Bank Marketing Analysis Project

D. Bracy , H.H. Nguyen, S.Purvis

04/02/2020

# Introduction

The retail banking industry provides financial services to families and individuals. Banks’ main functions are threefold; they issue credit in the forms of loans and credit lines, provide a secure location to deposit money, and allow a mechanism to manage finances in the form of checking and savings accounts. This analysis will focus specifically on the influential factors from direct marketing campaigns managed by a Portuguese banking institution in an attempt to get secure commitment for term deposits. Understanding not only which marketing campaigns were most effective, but also the timing of the campaign and the socioeconomic demographics will allow the retail banking industry to further target and tune their approach to securing term deposits.

# Data Description

The team was provided a substantial marketing dataset. It was comprised of categorical and continuous variables and a resulting binary result (Y/N). The data ranges from May 2008 to November 2010. As described in the table below, we have equal counts of numeric and categorical variables. There are demographics, data related to the depth and breadth of the marketing campaign, and market indicators included in this set.

|  |  |  |
| --- | --- | --- |
| Variable | Type | Description |
| Age | Numeric | Age of the Individual |
| Job | Categorical | Type of job held |
| Marital | Categorical | Marital Status |
| Education | Categorical | Level of Education of individual |
| Default | Categorical | Y/N/Unknown on whether the individual has credit in default |
| Housing | Categorical | Y/N/Unknown on whether the individual has a housing loan |
| Loan | Categorical | Y/N/Unknown on whether the individual has a personal loan |
| Contact | Categorical | Contact Communication Type |
| Month | Categorical | Month of last contact |
| Day\_of\_Week | Categorical | Day of the week of last contact – Weekdays Only |
| Duration | Numeric | Duration of last contact, in seconds. \*should only be used as a benchmark, since it can’t be known beforehand |
| Campaign | Numeric | Number of contacts performed during this campaign for this client |
| Pdays | Numeric | Number of days that passed by after a client was contacted from a previous campaign (999 means not contacted previously) |
| Previous | Numeric | Number of contacts performed before this campaign for this client |
| Poutcome | Categorical | Outcome of previous marketing campaign |
| Emp.var.rate | Numeric | Employment variation rate – quarterly indicator |
| Cons.price.idx | Numeric | Consumer Price Index – monthly indicator |
| Cons.conf.idx | Numeric | Consumer confidence index – monthly indicator |
| Euribor3m | Numeric | Euribor (Euro Interbank Offered Rate) 3 month rate – daily indicator |
| Nr.employed | Numeric | Number of employees – quarterly indicator |
| Y | Binary | Did Client subscribe to a term deposit |

# Exploratory Data Analysis (EDA)

## Objective 1:

Display the ability to perform EDA and build a logisitc regression model.

* Perform your logistic regression analysis and provide interpretation of the regression coefficients including hypothesis testing, and confidence intervals. For simplicity sake, you do not need to include interactions with this model. Comment on the practical vs statistical significance of the deemed important factors.  
  Logistical Considerations.
* Just like last time, this does not have to be extremely fancy in terms of the model building approach, let EDA, feature selection, and overall intuition guide you.
* If you feel like interactions are absolutely necessary to capture what is going on, then contact me so we can discuss an overall strategy of how to provide interpretations.

### Model Selection

#### Type of Selection

Any or all: LASSO, RIDGE, ELASTIC NET,  
 Stepwise, Forward, Backward   
 Manual / Intuition

#### Checking Assumptions

Lack of fit test  
 Influential point analysis (Cook’s D and Leverage)  
 Optional Residual Plots

#### Parameter Interpretation

Interpretation Required  
 Confidence Intervals Required

### Objective 1 Conclusion

## Objective 2

With a simple logistic regression model as a baseline, perform additional competing models to improve on prediction performance metrics. Which metrics are up to you and your given data set.

* Record the predictive performance metrics from your simple, highly interpretable model from Objective 1.
* You must include one additional logistic regression model which is also a more complicated logistic regression model than in Objective 1. By complicated, I do not mean that you include more predictors (that will be somewhat sorted out in Objective 1), but rather model complexity through interaction terms, new variables created by the group, transformations or additions through polynomials.
* Create another competing model using just the continuous predictors and use LDA or QDA.
* (Optional) Use a nonparameteric model approach as a competing model. Random forest or decision tree for predictors that are both categorical and continuous or a k-nearest neighbors approach if just working with continuous predictors.
* Provide a summary table of the performance across the competing methods. Summarize the overall findings. A really great report will also give insight as to why the “best” model won out. This is where a thorough EDA will always help. Logistical Considerations.
* Don’t forget PCA can be helpful in various ways throughout your analysis as well as other unsupervised tools such as heatmaps and cluster analysis from Unit 13.
* I think a good course of action is to tackle Objective 1 in SAS. The selection tools are really straight forward to run and the output is a little bit easier to grab. For objective 2, its better to go with R for this reason…..to ensure performance metrics are comparable make sure that the models are run on the exact same training and test sets (or through a CV approach). This can be done in SAS, it’ll just take a some additional coding to make sure it gets done properly. Additional details

NOTE 1: ALL ANALYSIS MUST BE DONE IN SAS OR R and all code must be placed in the appendix of your report. I’m okay with data cleaning steps and EDA being provided using other tools such as Python.

NOTE 2: Do not forget about organization among your group. Divide and conquer is always great, but there is “one report to rule them all” so make sure that it flows as you are stitching things together.

* Make sure it is clear how many models were created to compete against the one in Objective 1. Make note of any tuning parameters that were used and how you came up with them (knn and random forest logistics) Required

# Main Analysis Content

* Overall report of the error metrics on a test set or CV run. Also if the two best models have error rates of .05 and .045, can we really say that one model is outperforming the other? For the ambitious, McNemar’s test could be helpful in answering that.

# Conclusion/Discussion

* The conclusion should reprise the questions and conclusions of objective 2 with recommendations of the final model, what could be done to help analysis and model building in the future, and any insight as to why one method outshined all the rest if that is indeed the case. If they all are similar why did you go with your final model?

# Appendix

## EDA Plots

### Correlation Plot

##### 

### Scatterplot Matrix

##### 

## Code Section

### Read Data

#reading in 'Bank Additional' file  
bank = read.csv("./DataSets/bank-additional.csv",header = TRUE, sep = ";")  
str(bank)

## 'data.frame': 4119 obs. of 21 variables:  
## $ age : int 30 39 25 38 47 32 32 41 31 35 ...  
## $ job : Factor w/ 12 levels "admin.","blue-collar",..: 2 8 8 8 1 8 1 3 8 2 ...  
## $ marital : Factor w/ 4 levels "divorced","married",..: 2 3 2 2 2 3 3 2 1 2 ...  
## $ education : Factor w/ 8 levels "basic.4y","basic.6y",..: 3 4 4 3 7 7 7 7 6 3 ...  
## $ default : Factor w/ 3 levels "no","unknown",..: 1 1 1 1 1 1 1 2 1 2 ...  
## $ housing : Factor w/ 3 levels "no","unknown",..: 3 1 3 2 3 1 3 3 1 1 ...  
## $ loan : Factor w/ 3 levels "no","unknown",..: 1 1 1 2 1 1 1 1 1 1 ...  
## $ contact : Factor w/ 2 levels "cellular","telephone": 1 2 2 2 1 1 1 1 1 2 ...  
## $ month : Factor w/ 10 levels "apr","aug","dec",..: 7 7 5 5 8 10 10 8 8 7 ...  
## $ day\_of\_week : Factor w/ 5 levels "fri","mon","thu",..: 1 1 5 1 2 3 2 2 4 3 ...  
## $ duration : int 487 346 227 17 58 128 290 44 68 170 ...  
## $ campaign : int 2 4 1 3 1 3 4 2 1 1 ...  
## $ pdays : int 999 999 999 999 999 999 999 999 999 999 ...  
## $ previous : int 0 0 0 0 0 2 0 0 1 0 ...  
## $ poutcome : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 1 2 2 1 2 ...  
## $ emp.var.rate : num -1.8 1.1 1.4 1.4 -0.1 -1.1 -1.1 -0.1 -0.1 1.1 ...  
## $ cons.price.idx: num 92.9 94 94.5 94.5 93.2 ...  
## $ cons.conf.idx : num -46.2 -36.4 -41.8 -41.8 -42 -37.5 -37.5 -42 -42 -36.4 ...  
## $ euribor3m : num 1.31 4.86 4.96 4.96 4.19 ...  
## $ nr.employed : num 5099 5191 5228 5228 5196 ...  
## $ y : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...

summary(bank)

