# Harvard Data Science Capstone: CYO - Bank Marketing Data Set

Jason Baird

May 2, 2021

## Contents

```
## Loading required package: tidyverse

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

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

## Loading required package: caret

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

## Loading required package: lattice

##
## Attaching package: 'caret'

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

## Loading required package: corrplot

## Warning: package 'corrplot' was built under R version 4.0.5

## corrplot 0.84 loaded

## Loading required package: pROC

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

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

##
## Attaching package: 'pROC'

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

## Loading required package: class

## Loading required package: 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:dplyr':
##
##     combine

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

# 1 Executive Summary

The purpose of this project is to generate a classification algorithm that identifies whether or not specific bank clients will subscribe to a term deposit. The bank marketing data set contains slightly less than 42,000 observations and 21 features spanning demographic client information, marketing campaign specific data, socioeconomic data, and a few additional data points on the client. As is the case with many marketing campaigns, the success rate of subscribing to a term deposit is much lower than 50%. In fact, the average subscription rate among this population was 11.2%.

Due to the disparity between subscribing and non-subscribing clients, I did not want to use accuracy as the primary metric gauging model performance. Instead, I gauged each model's performance by the Area Under the ROC Curve (AUC) which evaluates sensitivity and specificity pairs.

In short, I ran the classification analysis on three models: 1) logistic regression, 2) KNN Clustering algorithm, and 3) Random Forest model. **The logistic regression model** was the most effective model for classifying whether a client would subscribe or not to a term deposit. **The AUC for the logistic regression was 92.4%**.

# 2 Project Outline

## 2.1 Objective

The bank marketing data set presents a classification analysis where the goal is to predict whether a client will subscribe a term deposit. The data provided is from a marketing campaign launched by the bank where they called various clients to promote their term deposit offering. In total, the data set contains 20 explanatory variables, an output variable, and 41,188 client observations.

The goal of this project is to uncover the explanatory variables that best predict whether a client will subscribe a term deposit, engineer those features so they are best suited for a machine learning algorithm, and then select a model that best predicts whether the client will or will not subscribe a term deposit.

## 2.2 Key Metrics: Area Under Curve (AUC)

In order to attain the objective for this project, I will be using Area Under Curve to determine model selection.

***Possible Metrics for Classification Analysis***:

**Accuracy** = (TP + TN)/(TP + FP + FN + TN) – this metric refers to the ratio of correctly predicted records as compared to the total number of records

**Precision** = TP/(TP + FP) – this metric refers to the ratio between the correct number of positive predictions and the total number of positive labels.

**Recall (AKA Sensitivity)** – TP/(TP + FN) - this metric refers to the ratio between the correct number of positive predictions and the sum of correct positive predictions plus incorrect positive predictions.

**Specificity** – TN/(TN + FP) – this metric refers to the ratio between correctly labeled negative values and the sum of correctly labeled negative values plus incorrectly labeled negative values

**Area Under Curve (AUC)** – line chart where the x-axis refers to the false positive rate (1 – Specificity) and y-axis representing the true positive rate (AKA Sensitivity). The AUC is an effective metric that highlights the efficacy of a classification algorithm when it is important to balance between correctly and incorrectly classifying the target variable.

## 2.3 A Quick Look at our Output Variable

Since marketing campaigns are often considered extremely successful when target goals are achieved among 10 – 20% of the audience, my assumption is that the bank marketing data set's target outcome (Term Deposit Subscription) will be disproportionate.

Prior to splitting our data, it is important to see if there is a disparity between the outcomes of our predictor variable.

