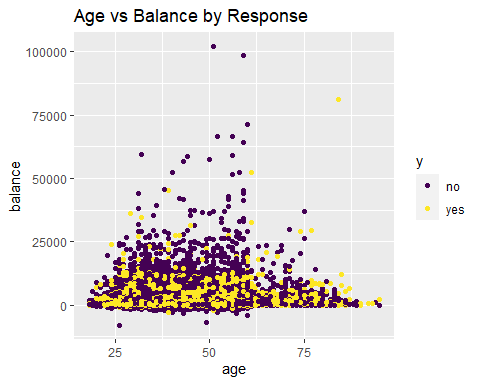
#The number of contacts performed before this campaign and for this client shows on an average the same for yes and no for having a term deposit.# outlier over 250  
# let's look at it w/o  
  
Bank\_Full %>%  
 filter(previous < 100) %>%  
 ggplot(aes(x=y, y=previous, color=y)) +  
 geom\_boxplot() +  
 scale\_fill\_viridis\_d() +  
 ggtitle("Term Deposit vs Previous") +  
 xlab("Term Deposit")+scale\_y\_log10()

## Warning: Transformation introduced infinite values in continuous y-axis  
  
## Warning: Removed 36954 rows containing non-finite values (stat\_boxplot).

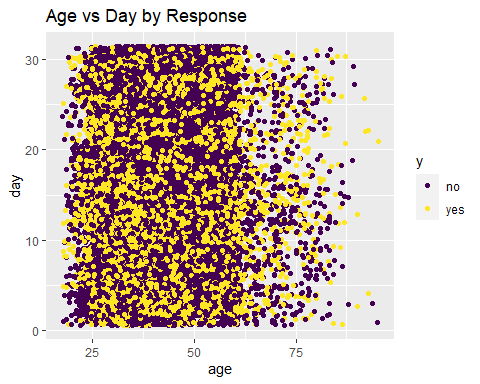


# hard to visualize. in the table, the mean for yes is higher though.

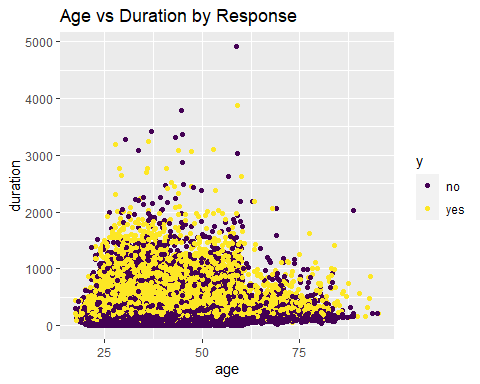
# paired scatterplots coded by response: AGE  
# (looking for possible interactions)  
# Age vs Balance  
Bank\_Full %>%  
 ggplot(aes(x=age, y=balance, color=y)) +  
 geom\_point()+  
#+scale\_y\_log10()+scale\_x\_log10()  
 scale\_color\_viridis\_d() +  
 ggtitle("Age vs Balance by Response")



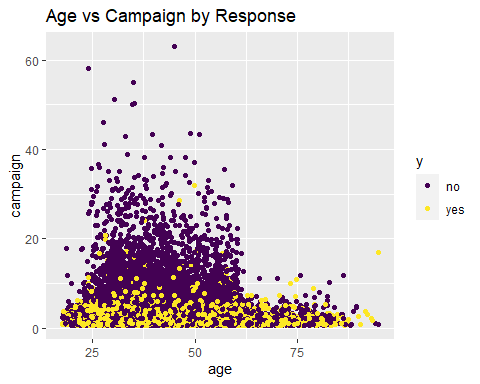
#Looks like they are linearly correlated to each other. We also see few outlier that are high leverage points for no (2 observations) and yes (1 observation).This could be a valid interaction as age and balance are linearly correlated.  
  
# Age vs Day  
Bank\_Full %>%  
 ggplot(aes(x=age, y=day, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Age vs Day by Response")



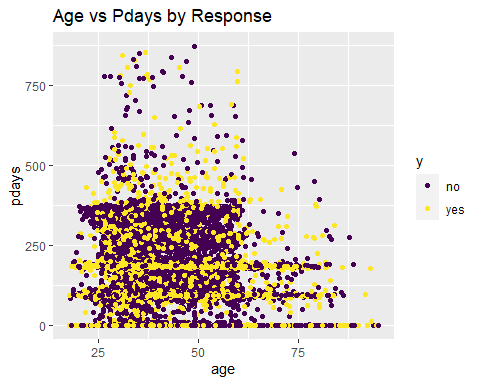
#Age and last contact day of the month dont seem to be valid interactions  
  
# Age vs Duration  
Bank\_Full %>%  
 ggplot(aes(x=age, y=duration, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Age vs Duration by Response")



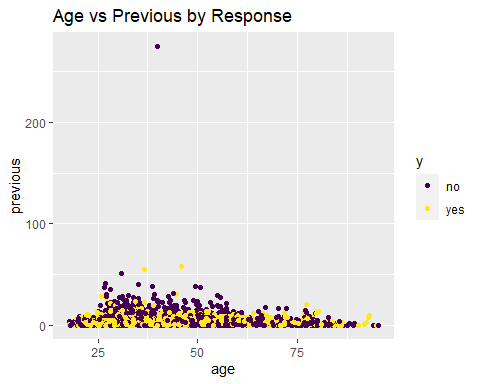
#Age and last contact duration dont seem to be valid interactions  
  
# Age vs Campaign  
Bank\_Full %>%  
 ggplot(aes(x=age, y=campaign, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Age vs Campaign by Response")



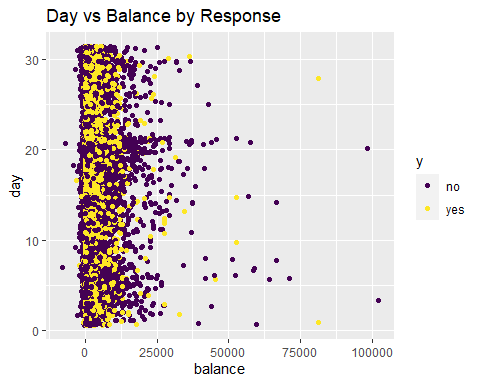
# There's a cut-off around 60 where people are no longer contacted by the campaign a ton of times.Age and campaign could play an import role in identifying the right candidate eligible for a term deposit.  
  
# Age vs Pdays  
Bank\_Full %>%  
 ggplot(aes(x=age, y=pdays, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Age vs Pdays by Response")



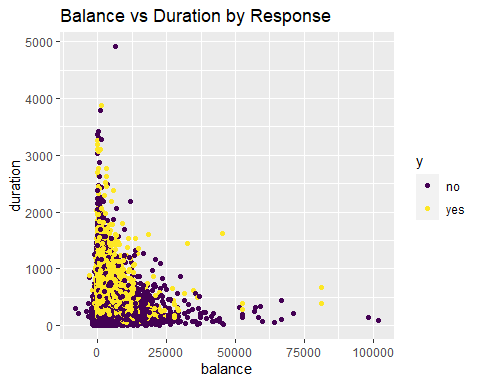
#Age and number of contacts performed during this campaign could be correlated and eligible for an interaction.  
  
# Age vs Previous  
Bank\_Full %>%  
 ggplot(aes(x=age, y=previous, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Age vs Previous by Response")



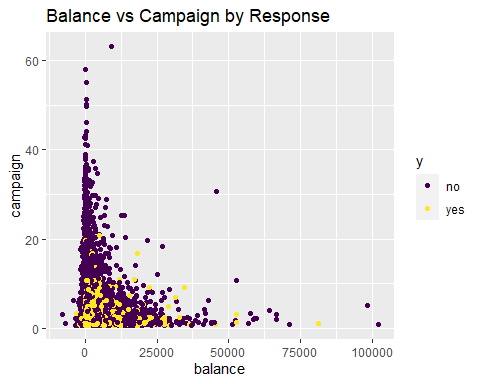
# note the outlier: previous = 275. Next closest is 58.  
#Age and number of contacts performed before this campaign could be potential for interactions.  
  
# paired scatterplots coded by response: BALANCE  
# (looking for possible interactions)  
# Balance vs Day  
Bank\_Full %>%  
 ggplot(aes(x=balance, y=day, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Day vs Balance by Response")



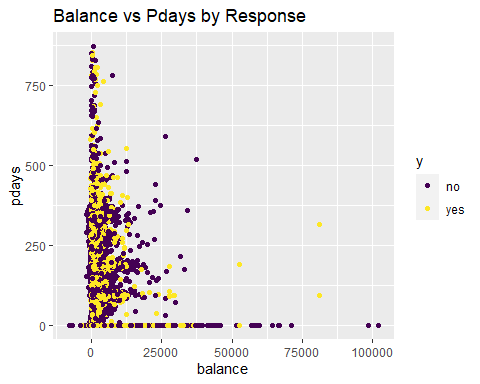
#Balance and last contact of the month could be potential for interaction.  
  
# Balance vs Duration  
Bank\_Full %>%  
 ggplot(aes(x=balance, y=duration, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Balance vs Duration by Response")



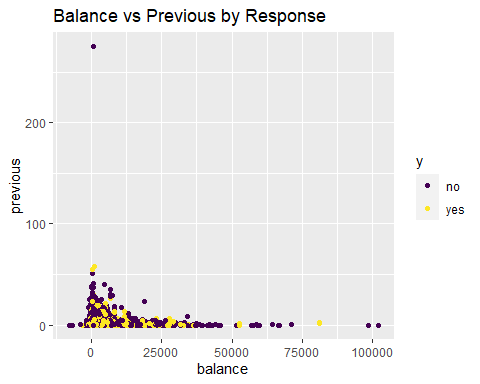
#Balance vs last contact duration dont se to be potential for interactionsS  
  
# Balance vs Campaign  
Bank\_Full %>%  
 ggplot(aes(x=balance, y=campaign, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Balance vs Campaign by Response")



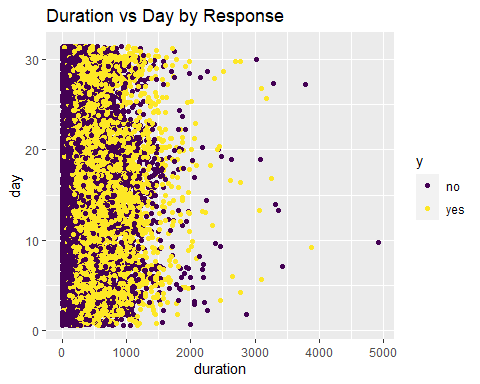
# As noted previously, there aren't any Yes's above a certain # in campaign  
  
# Balance vs Pdays  
Bank\_Full %>%  
 ggplot(aes(x=balance, y=pdays, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Balance vs Pdays by Response")



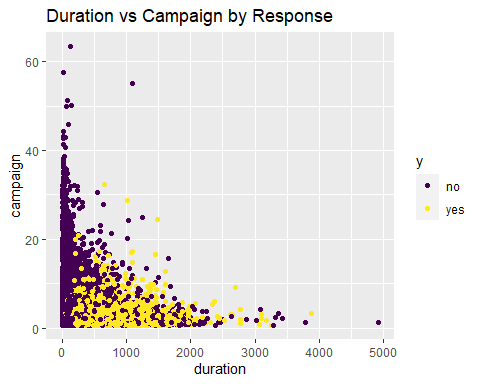
# note the outlier.Not potential for interactionsS  
  
# Balance vs Previous  
Bank\_Full %>%  
 ggplot(aes(x=balance, y=previous, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Balance vs Previous by Response")



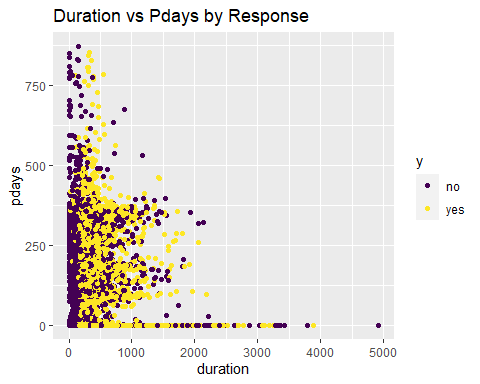
# note the outlier: previous = 275. Next closest is 58.  
  
# paired scatterplots coded by response: DURATION  
# (looking for possible interactions)  
# Duration vs Day  
Bank\_Full %>%  
 ggplot(aes(x=duration, y=day, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Duration vs Day by Response")



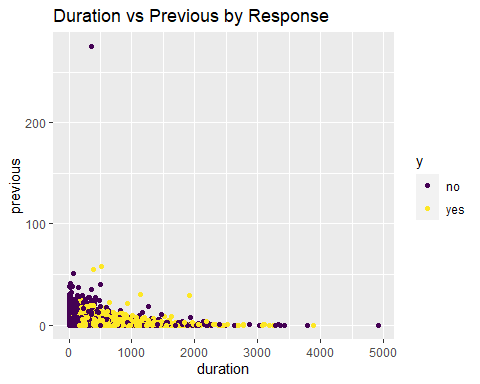
# note the outlier:  
  
# Duration vs Campaign  
Bank\_Full %>%  
 ggplot(aes(x=duration, y=campaign, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Duration vs Campaign by Response")



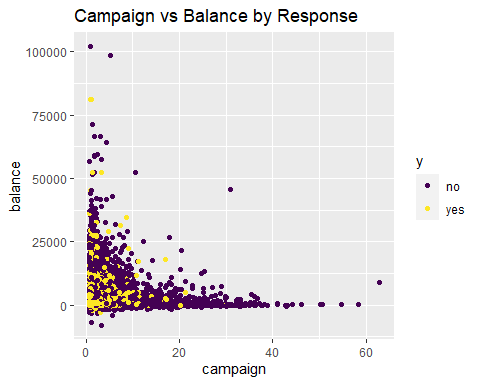
#Last contact duration and campain contacts dont seem to have an interaction.There are 1 outliers on Duration and 2 on campaign  
  
# Duration vs Pdays  
Bank\_Full %>%  
 ggplot(aes(x=duration, y=pdays, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Duration vs Pdays by Response")



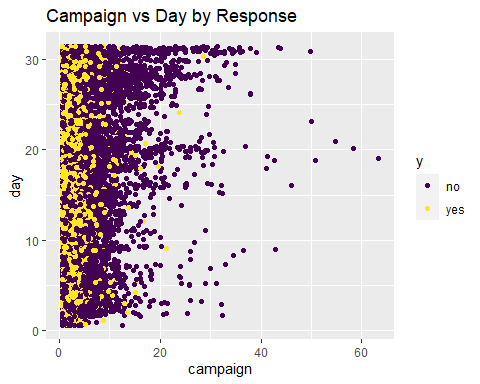
#Last contacted duration and pdays passed after client was last contacted could have an interaction.  
  
# Duration vs Previous  
Bank\_Full %>%  
 ggplot(aes(x=duration, y=previous, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Duration vs Previous by Response")



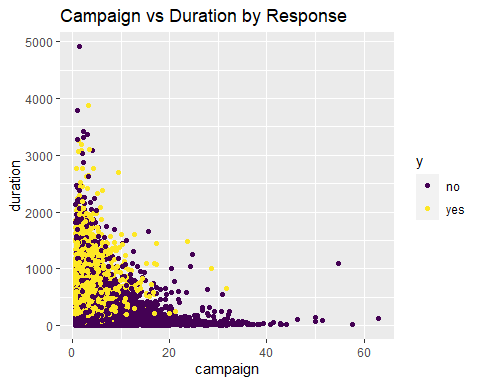
# note the outlier: previous = 275. Next closest is 58.  
  
# paired scatterplots coded by response: Campaign  
# (looking for possible interactions)  
# Campaign vs Balance  
Bank\_Full %>%  
 ggplot(aes(x=campaign, y=balance, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Campaign vs Balance by Response")



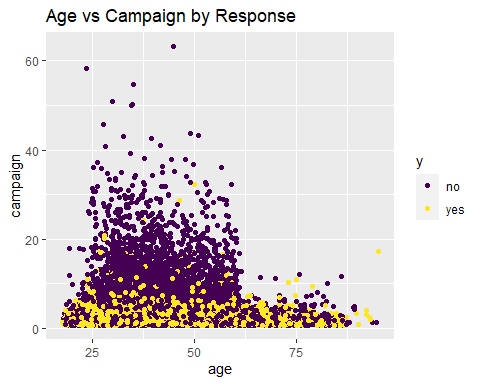
#Look at the outlier  
  
# Campaign vs Day  
Bank\_Full %>%  
 ggplot(aes(x=campaign, y=day, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Campaign vs Day by Response")



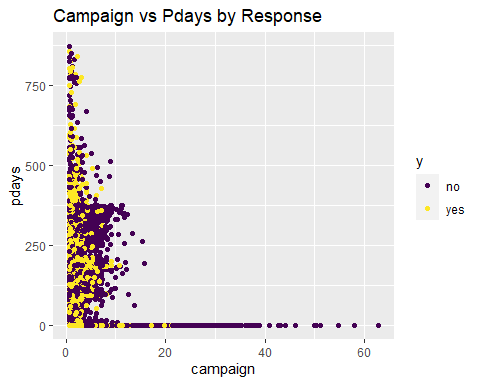
# Campaign vs Duration  
Bank\_Full %>%  
 ggplot(aes(x=campaign, y=duration, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Campaign vs Duration by Response")



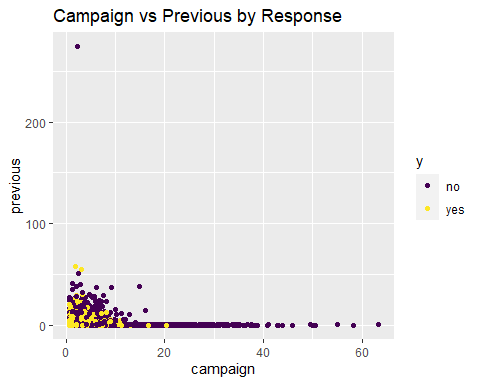
# Age vs Campaign  
Bank\_Full %>%  
 ggplot(aes(x=age, y=campaign, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Age vs Campaign by Response")



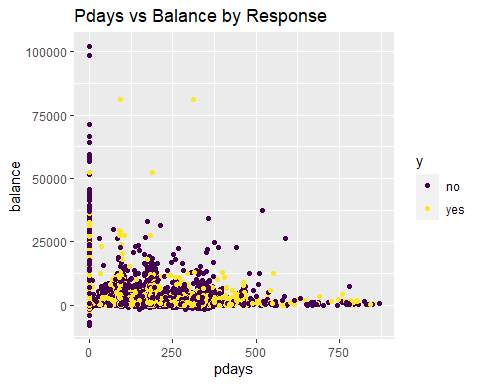
# There's a cut-off around 60 where people are no longer contacted by the campaign a ton of times.  
  