## age job marital education   
## Min. :18.00 admin. :1012 divorced: 446 university.degree :1264   
## 1st Qu.:32.00 blue-collar: 884 married :2509 high.school : 921   
## Median :38.00 technician : 691 single :1153 basic.9y : 574   
## Mean :40.11 services : 393 unknown : 11 professional.course: 535   
## 3rd Qu.:47.00 management : 324 basic.4y : 429   
## Max. :88.00 retired : 166 basic.6y : 228   
## (Other) : 649 (Other) : 168   
## default housing loan contact month   
## no :3315 no :1839 no :3349 cellular :2652 may :1378   
## unknown: 803 unknown: 105 unknown: 105 telephone:1467 jul : 711   
## yes : 1 yes :2175 yes : 665 aug : 636   
## jun : 530   
## nov : 446   
## apr : 215   
## (Other): 203   
## day\_of\_week duration campaign pdays previous   
## fri:768 Min. : 0.0 Min. : 1.000 Min. : 0.0 Min. :0.0000   
## mon:855 1st Qu.: 103.0 1st Qu.: 1.000 1st Qu.:999.0 1st Qu.:0.0000   
## thu:860 Median : 181.0 Median : 2.000 Median :999.0 Median :0.0000   
## tue:841 Mean : 256.8 Mean : 2.537 Mean :960.4 Mean :0.1903   
## wed:795 3rd Qu.: 317.0 3rd Qu.: 3.000 3rd Qu.:999.0 3rd Qu.:0.0000   
## Max. :3643.0 Max. :35.000 Max. :999.0 Max. :6.0000   
##   
## poutcome emp.var.rate cons.price.idx cons.conf.idx   
## failure : 454 Min. :-3.40000 Min. :92.20 Min. :-50.8   
## nonexistent:3523 1st Qu.:-1.80000 1st Qu.:93.08 1st Qu.:-42.7   
## success : 142 Median : 1.10000 Median :93.75 Median :-41.8   
## Mean : 0.08497 Mean :93.58 Mean :-40.5   
## 3rd Qu.: 1.40000 3rd Qu.:93.99 3rd Qu.:-36.4   
## Max. : 1.40000 Max. :94.77 Max. :-26.9   
##   
## euribor3m nr.employed y   
## Min. :0.635 Min. :4964 no :3668   
## 1st Qu.:1.334 1st Qu.:5099 yes: 451   
## Median :4.857 Median :5191   
## Mean :3.621 Mean :5166   
## 3rd Qu.:4.961 3rd Qu.:5228   
## Max. :5.045 Max. :5228   
##

sum(is.na(bank))

## [1] 0

#reading in 'Bank Additional Full' file  
bankfull = read.csv("./DataSets/bank-additional-full.csv",header = TRUE, sep = ";")  
str(bankfull)

## 'data.frame': 41188 obs. of 21 variables:  
## $ age : int 56 57 37 40 56 45 59 41 24 25 ...  
## $ job : Factor w/ 12 levels "admin.","blue-collar",..: 4 8 8 1 8 8 1 2 10 8 ...  
## $ marital : Factor w/ 4 levels "divorced","married",..: 2 2 2 2 2 2 2 2 3 3 ...  
## $ education : Factor w/ 8 levels "basic.4y","basic.6y",..: 1 4 4 2 4 3 6 8 6 4 ...  
## $ default : Factor w/ 3 levels "no","unknown",..: 1 2 1 1 1 2 1 2 1 1 ...  
## $ housing : Factor w/ 3 levels "no","unknown",..: 1 1 3 1 1 1 1 1 3 3 ...  
## $ loan : Factor w/ 3 levels "no","unknown",..: 1 1 1 1 3 1 1 1 1 1 ...  
## $ contact : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2 2 2 2 2 ...  
## $ month : Factor w/ 10 levels "apr","aug","dec",..: 7 7 7 7 7 7 7 7 7 7 ...  
## $ day\_of\_week : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ duration : int 261 149 226 151 307 198 139 217 380 50 ...  
## $ campaign : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ pdays : int 999 999 999 999 999 999 999 999 999 999 ...  
## $ previous : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ poutcome : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ emp.var.rate : num 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...  
## $ cons.price.idx: num 94 94 94 94 94 ...  
## $ cons.conf.idx : num -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 ...  
## $ euribor3m : num 4.86 4.86 4.86 4.86 4.86 ...  
## $ nr.employed : num 5191 5191 5191 5191 5191 ...  
## $ y : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...

summary(bankfull)

## age job marital   
## Min. :17.00 admin. :10422 divorced: 4612   
## 1st Qu.:32.00 blue-collar: 9254 married :24928   
## Median :38.00 technician : 6743 single :11568   
## Mean :40.02 services : 3969 unknown : 80   
## 3rd Qu.:47.00 management : 2924   
## Max. :98.00 retired : 1720   
## (Other) : 6156   
## education default housing loan   
## university.degree :12168 no :32588 no :18622 no :33950   
## high.school : 9515 unknown: 8597 unknown: 990 unknown: 990   
## basic.9y : 6045 yes : 3 yes :21576 yes : 6248   
## professional.course: 5243   
## basic.4y : 4176   
## basic.6y : 2292   
## (Other) : 1749   
## contact month day\_of\_week duration   
## cellular :26144 may :13769 fri:7827 Min. : 0.0   
## telephone:15044 jul : 7174 mon:8514 1st Qu.: 102.0   
## aug : 6178 thu:8623 Median : 180.0   
## jun : 5318 tue:8090 Mean : 258.3   
## nov : 4101 wed:8134 3rd Qu.: 319.0   
## apr : 2632 Max. :4918.0   
## (Other): 2016   
## campaign pdays previous poutcome   
## Min. : 1.000 Min. : 0.0 Min. :0.000 failure : 4252   
## 1st Qu.: 1.000 1st Qu.:999.0 1st Qu.:0.000 nonexistent:35563   
## Median : 2.000 Median :999.0 Median :0.000 success : 1373   
## Mean : 2.568 Mean :962.5 Mean :0.173   
## 3rd Qu.: 3.000 3rd Qu.:999.0 3rd Qu.:0.000   
## Max. :56.000 Max. :999.0 Max. :7.000   
##   
## emp.var.rate cons.price.idx cons.conf.idx euribor3m   
## Min. :-3.40000 Min. :92.20 Min. :-50.8 Min. :0.634   
## 1st Qu.:-1.80000 1st Qu.:93.08 1st Qu.:-42.7 1st Qu.:1.344   
## Median : 1.10000 Median :93.75 Median :-41.8 Median :4.857   
## Mean : 0.08189 Mean :93.58 Mean :-40.5 Mean :3.621   
## 3rd Qu.: 1.40000 3rd Qu.:93.99 3rd Qu.:-36.4 3rd Qu.:4.961   
## Max. : 1.40000 Max. :94.77 Max. :-26.9 Max. :5.045   
##   
## nr.employed y   
## Min. :4964 no :36548   
## 1st Qu.:5099 yes: 4640   
## Median :5191   
## Mean :5167   
## 3rd Qu.:5228   
## Max. :5228   
##