```
##
##    no   yes
## 36548  4640
```

As you can see from the data above, slightly more than 11% of the bank's clients actually subscribed to the term deposit. This means that approximately 89% of their client's did not subscribe. Since the goal of the marketing campaign algorithm is to identify clients that WILL subscribe to the bank's term deposit but also help the marketing team prioritize clients that have a higher probability of success, it is essential that our primary metric for evaluating our algorithm focuses on both the true positive rate but not at the expense of a high false positive rate.

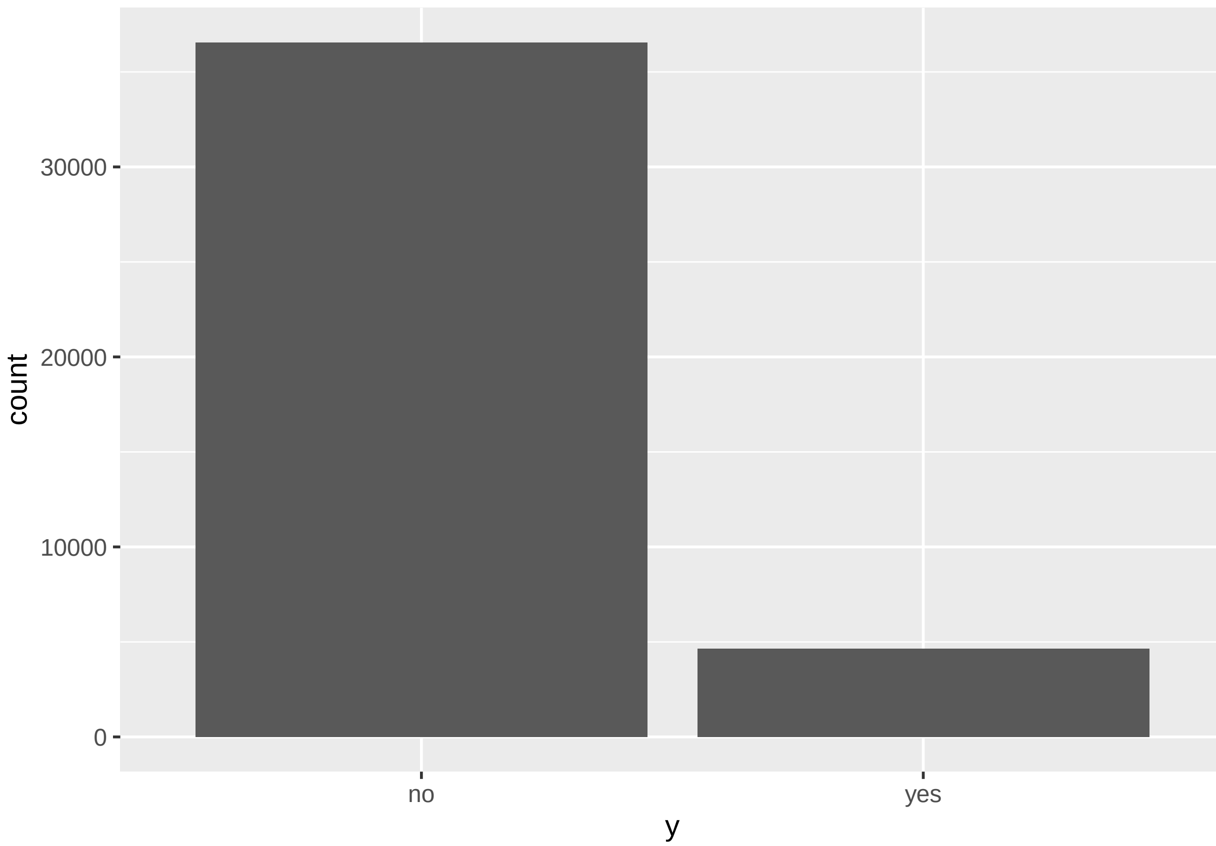



Figure 1: Term Deposit Subscription Counts

## 2.4 Model Performance & Selection:

Due to the disparity described above, accuracy is not the best measure for our algorithm. For example, **A Naïve model** using accuracy would simply classify every client as not subscribing to a term deposit and maintain an accuracy of approximately 89%.

Since the **Area Under the Curve (AUC)** provides a single metric to evaluate each model that balances Sensitivity and Specificity, it is the ideal metric to be used for this classification problem.

### 2.4.1 Types of Models to be Tested

I plan on evaluating three algorithms for this project.

1) Logistic Regression
2) K Nearest Neighbors
3) Random Forest

Each will be graded on the AUC on the test set.

# 3 Introducing the Data

The Bank Marketing Data Set from the UCI Machine Learning Repository is a popular data set related with the efficacy of direct marketing campaigns on bank consumers. In short, the goal of the direct marketing campaign was to contact bank clients in hopes that they would open a new term deposit.

## 3.1 Target Variable

Our output variable is a binary categorical variable. It specifies whether a client has subscribed to a term deposit or not.

21 - y - has the client subscribed a term deposit? (binary: 'yes','no')

## 3.2 Explanatory Variables

Since we are working with marketing data, it is important to understand how our twenty explanatory variables can be grouped into different types of relevancy to our marketing campaign dataset:

### 3.2.1 Audience Demographic Data

Every marketing campaign deals with audience segmentation. Audience segmentation refers to the different demographics groups of an audience and how those groups are affected by the marketing campaign. It is in this component that the bank will determine which members of their client base should receive the marketing campaign's message. Here are some of the relevant audience demographics from the data set.

1 - age (numeric)
2 - job: type of job (categorical: 'admin.', 'blue-collar', 'entrepreneur', 'housemaid', 'management', 'retired', 'self-employed', 'services', 'student', 'technician', 'unemployed', 'unknown')
3 - marital: marital status (categorical: 'divorced', 'married', 'single', 'unknown'; note: 'divorced' means divorced or widowed)
4 - education (categorical: 'basic.4y', 'basic.6y', 'basic.9y', 'high.school', 'illiterate', 'professional.course', 'university.degree', 'unknown')

Additionally, the bank has unique information on its clients which it leverages for this marketing campaign. By adding the client-specific data to the audience demographics, the bank can further segment its audience.

5 - default: has credit in default? (categorical: 'no', 'yes','unknown')
6 - housing: has housing loan? (categorical: 'no', 'yes', 'unknown')
7 - loan: has personal loan? (categorical: 'no', 'yes', 'unknown')
### Campaign-Specific Features

In addition to the demographic and client data, the bank has recorded various metrics on their marketing campaign. This following variables provided in the data set reference the campaign-specific features.

8 - contact: contact communication type (categorical: 'cellular', 'telephone')
9 - month: last contact month of year (categorical: 'jan', 'feb', 'mar', . . . , 'nov', 'dec')
10 - day_of_week: last contact day of the week (categorical: 'mon','tue','wed','thu','fri')
11 - duration: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.
### Additional attributes

The following additional attributes are associated with the data set.

12 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
13 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
14 - previous: number of contacts performed before this campaign and for this client (numeric)
15 - poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')
### Social and Economic Context Variables

16 - emp.var.rate: employment variation rate - quarterly indicator (numeric)
17 - cons.price.idx: consumer price index - monthly indicator (numeric)
18 - cons.conf.idx: consumer confidence index - monthly indicator (numeric)
19 - euribor3m: euribor 3 month rate - daily indicator (numeric)
20 - nr.employed: number of employees - quarterly indicator (numeric)

# 4 Obtaining the Data

## 4.1 Loading the Data

The data can be obtained by searching for the bank marketing data set on the UCI Machine Learning Repository. I am using the most updated data set from the data folder, the "bank-additional" file.

## 4.2 Splitting the Data into Train and Test Data Sets

Since we are dealing with a data set that has a significant amount of observations, we will split the training data set into 80% of the total observations and test set will be comprised of 20% of the data from the original data set.