# Campaign vs Pdays  
Bank\_Full %>%  
 ggplot(aes(x=campaign, y=pdays, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Campaign vs Pdays by Response")



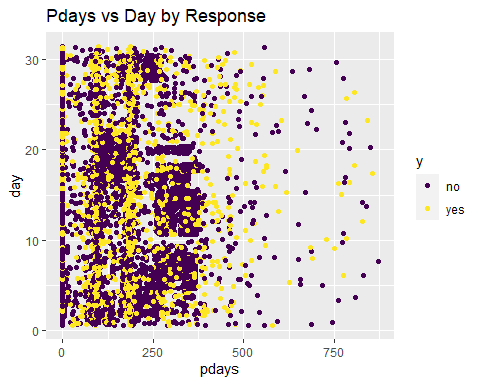
# Campaign vs Previous  
Bank\_Full %>%  
 ggplot(aes(x=campaign, y=previous, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Campaign vs Previous by Response")



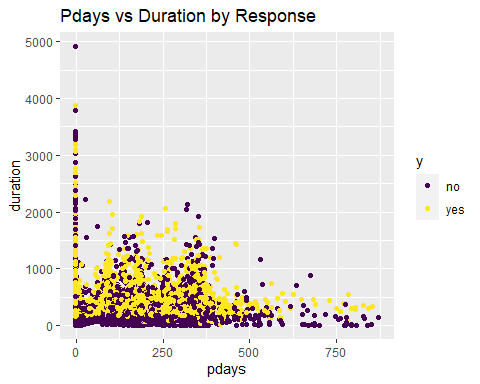
# So the people who were contacted a ton of times by this campaign were all people who had not been previously contacted before  
  
# paired scatterplots coded by response: PDAYS  
# (looking for possible interactions)  
# Pdays vs Balance  
Bank\_Full %>%  
 ggplot(aes(x=pdays, y=balance, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Pdays vs Balance by Response")



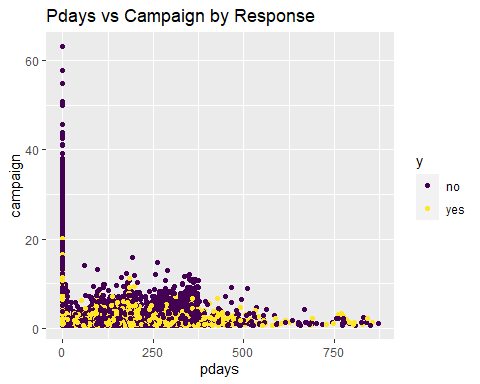
# Pdays vs Day  
Bank\_Full %>%  
 ggplot(aes(x=pdays, y=day, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Pdays vs Day by Response")



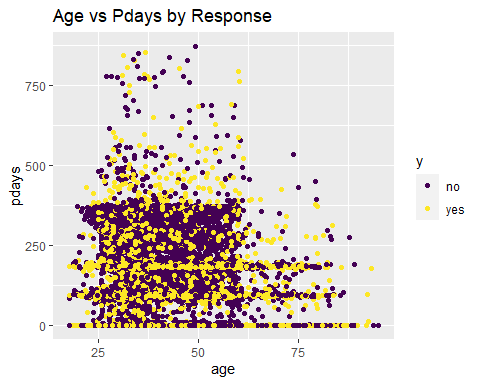
# Pdays vs Duration  
Bank\_Full %>%  
 ggplot(aes(x=pdays, y=duration, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Pdays vs Duration by Response")



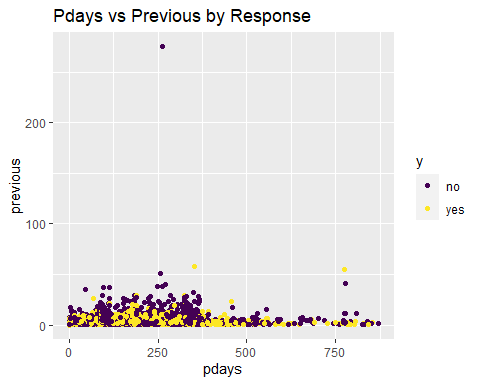
# Pdays vs Campaign  
Bank\_Full %>%  
 ggplot(aes(x=pdays, y=campaign, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Pdays vs Campaign by Response")



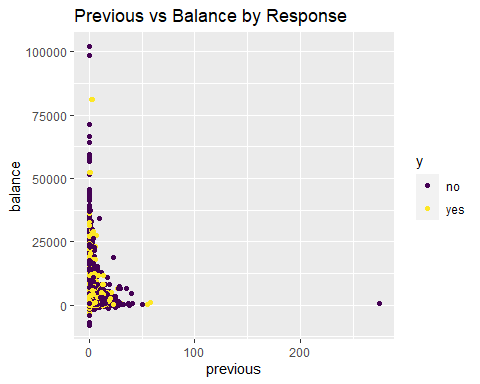
# Age vs Pdays  
Bank\_Full %>%  
 ggplot(aes(x=age, y=pdays, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Age vs Pdays by Response")



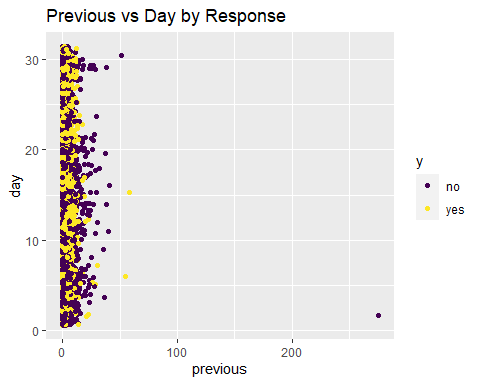
# Pdays vs Previous  
Bank\_Full %>%  
 ggplot(aes(x=pdays, y=previous, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Pdays vs Previous by Response")



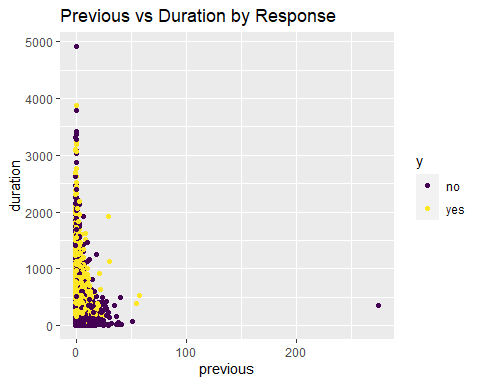
# note the outlier: previous = 275. Next closest is 58.  
  
# paired scatterplots coded by response: PREVIOUS  
# (looking for possible interactions)  
# Previous vs Balance  
Bank\_Full %>%  
 ggplot(aes(x=previous, y=balance, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Previous vs Balance by Response")



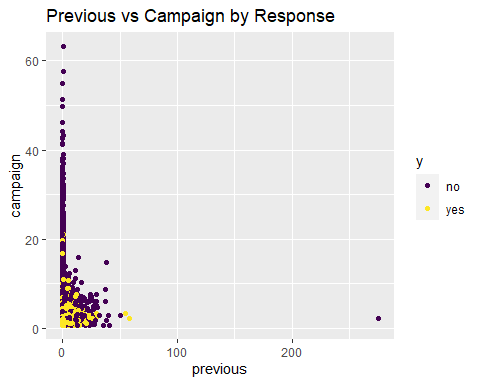
#There is the outlier.  
  
# Previous vs Day  
Bank\_Full %>%  
 ggplot(aes(x=previous, y=day, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Previous vs Day by Response")



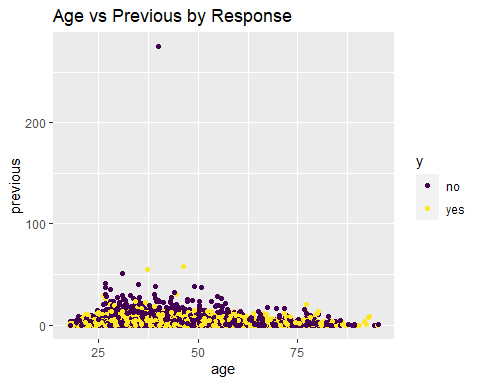
#The outlier is seen  
  
# Previous vs Duration  
Bank\_Full %>%  
 ggplot(aes(x=previous, y=duration, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Previous vs Duration by Response")



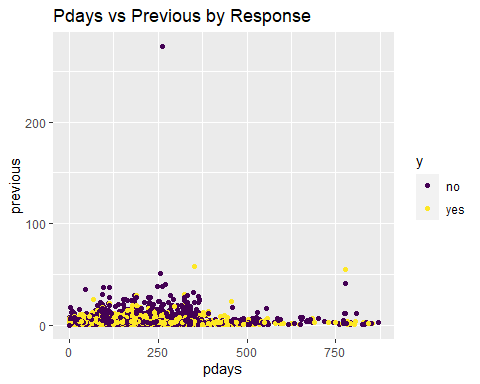
#The outlier is seen  
  
# Previous vs Campaign  
Bank\_Full %>%  
 ggplot(aes(x=previous, y=campaign, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Previous vs Campaign by Response")



#The outlier is seen  
  
# Age vs Previous  
Bank\_Full %>%  
 ggplot(aes(x=age, y=previous, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Age vs Previous by Response")

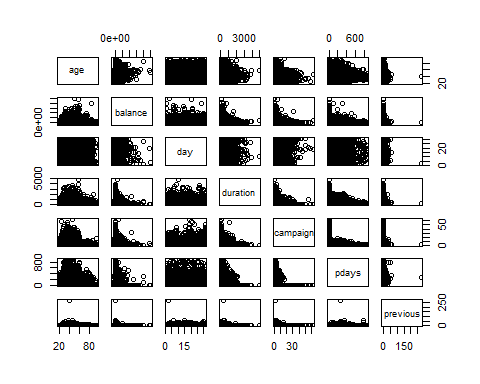


# note the outlier: previous = 275. Next closest is 58.  
  
# Pdays vs Previous  
Bank\_Full %>%  
 ggplot(aes(x=pdays, y=previous, color=y)) +  
 geom\_point(position="jitter") +  
 scale\_color\_viridis\_d() +  
 ggtitle("Pdays vs Previous by Response")



# note the outlier: previous = 275. Next closest is 58.

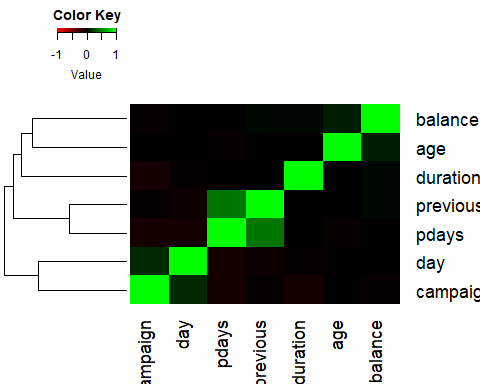
# Correlations between continuous variable  
# Exploring multicollinearity  
pairs(Bank\_Full[,c(1,6,10,12,13,14,15)])



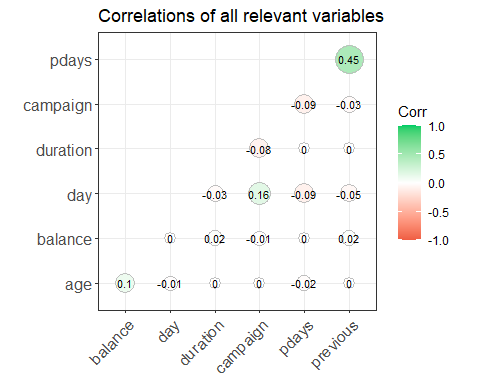
my.cor<-cor(Bank\_Full[,c(1,6,10,12,13,14,15)])  
my.cor

## age balance day duration campaign  
## age 1.000000000 0.097782739 -0.009120046 -0.004648428 0.004760312  
## balance 0.097782739 1.000000000 0.004502585 0.021560380 -0.014578279  
## day -0.009120046 0.004502585 1.000000000 -0.030206341 0.162490216  
## duration -0.004648428 0.021560380 -0.030206341 1.000000000 -0.084569503  
## campaign 0.004760312 -0.014578279 0.162490216 -0.084569503 1.000000000  
## pdays -0.023758014 0.003435322 -0.093044074 -0.001564770 -0.088627668  
## previous 0.001288319 0.016673637 -0.051710497 0.001203057 -0.032855290  
## pdays previous  
## age -0.023758014 0.001288319  
## balance 0.003435322 0.016673637  
## day -0.093044074 -0.051710497  
## duration -0.001564770 0.001203057  
## campaign -0.088627668 -0.032855290  
## pdays 1.000000000 0.454819635  
## previous 0.454819635 1.000000000

#pairs(Bank\_Full[,c(1,6,10,12,13,14,15)],col=Bank\_Full$y)  
# Heatmap  
heatmap.2(my.cor,col=redgreen(75),   
 density.info="none", trace="none", dendrogram=c("row"),   
 symm=F,symkey=T,symbreaks=T, scale="none")



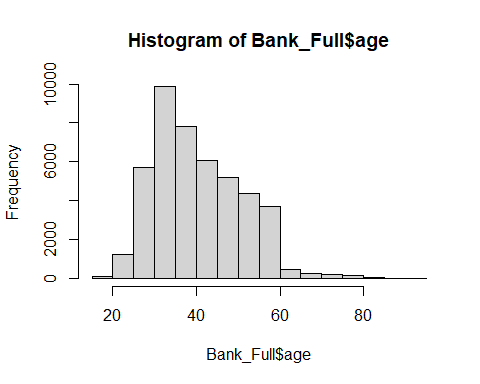
corr <- Bank\_Full %>% dplyr::select(age,balance,day,duration,campaign,pdays,previous)   
corr <- round(cor(corr), 2)  
ggcorrplot(corr, type = "lower",  
 lab = TRUE, lab\_size = 3, method = "circle",  
 colors = c("tomato2", "white", "springgreen3"),  
 title = "Correlations of all relevant variables",  
 ggtheme = theme\_bw())



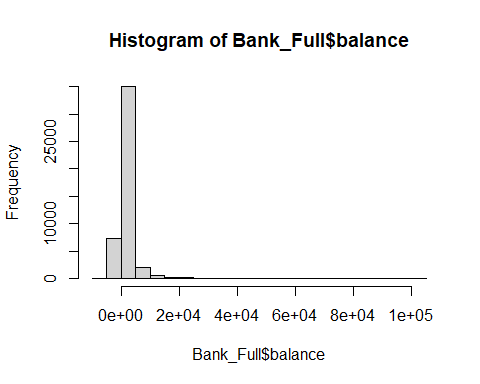
# previous and pdays correlation = 0.45. Will also be correlated with the dummy variable we build  
  
# multicollinearity w/ categorical speculation  
# Hyp: older people would be more likely to have landline  
#plot(Bank\_Full$age~Bank\_Full$contact,col=c("red","blue"))  
# Yes, but there's overlap  
# Hyp: balance and education   
#plot(Bank\_Full$balance~Bank\_Full$education,col=c("red","blue"))  
#plot(log(Bank\_Full$balance)~Bank\_Full$education,col=c("red","blue"))  
# note I know there's zeros, just looking   
# slight but not dramatic  
# age and marital  
#plot(Bank\_Full$age~Bank\_Full$marital,col=c("red","blue"))  
# single people tend to be younger, as you'd expect  
# housing and loan  
#plot(Bank\_Full$housing~Bank\_Full$loan,col=c("red","blue")) # similar ratios

#model.main<-glm(age~., data=Bank\_Full)  
#vif(model.main)  
#Using this tool, GVIF is the same as VIF for continuous predictors only  
#For categorical predictors, the value GVIG^(1/(2\*df)) should be squared and interpreted  
#as a usual vif type metric.The following code can be used to interpret VIFs like we   
#discussed in class.  
#(vif(model.main)[,3])^2  
#VIF look good.Nothing seems to be greater tha 10

# Normality will be a concern for LDA/QDA:  
hist(Bank\_Full$age) #fine



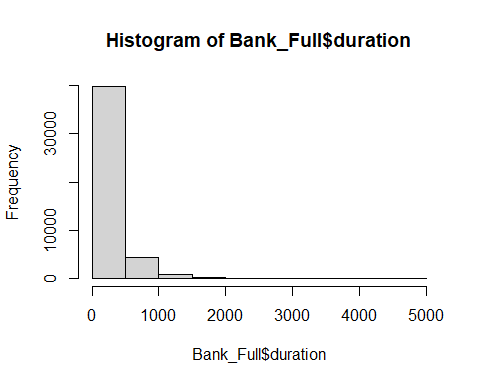
hist(Bank\_Full$balance) #skew



range(Bank\_Full$balance) # This is skewed but has negative numbers

## [1] -8019 102127

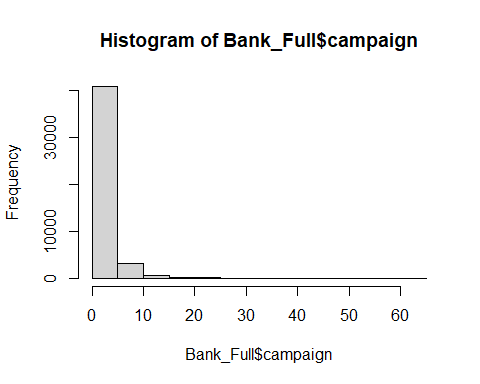
hist(Bank\_Full$duration) #skew



range(Bank\_Full$duration) # There are 3 zeros...

## [1] 0 4918

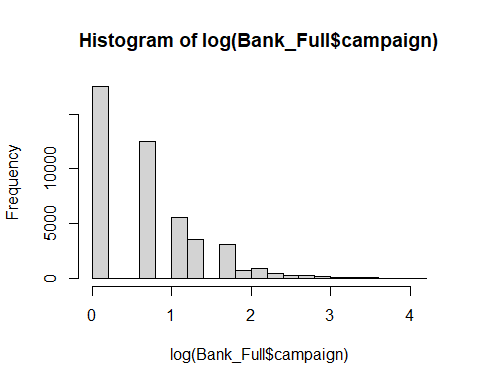
hist(Bank\_Full$campaign)#skew



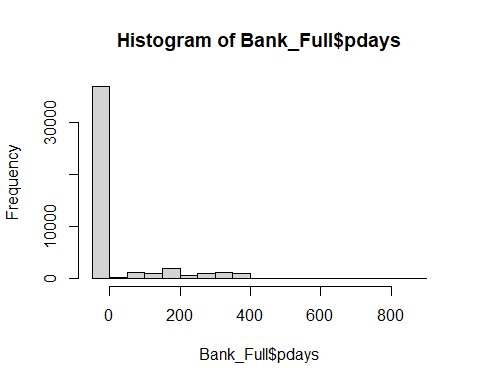
range(Bank\_Full$campaign) # 1-3

## [1] 1 63

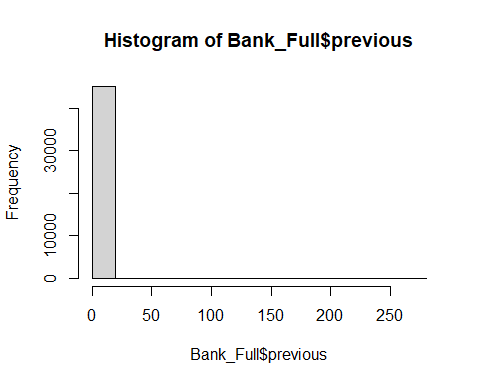
hist(log(Bank\_Full$campaign)) # still doesn't look great



hist(Bank\_Full$pdays) #skewed, but there's negative -1's that we will likely change to 0



hist(Bank\_Full$previous)

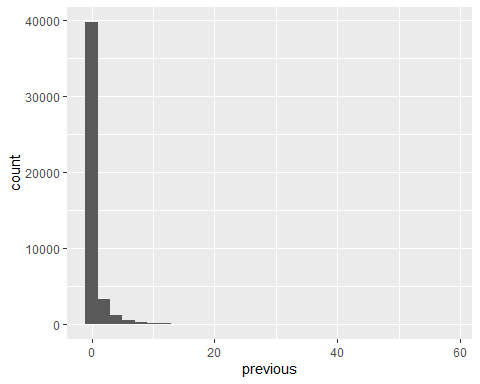


range(Bank\_Full$previous) #0-275 (but next lowest is 58)

## [1] 0 275

Bank\_Full %>%  
 filter(previous <60) %>%  
 ggplot(aes(x=previous)) +  
 geom\_histogram()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



# Make indicator variable for pdays variable  
# Step 1: change pdays -1 values to 0  
# Note: There are no pre-existing 0 values in pdays. -1 was arbitrary. By changing it to 0, we can have the beta for this coefficient not turn on when people not previously contacted  
# Step 2: make indicator variable that is 1 if pdays is 0, and 0 if pdays is any other value  
# Note: This turns on the dummy variable if someone wasn't previously contacted. If they were, it turns it off because the pdays variable is modeling it  
# Double checking my work:  
#Bank\_Full %>%  
 #filter(pdays == -1) %>%  
 #count()  
# there are 36954 -1 values in pdays  
# Change pdays -1 to 0:  
#Bank\_Full$pdays[Bank\_Full$pdays==-1] = 0  
# check the count  
#Bank\_Full %>%  
 #filter(pdays == 0) %>%  
 #count() #looks right  
# Make dummy variable and change to factor  
#Bank\_Full = Bank\_Full %>%  
 #mutate(pcontact = case\_when(  
 #pdays == 0 ~ 1,  
 #pdays != 0 ~ 0  
 #))  
#Bank\_Full$pcontact = factor(Bank\_Full$pcontact)  
# Note: pcontact = 1 means "not previously contacted", pcontact = 0 means "previously contacted"

# look at the identified possible outliers  
# Duration  
summary(Bank\_Full$duration)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0 103.0 180.0 258.2 319.0 4918.0

Bank\_Full %>%   
 filter(Bank\_Full$duration == 4918)

## age job marital education default balance housing loan contact day  
## 1 59 technician married tertiary no 6573 yes no telephone 10  
## month duration campaign pdays previous poutcome y id year  
## 1 nov 4918 1 -1 0 unknown no 24149 2008