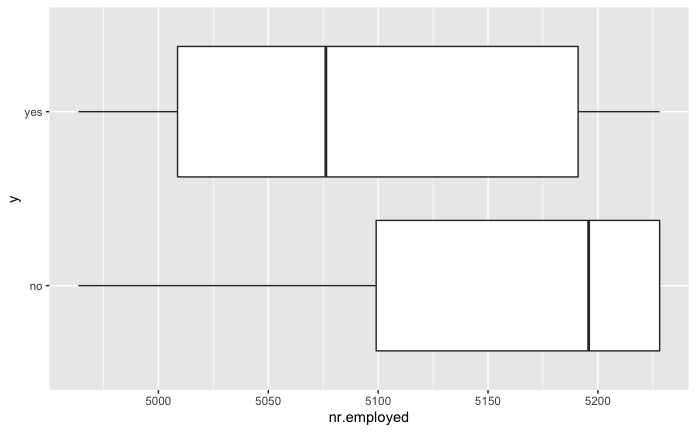
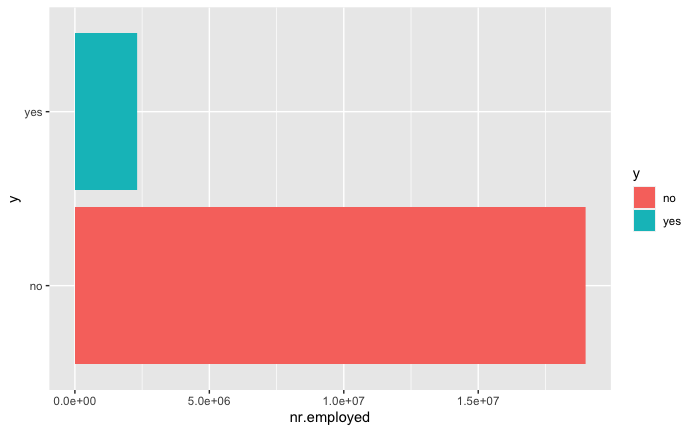
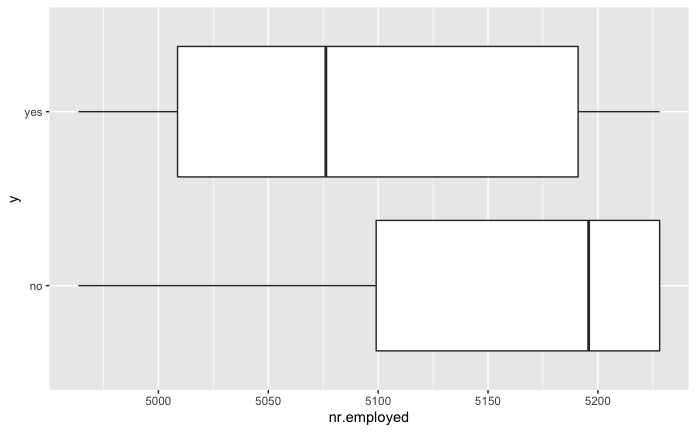
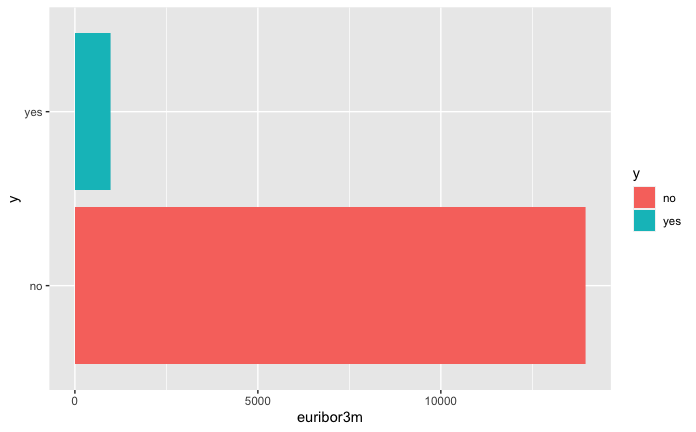
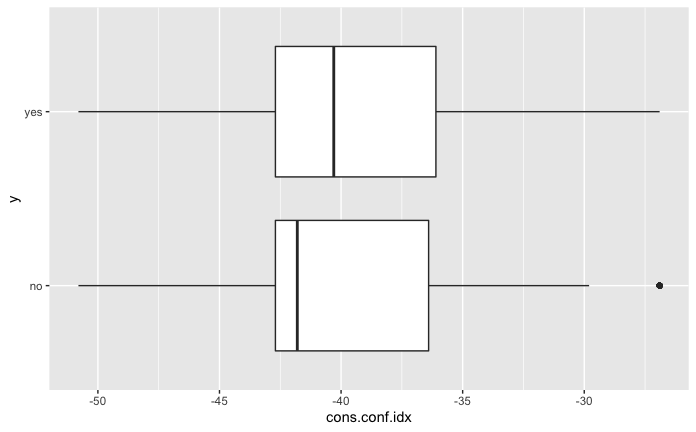
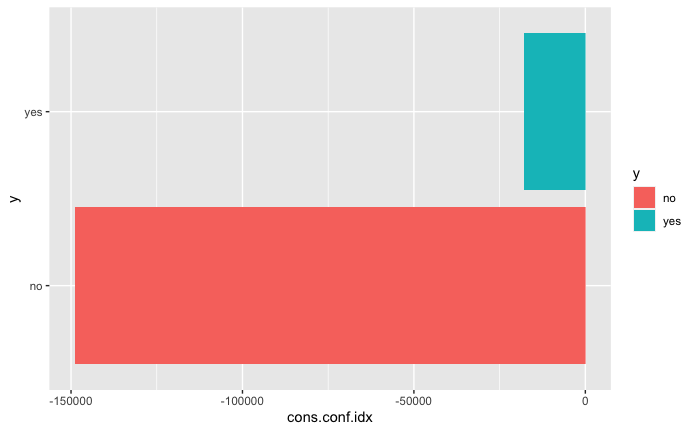
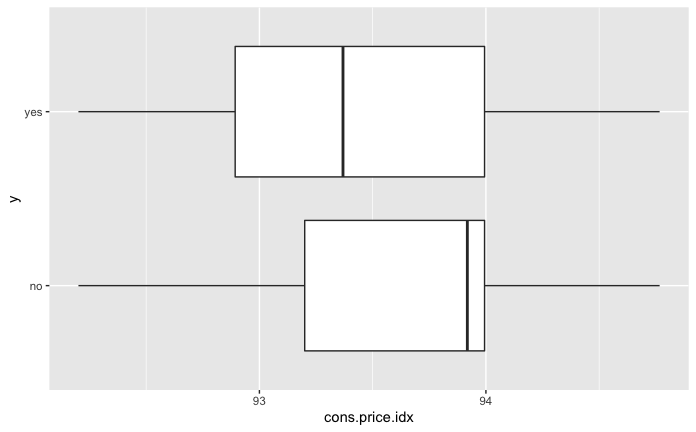
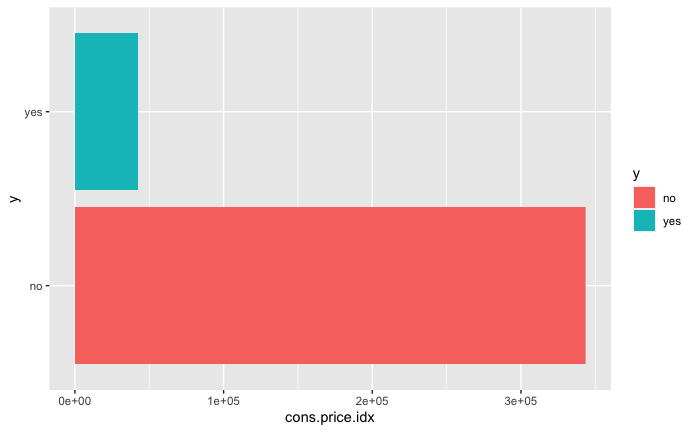
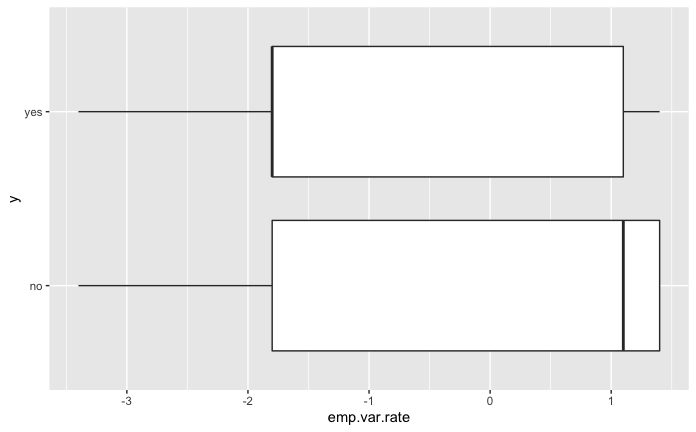
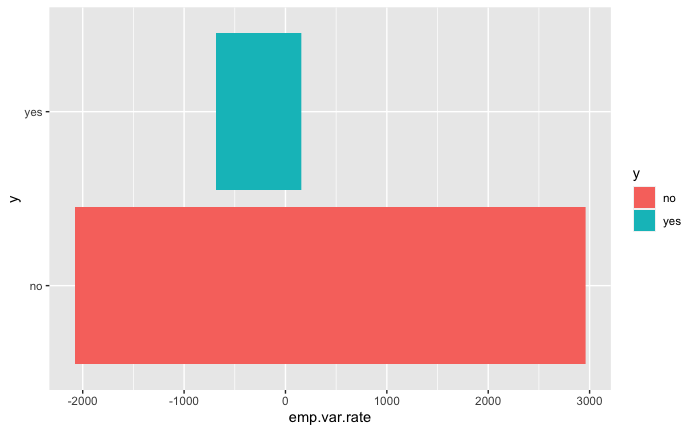
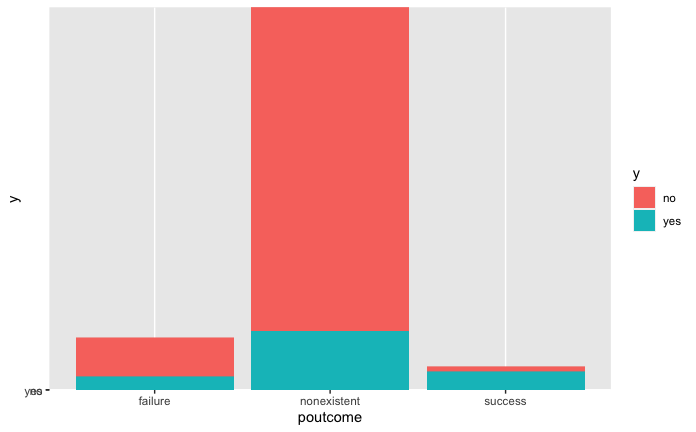
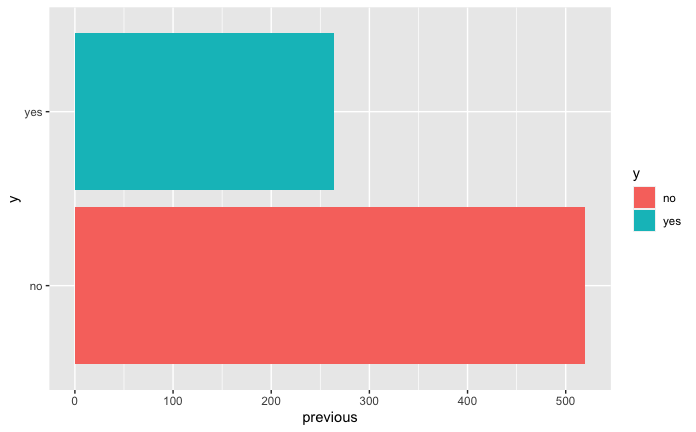
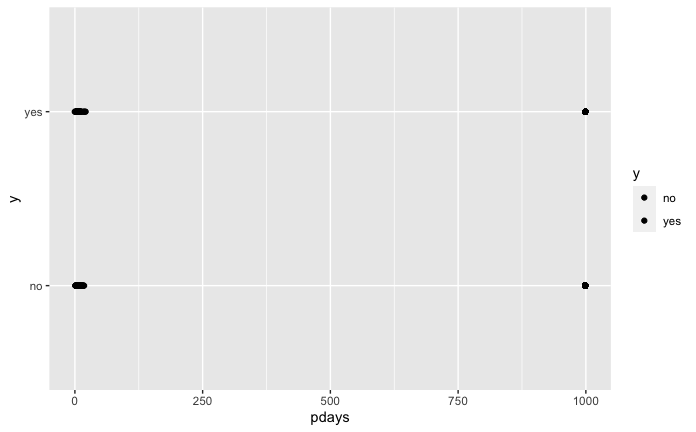
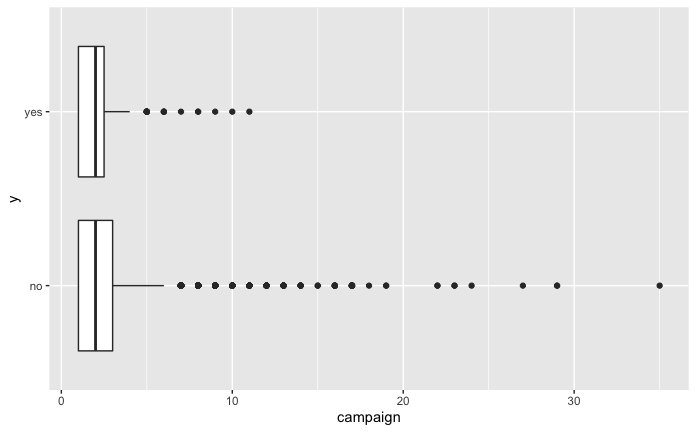
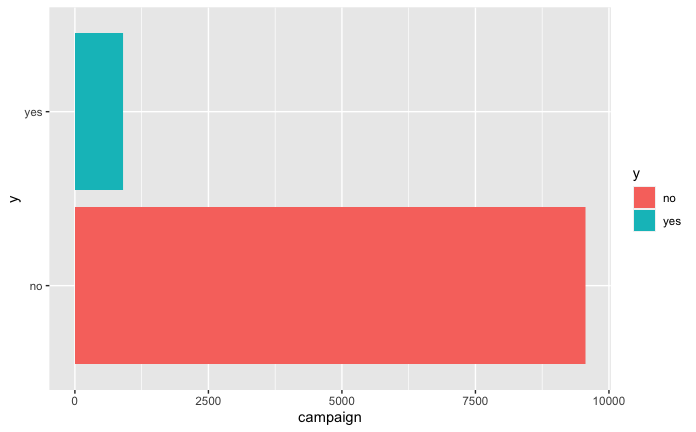
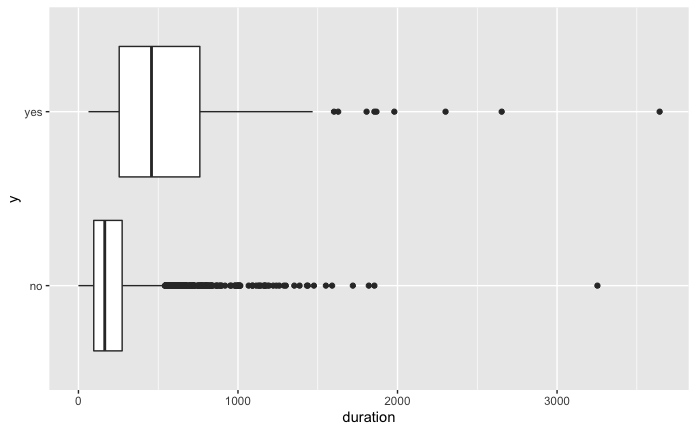
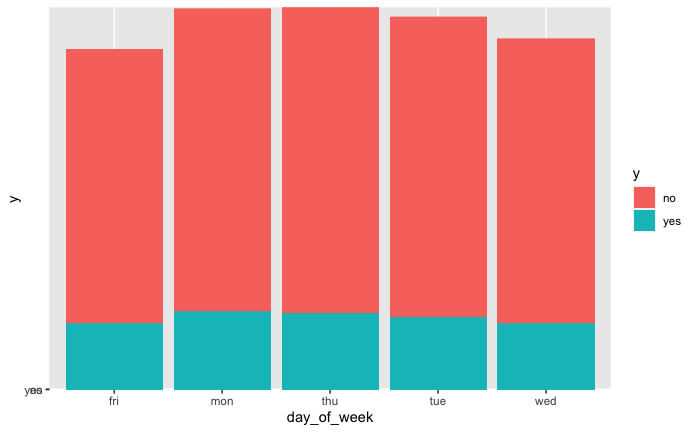
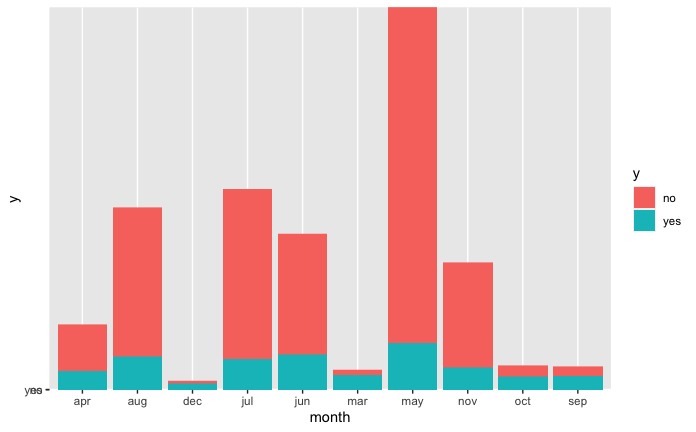