# 4918 is max (response is no), next value following it is 3881  
# 4918/60 = 81.9 minutes. 3881/60 = 64.68 minutes. Median = 3 min, Mean = 4.3 min, 3rd Q = 5.31 min  
# Can we justify throwing this out by saying 81.9 minutes is not a good use of time? That's 19 average phone calls...  
# Also this is Nov 2008. Depending on what we do, it may be a non-issue  
# Previous  
summary(Bank\_Full$previous)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0000 0.0000 0.5803 0.0000 275.0000

# Might make more sense to look at it w/o the 0s:  
Bank\_Full %>%  
 filter(previous >0) %>%  
 summarise(mean = mean(previous),   
 median = median(previous),   
 min = min(previous),  
 max = max(previous),  
 quantile = quantile(previous))

## mean median min max quantile  
## 1 3.177546 2 1 275 1  
## 2 3.177546 2 1 275 1  
## 3 3.177546 2 1 275 2  
## 4 3.177546 2 1 275 4  
## 5 3.177546 2 1 275 275

# 275 (response is no), next value is 58   
Bank\_Full %>%  
 filter(Bank\_Full$previous == 275)

## age job marital education default balance housing loan contact day  
## 1 40 management married tertiary no 543 yes no cellular 2  
## month duration campaign pdays previous poutcome y id year  
## 1 feb 349 2 262 275 other no 29183 2009

# I suppose we could say that we don't have enough data in this date range to be useful  
# So we can restrict the range down to <100 or something.  
# Removing the 2 outliers:  
# double checking my work  
nrow(Bank\_Full) # 45211

## [1] 45211

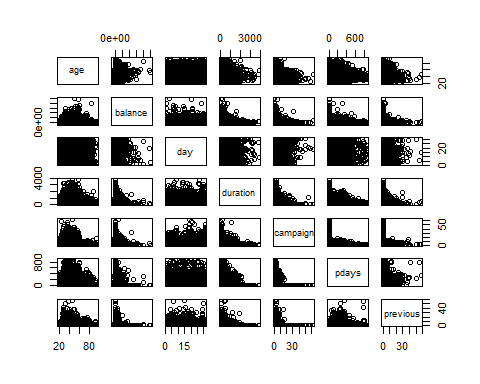
Bank\_Full = subset(Bank\_Full, duration != 4918)  
nrow(Bank\_Full) #45210

## [1] 45210

Bank\_Full = subset(Bank\_Full, previous != 275)  
nrow(Bank\_Full) #45209

## [1] 45209

#PCA as part of the EDA  
# Continuous predictors  
#For the 3 Years  
# This includes: age, balance, day, duration, campaign, pdays, previous  
reduced<-Bank\_Full[,c(1,6,10,12,13,14,15)]  
pairs(reduced)



#Let's take a quick look at the summary statistics and in particular lets calculate the variance of each variable and add them up to obtain the total variance.  
apply(reduced,2,summary)

## age balance day duration campaign pdays previous  
## Min. 18.00000 -8019.000 1.00000 0.000 1.000000 -1.00000 0.0000000  
## 1st Qu. 33.00000 72.000 8.00000 103.000 1.000000 -1.00000 0.0000000  
## Median 39.00000 448.000 16.00000 180.000 2.000000 -1.00000 0.0000000  
## Mean 40.93583 1362.175 15.80685 258.058 2.763897 40.19383 0.5742662  
## 3rd Qu. 48.00000 1428.000 21.00000 319.000 3.000000 -1.00000 0.0000000  
## Max. 95.00000 102127.000 31.00000 3881.000 63.000000 871.00000 58.0000000

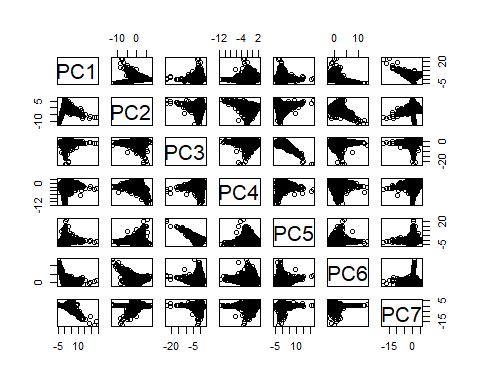
var.raw<-apply(reduced,2,var)  
var.raw

## age balance day duration campaign pdays   
## 1.127559e+02 9.270394e+06 6.926171e+01 6.584300e+04 9.598076e+00 1.002508e+04   
## previous   
## 3.640261e+00

#Total variance  
sum(var.raw)

## [1] 9346457

#Running PCA is relatively straight forward. The following script conducts a PCA using the covariance matrix (nonstandardarized #variables) and stores the results in an object. This object contains the eigenvectors, eigenvalue, and the new principle #component vectors. Lets start by producing a correlation matrix to verify that new principle component variables are #uncorrelated.  
pc.result<-prcomp(Bank\_Full[,c(1,6,10,12,13,14,15)],scale.=TRUE)  
pc.scores<-pc.result$x  
pairs(pc.scores)



cor(pc.scores)

## PC1 PC2 PC3 PC4 PC5  
## PC1 1.000000e+00 1.646821e-15 7.240806e-16 7.513190e-16 1.061386e-15  
## PC2 1.646821e-15 1.000000e+00 -4.098392e-15 5.588093e-15 -2.452901e-15  
## PC3 7.240806e-16 -4.098392e-15 1.000000e+00 -1.011659e-14 3.121267e-14  
## PC4 7.513190e-16 5.588093e-15 -1.011659e-14 1.000000e+00 -1.358345e-14  
## PC5 1.061386e-15 -2.452901e-15 3.121267e-14 -1.358345e-14 1.000000e+00  
## PC6 4.651381e-16 -1.817970e-15 5.957578e-15 1.881012e-15 6.553160e-15  
## PC7 -1.156057e-15 -9.565370e-17 -1.828315e-15 1.509032e-15 -2.007877e-15  
## PC6 PC7  
## PC1 4.651381e-16 -1.156057e-15  
## PC2 -1.817970e-15 -9.565370e-17  
## PC3 5.957578e-15 -1.828315e-15  
## PC4 1.881012e-15 1.509032e-15  
## PC5 6.553160e-15 -2.007877e-15  
## PC6 1.000000e+00 6.575485e-16  
## PC7 6.575485e-16 1.000000e+00

#We can again verify that the total variance in the new PC variables is exactly the same as the original data. The eigenvectors are stored inside of "pc.result" as well in the "rotation" object.  
var.pca<-apply(pc.scores,2,var)  
var.pca

## PC1 PC2 PC3 PC4 PC5 PC6 PC7   
## 1.5895869 1.1600390 1.0974117 0.9754226 0.8972295 0.8260487 0.4542615

#Total Variance of PC's  
sum(var.pca)

## [1] 7

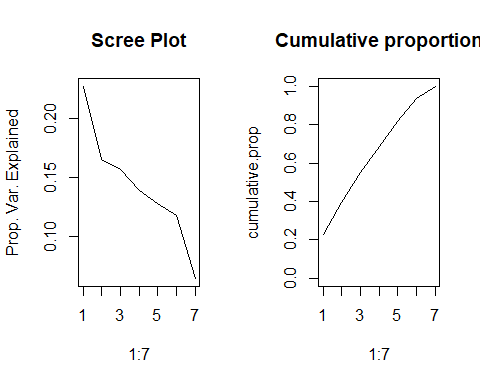
#Total Variance of Original Variables.  
sum(var.raw)

## [1] 9346457

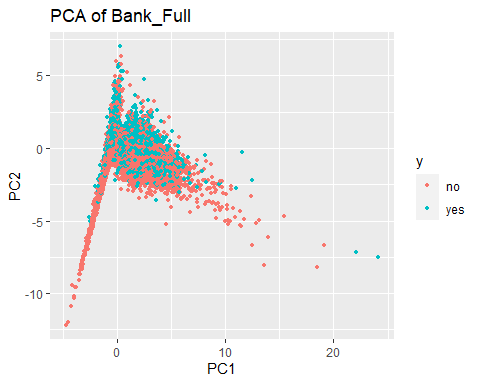
#List of eigenvectors  
pc.result$rotation

## PC1 PC2 PC3 PC4 PC5  
## age -0.01889173 0.09584663 -0.696026622 0.23882876 -0.659588776  
## balance 0.02948460 0.12361649 -0.697442213 -0.15340804 0.678698184  
## day -0.23229130 -0.53523432 -0.114209259 -0.47634785 -0.009841468  
## duration 0.04300790 0.44775037 0.017235867 -0.81694076 -0.287173990  
## campaign -0.21653157 -0.63044363 -0.108307865 -0.12096282 -0.135356238  
## pdays 0.67768911 -0.17892257 -0.001868909 -0.04901571 -0.024472952  
## previous 0.66092086 -0.24311265 -0.063611200 -0.08995985 -0.053154775  
## PC6 PC7  
## age -0.113984599 0.03023984  
## balance 0.113764723 0.01778535  
## day -0.646934159 0.03151640  
## duration 0.217593428 0.01283948  
## campaign 0.712431968 0.05840970  
## pdays -0.023886592 0.71073810  
## previous 0.008032136 -0.69932164

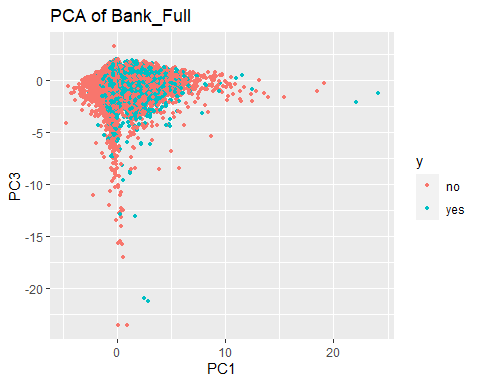
#A scree plot of the eigenvalues used to determine how many pc's to keep can be plotted in the following way:  
par(mfrow=c(1,2))  
eigenvals<-(pc.result$sdev)^2  
plot(1:7,eigenvals/sum(eigenvals),type="l",main="Scree Plot",ylab="Prop. Var. Explained")  
cumulative.prop<-cumsum(eigenvals/sum(eigenvals))  
plot(1:7,cumulative.prop,type="l",main="Cumulative proportion",ylim=c(0,1))



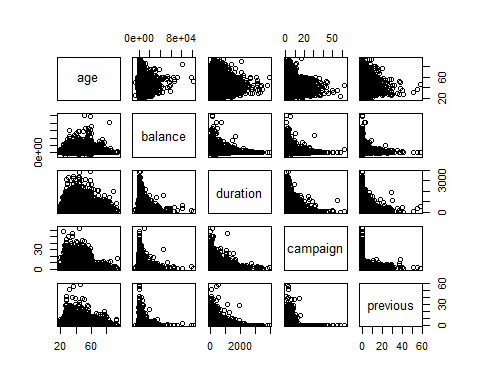
par(mfrow=c(1,1))  
#The scree plots show the elbow around 2.The variance explained for .9 is 2.   
#Use ggplot2 to plot the first few pc's  
pc.scores<-data.frame(pc.scores)  
pc.scores$y<-Bank\_Full$y  
ggplot(data = pc.scores, aes(x = PC1, y = PC2)) +  
 geom\_point(aes(col=y), size=1)+  
 ggtitle("PCA of Bank\_Full")



#Looking at PC1 and PC3  
pc.scores<-data.frame(pc.scores)  
pc.scores$y<-Bank\_Full$y  
ggplot(data = pc.scores, aes(x = PC1, y = PC3)) +  
 geom\_point(aes(col=y), size=1)+  
 ggtitle("PCA of Bank\_Full")



# Not great. Some separation. Still debating using day in there.  
# PCA w/o day:  
# This includes: age, balance, duration, campaign, pdays, previous  
# We might want to do this again when we address normality for LDA/QDA.  
# Not necessary for PCA but slides indicate it may improve the results.  
reduced<-Bank\_Full[,c(1,6,12,13,15)]  
pairs(reduced)



apply(reduced,2,summary)

## age balance duration campaign previous  
## Min. 18.00000 -8019.000 0.000 1.000000 0.0000000  
## 1st Qu. 33.00000 72.000 103.000 1.000000 0.0000000  
## Median 39.00000 448.000 180.000 2.000000 0.0000000  
## Mean 40.93583 1362.175 258.058 2.763897 0.5742662  
## 3rd Qu. 48.00000 1428.000 319.000 3.000000 0.0000000  
## Max. 95.00000 102127.000 3881.000 63.000000 58.0000000

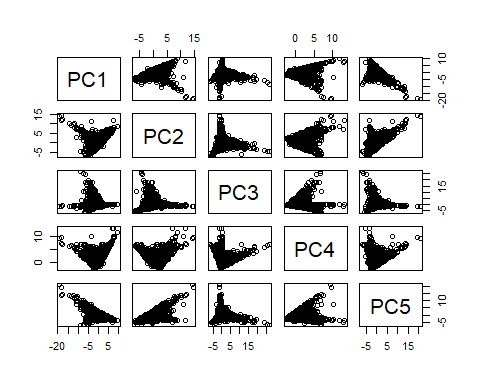
var.raw<-apply(reduced,2,var)  
var.raw

## age balance duration campaign previous   
## 1.127559e+02 9.270394e+06 6.584300e+04 9.598076e+00 3.640261e+00

#Total variance  
sum(var.raw)

## [1] 9336363

pc.result<-prcomp(Bank\_Full[,c(1,6,12,13,15)],scale.=TRUE)  
pc.scores<-pc.result$x  
pairs(pc.scores)



cor(pc.scores)

## PC1 PC2 PC3 PC4 PC5  
## PC1 1.000000e+00 -1.892153e-14 3.215478e-15 1.076578e-14 1.790066e-14  
## PC2 -1.892153e-14 1.000000e+00 -4.064115e-15 -1.361522e-14 -2.300901e-14  
## PC3 3.215478e-15 -4.064115e-15 1.000000e+00 2.339893e-15 4.466491e-15  
## PC4 1.076578e-14 -1.361522e-14 2.339893e-15 1.000000e+00 1.848995e-14  
## PC5 1.790066e-14 -2.300901e-14 4.466491e-15 1.848995e-14 1.000000e+00

var.pca<-apply(pc.scores,2,var)  
var.pca

## PC1 PC2 PC3 PC4 PC5   
## 1.1145237 1.0811957 0.9988439 0.9101864 0.8952504

#Total Variance of PC's  
sum(var.pca)

## [1] 5

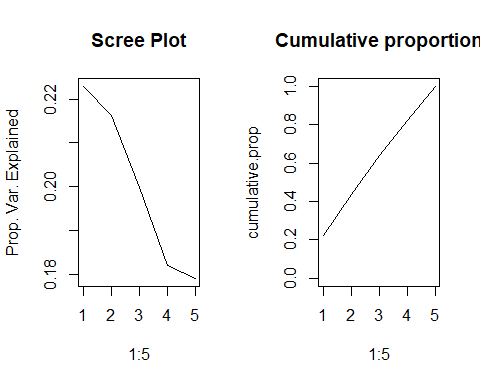
#Total Variance of Original Variables.  
sum(var.raw)

## [1] 9336363

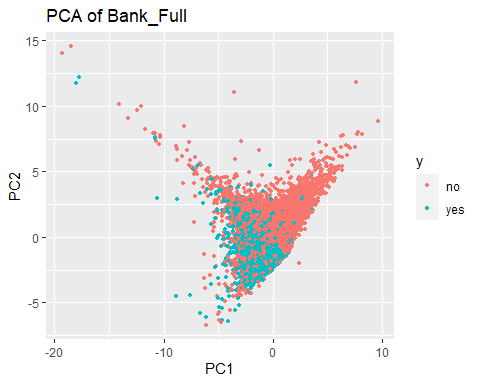
#List of eigenvectors  
pc.result$rotation

## PC1 PC2 PC3 PC4 PC5  
## age -0.4549616 0.5505030 -0.105325051 -0.3530449 -0.5951658  
## balance -0.5737513 0.4077508 -0.038322792 0.3168371 0.6345820  
## duration -0.4242188 -0.4824403 -0.414327159 0.5319689 -0.3641867  
## campaign 0.4592187 0.5281887 -0.003664299 0.6663462 -0.2571074  
## previous -0.2701408 -0.1376724 0.903192989 0.2190102 -0.2105878

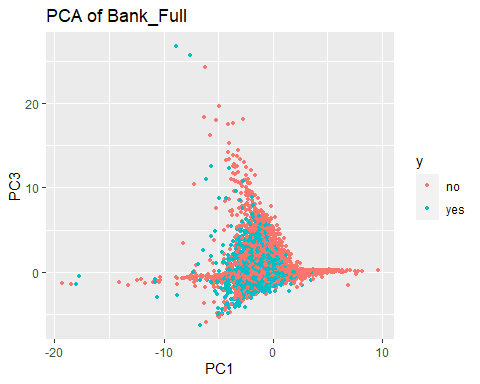
#A scree plot of the eigenvalues used to determine how many pc's to keep can be plotted in the following way:  
par(mfrow=c(1,2))  
eigenvals<-(pc.result$sdev)^2  
plot(1:5,eigenvals/sum(eigenvals),type="l",main="Scree Plot",ylab="Prop. Var. Explained")  
cumulative.prop<-cumsum(eigenvals/sum(eigenvals))  
plot(1:5,cumulative.prop,type="l",main="Cumulative proportion",ylim=c(0,1))



par(mfrow=c(1,1))  
#Removing day and pdays The scree plots show the elbow around 4 predictors.The variance explained for 80 of the variance is 4.   
#Use ggplot2 to plot the first few pc's  
pc.scores<-data.frame(pc.scores)  
pc.scores$y<-Bank\_Full$y  
ggplot(data = pc.scores, aes(x = PC1, y = PC2)) +  
 geom\_point(aes(col=y), size=1)+  
 ggtitle("PCA of Bank\_Full")



#Looking at PC1 and PC3  
pc.scores<-data.frame(pc.scores)  
pc.scores$y<-Bank\_Full$y  
ggplot(data = pc.scores, aes(x = PC1, y = PC3)) +  
 geom\_point(aes(col=y), size=1)+  
 ggtitle("PCA of Bank\_Full")



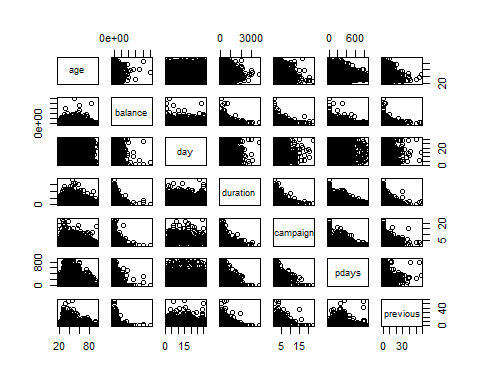
#Without day and pday we find that the model requires first 4 principle components or combinations of these to explain about 80% of the variability.  
  