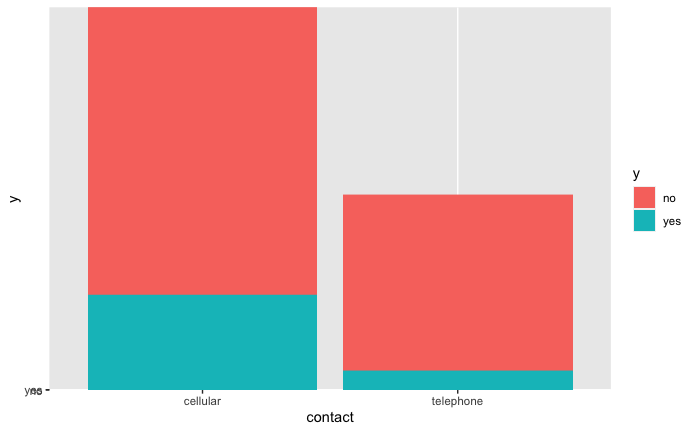
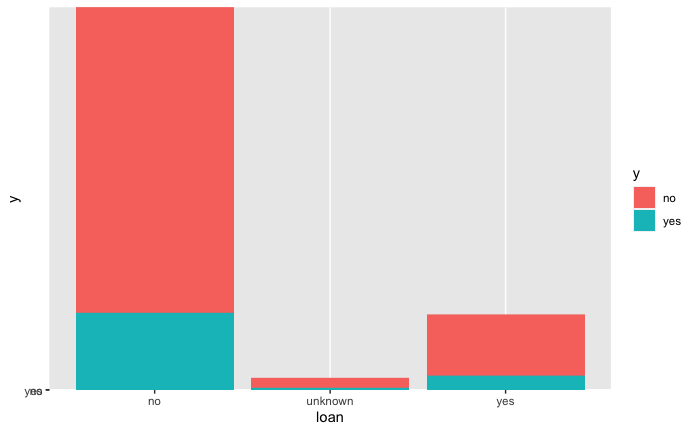
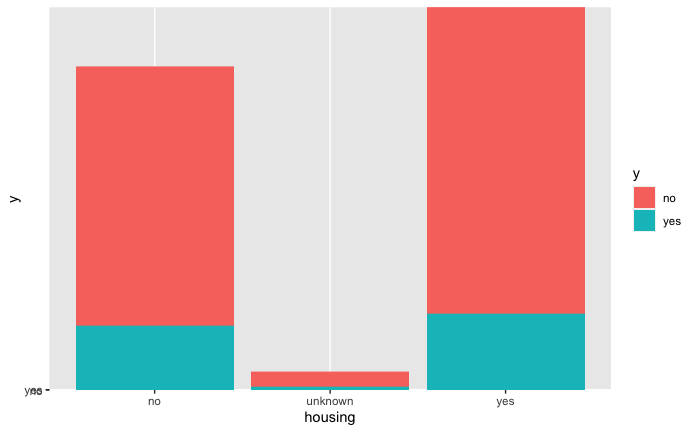
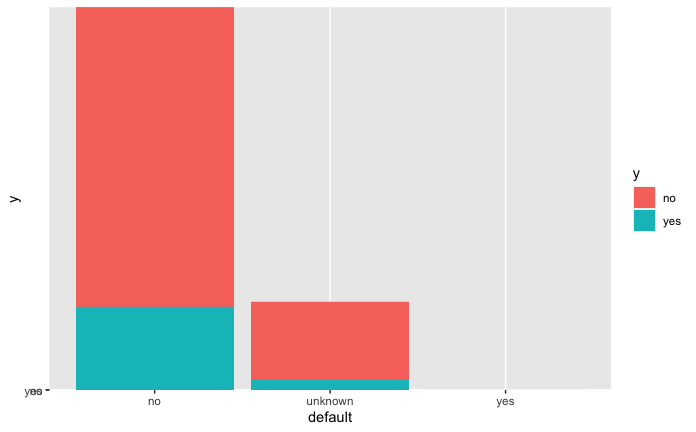
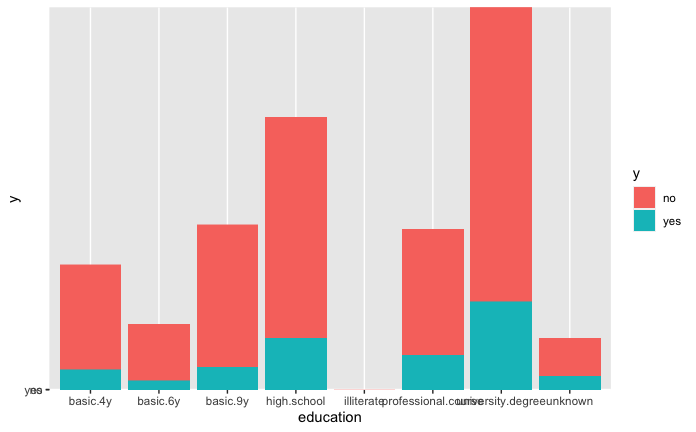
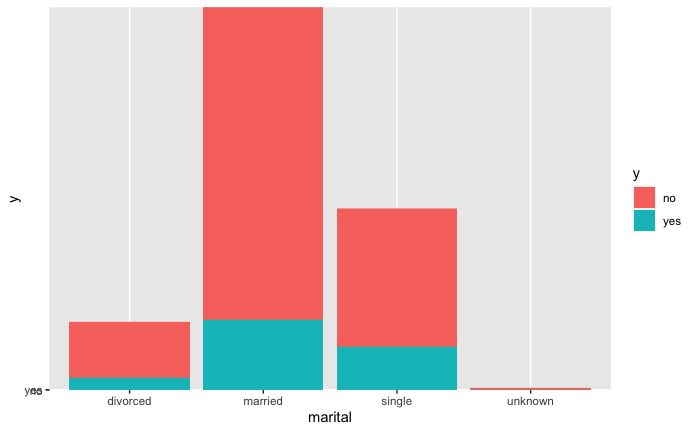
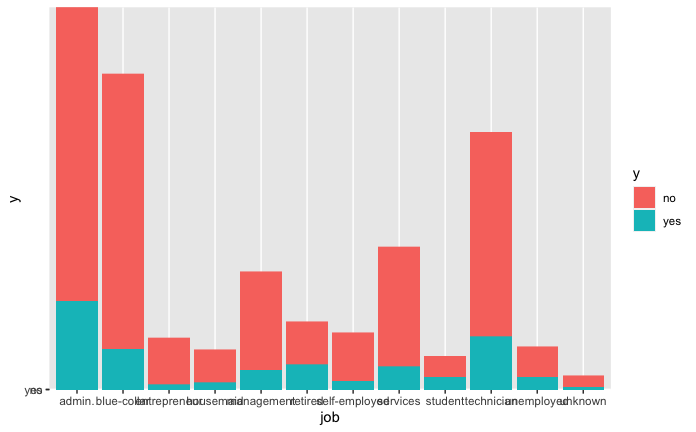
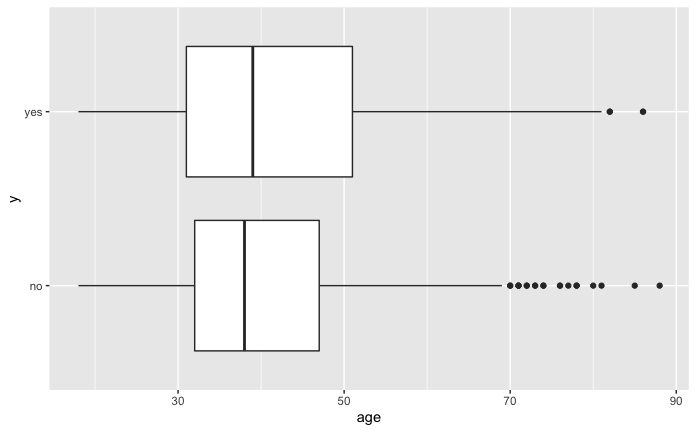
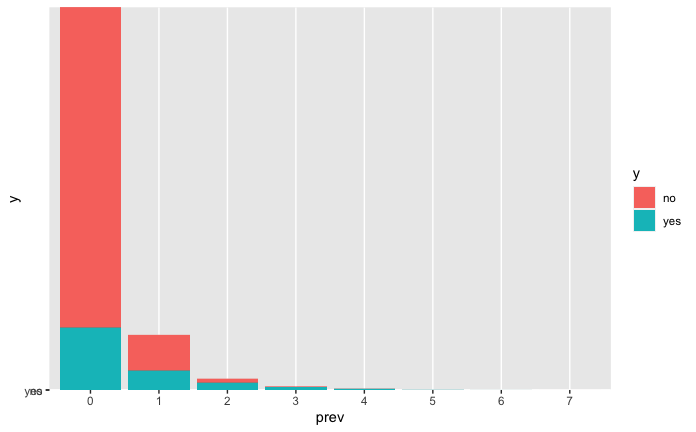
sum(is.na(bankfull))

## [1] 0

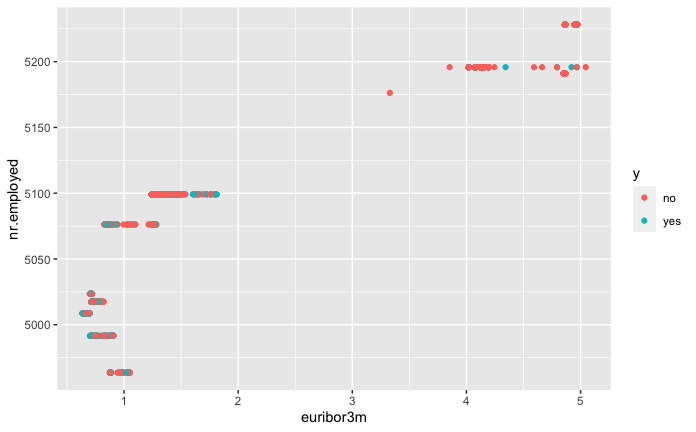
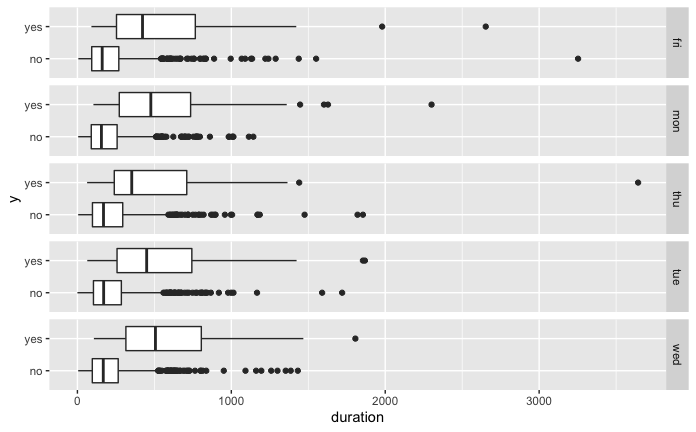
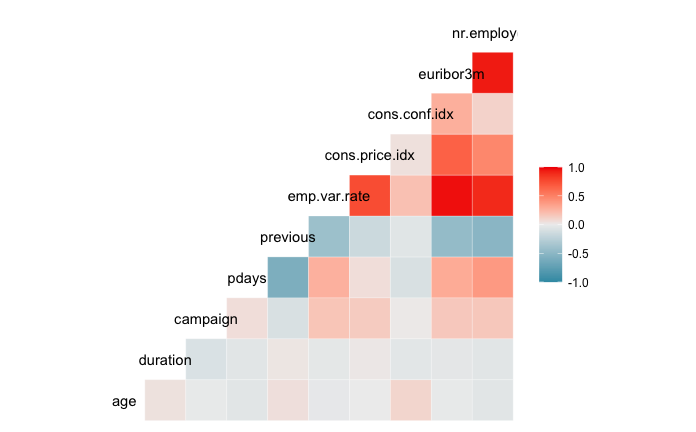
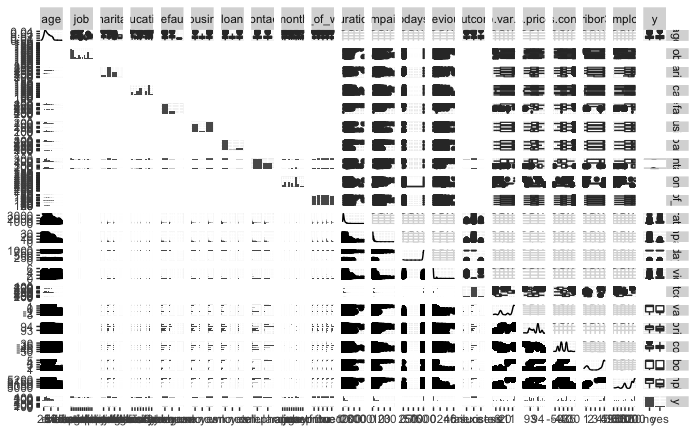
### Feature Engineering

#Create id to merge later  
#bankfull$id <- seq.int(nrow(bankfull))  
#bankfull$pdays <- ifelse(bankfull$pdays == 999, NA, bankfull$pdays)  
  
  
#Onehot encode categorical variables to binary:  
dmy <- dummyVars(" ~ .", data = bankfull)  
trsf <- data.frame(predict(dmy, newdata = bankfull))  
#bank2 <- inner\_join(bankfull, trsf, by = "id")  
  
#Remove binary encoded response  
trsf$y <- ifelse(trsf$y.no == 1, 0, 1)  
bankbin <- subset(trsf, select = -c(y.no, y.yes))

### Correlation Plot Code:



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



# Select only continuous variables:  
bankCont <- dplyr::select(bankfull, c('age','duration', 'campaign', 'pdays', 'cons.price.idx', 'cons.conf.idx', 'euribor3m', 'nr.employed','y'))  
bankCont$age <- as.numeric(bankCont$age)  
bankCont$duration <- as.numeric(bankCont$duration)  
bankCont$campaign <- as.numeric(bankCont$campaign)  
bankCont$pdays <- as.numeric(bankCont$pdays)  
  
  
mylda<-lda(y~age+duration+ campaign+ pdays+ cons.price.idx+ cons.conf.idx+ euribor3m+ nr.employed, data=bankCont)  
pred<-predict(mylda,newdata=bankCont)$class #Predictions can come in many forms, the class form provides the categorical level of your response.  
Truth<-bankCont$y  
x<-table(pred,Truth) # Creating a confusion matrix  
x

## Truth  
## pred no yes  
## no 35292 2582  
## yes 1256 2058

#Missclassification Error  
ME<-(x[2,1]+x[1,2])/1000  
ME

## [1] 3.838

mylda<-lda(y~., data=bankbin)

## Warning in lda.default(x, grouping, ...): variables are collinear

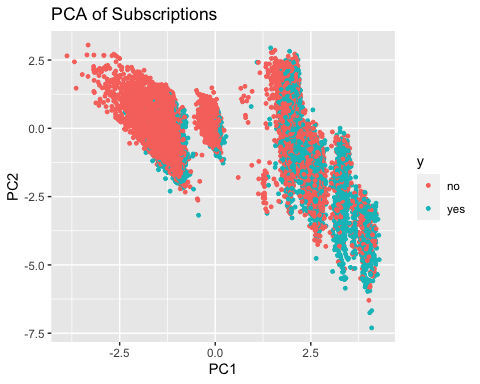
pred<-predict(mylda,newdata=bankbin)$class #Predictions can come in many forms, the class form provides the categorical level of your response.  
Truth<-bankbin$y  
x<-table(pred,Truth) # Creating a confusion matrix  
x

## Truth  
## pred 0 1  
## 0 35111 2264  
## 1 1437 2376

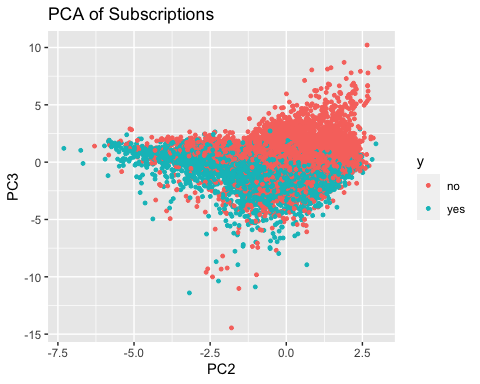
#Missclassification Error  
ME<-(x[2,1]+x[1,2])/1000  
ME

## [1] 3.701

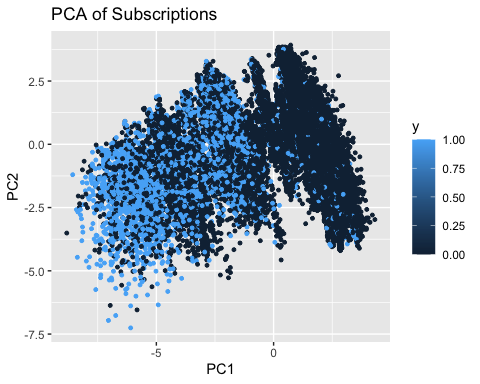
pc.bankCont<-prcomp(bankCont[,-9],scale.=TRUE)  
pc.bankCont.scores<-pc.bankCont$x  
  
#Adding the response column to the PC's data frame  
pc.bankCont.scores<-data.frame(pc.bankCont.scores)  
pc.bankCont.scores$y<-bankCont$y  
  
#Use ggplot2 to plot the first few pc's  
ggplot(data = pc.bankCont.scores, aes(x = PC1, y = PC2)) +  
 geom\_point(aes(col=y), size=1)+  
 ggtitle("PCA of Subscriptions")