#Removing year 2008  
# Explore what the data looks like if we remove 2008  
prop.table(table(Bank\_Full$y,Bank\_Full$year),2)

##   
## 2008 2009 2010  
## no 0.94947346 0.82935200 0.48396947  
## yes 0.05052654 0.17064800 0.51603053

#plot(Bank\_Full$y~Bank\_Full$year,col=c("red","blue"))  
Bank\_Full.new = Bank\_Full %>%  
 filter(year != 2008)  
summary(Bank\_Full.new)

## age job marital education   
## Min. :18.00 Length:17481 Length:17481 Length:17481   
## 1st Qu.:31.00 Class :character Class :character Class :character   
## Median :37.00 Mode :character Mode :character Mode :character   
## Mean :40.44   
## 3rd Qu.:48.00   
## Max. :95.00   
##   
## default balance housing loan   
## Length:17481 Min. : -4057 Length:17481 Length:17481   
## Class :character 1st Qu.: 144 Class :character Class :character   
## Mode :character Median : 539 Mode :character Mode :character   
## Mean : 1437   
## 3rd Qu.: 1536   
## Max. :102127   
##   
## contact day month duration   
## Length:17481 Min. : 1.00 may :5809 Min. : 0.0   
## Class :character 1st Qu.: 6.00 apr :2932 1st Qu.: 112.0   
## Mode :character Median :13.00 feb :2648 Median : 197.0   
## Mean :13.73 jan :1403 Mean : 267.6   
## 3rd Qu.:19.00 aug :1032 3rd Qu.: 336.0   
## Max. :31.00 jun : 855 Max. :3785.0   
## (Other):2802   
## campaign pdays previous poutcome   
## Min. : 1.000 Min. : -1.00 Min. : 0.000 Length:17481   
## 1st Qu.: 1.000 1st Qu.: -1.00 1st Qu.: 0.000 Class :character   
## Median : 2.000 Median : -1.00 Median : 0.000 Mode :character   
## Mean : 2.094 Mean : 97.58 Mean : 1.333   
## 3rd Qu.: 2.000 3rd Qu.:185.00 3rd Qu.: 2.000   
## Max. :23.000 Max. :871.00 Max. :58.000   
##   
## y id year   
## Length:17481 Min. :27730 2008: 0   
## Class :character 1st Qu.:32101 2009:14861   
## Mode :character Median :36471 2010: 2620   
## Mean :36471   
## 3rd Qu.:40841   
## Max. :45211   
##

# 17481 rows. 13593 no to 3888 yes. (3.5:1) Much less unbalanced.  
#For the 3 Years  
# This includes: age, balance, day, duration, campaign, pdays, previous  
reduced<-Bank\_Full.new[,c(1,6,10,12,13,14,15)]  
pairs(reduced)



#Let's take a quick look at the summary statistics and in particular lets calculate the variance of each variable and add them up to obtain the total variance.  
apply(reduced,2,summary)

## age balance day duration campaign pdays previous  
## Min. 18.00000 -4057.000 1.00000 0.000 1.00000 -1.00000 0.000000  
## 1st Qu. 31.00000 144.000 6.00000 112.000 1.00000 -1.00000 0.000000  
## Median 37.00000 539.000 13.00000 197.000 2.00000 -1.00000 0.000000  
## Mean 40.43602 1436.968 13.73222 267.627 2.09353 97.58469 1.333505  
## 3rd Qu. 48.00000 1536.000 19.00000 336.000 2.00000 185.00000 2.000000  
## Max. 95.00000 102127.000 31.00000 3785.000 23.00000 871.00000 58.000000

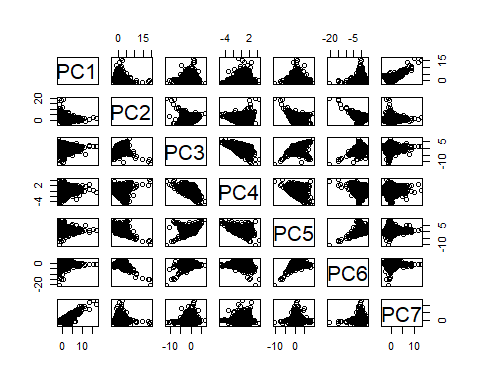
var.raw<-apply(reduced,2,var)  
var.raw

## age balance day duration campaign pdays   
## 1.512222e+02 9.346700e+06 7.194780e+01 6.357241e+04 2.762934e+00 1.932927e+04   
## previous   
## 7.568860e+00

#Total variance  
sum(var.raw)

## [1] 9429835

#Running PCA is relatively straight forward. The following script conducts a PCA using the covariance matrix (nonstandardarized #variables) and stores the results in an object. This object contains the eigenvectors, eigenvalue, and the new principle #component vectors. Lets start by producing a correlation matrix to verify that new principle component variables are #uncorrelated.  
pc.result<-prcomp(Bank\_Full.new[,c(1,6,10,12,13,14,15)],scale.=TRUE)  
pc.scores<-pc.result$x  
pairs(pc.scores)



cor(pc.scores)

## PC1 PC2 PC3 PC4 PC5  
## PC1 1.000000e+00 -1.427296e-16 -9.148076e-16 -1.202620e-16 9.366677e-16  
## PC2 -1.427296e-16 1.000000e+00 -5.497426e-15 -2.570594e-16 7.589130e-16  
## PC3 -9.148076e-16 -5.497426e-15 1.000000e+00 -1.525847e-15 -6.003010e-16  
## PC4 -1.202620e-16 -2.570594e-16 -1.525847e-15 1.000000e+00 -7.099381e-16  
## PC5 9.366677e-16 7.589130e-16 -6.003010e-16 -7.099381e-16 1.000000e+00  
## PC6 -1.032868e-16 1.785161e-14 -9.561581e-15 1.287745e-15 -1.103208e-15  
## PC7 3.088198e-15 -3.035899e-15 1.328950e-15 3.270716e-16 -6.093781e-16  
## PC6 PC7  
## PC1 -1.032868e-16 3.088198e-15  
## PC2 1.785161e-14 -3.035899e-15  
## PC3 -9.561581e-15 1.328950e-15  
## PC4 1.287745e-15 3.270716e-16  
## PC5 -1.103208e-15 -6.093781e-16  
## PC6 1.000000e+00 -1.512648e-15  
## PC7 -1.512648e-15 1.000000e+00

#We can again verify that the total variance in the new PC variables is exactly the same as the original data. The eigenvectors are stored inside of "pc.result" as well in the "rotation" object.  
var.pca<-apply(pc.scores,2,var)  
var.pca

## PC1 PC2 PC3 PC4 PC5 PC6 PC7   
## 1.4707498 1.1539191 1.0635289 0.9943371 0.9088628 0.8764210 0.5321814

#Total Variance of PC's  
sum(var.pca)

## [1] 7

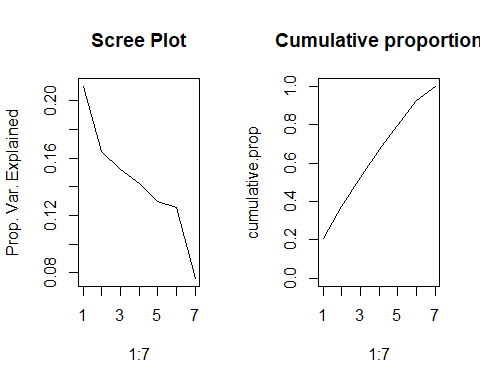
#Total Variance of Original Variables.  
sum(var.raw)

## [1] 9429835

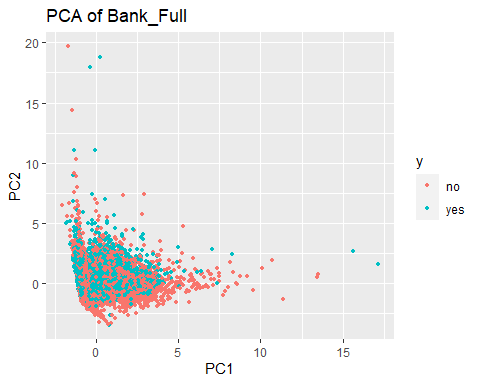
#List of eigenvectors  
pc.result$rotation

## PC1 PC2 PC3 PC4 PC5  
## age -0.007038238 0.61876837 -0.30570311 0.02864049 0.002653833  
## balance -0.026411505 0.59292986 -0.34132501 0.13541651 -0.276511458  
## day -0.045536753 0.10334149 0.33534634 0.90698451 0.225902049  
## duration -0.077477477 0.43916842 0.43889255 -0.37809106 0.657217710  
## campaign 0.147971974 -0.22767737 -0.67677310 0.11806214 0.655109872  
## pdays 0.691815968 0.03974483 0.15912307 -0.02711785 -0.100011876  
## previous 0.700479480 0.09270859 0.04021234 0.02437815 0.037365723  
## PC6 PC7  
## age 0.72248193 -0.02853618  
## balance -0.65879419 -0.04927316  
## day 0.01496671 -0.03021812  
## duration -0.17997798 -0.03495188  
## campaign -0.10181403 -0.12153918  
## pdays 0.01400625 -0.69538258  
## previous -0.02883964 0.70448163

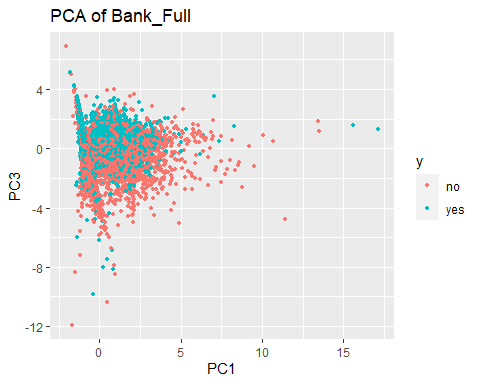
#A scree plot of the eigenvalues used to determine how many pc's to keep can be plotted in the following way:  
par(mfrow=c(1,2))  
eigenvals<-(pc.result$sdev)^2  
plot(1:7,eigenvals/sum(eigenvals),type="l",main="Scree Plot",ylab="Prop. Var. Explained")  
cumulative.prop<-cumsum(eigenvals/sum(eigenvals))  
plot(1:7,cumulative.prop,type="l",main="Cumulative proportion",ylim=c(0,1))



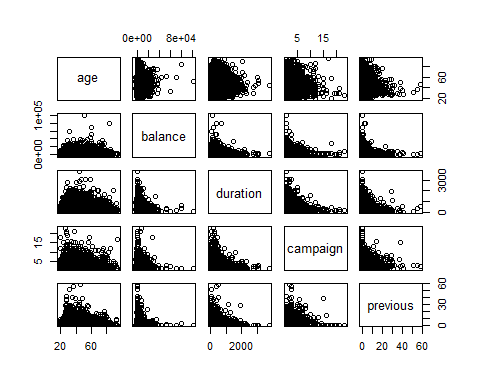
par(mfrow=c(1,1))  
#The scree plots show the elbow around 5-6.The variance explained for .8 is 6.   
#Use ggplot2 to plot the first few pc's  
pc.scores<-data.frame(pc.scores)  
pc.scores$y<-Bank\_Full.new$y  
ggplot(data = pc.scores, aes(x = PC1, y = PC2)) +  
 geom\_point(aes(col=y), size=1)+  
 ggtitle("PCA of Bank\_Full")



#Looking at PC1 and PC3  
pc.scores<-data.frame(pc.scores)  
pc.scores$y<-Bank\_Full.new$y  
ggplot(data = pc.scores, aes(x = PC1, y = PC3)) +  
 geom\_point(aes(col=y), size=1)+  
 ggtitle("PCA of Bank\_Full")



# Not great. Some separation. Still debating using day in there.  
# PCA w/o day & pday:  
# This includes: age, balance, duration, campaign, pdays, previous  
# We might want to do this again when we address normality for LDA/QDA.  
# Not necessary for PCA but slides indicate it may improve the results.  
reduced<-Bank\_Full.new[,c(1,6,12,13,15)]  
pairs(reduced)



apply(reduced,2,summary)

## age balance duration campaign previous  
## Min. 18.00000 -4057.000 0.000 1.00000 0.000000  
## 1st Qu. 31.00000 144.000 112.000 1.00000 0.000000  
## Median 37.00000 539.000 197.000 2.00000 0.000000  
## Mean 40.43602 1436.968 267.627 2.09353 1.333505  
## 3rd Qu. 48.00000 1536.000 336.000 2.00000 2.000000  
## Max. 95.00000 102127.000 3785.000 23.00000 58.000000

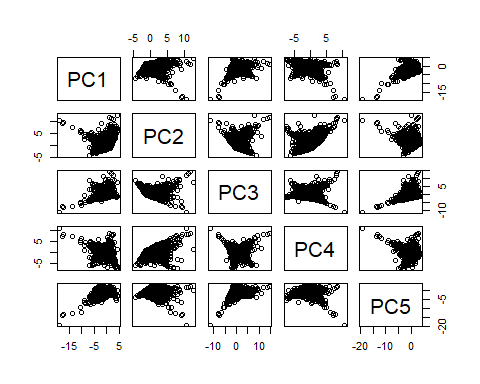
var.raw<-apply(reduced,2,var)  
var.raw

## age balance duration campaign previous   
## 1.512222e+02 9.346700e+06 6.357241e+04 2.762934e+00 7.568860e+00

#Total variance  
sum(var.raw)

## [1] 9410434

pc.result<-prcomp(Bank\_Full.new[,c(1,6,12,13,15)],scale.=TRUE)  
pc.scores<-pc.result$x  
pairs(pc.scores)



cor(pc.scores)

## PC1 PC2 PC3 PC4 PC5  
## PC1 1.000000e+00 -6.805662e-15 5.950315e-15 4.730126e-15 -1.778636e-14  
## PC2 -6.805662e-15 1.000000e+00 -2.460516e-16 -3.073818e-15 8.955968e-15  
## PC3 5.950315e-15 -2.460516e-16 1.000000e+00 5.036443e-15 -4.070655e-15  
## PC4 4.730126e-15 -3.073818e-15 5.036443e-15 1.000000e+00 -7.440006e-15  
## PC5 -1.778636e-14 8.955968e-15 -4.070655e-15 -7.440006e-15 1.000000e+00

var.pca<-apply(pc.scores,2,var)  
var.pca

## PC1 PC2 PC3 PC4 PC5   
## 1.1546225 1.1099629 0.9705134 0.8895309 0.8753704

#Total Variance of PC's  
sum(var.pca)

## [1] 5

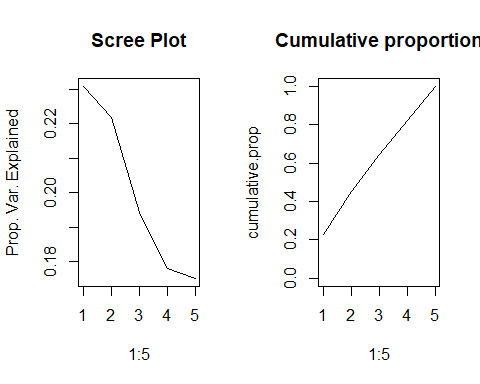
#Total Variance of Original Variables.  
sum(var.raw)

## [1] 9410434

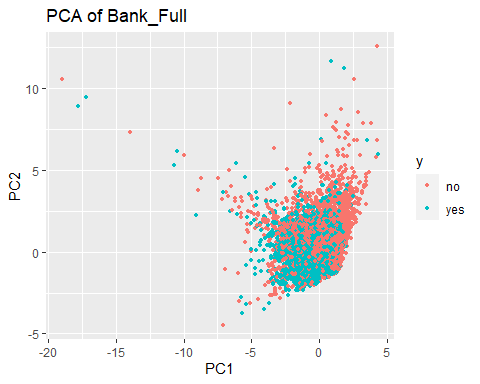
#List of eigenvectors  
pc.result$rotation

## PC1 PC2 PC3 PC4 PC5  
## age -0.58942529 0.3193297 -0.1749158 -0.2344791 0.6819313  
## balance -0.56345708 0.3263786 -0.3163908 0.3208739 -0.6106799  
## duration -0.48010419 -0.2769614 0.6434495 -0.4506033 -0.2751760  
## campaign 0.31408200 0.6204137 -0.0768656 -0.6628779 -0.2666906  
## previous 0.07711298 0.5743560 0.6703475 0.4467763 0.1232638

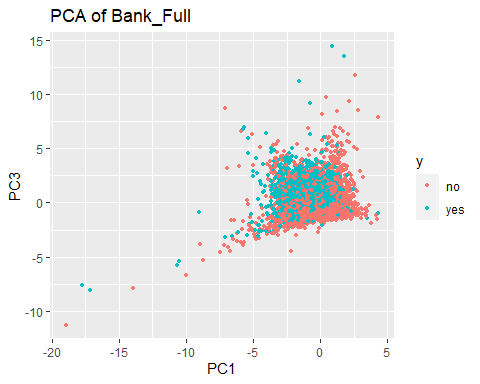
#A scree plot of the eigenvalues used to determine how many pc's to keep can be plotted in the following way:  
par(mfrow=c(1,2))  
eigenvals<-(pc.result$sdev)^2  
plot(1:5,eigenvals/sum(eigenvals),type="l",main="Scree Plot",ylab="Prop. Var. Explained")  
cumulative.prop<-cumsum(eigenvals/sum(eigenvals))  
plot(1:5,cumulative.prop,type="l",main="Cumulative proportion",ylim=c(0,1))



par(mfrow=c(1,1))  
#Removing day and pdays The scree plots show the elbow around 4 predictors.The variance explained for 80 of the variance is 4.   
#Use ggplot2 to plot the first few pc's  
pc.scores<-data.frame(pc.scores)  
pc.scores$y<-Bank\_Full.new$y  
ggplot(data = pc.scores, aes(x = PC1, y = PC2)) +  
 geom\_point(aes(col=y), size=1)+  
 ggtitle("PCA of Bank\_Full")



#Looking at PC1 and PC3  
pc.scores<-data.frame(pc.scores)  
pc.scores$y<-Bank\_Full.new$y  
ggplot(data = pc.scores, aes(x = PC1, y = PC3)) +  
 geom\_point(aes(col=y), size=1)+  
 ggtitle("PCA of Bank\_Full")



#Without day and pday we find that the model requires first 4 principle components or combinations of these to explain about 80% of the variability withot considering year 2008 data.  
#This looks better

# train test split  
#Using the data without 2008 as excluding it makes the dataset more balanced than when we have 2008 as we know that 2008 had financial crisis and because of which we see that the data is not good to include in prediction.  
# 80/20 would be: 13985:3496  
set.seed(1234)  
index<-sample(1:dim(Bank\_Full.new)[1],3496,replace=F)  
test<-Bank\_Full.new[index,]  
train<-Bank\_Full.new[-index,]

# Using glm  
train<-na.omit(train)  
Model\_Full<-glm(as.factor(y) ~ age+job+marital+education+default+balance+housing+loan+contact+day+month+duration+campaign+pdays+previous+poutcome+year,data=train,family= binomial(link="logit"))  
#(vif(Model\_Full)[,3])^2  
#On letting all the predictors in the model and running the vif function, nothing stands out in terms of excluding the predictors from the modle with a VIF score > 10  
  
#Summary of current fit  
summary(Model\_Full)

##   
## Call:  
## glm(formula = as.factor(y) ~ age + job + marital + education +   
## default + balance + housing + loan + contact + day + month +   
## duration + campaign + pdays + previous + poutcome + year,   
## family = binomial(link = "logit"), data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.8406 -0.5363 -0.3265 -0.1781 3.1443   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.882e+00 2.728e-01 -17.897 < 2e-16 \*\*\*  
## age 5.808e-03 3.030e-03 1.917 0.05525 .   
## jobblue-collar -3.136e-01 1.043e-01 -3.008 0.00263 \*\*   
## jobentrepreneur -5.755e-01 1.963e-01 -2.931 0.00337 \*\*   
## jobhousemaid -1.675e-01 1.912e-01 -0.876 0.38097   
## jobmanagement -1.316e-03 9.824e-02 -0.013 0.98931   
## jobretired -8.005e-02 1.343e-01 -0.596 0.55116   
## jobself-employed -8.712e-02 1.590e-01 -0.548 0.58370   
## jobservices -1.362e-01 1.178e-01 -1.156 0.24761   
## jobstudent 9.258e-02 1.309e-01 0.707 0.47929   
## jobtechnician -7.167e-02 9.524e-02 -0.753 0.45174   
## jobunemployed -2.062e-01 1.466e-01 -1.407 0.15945   
## jobunknown -2.238e-01 3.203e-01 -0.699 0.48466   
## maritalmarried -7.640e-02 8.710e-02 -0.877 0.38041   
## maritalsingle 3.045e-02 9.897e-02 0.308 0.75836   
## educationsecondary 2.277e-01 9.301e-02 2.448 0.01437 \*   
## educationtertiary 4.281e-01 1.058e-01 4.048 5.17e-05 \*\*\*  
## educationunknown 2.473e-01 1.424e-01 1.737 0.08240 .   
## defaultyes 7.815e-02 3.298e-01 0.237 0.81267   
## balance 9.688e-06 7.221e-06 1.342 0.17974   
## housingyes -6.916e-01 6.287e-02 -11.001 < 2e-16 \*\*\*  
## loanyes -2.739e-01 9.573e-02 -2.862 0.00421 \*\*   
## contacttelephone -5.323e-01 1.014e-01 -5.247 1.54e-07 \*\*\*  
## contactunknown -1.653e+00 2.532e-01 -6.530 6.57e-11 \*\*\*  
## day 2.676e-02 3.389e-03 7.895 2.91e-15 \*\*\*  
## monthfeb 1.394e+00 1.479e-01 9.426 < 2e-16 \*\*\*  
## monthmar 2.745e+00 1.697e-01 16.176 < 2e-16 \*\*\*  
## monthapr 1.494e+00 1.328e-01 11.245 < 2e-16 \*\*\*  
## monthmay 1.141e+00 1.363e-01 8.368 < 2e-16 \*\*\*  
## monthjun 2.522e+00 1.579e-01 15.970 < 2e-16 \*\*\*  
## monthjul 2.226e+00 1.756e-01 12.679 < 2e-16 \*\*\*  
## monthaug 2.179e+00 1.477e-01 14.749 < 2e-16 \*\*\*  
## monthsep 2.228e+00 1.692e-01 13.167 < 2e-16 \*\*\*  
## monthoct 2.002e+00 1.575e-01 12.711 < 2e-16 \*\*\*  
## monthnov 2.622e+00 1.847e-01 14.192 < 2e-16 \*\*\*  
## monthdec 2.628e+00 2.241e-01 11.726 < 2e-16 \*\*\*  
## duration 3.648e-03 1.067e-04 34.181 < 2e-16 \*\*\*  
## campaign -8.171e-02 1.871e-02 -4.368 1.25e-05 \*\*\*  
## pdays 1.008e-05 3.378e-04 0.030 0.97619   
## previous -1.648e-03 1.281e-02 -0.129 0.89766   
## poutcomeother 1.676e-01 1.062e-01 1.579 0.11445   
## poutcomesuccess 1.811e+00 9.535e-02 18.994 < 2e-16 \*\*\*  
## poutcomeunknown 4.747e-01 1.115e-01 4.257 2.07e-05 \*\*\*  
## year2010 1.065e+00 7.042e-02 15.121 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 14816 on 13984 degrees of freedom  
## Residual deviance: 10245 on 13941 degrees of freedom  
## AIC: 10333  
##   
## Number of Fisher Scoring iterations: 5