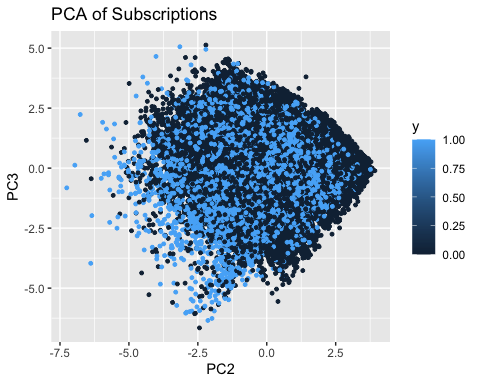
ggplot(data = pc.bankCont.scores, aes(x = PC2, y = PC3)) +  
 geom\_point(aes(col=y), size=1)+  
 ggtitle("PCA of Subscriptions")



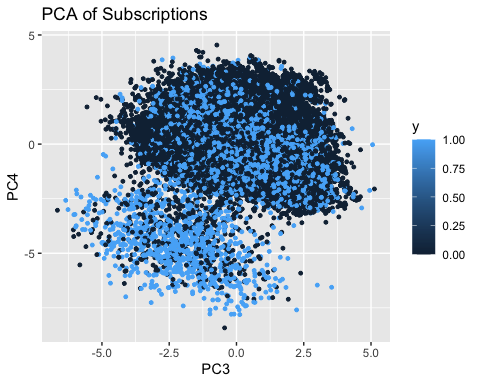
pc.bankbin<-prcomp(bankbin[,-64],scale.=TRUE)  
pc.bankbin.scores<-pc.bankbin$x  
  
#Adding the response column to the PC's data frame  
pc.bankbin.scores<-data.frame(pc.bankbin.scores)  
pc.bankbin.scores$y<-bankbin$y  
  
#Use ggplot2 to plot the first few pc's  
ggplot(data = pc.bankbin.scores, aes(x = PC1, y = PC2)) +  
 geom\_point(aes(col=y), size=1)+  
 ggtitle("PCA of Subscriptions")



ggplot(data = pc.bankbin.scores, aes(x = PC2, y = PC3)) +  
 geom\_point(aes(col=y), size=1)+  
 ggtitle("PCA of Subscriptions")



ggplot(data = pc.bankbin.scores, aes(x = PC3, y = PC4)) +  
 geom\_point(aes(col=y), size=1)+  
 ggtitle("PCA of Subscriptions")



# This chunk need a label. Is this random forest?? What is algo name?  
  
#########################################################  
# Sampling the dataset into training data and test data:  
#############################################################  
  
set.seed(1234)  
splitPerc = .75  
index = sample(1:dim(bankCont)[1],round(splitPerc \* dim(bankCont)[1]))  
#index<-sample(1:dim(Auto2)[1],192,replace=F)  
traindata<-bankCont[index,]  
testdata<-bankCont[-index,]  
  
   
  
  
# Classification and Regression Trees  
bank.cart<-rpart(y ~ ., traindata , method = 'class')  
  
#par(mfrow=c(1,1))  
#fancyRpartPlot(bank.cart , digits=2 , palettes = c("Purples", "Oranges"))  
  
#predict  
cart\_pred <- predict( bank.cart , testdata , type = "class")  
cart\_prob <- predict( bank.cart , testdata , type = "prob")  
  
# Confusion matrix  
confusionMatrix(cart\_pred , testdata$y)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 8801 544  
## yes 355 597  
##   
## Accuracy : 0.9127   
## 95% CI : (0.9071, 0.9181)  
## No Information Rate : 0.8892   
## P-Value [Acc > NIR] : 2.321e-15   
##   
## Kappa : 0.5223   
##   
## Mcnemar's Test P-Value : 3.607e-10   
##   
## Sensitivity : 0.9612   
## Specificity : 0.5232   
## Pos Pred Value : 0.9418   
## Neg Pred Value : 0.6271   
## Prevalence : 0.8892   
## Detection Rate : 0.8547   
## Detection Prevalence : 0.9075   
## Balanced Accuracy : 0.7422   
##   
## 'Positive' Class : no   
##

### Cross table validation for CART  
CrossTable(testdata$y, cart\_pred,  
 prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE,  
 dnn = c('actual default', 'predicted default'))

## Cell Contents   
## |-------------------------|  
## | N |   
## | N / Table Total |   
## |-------------------------|  
##   
## =======================================  
## predicted default  
## actual default no yes Total  
## ---------------------------------------  
## no 8801 355 9156  
## 0.855 0.034   
## ---------------------------------------  
## yes 544 597 1141  
## 0.053 0.058   
## ---------------------------------------  
## Total 9345 952 10297  
## =======================================

#Implementing KNN  
###########################################  
bank.knn <- train(y ~ ., data = traindata, method = "knn",   
 maximize = TRUE,  
 trControl = trainControl(method = "cv", number = 10),  
 preProcess=c("center", "scale"))  
  
predictedkNN <- predict(bank.knn , newdata = testdata)  
confusionMatrix(predictedkNN , testdata$y)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction no yes  
## no 8785 583  
## yes 371 558  
##   
## Accuracy : 0.9074   
## 95% CI : (0.9016, 0.9129)  
## No Information Rate : 0.8892   
## P-Value [Acc > NIR] : 9.774e-10   
##   
## Kappa : 0.4882   
##   
## Mcnemar's Test P-Value : 8.410e-12   
##   
## Sensitivity : 0.9595   
## Specificity : 0.4890   
## Pos Pred Value : 0.9378   
## Neg Pred Value : 0.6006   
## Prevalence : 0.8892   
## Detection Rate : 0.8532   
## Detection Prevalence : 0.9098   
## Balanced Accuracy : 0.7243   
##   
## 'Positive' Class : no   
##

### Cross table validation for KNN  
CrossTable(testdata$y, predictedkNN,  
 prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE,  
 dnn = c('actual default', 'predicted default'))

## Cell Contents   
## |-------------------------|  
## | N |   
## | N / Table Total |   
## |-------------------------|  
##   
## =======================================  
## predicted default  
## actual default no yes Total  
## ---------------------------------------  
## no 8785 371 9156  
## 0.853 0.036   
## ---------------------------------------  
## yes 583 558 1141  
## 0.057 0.054   
## ---------------------------------------  
## Total 9368 929 10297  
## =======================================

# Build feature list:  
x<-colnames(bankfull)  
x<-x[x != "y"]  
x<-paste(x, collapse='+')  
x # copy this printed value into the model

## [1] "age+job+marital+education+default+housing+loan+contact+month+day\_of\_week+duration+campaign+pdays+previous+poutcome+emp.var.rate+cons.price.idx+cons.conf.idx+euribor3m+nr.employed"

rm(x)  
  
# Build model:  
mylogit <- glm(y ~ age+job+marital+education+default+housing+loan+contact+month+day\_of\_week+duration+campaign+pdays+previous+poutcome+emp.var.rate+cons.price.idx+cons.conf.idx+euribor3m+nr.employed  
 ,data = bankfull, family = "binomial")  
  