#Null deviance: 14843 on 13983 degrees of freedom  
#Residual deviance: 10230 on 13940 degrees of freedom  
#AIC: 10318  
  
#Hosmer Lemeshow test for lack of fit. Use as needed. The g=10 is an option that deals with the continuous predictors if any are there.  
#This should be increased with caution.   
hoslem.test(Model\_Full$y, fitted(Model\_Full), g=10)

##   
## Hosmer and Lemeshow goodness of fit (GOF) test  
##   
## data: Model\_Full$y, fitted(Model\_Full)  
## X-squared = 219.52, df = 8, p-value < 2.2e-16

#p-value < 2.2e-16  
# rejects. But the sample size is large, so I don't think it's an issue  
  
# Using the summary coefficients we can generate CI for each one in the table  
exp(cbind("Odds ratio" = coef(Model\_Full), confint.default(Model\_Full, level = 0.95)))

## Odds ratio 2.5 % 97.5 %  
## (Intercept) 0.007583663 0.004443196 0.01294382  
## age 1.005825187 0.999869512 1.01181634  
## jobblue-collar 0.730836556 0.595769994 0.89652395  
## jobentrepreneur 0.562418813 0.382782507 0.82635678  
## jobhousemaid 0.845771376 0.581444376 1.23026252  
## jobmanagement 0.998684436 0.823774548 1.21073248  
## jobretired 0.923068300 0.709430057 1.20104171  
## jobself-employed 0.916564488 0.671169443 1.25168163  
## jobservices 0.872659120 0.692730276 1.09932244  
## jobstudent 1.096996163 0.848821602 1.41773086  
## jobtechnician 0.930835186 0.772326437 1.12187555  
## jobunemployed 0.813678710 0.610528756 1.08442565  
## jobunknown 0.799465519 0.426761665 1.49766291  
## maritalmarried 0.926447668 0.781056261 1.09890327  
## maritalsingle 1.030916023 0.849136746 1.25160977  
## educationsecondary 1.255678389 1.046432897 1.50676476  
## educationtertiary 1.534311734 1.247080681 1.88769863  
## educationunknown 1.280541578 0.968746372 1.69268942  
## defaultyes 1.081281962 0.566564774 2.06361344  
## balance 1.000009688 0.999995534 1.00002384  
## housingyes 0.500754366 0.442700651 0.56642098  
## loanyes 0.760370413 0.630294508 0.91729050  
## contacttelephone 0.587240519 0.481356184 0.71641632  
## contactunknown 0.191408873 0.116533764 0.31439263  
## day 1.027116267 1.020316455 1.03396139  
## monthfeb 4.029847896 3.015951455 5.38459399  
## monthmar 15.562428899 11.159438616 21.70263233  
## monthapr 4.454200354 3.433148888 5.77892234  
## monthmay 3.129213698 2.395479949 4.08768955  
## monthjun 12.452361312 9.137651097 16.96949255  
## monthjul 9.262352423 6.565670908 13.06662695  
## monthaug 8.836894451 6.615295527 11.80456765  
## monthsep 9.281979633 6.661905564 12.93250784  
## monthoct 7.404515826 5.437839436 10.08247030  
## monthnov 13.757116930 9.578455752 19.75874516  
## monthdec 13.848351398 8.925138578 21.48726709  
## duration 1.003654879 1.003444945 1.00386486  
## campaign 0.921539143 0.888362647 0.95595464  
## pdays 1.000010083 0.999348202 1.00067240  
## previous 0.998353671 0.973598547 1.02373823  
## poutcomeother 1.182459078 0.960310045 1.45599796  
## poutcomesuccess 6.116638785 5.074030884 7.37348094  
## poutcomeunknown 1.607523256 1.291919406 2.00022618  
## year2010 2.900353276 2.526436918 3.32960980

#stepwise:  
#This starts with a null model and then builds up using forward selection up to all the predictors that were specified in main model previously.  
# Code from AutoClassify.R  
Model\_Step<-Model\_Full %>%stepAIC(trace=FALSE)  
summary(Model\_Step)

##   
## Call:  
## glm(formula = as.factor(y) ~ age + job + education + housing +   
## loan + contact + day + month + duration + campaign + poutcome +   
## year, family = binomial(link = "logit"), data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.8561 -0.5369 -0.3272 -0.1782 3.1320   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.8405275 0.2340604 -20.681 < 2e-16 \*\*\*  
## age 0.0042197 0.0026812 1.574 0.11554   
## jobblue-collar -0.3265862 0.1038092 -3.146 0.00166 \*\*   
## jobentrepreneur -0.5818199 0.1961553 -2.966 0.00302 \*\*   
## jobhousemaid -0.1682360 0.1907008 -0.882 0.37767   
## jobmanagement -0.0062541 0.0979995 -0.064 0.94912   
## jobretired -0.0602656 0.1335506 -0.451 0.65180   
## jobself-employed -0.0835345 0.1586317 -0.527 0.59848   
## jobservices -0.1385329 0.1177730 -1.176 0.23949   
## jobstudent 0.1189071 0.1300069 0.915 0.36039   
## jobtechnician -0.0738785 0.0951709 -0.776 0.43759   
## jobunemployed -0.1980792 0.1463805 -1.353 0.17600   
## jobunknown -0.2273303 0.3204025 -0.710 0.47801   
## educationsecondary 0.2291422 0.0928646 2.467 0.01361 \*   
## educationtertiary 0.4399502 0.1054116 4.174 3.00e-05 \*\*\*  
## educationunknown 0.2509384 0.1421357 1.765 0.07748 .   
## housingyes -0.7012453 0.0621319 -11.286 < 2e-16 \*\*\*  
## loanyes -0.2862562 0.0955208 -2.997 0.00273 \*\*   
## contacttelephone -0.5265140 0.1011343 -5.206 1.93e-07 \*\*\*  
## contactunknown -1.6583867 0.2525171 -6.567 5.12e-11 \*\*\*  
## day 0.0268217 0.0033867 7.920 2.38e-15 \*\*\*  
## monthfeb 1.3991274 0.1477821 9.468 < 2e-16 \*\*\*  
## monthmar 2.7519347 0.1694256 16.243 < 2e-16 \*\*\*  
## monthapr 1.4986044 0.1325973 11.302 < 2e-16 \*\*\*  
## monthmay 1.1489027 0.1361363 8.439 < 2e-16 \*\*\*  
## monthjun 2.5289512 0.1576144 16.045 < 2e-16 \*\*\*  
## monthjul 2.2373107 0.1751887 12.771 < 2e-16 \*\*\*  
## monthaug 2.1820028 0.1475846 14.785 < 2e-16 \*\*\*  
## monthsep 2.2314918 0.1690511 13.200 < 2e-16 \*\*\*  
## monthoct 2.0099592 0.1573271 12.776 < 2e-16 \*\*\*  
## monthnov 2.6226931 0.1843656 14.225 < 2e-16 \*\*\*  
## monthdec 2.6334117 0.2239579 11.759 < 2e-16 \*\*\*  
## duration 0.0036533 0.0001067 34.244 < 2e-16 \*\*\*  
## campaign -0.0826160 0.0186840 -4.422 9.79e-06 \*\*\*  
## poutcomeother 0.1703772 0.1056344 1.613 0.10677   
## poutcomesuccess 1.8114790 0.0922438 19.638 < 2e-16 \*\*\*  
## poutcomeunknown 0.4803633 0.0694529 6.916 4.63e-12 \*\*\*  
## year2010 1.0654236 0.0695660 15.315 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 14816 on 13984 degrees of freedom  
## Residual deviance: 10250 on 13947 degrees of freedom  
## AIC: 10326  
##   
## Number of Fisher Scoring iterations: 5

#Null deviance: 14843 on 13983 degrees of freedom  
#Residual deviance: 10252 on 13955 degrees of freedom  
#AIC: 10310  
hoslem.test(Model\_Step$y, fitted(Model\_Step), g=10)

##   
## Hosmer and Lemeshow goodness of fit (GOF) test  
##   
## data: Model\_Step$y, fitted(Model\_Step)  
## X-squared = 221.15, df = 8, p-value < 2.2e-16

#p-value < 2.2e-16  
# rejects. But the sample size is large, so I don't think it's an issue  
exp(cbind("Odds ratio" = coef(Model\_Step), confint.default(Model\_Step, level = 0.95)))

## Odds ratio 2.5 % 97.5 %  
## (Intercept) 0.007902884 0.004995201 0.01250311  
## age 1.004228600 0.998965106 1.00951983  
## jobblue-collar 0.721382194 0.588576488 0.88415402  
## jobentrepreneur 0.558880344 0.380496919 0.82089295  
## jobhousemaid 0.845154342 0.581582373 1.22817660  
## jobmanagement 0.993765427 0.820099455 1.20420727  
## jobretired 0.941514403 0.724683715 1.22322242  
## jobself-employed 0.919859349 0.674052734 1.25530419  
## jobservices 0.870634564 0.691173917 1.09669148  
## jobstudent 1.126265252 0.872928431 1.45312419  
## jobtechnician 0.928784524 0.770735475 1.11924353  
## jobunemployed 0.820304866 0.615709634 1.09288541  
## jobunknown 0.796657574 0.425152366 1.49279021  
## educationsecondary 1.257520792 1.048259232 1.50855675  
## educationtertiary 1.552629969 1.262819910 1.90894980  
## educationunknown 1.285230885 0.972736121 1.69811565  
## housingyes 0.495967310 0.439102895 0.56019574  
## loanyes 0.751070159 0.622835074 0.90570748  
## contacttelephone 0.590660420 0.484453606 0.72015096  
## contactunknown 0.190445974 0.116099090 0.31240270  
## day 1.027184668 1.020389027 1.03402557  
## monthfeb 4.051662816 3.032780081 5.41284601  
## monthmar 15.672924662 11.244359102 21.84567081  
## monthapr 4.475438833 3.451187601 5.80366965  
## monthmay 3.154729352 2.415920841 4.11947160  
## monthjun 12.540346992 9.207632301 17.07934218  
## monthjul 9.368103616 6.645548519 13.20603786  
## monthaug 8.864041016 6.637545712 11.83739089  
## monthsep 9.313749675 6.686949427 12.97242247  
## monthoct 7.463012522 5.482732768 10.15853923  
## monthnov 13.772765652 9.595969408 19.76757799  
## monthdec 13.921184492 8.975173908 21.59282702  
## duration 1.003660014 1.003450173 1.00386990  
## campaign 0.920704651 0.887598329 0.95504580  
## poutcomeother 1.185752032 0.964001445 1.45851221  
## poutcomesuccess 6.119491481 5.107368974 7.33218535  
## poutcomeunknown 1.616661697 1.410914671 1.85241184  
## year2010 2.902067932 2.532170037 3.32600029

Model\_Null<-glm(as.factor(y) ~ 1, data=train,family = binomial(link="logit"))  
#Forward:  
Model\_FWD<-stepAIC(Model\_Null,  
 scope = list(upper=Model\_Full),  
 direction="forward",  
 test="Chisq",  
 data=train)

## Start: AIC=14818.44  
## as.factor(y) ~ 1  
##   
## Df Deviance AIC LRT Pr(Chi)   
## + month 11 13336 13360 1480.53 < 2.2e-16 \*\*\*  
## + duration 1 13380 13384 1436.46 < 2.2e-16 \*\*\*  
## + poutcome 3 13600 13608 1216.10 < 2.2e-16 \*\*\*  
## + year 1 13785 13789 1031.65 < 2.2e-16 \*\*\*  
## + housing 1 13874 13878 942.10 < 2.2e-16 \*\*\*  
## + job 11 14329 14353 487.11 < 2.2e-16 \*\*\*  
## + education 3 14650 14658 166.87 < 2.2e-16 \*\*\*  
## + loan 1 14674 14678 142.60 < 2.2e-16 \*\*\*  
## + campaign 1 14718 14722 98.15 < 2.2e-16 \*\*\*  
## + age 1 14726 14730 90.78 < 2.2e-16 \*\*\*  
## + balance 1 14731 14735 85.69 < 2.2e-16 \*\*\*  
## + day 1 14787 14791 29.36 6.023e-08 \*\*\*  
## + previous 1 14789 14793 27.44 1.623e-07 \*\*\*  
## + contact 2 14791 14797 25.03 3.665e-06 \*\*\*  
## + marital 2 14801 14807 15.19 0.0005023 \*\*\*  
## + default 1 14806 14810 10.92 0.0009510 \*\*\*  
## + pdays 1 14811 14815 5.56 0.0183985 \*   
## <none> 14816 14818   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Step: AIC=13359.91  
## as.factor(y) ~ month  
##   
## Df Deviance AIC LRT Pr(Chi)   
## + duration 1 11730 11756 1606.16 < 2.2e-16 \*\*\*  
## + poutcome 3 12549 12579 786.82 < 2.2e-16 \*\*\*  
## + year 1 12848 12874 488.20 < 2.2e-16 \*\*\*  
## + housing 1 13025 13051 310.68 < 2.2e-16 \*\*\*  
## + job 11 13201 13247 134.67 < 2.2e-16 \*\*\*  
## + contact 2 13237 13265 99.13 < 2.2e-16 \*\*\*  
## + day 1 13239 13265 96.87 < 2.2e-16 \*\*\*  
## + loan 1 13270 13296 65.53 5.729e-16 \*\*\*  
## + education 3 13273 13303 63.31 1.151e-13 \*\*\*  
## + campaign 1 13287 13313 48.59 3.151e-12 \*\*\*  
## + balance 1 13313 13339 23.25 1.421e-06 \*\*\*  
## + age 1 13322 13348 14.09 0.0001741 \*\*\*  
## + marital 2 13320 13348 15.70 0.0003896 \*\*\*  
## + previous 1 13325 13351 11.35 0.0007532 \*\*\*  
## <none> 13336 13360   
## + default 1 13334 13360 1.40 0.2362010   
## + pdays 1 13336 13362 0.30 0.5838742   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Step: AIC=11755.75  
## as.factor(y) ~ month + duration  
##   
## Df Deviance AIC LRT Pr(Chi)   
## + poutcome 3 11010 11042 719.69 < 2.2e-16 \*\*\*  
## + year 1 11252 11280 478.00 < 2.2e-16 \*\*\*  
## + housing 1 11397 11425 332.43 < 2.2e-16 \*\*\*  
## + day 1 11600 11628 129.71 < 2.2e-16 \*\*\*  
## + job 11 11581 11629 149.05 < 2.2e-16 \*\*\*  
## + education 3 11649 11681 80.55 < 2.2e-16 \*\*\*  
## + loan 1 11676 11704 54.25 1.761e-13 \*\*\*  
## + contact 2 11685 11715 44.94 1.747e-10 \*\*\*  
## + campaign 1 11699 11727 31.17 2.361e-08 \*\*\*  
## + balance 1 11714 11742 16.25 5.559e-05 \*\*\*  
## + marital 2 11713 11743 16.91 0.0002128 \*\*\*  
## + previous 1 11715 11743 14.75 0.0001228 \*\*\*  
## + age 1 11726 11754 3.27 0.0705068 .   
## <none> 11730 11756   
## + default 1 11728 11756 1.46 0.2261448   
## + pdays 1 11730 11758 0.02 0.8961829   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Step: AIC=11042.06  
## as.factor(y) ~ month + duration + poutcome  
##   
## Df Deviance AIC LRT Pr(Chi)   
## + year 1 10712 10746 297.995 < 2.2e-16 \*\*\*  
## + housing 1 10738 10772 271.936 < 2.2e-16 \*\*\*  
## + job 11 10875 10929 135.355 < 2.2e-16 \*\*\*  
## + day 1 10896 10930 114.410 < 2.2e-16 \*\*\*  
## + education 3 10945 10983 65.111 4.750e-14 \*\*\*  
## + loan 1 10971 11005 38.720 4.893e-10 \*\*\*  
## + contact 2 10976 11012 33.745 4.704e-08 \*\*\*  
## + campaign 1 10987 11021 23.352 1.349e-06 \*\*\*  
## + marital 2 10993 11029 17.474 0.0001606 \*\*\*  
## + balance 1 10997 11031 12.708 0.0003642 \*\*\*  
## + pdays 1 11007 11041 2.640 0.1042085   
## <none> 11010 11042   
## + age 1 11008 11042 1.677 0.1953021   
## + previous 1 11010 11044 0.481 0.4878077   
## + default 1 11010 11044 0.443 0.5057638   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Step: AIC=10746.07  
## as.factor(y) ~ month + duration + poutcome + year  
##   
## Df Deviance AIC LRT Pr(Chi)   
## + housing 1 10516 10552 196.176 < 2.2e-16 \*\*\*  
## + job 11 10605 10661 107.484 < 2.2e-16 \*\*\*  
## + contact 2 10623 10661 89.369 < 2.2e-16 \*\*\*  
## + day 1 10628 10664 83.564 < 2.2e-16 \*\*\*  
## + education 3 10648 10688 64.070 7.929e-14 \*\*\*  
## + loan 1 10682 10718 30.255 3.788e-08 \*\*\*  
## + campaign 1 10692 10728 20.487 6.003e-06 \*\*\*  
## + marital 2 10698 10736 13.981 0.0009206 \*\*\*  
## + balance 1 10701 10737 11.070 0.0008772 \*\*\*  
## + pdays 1 10704 10740 7.888 0.0049753 \*\*   
## <none> 10712 10746   
## + previous 1 10711 10747 1.456 0.2275573   
## + age 1 10712 10748 0.427 0.5135798   
## + default 1 10712 10748 0.152 0.6966637   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Step: AIC=10551.89  
## as.factor(y) ~ month + duration + poutcome + year + housing  
##   
## Df Deviance AIC LRT Pr(Chi)   
## + contact 2 10419 10459 96.741 < 2.2e-16 \*\*\*  
## + day 1 10455 10493 60.826 6.234e-15 \*\*\*  
## + education 3 10456 10498 59.928 6.089e-13 \*\*\*  
## + job 11 10446 10504 69.373 1.607e-10 \*\*\*  
## + campaign 1 10496 10534 19.754 8.810e-06 \*\*\*  
## + loan 1 10503 10541 12.441 0.0004199 \*\*\*  
## + marital 2 10507 10547 8.996 0.0111325 \*   
## + balance 1 10512 10550 4.432 0.0352806 \*   
## <none> 10516 10552   
## + pdays 1 10515 10553 1.054 0.3046656   
## + age 1 10515 10553 0.900 0.3426832   
## + previous 1 10516 10554 0.239 0.6251236   
## + default 1 10516 10554 0.026 0.8714786   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Step: AIC=10459.15  
## as.factor(y) ~ month + duration + poutcome + year + housing +   
## contact  
##   
## Df Deviance AIC LRT Pr(Chi)   
## + day 1 10354 10396 65.239 6.636e-16 \*\*\*  
## + education 3 10375 10421 43.919 1.570e-09 \*\*\*  
## + job 11 10361 10423 57.745 2.425e-08 \*\*\*  
## + campaign 1 10400 10442 19.251 1.146e-05 \*\*\*  
## + loan 1 10408 10450 11.176 0.0008286 \*\*\*  
## + marital 2 10412 10456 7.002 0.0301738 \*   
## + balance 1 10415 10457 3.708 0.0541395 .   
## <none> 10419 10459   
## + previous 1 10419 10461 0.424 0.5148065   
## + pdays 1 10419 10461 0.163 0.6868622   
## + age 1 10419 10461 0.044 0.8331325   
## + default 1 10419 10461 0.002 0.9671093   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Step: AIC=10395.91  
## as.factor(y) ~ month + duration + poutcome + year + housing +   
## contact + day  
##   
## Df Deviance AIC LRT Pr(Chi)   
## + education 3 10311 10359 43.034 2.420e-09 \*\*\*  
## + job 11 10301 10365 53.256 1.613e-07 \*\*\*  
## + campaign 1 10332 10376 22.244 2.401e-06 \*\*\*  
## + loan 1 10343 10387 10.616 0.001121 \*\*   
## + marital 2 10348 10394 6.133 0.046588 \*   
## + balance 1 10350 10394 3.711 0.054070 .   
## <none> 10354 10396   
## + previous 1 10354 10398 0.221 0.638085   
## + pdays 1 10354 10398 0.134 0.714436   
## + age 1 10354 10398 0.084 0.771311   
## + default 1 10354 10398 0.001 0.970270   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Step: AIC=10358.87  
## as.factor(y) ~ month + duration + poutcome + year + housing +   
## contact + day + education  
##   
## Df Deviance AIC LRT Pr(Chi)   
## + campaign 1 10289 10339 21.9773 2.759e-06 \*\*\*  
## + loan 1 10301 10351 10.1579 0.001437 \*\*   
## + job 11 10283 10353 28.0140 0.003221 \*\*   
## + age 1 10308 10358 3.3018 0.069205 .   
## + balance 1 10308 10358 2.8110 0.093619 .   
## <none> 10311 10359   
## + marital 2 10308 10360 3.1132 0.210855   
## + previous 1 10311 10361 0.1863 0.666055   
## + pdays 1 10311 10361 0.0305 0.861443   
## + default 1 10311 10361 0.0149 0.902914   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Step: AIC=10338.9  
## as.factor(y) ~ month + duration + poutcome + year + housing +   
## contact + day + education + campaign  
##   
## Df Deviance AIC LRT Pr(Chi)   
## + loan 1 10279 10331 9.9904 0.001574 \*\*  
## + job 11 10262 10334 26.7182 0.005067 \*\*  
## + age 1 10286 10338 3.2177 0.072847 .   
## + balance 1 10286 10338 2.7789 0.095513 .   
## <none> 10289 10339   
## + marital 2 10286 10340 2.7866 0.248252   
## + default 1 10289 10341 0.0269 0.869635   
## + pdays 1 10289 10341 0.0115 0.914681   
## + previous 1 10289 10341 0.0097 0.921497   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Step: AIC=10330.91  
## as.factor(y) ~ month + duration + poutcome + year + housing +   
## contact + day + education + campaign + loan  
##   
## Df Deviance AIC LRT Pr(Chi)   
## + job 11 10253 10327 26.0375 0.006407 \*\*  
## + age 1 10276 10330 3.2501 0.071421 .   
## + balance 1 10277 10331 2.2981 0.129530   
## <none> 10279 10331   
## + marital 2 10276 10332 2.4947 0.287269   
## + default 1 10279 10333 0.0427 0.836312   
## + pdays 1 10279 10333 0.0072 0.932173   
## + previous 1 10279 10333 0.0045 0.946371   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Step: AIC=10326.87  
## as.factor(y) ~ month + duration + poutcome + year + housing +   
## contact + day + education + campaign + loan + job  
##   
## Df Deviance AIC LRT Pr(Chi)  
## + age 1 10250 10326 2.47355 0.1158  
## + balance 1 10251 10327 2.13020 0.1444  
## <none> 10253 10327   
## + default 1 10253 10329 0.05445 0.8155  
## + previous 1 10253 10329 0.01248 0.9110  
## + pdays 1 10253 10329 0.00537 0.9416  
## + marital 2 10251 10329 1.52524 0.4664  
##   
## Step: AIC=10326.4  
## as.factor(y) ~ month + duration + poutcome + year + housing +   
## contact + day + education + campaign + loan + job + age  
##   
## Df Deviance AIC LRT Pr(Chi)  
## <none> 10250 10326   
## + balance 1 10249 10327 1.81672 0.1777  
## + marital 2 10247 10327 3.12076 0.2101  
## + default 1 10250 10328 0.05682 0.8116  
## + previous 1 10250 10328 0.01436 0.9046  
## + pdays 1 10250 10328 0.00457 0.9461

summary(Model\_FWD)

##   
## Call:  
## glm(formula = as.factor(y) ~ month + duration + poutcome + year +   
## housing + contact + day + education + campaign + loan + job +   
## age, family = binomial(link = "logit"), data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.8561 -0.5369 -0.3272 -0.1782 3.1320   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.8405275 0.2340604 -20.681 < 2e-16 \*\*\*  
## monthfeb 1.3991274 0.1477821 9.468 < 2e-16 \*\*\*  
## monthmar 2.7519347 0.1694256 16.243 < 2e-16 \*\*\*  
## monthapr 1.4986044 0.1325973 11.302 < 2e-16 \*\*\*  
## monthmay 1.1489027 0.1361363 8.439 < 2e-16 \*\*\*  
## monthjun 2.5289512 0.1576144 16.045 < 2e-16 \*\*\*  
## monthjul 2.2373107 0.1751887 12.771 < 2e-16 \*\*\*  
## monthaug 2.1820028 0.1475846 14.785 < 2e-16 \*\*\*  
## monthsep 2.2314918 0.1690511 13.200 < 2e-16 \*\*\*  
## monthoct 2.0099592 0.1573271 12.776 < 2e-16 \*\*\*  
## monthnov 2.6226931 0.1843656 14.225 < 2e-16 \*\*\*  
## monthdec 2.6334117 0.2239579 11.759 < 2e-16 \*\*\*  
## duration 0.0036533 0.0001067 34.244 < 2e-16 \*\*\*  
## poutcomeother 0.1703772 0.1056344 1.613 0.10677   
## poutcomesuccess 1.8114790 0.0922438 19.638 < 2e-16 \*\*\*  
## poutcomeunknown 0.4803633 0.0694529 6.916 4.63e-12 \*\*\*  
## year2010 1.0654236 0.0695660 15.315 < 2e-16 \*\*\*  
## housingyes -0.7012453 0.0621319 -11.286 < 2e-16 \*\*\*  
## contacttelephone -0.5265140 0.1011343 -5.206 1.93e-07 \*\*\*  
## contactunknown -1.6583867 0.2525171 -6.567 5.12e-11 \*\*\*  
## day 0.0268217 0.0033867 7.920 2.38e-15 \*\*\*  
## educationsecondary 0.2291422 0.0928646 2.467 0.01361 \*   
## educationtertiary 0.4399502 0.1054116 4.174 3.00e-05 \*\*\*  
## educationunknown 0.2509384 0.1421357 1.765 0.07748 .   
## campaign -0.0826160 0.0186840 -4.422 9.79e-06 \*\*\*  
## loanyes -0.2862562 0.0955208 -2.997 0.00273 \*\*   
## jobblue-collar -0.3265862 0.1038092 -3.146 0.00166 \*\*   
## jobentrepreneur -0.5818199 0.1961553 -2.966 0.00302 \*\*   
## jobhousemaid -0.1682360 0.1907008 -0.882 0.37767   
## jobmanagement -0.0062541 0.0979995 -0.064 0.94912   
## jobretired -0.0602656 0.1335506 -0.451 0.65180   
## jobself-employed -0.0835345 0.1586317 -0.527 0.59848   
## jobservices -0.1385329 0.1177730 -1.176 0.23949   
## jobstudent 0.1189071 0.1300069 0.915 0.36039   
## jobtechnician -0.0738785 0.0951709 -0.776 0.43759   
## jobunemployed -0.1980792 0.1463805 -1.353 0.17600   
## jobunknown -0.2273303 0.3204025 -0.710 0.47801   
## age 0.0042197 0.0026812 1.574 0.11554   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 14816 on 13984 degrees of freedom  
## Residual deviance: 10250 on 13947 degrees of freedom  
## AIC: 10326  
##   
## Number of Fisher Scoring iterations: 5

#Degrees of Freedom: 13983 Total (i.e. Null); 13955 Residual  
#Null Deviance: 14840   
#Residual Deviance: 10250 AIC: 10310  
hoslem.test(Model\_FWD$y, fitted(Model\_FWD), g=10)

##   
## Hosmer and Lemeshow goodness of fit (GOF) test  
##   
## data: Model\_FWD$y, fitted(Model\_FWD)  
## X-squared = 221.15, df = 8, p-value < 2.2e-16

#p-value < 2.2e-16  
# rejects. But the sample size is large, so I don't think it's an issue  
exp(cbind("Odds ratio" = coef(Model\_FWD), confint.default(Model\_FWD, level = 0.95)))

## Odds ratio 2.5 % 97.5 %  
## (Intercept) 0.007902884 0.004995201 0.01250311  
## monthfeb 4.051662816 3.032780081 5.41284601  
## monthmar 15.672924662 11.244359102 21.84567081  
## monthapr 4.475438833 3.451187601 5.80366965  
## monthmay 3.154729352 2.415920841 4.11947160  
## monthjun 12.540346992 9.207632301 17.07934218  
## monthjul 9.368103616 6.645548519 13.20603786  
## monthaug 8.864041016 6.637545712 11.83739089  
## monthsep 9.313749675 6.686949427 12.97242247  
## monthoct 7.463012522 5.482732768 10.15853923  
## monthnov 13.772765652 9.595969408 19.76757799  
## monthdec 13.921184492 8.975173908 21.59282702  
## duration 1.003660014 1.003450173 1.00386990  
## poutcomeother 1.185752032 0.964001445 1.45851221  
## poutcomesuccess 6.119491481 5.107368974 7.33218535  
## poutcomeunknown 1.616661697 1.410914671 1.85241184  
## year2010 2.902067932 2.532170037 3.32600029  
## housingyes 0.495967310 0.439102895 0.56019574  
## contacttelephone 0.590660420 0.484453606 0.72015096  
## contactunknown 0.190445974 0.116099090 0.31240270  
## day 1.027184668 1.020389027 1.03402557  
## educationsecondary 1.257520792 1.048259232 1.50855675  
## educationtertiary 1.552629969 1.262819910 1.90894980  
## educationunknown 1.285230885 0.972736121 1.69811565  
## campaign 0.920704651 0.887598329 0.95504580  
## loanyes 0.751070159 0.622835074 0.90570748  
## jobblue-collar 0.721382194 0.588576488 0.88415402  
## jobentrepreneur 0.558880344 0.380496919 0.82089295  
## jobhousemaid 0.845154342 0.581582373 1.22817660  
## jobmanagement 0.993765427 0.820099455 1.20420727  
## jobretired 0.941514403 0.724683715 1.22322242  
## jobself-employed 0.919859349 0.674052734 1.25530419  
## jobservices 0.870634564 0.691173917 1.09669148  
## jobstudent 1.126265252 0.872928431 1.45312419  
## jobtechnician 0.928784524 0.770735475 1.11924353  
## jobunemployed 0.820304866 0.615709634 1.09288541  
## jobunknown 0.796657574 0.425152366 1.49279021  
## age 1.004228600 0.998965106 1.00951983

#Backward:  
Model\_Bwd<-stepAIC(Model\_Full,direction="backward",trace=FALSE)  
summary(Model\_Bwd)

##   
## Call:  
## glm(formula = as.factor(y) ~ age + job + education + housing +   
## loan + contact + day + month + duration + campaign + poutcome +   
## year, family = binomial(link = "logit"), data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.8561 -0.5369 -0.3272 -0.1782 3.1320   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.8405275 0.2340604 -20.681 < 2e-16 \*\*\*  
## age 0.0042197 0.0026812 1.574 0.11554   
## jobblue-collar -0.3265862 0.1038092 -3.146 0.00166 \*\*   
## jobentrepreneur -0.5818199 0.1961553 -2.966 0.00302 \*\*   
## jobhousemaid -0.1682360 0.1907008 -0.882 0.37767   
## jobmanagement -0.0062541 0.0979995 -0.064 0.94912   
## jobretired -0.0602656 0.1335506 -0.451 0.65180   
## jobself-employed -0.0835345 0.1586317 -0.527 0.59848   
## jobservices -0.1385329 0.1177730 -1.176 0.23949   
## jobstudent 0.1189071 0.1300069 0.915 0.36039   
## jobtechnician -0.0738785 0.0951709 -0.776 0.43759   
## jobunemployed -0.1980792 0.1463805 -1.353 0.17600   
## jobunknown -0.2273303 0.3204025 -0.710 0.47801   
## educationsecondary 0.2291422 0.0928646 2.467 0.01361 \*   
## educationtertiary 0.4399502 0.1054116 4.174 3.00e-05 \*\*\*  
## educationunknown 0.2509384 0.1421357 1.765 0.07748 .   
## housingyes -0.7012453 0.0621319 -11.286 < 2e-16 \*\*\*  
## loanyes -0.2862562 0.0955208 -2.997 0.00273 \*\*   
## contacttelephone -0.5265140 0.1011343 -5.206 1.93e-07 \*\*\*  
## contactunknown -1.6583867 0.2525171 -6.567 5.12e-11 \*\*\*  
## day 0.0268217 0.0033867 7.920 2.38e-15 \*\*\*  
## monthfeb 1.3991274 0.1477821 9.468 < 2e-16 \*\*\*  
## monthmar 2.7519347 0.1694256 16.243 < 2e-16 \*\*\*  
## monthapr 1.4986044 0.1325973 11.302 < 2e-16 \*\*\*  
## monthmay 1.1489027 0.1361363 8.439 < 2e-16 \*\*\*  
## monthjun 2.5289512 0.1576144 16.045 < 2e-16 \*\*\*  
## monthjul 2.2373107 0.1751887 12.771 < 2e-16 \*\*\*  
## monthaug 2.1820028 0.1475846 14.785 < 2e-16 \*\*\*  
## monthsep 2.2314918 0.1690511 13.200 < 2e-16 \*\*\*  
## monthoct 2.0099592 0.1573271 12.776 < 2e-16 \*\*\*  
## monthnov 2.6226931 0.1843656 14.225 < 2e-16 \*\*\*  
## monthdec 2.6334117 0.2239579 11.759 < 2e-16 \*\*\*  
## duration 0.0036533 0.0001067 34.244 < 2e-16 \*\*\*  
## campaign -0.0826160 0.0186840 -4.422 9.79e-06 \*\*\*  
## poutcomeother 0.1703772 0.1056344 1.613 0.10677   
## poutcomesuccess 1.8114790 0.0922438 19.638 < 2e-16 \*\*\*  
## poutcomeunknown 0.4803633 0.0694529 6.916 4.63e-12 \*\*\*  
## year2010 1.0654236 0.0695660 15.315 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 14816 on 13984 degrees of freedom  
## Residual deviance: 10250 on 13947 degrees of freedom  
## AIC: 10326  
##   
## Number of Fisher Scoring iterations: 5

#Degrees of Freedom: 13983 Total (i.e. Null); 13955 Residual  
#Null Deviance: 14840   
#Residual Deviance: 10250 AIC: 10310  
hoslem.test(Model\_Bwd$y, fitted(Model\_Bwd), g=10)

##   
## Hosmer and Lemeshow goodness of fit (GOF) test  
##   
## data: Model\_Bwd$y, fitted(Model\_Bwd)  
## X-squared = 221.15, df = 8, p-value < 2.2e-16

#p-value < 2.2e-16  
# rejects. But the sample size is large, so I don't think it's an issue  
exp(cbind("Odds ratio" = coef(Model\_Bwd), confint.default(Model\_Bwd, level = 0.95)))

## Odds ratio 2.5 % 97.5 %  
## (Intercept) 0.007902884 0.004995201 0.01250311  
## age 1.004228600 0.998965106 1.00951983  
## jobblue-collar 0.721382194 0.588576488 0.88415402  
## jobentrepreneur 0.558880344 0.380496919 0.82089295  
## jobhousemaid 0.845154342 0.581582373 1.22817660  
## jobmanagement 0.993765427 0.820099455 1.20420727  
## jobretired 0.941514403 0.724683715 1.22322242  
## jobself-employed 0.919859349 0.674052734 1.25530419  
## jobservices 0.870634564 0.691173917 1.09669148  
## jobstudent 1.126265252 0.872928431 1.45312419  
## jobtechnician 0.928784524 0.770735475 1.11924353  
## jobunemployed 0.820304866 0.615709634 1.09288541  
## jobunknown 0.796657574 0.425152366 1.49279021  
## educationsecondary 1.257520792 1.048259232 1.50855675  
## educationtertiary 1.552629969 1.262819910 1.90894980  
## educationunknown 1.285230885 0.972736121 1.69811565  
## housingyes 0.495967310 0.439102895 0.56019574  
## loanyes 0.751070159 0.622835074 0.90570748  
## contacttelephone 0.590660420 0.484453606 0.72015096  
## contactunknown 0.190445974 0.116099090 0.31240270  
## day 1.027184668 1.020389027 1.03402557  
## monthfeb 4.051662816 3.032780081 5.41284601  
## monthmar 15.672924662 11.244359102 21.84567081  
## monthapr 4.475438833 3.451187601 5.80366965  
## monthmay 3.154729352 2.415920841 4.11947160  
## monthjun 12.540346992 9.207632301 17.07934218  
## monthjul 9.368103616 6.645548519 13.20603786  
## monthaug 8.864041016 6.637545712 11.83739089  
## monthsep 9.313749675 6.686949427 12.97242247  
## monthoct 7.463012522 5.482732768 10.15853923  
## monthnov 13.772765652 9.595969408 19.76757799  
## monthdec 13.921184492 8.975173908 21.59282702  
## duration 1.003660014 1.003450173 1.00386990  
## campaign 0.920704651 0.887598329 0.95504580  
## poutcomeother 1.185752032 0.964001445 1.45851221  
## poutcomesuccess 6.119491481 5.107368974 7.33218535  
## poutcomeunknown 1.616661697 1.410914671 1.85241184  
## year2010 2.902067932 2.532170037 3.32600029

#Both:  
Model\_Both<-stepAIC(Model\_Full,direction="both",trace=FALSE)  
summary(Model\_Both)

##   
## Call:  
## glm(formula = as.factor(y) ~ age + job + education + housing +   
## loan + contact + day + month + duration + campaign + poutcome +   
## year, family = binomial(link = "logit"), data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.8561 -0.5369 -0.3272 -0.1782 3.1320   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.8405275 0.2340604 -20.681 < 2e-16 \*\*\*  
## age 0.0042197 0.0026812 1.574 0.11554   
## jobblue-collar -0.3265862 0.1038092 -3.146 0.00166 \*\*   
## jobentrepreneur -0.5818199 0.1961553 -2.966 0.00302 \*\*   
## jobhousemaid -0.1682360 0.1907008 -0.882 0.37767   
## jobmanagement -0.0062541 0.0979995 -0.064 0.94912   
## jobretired -0.0602656 0.1335506 -0.451 0.65180   
## jobself-employed -0.0835345 0.1586317 -0.527 0.59848   
## jobservices -0.1385329 0.1177730 -1.176 0.23949   
## jobstudent 0.1189071 0.1300069 0.915 0.36039   
## jobtechnician -0.0738785 0.0951709 -0.776 0.43759   
## jobunemployed -0.1980792 0.1463805 -1.353 0.17600   
## jobunknown -0.2273303 0.3204025 -0.710 0.47801   
## educationsecondary 0.2291422 0.0928646 2.467 0.01361 \*   
## educationtertiary 0.4399502 0.1054116 4.174 3.00e-05 \*\*\*  
## educationunknown 0.2509384 0.1421357 1.765 0.07748 .   
## housingyes -0.7012453 0.0621319 -11.286 < 2e-16 \*\*\*  
## loanyes -0.2862562 0.0955208 -2.997 0.00273 \*\*   
## contacttelephone -0.5265140 0.1011343 -5.206 1.93e-07 \*\*\*  
## contactunknown -1.6583867 0.2525171 -6.567 5.12e-11 \*\*\*  
## day 0.0268217 0.0033867 7.920 2.38e-15 \*\*\*  
## monthfeb 1.3991274 0.1477821 9.468 < 2e-16 \*\*\*  
## monthmar 2.7519347 0.1694256 16.243 < 2e-16 \*\*\*  
## monthapr 1.4986044 0.1325973 11.302 < 2e-16 \*\*\*  
## monthmay 1.1489027 0.1361363 8.439 < 2e-16 \*\*\*  
## monthjun 2.5289512 0.1576144 16.045 < 2e-16 \*\*\*  
## monthjul 2.2373107 0.1751887 12.771 < 2e-16 \*\*\*  
## monthaug 2.1820028 0.1475846 14.785 < 2e-16 \*\*\*  
## monthsep 2.2314918 0.1690511 13.200 < 2e-16 \*\*\*  
## monthoct 2.0099592 0.1573271 12.776 < 2e-16 \*\*\*  
## monthnov 2.6226931 0.1843656 14.225 < 2e-16 \*\*\*  
## monthdec 2.6334117 0.2239579 11.759 < 2e-16 \*\*\*  
## duration 0.0036533 0.0001067 34.244 < 2e-16 \*\*\*  
## campaign -0.0826160 0.0186840 -4.422 9.79e-06 \*\*\*  
## poutcomeother 0.1703772 0.1056344 1.613 0.10677   
## poutcomesuccess 1.8114790 0.0922438 19.638 < 2e-16 \*\*\*  
## poutcomeunknown 0.4803633 0.0694529 6.916 4.63e-12 \*\*\*  
## year2010 1.0654236 0.0695660 15.315 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 14816 on 13984 degrees of freedom  
## Residual deviance: 10250 on 13947 degrees of freedom  
## AIC: 10326  
##   
## Number of Fisher Scoring iterations: 5