# Summary of everything model:  
summary(mylogit)

##   
## Call:  
## glm(formula = y ~ age + job + marital + education + default +   
## housing + loan + contact + month + day\_of\_week + duration +   
## campaign + pdays + previous + poutcome + emp.var.rate + cons.price.idx +   
## cons.conf.idx + euribor3m + nr.employed, family = "binomial",   
## data = bankfull)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -6.0022 -0.2984 -0.1855 -0.1344 3.3804   
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.366e+02 3.831e+01 -6.176 6.56e-10 \*\*\*  
## age 1.966e-04 2.434e-03 0.081 0.935624   
## jobblue-collar -2.347e-01 7.988e-02 -2.939 0.003295 \*\*   
## jobentrepreneur -1.780e-01 1.260e-01 -1.413 0.157566   
## jobhousemaid -2.432e-02 1.478e-01 -0.165 0.869320   
## jobmanagement -5.614e-02 8.536e-02 -0.658 0.510710   
## jobretired 2.858e-01 1.071e-01 2.669 0.007606 \*\*   
## jobself-employed -1.578e-01 1.178e-01 -1.340 0.180396   
## jobservices -1.399e-01 8.610e-02 -1.624 0.104286   
## jobstudent 2.034e-01 1.115e-01 1.823 0.068230 .   
## jobtechnician -1.401e-02 7.113e-02 -0.197 0.843838   
## jobunemployed 2.116e-02 1.279e-01 0.165 0.868636   
## jobunknown -7.049e-02 2.386e-01 -0.295 0.767646   
## maritalmarried -2.734e-03 6.854e-02 -0.040 0.968188   
## maritalsingle 5.591e-02 7.826e-02 0.714 0.474969   
## maritalunknown 2.949e-02 4.167e-01 0.071 0.943587   
## educationbasic.6y 1.226e-01 1.206e-01 1.017 0.309241   
## educationbasic.9y -9.772e-04 9.535e-02 -0.010 0.991823   
## educationhigh.school 4.861e-02 9.210e-02 0.528 0.597676   
## educationilliterate 1.067e+00 7.550e-01 1.413 0.157706   
## educationprofessional.course 1.150e-01 1.015e-01 1.133 0.257261   
## educationuniversity.degree 1.958e-01 9.220e-02 2.124 0.033667 \*   
## educationunknown 1.497e-01 1.195e-01 1.253 0.210204   
## defaultunknown -3.003e-01 6.740e-02 -4.456 8.36e-06 \*\*\*  
## defaultyes -7.298e+00 1.134e+02 -0.064 0.948704   
## housingunknown -9.331e-02 1.398e-01 -0.668 0.504354   
## housingyes -4.731e-03 4.135e-02 -0.114 0.908909   
## loanunknown NA NA NA NA   
## loanyes -5.160e-02 5.747e-02 -0.898 0.369227   
## contacttelephone -6.460e-01 7.688e-02 -8.402 < 2e-16 \*\*\*  
## monthaug 8.653e-01 1.206e-01 7.174 7.26e-13 \*\*\*  
## monthdec 3.192e-01 2.093e-01 1.525 0.127236   
## monthjul 1.346e-01 9.630e-02 1.398 0.162205   
## monthjun -5.243e-01 1.262e-01 -4.154 3.27e-05 \*\*\*  
## monthmar 2.014e+00 1.444e-01 13.949 < 2e-16 \*\*\*  
## monthmay -4.439e-01 8.261e-02 -5.373 7.74e-08 \*\*\*  
## monthnov -4.180e-01 1.210e-01 -3.454 0.000552 \*\*\*  
## monthoct 1.940e-01 1.538e-01 1.262 0.207028   
## monthsep 3.740e-01 1.795e-01 2.083 0.037226 \*   
## day\_of\_weekmon -1.168e-01 6.613e-02 -1.767 0.077304 .   
## day\_of\_weekthu 5.600e-02 6.409e-02 0.874 0.382274   
## day\_of\_weektue 9.719e-02 6.587e-02 1.476 0.140067   
## day\_of\_weekwed 1.753e-01 6.567e-02 2.669 0.007603 \*\*   
## duration 4.706e-03 7.457e-05 63.108 < 2e-16 \*\*\*  
## campaign -4.015e-02 1.156e-02 -3.473 0.000514 \*\*\*  
## pdays -9.388e-04 2.170e-04 -4.326 1.52e-05 \*\*\*  
## previous -6.277e-02 5.912e-02 -1.062 0.288383   
## poutcomenonexistent 4.258e-01 9.423e-02 4.519 6.23e-06 \*\*\*  
## poutcomesuccess 9.597e-01 2.115e-01 4.538 5.67e-06 \*\*\*  
## emp.var.rate -1.758e+00 1.420e-01 -12.380 < 2e-16 \*\*\*  
## cons.price.idx 2.190e+00 2.524e-01 8.679 < 2e-16 \*\*\*  
## cons.conf.idx 2.069e-02 7.768e-03 2.664 0.007733 \*\*   
## euribor3m 3.316e-01 1.300e-01 2.551 0.010737 \*   
## nr.employed 5.413e-03 3.115e-03 1.738 0.082275 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 28999 on 41187 degrees of freedom  
## Residual deviance: 17078 on 41135 degrees of freedom  
## AIC: 17184  
##   
## Number of Fisher Scoring iterations: 10

# Recommend we drop: marital, age, loan, previous, housing, education  
mylogit2 <- glm(y ~ job+default+contact+month+day\_of\_week+duration+campaign+pdays+poutcome+emp.var.rate+cons.price.idx+cons.conf.idx+euribor3m+nr.employed  
 ,data = bankfull, family = "binomial")  
summary(mylogit2)

##   
## Call:  
## glm(formula = y ~ job + default + contact + month + day\_of\_week +   
## duration + campaign + pdays + poutcome + emp.var.rate + cons.price.idx +   
## cons.conf.idx + euribor3m + nr.employed, family = "binomial",   
## data = bankfull)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -5.9947 -0.2991 -0.1855 -0.1348 3.3485   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.323e+02 3.822e+01 -6.078 1.22e-09 \*\*\*  
## jobblue-collar -3.329e-01 6.586e-02 -5.055 4.31e-07 \*\*\*  
## jobentrepreneur -2.029e-01 1.244e-01 -1.631 0.102947   
## jobhousemaid -1.118e-01 1.409e-01 -0.793 0.427695   
## jobmanagement -4.317e-02 8.341e-02 -0.518 0.604744   
## jobretired 2.047e-01 8.381e-02 2.442 0.014610 \*   
## jobself-employed -1.509e-01 1.169e-01 -1.291 0.196739   
## jobservices -2.128e-01 8.168e-02 -2.605 0.009185 \*\*   
## jobstudent 1.770e-01 1.018e-01 1.739 0.081968 .   
## jobtechnician -2.728e-02 6.348e-02 -0.430 0.667323   
## jobunemployed -3.686e-02 1.261e-01 -0.292 0.769998   
## jobunknown -9.286e-02 2.344e-01 -0.396 0.691981   
## defaultunknown -3.106e-01 6.636e-02 -4.681 2.86e-06 \*\*\*  
## defaultyes -7.326e+00 1.134e+02 -0.065 0.948510   
## contacttelephone -6.421e-01 7.674e-02 -8.368 < 2e-16 \*\*\*  
## monthaug 8.674e-01 1.202e-01 7.217 5.32e-13 \*\*\*  
## monthdec 3.019e-01 2.088e-01 1.446 0.148150   
## monthjul 1.361e-01 9.598e-02 1.418 0.156227   
## monthjun -5.115e-01 1.258e-01 -4.068 4.75e-05 \*\*\*  
## monthmar 2.019e+00 1.441e-01 14.007 < 2e-16 \*\*\*  
## monthmay -4.553e-01 8.233e-02 -5.530 3.20e-08 \*\*\*  
## monthnov -4.253e-01 1.208e-01 -3.522 0.000429 \*\*\*  
## monthoct 1.803e-01 1.535e-01 1.175 0.240163   
## monthsep 3.607e-01 1.793e-01 2.012 0.044220 \*   
## day\_of\_weekmon -1.172e-01 6.604e-02 -1.775 0.075831 .   
## day\_of\_weekthu 5.858e-02 6.401e-02 0.915 0.360104   
## day\_of\_weektue 9.500e-02 6.575e-02 1.445 0.148527   
## day\_of\_weekwed 1.727e-01 6.562e-02 2.632 0.008488 \*\*   
## duration 4.702e-03 7.450e-05 63.116 < 2e-16 \*\*\*  
## campaign -3.981e-02 1.155e-02 -3.448 0.000564 \*\*\*  
## pdays -8.445e-04 2.036e-04 -4.149 3.34e-05 \*\*\*  
## poutcomenonexistent 5.026e-01 6.411e-02 7.840 4.52e-15 \*\*\*  
## poutcomesuccess 1.036e+00 2.040e-01 5.081 3.76e-07 \*\*\*  
## emp.var.rate -1.752e+00 1.419e-01 -12.350 < 2e-16 \*\*\*  
## cons.price.idx 2.160e+00 2.516e-01 8.586 < 2e-16 \*\*\*  
## cons.conf.idx 2.039e-02 7.734e-03 2.637 0.008377 \*\*   
## euribor3m 3.422e-01 1.297e-01 2.638 0.008333 \*\*   
## nr.employed 5.116e-03 3.108e-03 1.646 0.099749 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 28999 on 41187 degrees of freedom  
## Residual deviance: 17094 on 41150 degrees of freedom  
## AIC: 17170  
##   
## Number of Fisher Scoring iterations: 10

# Rmd Examples

Suggested Downtime Activities:

1. Data cleaning (handle missing?, train/test split logistics: you want a good balance of Y/N in your train)
2. summary statistics (the ones I kept complaining about on Project 1)
3. EDA (basic boxplots, scatterplots, bar charts etc to see how things relate to the response, PCA tool),
4. Maybe play around and fit a few LDA models with continuous stuff.

### Centered Img

##### caption

Note this is just a syntax example. Since we don’t have a local picture to use, this will show nothing in the word doc. Also worth noting is the H5 marking, which centers the image.

#### Latex examples