#Degrees of Freedom: 13983 Total (i.e. Null); 13955 Residual  
#Null Deviance: 14840   
#Residual Deviance: 10250 AIC: 10310  
hoslem.test(Model\_Both$y, fitted(Model\_Both), g=10)

##   
## Hosmer and Lemeshow goodness of fit (GOF) test  
##   
## data: Model\_Both$y, fitted(Model\_Both)  
## X-squared = 221.15, df = 8, p-value < 2.2e-16

#p-value < 2.2e-16  
# rejects. But the sample size is large, so I don't think it's an issue  
exp(cbind("Odds ratio" = coef(Model\_Both), confint.default(Model\_Both, level = 0.95)))

## Odds ratio 2.5 % 97.5 %  
## (Intercept) 0.007902884 0.004995201 0.01250311  
## age 1.004228600 0.998965106 1.00951983  
## jobblue-collar 0.721382194 0.588576488 0.88415402  
## jobentrepreneur 0.558880344 0.380496919 0.82089295  
## jobhousemaid 0.845154342 0.581582373 1.22817660  
## jobmanagement 0.993765427 0.820099455 1.20420727  
## jobretired 0.941514403 0.724683715 1.22322242  
## jobself-employed 0.919859349 0.674052734 1.25530419  
## jobservices 0.870634564 0.691173917 1.09669148  
## jobstudent 1.126265252 0.872928431 1.45312419  
## jobtechnician 0.928784524 0.770735475 1.11924353  
## jobunemployed 0.820304866 0.615709634 1.09288541  
## jobunknown 0.796657574 0.425152366 1.49279021  
## educationsecondary 1.257520792 1.048259232 1.50855675  
## educationtertiary 1.552629969 1.262819910 1.90894980  
## educationunknown 1.285230885 0.972736121 1.69811565  
## housingyes 0.495967310 0.439102895 0.56019574  
## loanyes 0.751070159 0.622835074 0.90570748  
## contacttelephone 0.590660420 0.484453606 0.72015096  
## contactunknown 0.190445974 0.116099090 0.31240270  
## day 1.027184668 1.020389027 1.03402557  
## monthfeb 4.051662816 3.032780081 5.41284601  
## monthmar 15.672924662 11.244359102 21.84567081  
## monthapr 4.475438833 3.451187601 5.80366965  
## monthmay 3.154729352 2.415920841 4.11947160  
## monthjun 12.540346992 9.207632301 17.07934218  
## monthjul 9.368103616 6.645548519 13.20603786  
## monthaug 8.864041016 6.637545712 11.83739089  
## monthsep 9.313749675 6.686949427 12.97242247  
## monthoct 7.463012522 5.482732768 10.15853923  
## monthnov 13.772765652 9.595969408 19.76757799  
## monthdec 13.921184492 8.975173908 21.59282702  
## duration 1.003660014 1.003450173 1.00386990  
## campaign 0.920704651 0.887598329 0.95504580  
## poutcomeother 1.185752032 0.964001445 1.45851221  
## poutcomesuccess 6.119491481 5.107368974 7.33218535  
## poutcomeunknown 1.616661697 1.410914671 1.85241184  
## year2010 2.902067932 2.532170037 3.32600029

#Simple model by hand selecting the predictors  
Model\_SIM1<-glm(as.factor(y) ~ age+balance+loan+duration+campaign+previous+year,data=train,family= binomial(link="logit"))  
summary(Model\_SIM1)

##   
## Call:  
## glm(formula = as.factor(y) ~ age + balance + loan + duration +   
## campaign + previous + year, family = binomial(link = "logit"),   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.8033 -0.6016 -0.4732 -0.3104 2.8272   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.622e+00 8.861e-02 -29.585 < 2e-16 \*\*\*  
## age 7.198e-03 1.779e-03 4.045 5.23e-05 \*\*\*  
## balance 4.007e-05 7.020e-06 5.707 1.15e-08 \*\*\*  
## loanyes -7.230e-01 8.779e-02 -8.236 < 2e-16 \*\*\*  
## duration 3.190e-03 9.483e-05 33.644 < 2e-16 \*\*\*  
## campaign -1.112e-01 1.726e-02 -6.445 1.16e-10 \*\*\*  
## previous 2.815e-03 8.768e-03 0.321 0.748   
## year2010 1.658e+00 5.562e-02 29.819 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 14816 on 13984 degrees of freedom  
## Residual deviance: 12140 on 13977 degrees of freedom  
## AIC: 12156  
##   
## Number of Fisher Scoring iterations: 5

#Null deviance: 14843 on 13983 degrees of freedom  
#Residual deviance: 12141 on 13976 degrees of freedom  
#AIC: 12157  
hoslem.test(Model\_SIM1$y, fitted(Model\_SIM1), g=10)

##   
## Hosmer and Lemeshow goodness of fit (GOF) test  
##   
## data: Model\_SIM1$y, fitted(Model\_SIM1)  
## X-squared = 110.01, df = 8, p-value < 2.2e-16

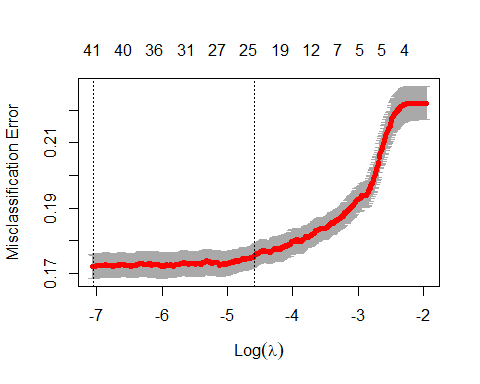
#p-value < 2.2e-16  
# rejects. But the sample size is large, so I don't think it's an issue  
  
#Simple model2 by hand seleting the predictors  
Model\_SIM2<-glm(as.factor(y) ~ age+job+education+balance+loan+duration+campaign+previous+year,data=train,family= binomial(link="logit"))  
summary(Model\_SIM2)

##   
## Call:  
## glm(formula = as.factor(y) ~ age + job + education + balance +   
## loan + duration + campaign + previous + year, family = binomial(link = "logit"),   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.8521 -0.6097 -0.4485 -0.2582 2.6665   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.942e+00 1.555e-01 -18.923 < 2e-16 \*\*\*  
## age 8.501e-03 2.460e-03 3.456 0.000548 \*\*\*  
## jobblue-collar -6.492e-01 9.536e-02 -6.808 9.93e-12 \*\*\*  
## jobentrepreneur -7.448e-01 1.804e-01 -4.129 3.65e-05 \*\*\*  
## jobhousemaid 3.602e-02 1.749e-01 0.206 0.836867   
## jobmanagement -1.495e-02 9.029e-02 -0.166 0.868513   
## jobretired 2.336e-01 1.222e-01 1.912 0.055837 .   
## jobself-employed -1.374e-01 1.453e-01 -0.946 0.344351   
## jobservices -3.371e-01 1.079e-01 -3.125 0.001776 \*\*   
## jobstudent 5.407e-01 1.195e-01 4.523 6.09e-06 \*\*\*  
## jobtechnician -8.444e-02 8.724e-02 -0.968 0.333094   
## jobunemployed -1.325e-01 1.323e-01 -1.001 0.316673   
## jobunknown 4.195e-02 3.092e-01 0.136 0.892082   
## educationsecondary 2.648e-01 8.624e-02 3.070 0.002140 \*\*   
## educationtertiary 6.321e-01 9.767e-02 6.472 9.69e-11 \*\*\*  
## educationunknown 3.762e-01 1.317e-01 2.857 0.004282 \*\*   
## balance 3.050e-05 7.007e-06 4.353 1.34e-05 \*\*\*  
## loanyes -6.448e-01 8.928e-02 -7.222 5.13e-13 \*\*\*  
## duration 3.327e-03 9.753e-05 34.110 < 2e-16 \*\*\*  
## campaign -1.049e-01 1.736e-02 -6.045 1.49e-09 \*\*\*  
## previous 5.199e-03 8.826e-03 0.589 0.555842   
## year2010 1.507e+00 5.689e-02 26.498 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 14816 on 13984 degrees of freedom  
## Residual deviance: 11846 on 13963 degrees of freedom  
## AIC: 11890  
##   
## Number of Fisher Scoring iterations: 5

#Null deviance: 14843 on 13983 degrees of freedom  
#Residual deviance: 11856 on 13962 degrees of freedom  
#AIC: 11900  
hoslem.test(Model\_SIM1$y, fitted(Model\_SIM2), g=10)

##   
## Hosmer and Lemeshow goodness of fit (GOF) test  
##   
## data: Model\_SIM1$y, fitted(Model\_SIM2)  
## X-squared = 138.99, df = 8, p-value < 2.2e-16

#p-value < 2.2e-16  
# rejects. But the sample size is large, so I don't think it's an issue  
  
#LASSO  
#LASSO is obtained in the exact same way for logistic as is MLR. The only difference is to let R know that our response is categorical through the "family="binomial"" option. Cross validation is used to obtain the optimal penalty value. A final refit using the entire data set can then be obtained once the optimal penalty value is determined. For this example, the object "finalmodel" produces the final lasso model  
dat.train.x <- model.matrix(y~age+job+marital+education+default+balance+housing+loan+contact+day+month+duration+campaign+pdays+previous+poutcome+year-1,train)  
dat.train.y<-train$y  
cvfit <- cv.glmnet(dat.train.x, dat.train.y, family = "binomial", type.measure = "class", nlambda = 1000)  
plot(cvfit)



coef(cvfit, s = "lambda.min")

## 46 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) -3.135940e+00  
## age 3.876147e-03  
## jobadmin. 5.970140e-02  
## jobblue-collar -2.446822e-01  
## jobentrepreneur -4.470396e-01  
## jobhousemaid -6.308227e-02  
## jobmanagement 6.372287e-02  
## jobretired 5.392798e-04  
## jobself-employed .   
## jobservices -4.151518e-02  
## jobstudent 1.574508e-01  
## jobtechnician .   
## jobunemployed -1.087220e-01  
## jobunknown -7.396316e-02  
## maritalmarried -7.315182e-02  
## maritalsingle 9.052000e-03  
## educationsecondary 1.195779e-01  
## educationtertiary 3.246692e-01  
## educationunknown 1.268397e-01  
## defaultyes .   
## balance 9.686831e-06  
## housingyes -6.741403e-01  
## loanyes -2.607401e-01  
## contacttelephone -4.860873e-01  
## contactunknown -1.498453e+00  
## day 2.071517e-02  
## monthfeb 8.721676e-01  
## monthmar 2.259378e+00  
## monthapr 1.037993e+00  
## monthmay 6.459701e-01  
## monthjun 2.007804e+00  
## monthjul 1.728842e+00  
## monthaug 1.697297e+00  
## monthsep 1.729026e+00  
## monthoct 1.542947e+00  
## monthnov 2.110781e+00  
## monthdec 2.117202e+00  
## duration 3.563375e-03  
## campaign -7.153557e-02  
## pdays -3.592797e-07  
## previous .   
## poutcomeother 8.929816e-02  
## poutcomesuccess 1.748185e+00  
## poutcomeunknown 4.046551e-01  
## year2009 -1.032423e+00  
## year2010 1.581718e-11

#CV misclassification error rate is little below .1  
print("CV Error Rate:")

## [1] "CV Error Rate:"

cvfit$cvm[which(cvfit$lambda==cvfit$lambda.min)]

## [1] 0.172113

#Optimal penalty  
print("Penalty Value:")

## [1] "Penalty Value:"

cvfit$lambda.min

## [1] 0.000865764

#For final model predictions go ahead and refit lasso using entire  
#data set  
finalmodel<-glmnet(dat.train.x, dat.train.y, family = "binomial",lambda=cvfit$lambda.min)

#Lets compare the stepwise and lasso models using the test set.However the true predictions from the models are predictive probabalities. To help get a handle on this, the following code makes predictions on the test set using the LASSO model.  
dat.test.x<-model.matrix(y~age+job+marital+education+default+balance+housing+loan+contact+day+month+duration+campaign+pdays+previous+poutcome+year-1,test)  
dat.test.y<-test$y  
fit.pred.lasso <- predict(finalmodel, newx = dat.test.x, type = "response")  
  
test$y[1:15]

## [1] "no" "no" "no" "no" "no" "no" "yes" "no" "no" "yes" "no" "no"   
## [13] "no" "no" "yes"

fit.pred.lasso[1:15]

## [1] 0.24703581 0.06148402 0.02339129 0.04356659 0.05119095 0.09347206  
## [7] 0.57613515 0.02279182 0.03580984 0.75344791 0.18823154 0.26835793  
## [13] 0.33391074 0.16994207 0.30028799

#Making predictions for stepwise as well for later  
fit.pred.step<-predict(Model\_Step,newdata=test,type="response")  
summary(fit.pred.step)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.004187 0.046168 0.109402 0.226842 0.332118 0.999994

#Confusion Matrix  
#Lets use the predicted probablities to classify the observations and make a final confusion matrix for the two models. We can use it to calculate error metrics.  
#Lets use a cutoff of 0.5 to make the classification.  
cutoff<-0.5  
class.lasso<-factor(ifelse(fit.pred.lasso>cutoff,"High","Low"),levels=c("Low","High"))  
class.step<-factor(ifelse(fit.pred.step>cutoff,"High","Low"),levels=c("Low","High"))  
  
#Confusion Matrix for Lasso  
conf.lasso<-table(class.lasso,test$y)  
print("Confusion matrix for LASSO")

## [1] "Confusion matrix for LASSO"

conf.lasso

##   
## class.lasso no yes  
## Low 2537 435  
## High 179 345

conf.step<-table(class.step,test$y)  
print("Confusion matrix for Stepwise")

## [1] "Confusion matrix for Stepwise"

conf.step

##   
## class.step no yes  
## Low 2529 429  
## High 187 351

#Accuracy of LASSO and Stepwise  
print("Overall accuracy for LASSO and Stepwise respectively")

## [1] "Overall accuracy for LASSO and Stepwise respectively"

sum(diag(conf.lasso))/sum(conf.lasso)

## [1] 0.8243707

sum(diag(conf.step))/sum(conf.step)

## [1] 0.8237986

print("Alternative calculations of accuracy")

## [1] "Alternative calculations of accuracy"

#Rather than making the calculations from the table, we can compute them more quickly using the following code which just checks if the prediction matches the truth and then computes the proportion.  
mean(class.lasso==test$y)

## [1] 0

mean(class.step==test$y)

## [1] 0

# Obviously these need some adjustment (ability to predict yes is not great) - look into better cutoff  
cutoff<-0.9  
class.lasso<-factor(ifelse(fit.pred.lasso>cutoff,"High","Low"),levels=c("Low","High"))  
class.step<-factor(ifelse(fit.pred.step>cutoff,"High","Low"),levels=c("Low","High"))  
  
#Confusion Matrix for Lasso  
conf.lasso<-table(class.lasso,test$y)  
print("Confusion matrix for LASSO")

## [1] "Confusion matrix for LASSO"

conf.lasso

##   
## class.lasso no yes  
## Low 2697 718  
## High 19 62

conf.step<-table(class.step,test$y)  
print("Confusion matrix for Stepwise")

## [1] "Confusion matrix for Stepwise"

conf.step

##   
## class.step no yes  
## Low 2694 717  
## High 22 63

#Accuracy of LASSO and Stepwise  
print("Overall accuracy for LASSO and Stepwise respectively")

## [1] "Overall accuracy for LASSO and Stepwise respectively"

sum(diag(conf.lasso))/sum(conf.lasso)

## [1] 0.7891876

sum(diag(conf.step))/sum(conf.step)

## [1] 0.7886156

print("Alternative calculations of accuracy")

## [1] "Alternative calculations of accuracy"

#Rather than making the calculations from the table, we can compute them more quickly using the following code which just checks if the prediction matches the truth and then computes the proportion.  
mean(class.lasso==test$y)

## [1] 0

mean(class.step==test$y)

## [1] 0

cutoff<-0.1  
class.lasso<-factor(ifelse(fit.pred.lasso>cutoff,"High","Low"),levels=c("Low","High"))  
class.step<-factor(ifelse(fit.pred.step>cutoff,"High","Low"),levels=c("Low","High"))  
  
#Confusion Matrix for Lasso  
conf.lasso<-table(class.lasso,test$y)  
print("Confusion matrix for LASSO")

## [1] "Confusion matrix for LASSO"

conf.lasso

##   
## class.lasso no yes  
## Low 1579 39  
## High 1137 741

conf.step<-table(class.step,test$y)  
print("Confusion matrix for Stepwise")

## [1] "Confusion matrix for Stepwise"

conf.step

##   
## class.step no yes  
## Low 1611 43  
## High 1105 737

#Accuracy of LASSO and Stepwise  
print("Overall accuracy for LASSO and Stepwise respectively")

## [1] "Overall accuracy for LASSO and Stepwise respectively"

sum(diag(conf.lasso))/sum(conf.lasso)

## [1] 0.6636156

sum(diag(conf.step))/sum(conf.step)

## [1] 0.6716247

print("Alternative calculations of accuracy")

## [1] "Alternative calculations of accuracy"

#Rather than making the calculations from the table, we can compute them more quickly using the following code which just checks if the prediction matches the truth and then computes the proportion.  
mean(class.lasso==test$y)

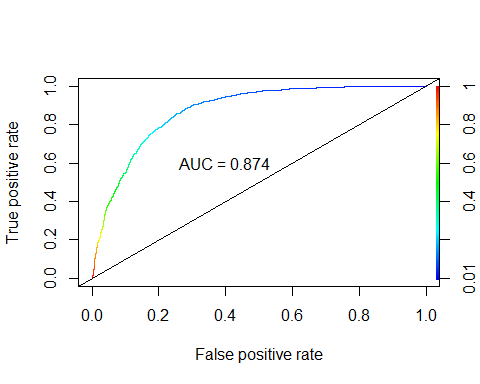
## [1] 0

mean(class.step==test$y)

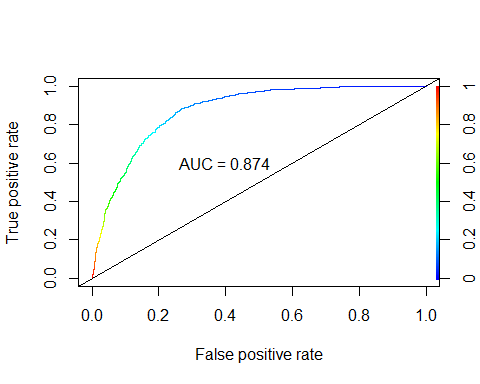
## [1] 0

#The 0.5 cut predicts 82% from LAZZO and 82% from stepwise

#ROC curves: LASSO model on test set  
pred1 <- prediction(fit.pred.lasso[,1], dat.test.y)  
roc.perf1 = performance(pred1, measure = "tpr", x.measure = "fpr")  
auc.val1 <- performance(pred1, measure = "auc")  
auc.val1 <- auc.val1@y.values  
plot(roc.perf1, colorize=TRUE)  
abline(a=0, b= 1)  
text(x = .40, y = .6,paste("AUC = ", round(auc.val1[[1]],3), sep = ""))

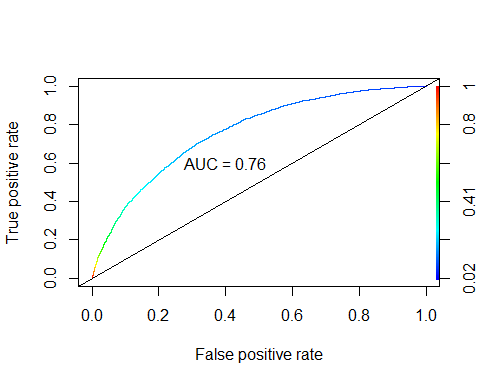


# AUC=0.87. Is there a problem? I thought the training set should get a better AUC?  
  
# Code from HW 12  
#ROC curves: Step model on test set  
results.step<-prediction(fit.pred.step, test$y,label.ordering=c("no","yes"))  
roc.step = performance(results.step, measure = "tpr", x.measure = "fpr")  
auc.val.step = performance(results.step, measure="auc")  
auc.val.step = auc.val.step@y.values  
plot(roc.step,colorize = TRUE)  
abline(a=0, b= 1)  
text(x = .40, y = .6,paste("AUC = ", round(auc.val.step[[1]],3), sep = ""))

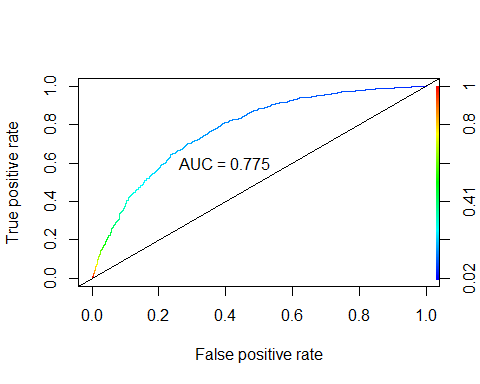


# AUC=0.872

#Training Set  
lda.train.x <- train[,c(1,6,10,12,13,14,15)]  
lda.train.y <- train$y  
fit.lda <- lda(lda.train.y ~ ., data = lda.train.x)  
pred.lda <- predict(fit.lda, newdata = lda.train.x)  
preds <- pred.lda$posterior  
preds <- as.data.frame(preds)  
# ROC for TRAINING data  
pred <- prediction(preds[,2],lda.train.y)  
roc.perf = performance(pred, measure = "tpr", x.measure = "fpr")  
auc.train <- performance(pred, measure = "auc")  
auc.train <- auc.train@y.values  
plot(roc.perf, colorize=TRUE)  
abline(a=0, b= 1)  
text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))

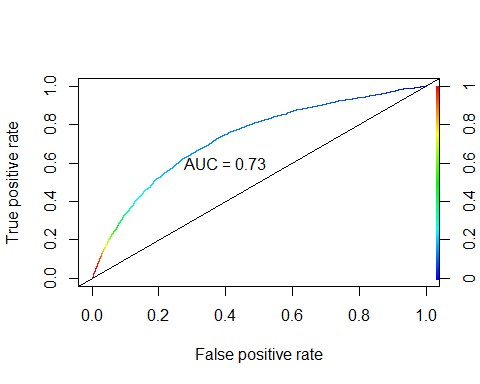


# AUC=0.763. Not surprising. Maybe better if we normalize and then weed out some variables  
  
#LDA for TEST data  
lda.test.x <- test[,c(1,6,10,12,13,14,15)]  
lda.test.y <- test$y  
pred.lda1 <- predict(fit.lda, newdata = lda.test.x)  
preds1 <- pred.lda1$posterior  
preds1 <- as.data.frame(preds1)  
pred1 <- prediction(preds1[,2],lda.test.y)  
roc.perf = performance(pred1, measure = "tpr", x.measure = "fpr")  
auc.train <- performance(pred1, measure = "auc")  
auc.train <- auc.train@y.values  
plot(roc.perf, colorize=TRUE)  
abline(a=0, b= 1)  
text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))

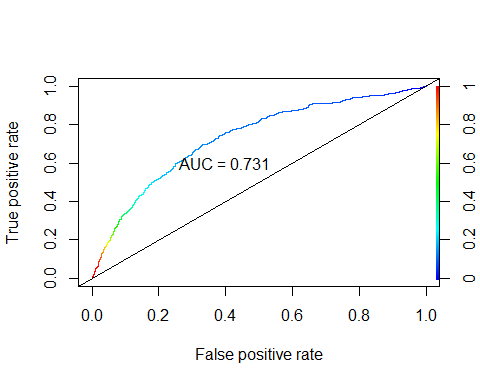


# AUC = 0.768

# Double check this Nicole  
fit.qda <- qda(lda.train.y ~ ., data = lda.train.x)  
pred.qda <- predict(fit.qda, newdata = lda.train.x)  
preds <- pred.qda$posterior  
preds <- as.data.frame(preds)  
# ROC for TRAINING data  
pred <- prediction(preds[,2],lda.train.y)  
roc.perf = performance(pred, measure = "tpr", x.measure = "fpr")  
auc.train <- performance(pred, measure = "auc")  
auc.train <- auc.train@y.values  
plot(roc.perf, colorize=TRUE)  
abline(a=0, b= 1)  
text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))



# AUC: 0.732  
  
# QDA for TEST data  
pred.qda1 <- predict(fit.qda, newdata = lda.test.x)  
preds1 <- pred.qda1$posterior  
preds1 <- as.data.frame(preds1)  
pred1 <- prediction(preds1[,2],lda.test.y)  
roc.perf = performance(pred1, measure = "tpr", x.measure = "fpr")  
auc.train <- performance(pred1, measure = "auc")  
auc.train <- auc.train@y.values  
plot(roc.perf, colorize=TRUE)  
abline(a=0, b= 1)  
text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))



# AUC = 0.726

# try some interactions  
#year&day and year&month  
model.complex<-glm(as.factor(y) ~ age+job+marital+education+default+balance+housing+loan+contact+day+month+duration+campaign+pdays+previous+poutcome+year+year:month+year:day, data=train,family = binomial(link="logit"))  
step(Model\_Full,  
 scope = list(upper=model.complex),  
 direction="forward",  
 test="Chisq",  
 data=newAuto)

## Start: AIC=10333.43  
## as.factor(y) ~ age + job + marital + education + default + balance +   
## housing + loan + contact + day + month + duration + campaign +   
## pdays + previous + poutcome + year  
##   
## Df Deviance AIC LRT Pr(>Chi)   
## + month:year 10 9938.5 10046 306.955 < 2.2e-16 \*\*\*  
## + day:year 1 10238.7 10329 6.752 0.009363 \*\*   
## <none> 10245.4 10333   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Step: AIC=10046.47  
## as.factor(y) ~ age + job + marital + education + default + balance +   
## housing + loan + contact + day + month + duration + campaign +   
## pdays + previous + poutcome + year + month:year  
##   
## Df Deviance AIC LRT Pr(>Chi)   
## + day:year 1 9901.0 10011 37.466 9.304e-10 \*\*\*  
## <none> 9938.5 10046   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Step: AIC=10011.01  
## as.factor(y) ~ age + job + marital + education + default + balance +   
## housing + loan + contact + day + month + duration + campaign +   
## pdays + previous + poutcome + year + month:year + day:year

##   
## Call: glm(formula = as.factor(y) ~ age + job + marital + education +   
## default + balance + housing + loan + contact + day + month +   
## duration + campaign + pdays + previous + poutcome + year +   
## month:year + day:year, family = binomial(link = "logit"),   
## data = train)  
##   
## Coefficients:  
## (Intercept) age jobblue-collar jobentrepreneur   
## -7.017e+00 3.415e-03 -2.509e-01 -4.968e-01   
## jobhousemaid jobmanagement jobretired jobself-employed   
## -1.904e-01 4.035e-02 -7.321e-02 -4.030e-02   
## jobservices jobstudent jobtechnician jobunemployed   
## -5.943e-02 1.034e-01 -4.941e-02 -1.121e-01   
## jobunknown maritalmarried maritalsingle educationsecondary   
## -1.526e-01 -1.428e-01 -1.683e-02 2.309e-01   
## educationtertiary educationunknown defaultyes balance   
## 3.973e-01 2.103e-01 2.260e-01 5.934e-06   
## housingyes loanyes contacttelephone contactunknown   
## -5.879e-01 -2.346e-01 -5.353e-01 -1.392e+00   
## day monthfeb monthmar monthapr   
## 4.774e-02 3.112e+00 5.049e+00 2.999e+00   
## monthmay monthjun monthjul monthaug   
## 2.689e+00 4.647e+00 4.470e+00 3.928e+00   
## monthsep monthoct monthnov monthdec   
## 4.454e+00 4.083e+00 4.725e+00 4.395e+00   
## duration campaign pdays previous   
## 3.896e-03 -8.309e-02 4.912e-04 1.091e-02   
## poutcomeother poutcomesuccess poutcomeunknown year2010   
## 2.103e-01 1.754e+00 6.207e-01 5.233e+00   
## monthfeb:year2010 monthmar:year2010 monthapr:year2010 monthmay:year2010   
## -3.078e+00 -4.773e+00 -2.776e+00 -2.513e+00   
## monthjun:year2010 monthjul:year2010 monthaug:year2010 monthsep:year2010   
## -4.599e+00 -4.222e+00 -3.572e+00 -4.451e+00   
## monthoct:year2010 monthnov:year2010 monthdec:year2010 day:year2010   
## -4.601e+00 -4.988e+00 NA -4.688e-02   
##   
## Degrees of Freedom: 13984 Total (i.e. Null); 13930 Residual  
## Null Deviance: 14820   
## Residual Deviance: 9901 AIC: 10010

#Degrees of Freedom: 13983 Total (i.e. Null); 13929 Residual  
#Null Deviance: 14840   
#Residual Deviance: 9900 AIC: 10010  
# Success! looks like yearxday and yearxmonth were significant  
# AIC: 10010  
  
#age&balance  
model.complex<-glm(as.factor(y) ~ age+job+marital+education+default+balance+housing+loan+contact+day+month+duration+campaign+pdays+previous+poutcome+year+age:balance, data=train,family = binomial(link="logit"))  
step(Model\_Full,  
 scope = list(upper=model.complex),  
 direction="forward",  
 test="Chisq",  
 data=newAuto)

## Start: AIC=10333.43  
## as.factor(y) ~ age + job + marital + education + default + balance +   
## housing + loan + contact + day + month + duration + campaign +   
## pdays + previous + poutcome + year  
##   
## Df Deviance AIC LRT Pr(>Chi)  
## <none> 10245 10333   
## + age:balance 1 10245 10335 0.77881 0.3775

##   
## Call: glm(formula = as.factor(y) ~ age + job + marital + education +   
## default + balance + housing + loan + contact + day + month +   
## duration + campaign + pdays + previous + poutcome + year,   
## family = binomial(link = "logit"), data = train)  
##   
## Coefficients:  
## (Intercept) age jobblue-collar jobentrepreneur   
## -4.882e+00 5.808e-03 -3.136e-01 -5.755e-01   
## jobhousemaid jobmanagement jobretired jobself-employed   
## -1.675e-01 -1.316e-03 -8.005e-02 -8.712e-02   
## jobservices jobstudent jobtechnician jobunemployed   
## -1.362e-01 9.258e-02 -7.167e-02 -2.062e-01   
## jobunknown maritalmarried maritalsingle educationsecondary   
## -2.238e-01 -7.640e-02 3.045e-02 2.277e-01   
## educationtertiary educationunknown defaultyes balance   
## 4.281e-01 2.473e-01 7.815e-02 9.688e-06   
## housingyes loanyes contacttelephone contactunknown   
## -6.916e-01 -2.739e-01 -5.323e-01 -1.653e+00   
## day monthfeb monthmar monthapr   
## 2.676e-02 1.394e+00 2.745e+00 1.494e+00   
## monthmay monthjun monthjul monthaug   
## 1.141e+00 2.522e+00 2.226e+00 2.179e+00   
## monthsep monthoct monthnov monthdec   
## 2.228e+00 2.002e+00 2.622e+00 2.628e+00   
## duration campaign pdays previous   
## 3.648e-03 -8.171e-02 1.008e-05 -1.648e-03   
## poutcomeother poutcomesuccess poutcomeunknown year2010   
## 1.676e-01 1.811e+00 4.747e-01 1.065e+00   
##   
## Degrees of Freedom: 13984 Total (i.e. Null); 13941 Residual  
## Null Deviance: 14820   
## Residual Deviance: 10250 AIC: 10330

#Degrees of Freedom: 13983 Total (i.e. Null); 13940 Residual  
#Null Deviance: 14840   
#Residual Deviance: 10230 AIC: 10320  
  
#job&balance  
model.complex<-glm(as.factor(y) ~ age+job+marital+education+default+balance+housing+loan+contact+day+month+duration+campaign+pdays+previous+poutcome+year+job:balance, data=train,family = binomial(link="logit"))  
step(Model\_Full,  
 scope = list(upper=model.complex),  
 direction="forward",  
 test="Chisq",  
 data=newAuto)

## Start: AIC=10333.43  
## as.factor(y) ~ age + job + marital + education + default + balance +   
## housing + loan + contact + day + month + duration + campaign +   
## pdays + previous + poutcome + year  
##   
## Df Deviance AIC LRT Pr(>Chi)  
## <none> 10245 10333   
## + job:balance 11 10237 10347 8.0642 0.7075

##   
## Call: glm(formula = as.factor(y) ~ age + job + marital + education +   
## default + balance + housing + loan + contact + day + month +   
## duration + campaign + pdays + previous + poutcome + year,   
## family = binomial(link = "logit"), data = train)  
##   
## Coefficients:  
## (Intercept) age jobblue-collar jobentrepreneur   
## -4.882e+00 5.808e-03 -3.136e-01 -5.755e-01   
## jobhousemaid jobmanagement jobretired jobself-employed   
## -1.675e-01 -1.316e-03 -8.005e-02 -8.712e-02   
## jobservices jobstudent jobtechnician jobunemployed   
## -1.362e-01 9.258e-02 -7.167e-02 -2.062e-01   
## jobunknown maritalmarried maritalsingle educationsecondary   
## -2.238e-01 -7.640e-02 3.045e-02 2.277e-01   
## educationtertiary educationunknown defaultyes balance   
## 4.281e-01 2.473e-01 7.815e-02 9.688e-06   
## housingyes loanyes contacttelephone contactunknown   
## -6.916e-01 -2.739e-01 -5.323e-01 -1.653e+00   
## day monthfeb monthmar monthapr   
## 2.676e-02 1.394e+00 2.745e+00 1.494e+00   
## monthmay monthjun monthjul monthaug   
## 1.141e+00 2.522e+00 2.226e+00 2.179e+00   
## monthsep monthoct monthnov monthdec   
## 2.228e+00 2.002e+00 2.622e+00 2.628e+00   
## duration campaign pdays previous   
## 3.648e-03 -8.171e-02 1.008e-05 -1.648e-03   
## poutcomeother poutcomesuccess poutcomeunknown year2010   
## 1.676e-01 1.811e+00 4.747e-01 1.065e+00   
##   
## Degrees of Freedom: 13984 Total (i.e. Null); 13941 Residual  
## Null Deviance: 14820   
## Residual Deviance: 10250 AIC: 10330

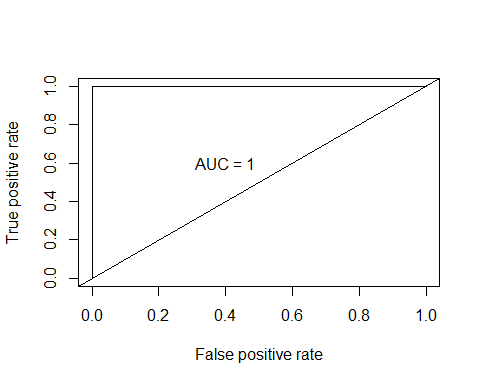
#Degrees of Freedom: 13983 Total (i.e. Null); 13940 Residual  
#Null Deviance: 14840   
#Residual Deviance: 10230 AIC: 10320  
  
#education&balance  
model.complex<-glm(as.factor(y) ~ age+job+marital+education+default+balance+housing+loan+contact+day+month+duration+campaign+pdays+previous+poutcome+year+education:balance, data=train,family = binomial(link="logit"))  
step(Model\_Full,  
 scope = list(upper=model.complex),  
 direction="forward",  
 test="Chisq",  
 data=newAuto)

## Start: AIC=10333.43  
## as.factor(y) ~ age + job + marital + education + default + balance +   
## housing + loan + contact + day + month + duration + campaign +   
## pdays + previous + poutcome + year  
##   
## Df Deviance AIC LRT Pr(>Chi)  
## <none> 10245 10333   
## + education:balance 3 10244 10338 0.9391 0.816

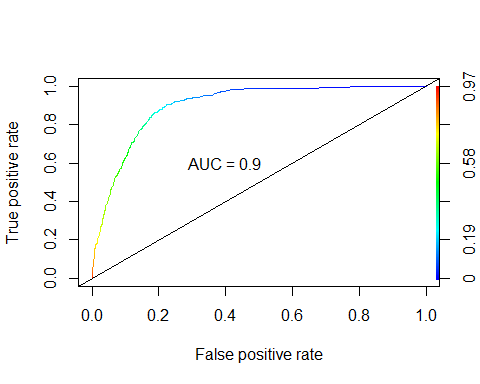
##   
## Call: glm(formula = as.factor(y) ~ age + job + marital + education +   
## default + balance + housing + loan + contact + day + month +   
## duration + campaign + pdays + previous + poutcome + year,   
## family = binomial(link = "logit"), data = train)  
##   
## Coefficients:  
## (Intercept) age jobblue-collar jobentrepreneur   
## -4.882e+00 5.808e-03 -3.136e-01 -5.755e-01   
## jobhousemaid jobmanagement jobretired jobself-employed   
## -1.675e-01 -1.316e-03 -8.005e-02 -8.712e-02   
## jobservices jobstudent jobtechnician jobunemployed   
## -1.362e-01 9.258e-02 -7.167e-02 -2.062e-01   
## jobunknown maritalmarried maritalsingle educationsecondary   
## -2.238e-01 -7.640e-02 3.045e-02 2.277e-01   
## educationtertiary educationunknown defaultyes balance   
## 4.281e-01 2.473e-01 7.815e-02 9.688e-06   
## housingyes loanyes contacttelephone contactunknown   
## -6.916e-01 -2.739e-01 -5.323e-01 -1.653e+00   
## day monthfeb monthmar monthapr   
## 2.676e-02 1.394e+00 2.745e+00 1.494e+00   
## monthmay monthjun monthjul monthaug   
## 1.141e+00 2.522e+00 2.226e+00 2.179e+00   
## monthsep monthoct monthnov monthdec   
## 2.228e+00 2.002e+00 2.622e+00 2.628e+00   
## duration campaign pdays previous   
## 3.648e-03 -8.171e-02 1.008e-05 -1.648e-03   
## poutcomeother poutcomesuccess poutcomeunknown year2010   
## 1.676e-01 1.811e+00 4.747e-01 1.065e+00   
##   
## Degrees of Freedom: 13984 Total (i.e. Null); 13941 Residual  
## Null Deviance: 14820   
## Residual Deviance: 10250 AIC: 10330

#Degrees of Freedom: 13983 Total (i.e. Null); 13940 Residual  
#Null Deviance: 14840   
#Residual Deviance: 10230 AIC: 10320

# Random Forest (training data)  
# Remove id variable as it's just for reference  
dat.train.rf <- train[,-18]  
train.rf<-randomForest(as.factor(y)~.,data=dat.train.rf,mtry=4,ntree=500,importance=T)  
fit.pred<-predict(train.rf,newdata=dat.train.rf,type="prob")  
pred <- prediction(fit.pred[,2], dat.train.rf$y)  
roc.perf = performance(pred, measure = "tpr", x.measure = "fpr")  
auc.train <- performance(pred, measure = "auc")  
auc.train <- auc.train@y.values  
plot(roc.perf)  
abline(a=0, b= 1)  
text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))



#AUC=1  
  
# Random Forest (test data)  
#Predict test set  
dat.val1.rf <- test[,-18]  
pred.val1<-predict(train.rf,newdata=dat.val1.rf,type="prob")  
pred <- prediction(pred.val1[,2], dat.val1.rf$y)  
roc.perf = performance(pred, measure = "tpr", x.measure = "fpr")  
auc.train <- performance(pred, measure = "auc")  
auc.train <- auc.train@y.values  
plot(roc.perf, colorize=TRUE)  
abline(a=0, b= 1)  
text(x = .40, y = .6,paste("AUC = ", round(auc.train[[1]],3), sep = ""))



# AUC = 0.9