```r
set.seed(1, sample.kind = "Rounding")
test_index <- createDataPartition(y = bank_marketing_data$y, times = 1, p = 0.2, list = FALSE)
train_set <- bank_marketing_data[-test_index,]
test_set <- bank_marketing_data[test_index,]
```

The 80/20 split provides the following dimensions for each of the new data sets.

```
dim(train_set)
```

```
## [1] 32950    21
```

```
dim(test_set)
```

```
## [1] 8238   21
```

Important: It is essential that we check that the split is stratified. This is an important concept that helps us ensure our data is not overfit. In both the training and test dataset we can see that 11% of the observations subscribe to a term deposit.

```
##
##    no   yes
## 29238  3712
```

```
##
##   no  yes
## 7310  928
```

# 5 Exploratory Data Analysis

## 5.1 Check for missing values

The first step I take in any data analysis is determining whether there is or is not missing values. This is a critical step since machine learning algorithms cannot work when there are missing values.

```
## [1] 32950
```

```
## [1] 8238
```

Fortunately, there is no missing values in this data set. Therefore, we will not need to impute any of the values for the data.

## 5.2 Look at the first several rows of data

```
##   age       job marital   education default housing loan   contact month
## 1  56 housemaid married    basic.4y      no      no   no telephone   may
## 2  57  services married high.school unknown      no   no telephone   may
## 3  37  services married high.school      no     yes   no telephone   may
## 4  40    admin. married    basic.6y      no      no   no telephone   may
## 5  56  services married high.school      no      no  yes telephone   may
## 6  45  services married    basic.9y unknown      no   no telephone   may
##   day_of_week duration campaign pdays previous    poutcome emp.var.rate
## 1         mon      261        1   999        0 nonexistent          1.1
## 2         mon      149        1   999        0 nonexistent          1.1
## 3         mon      226        1   999        0 nonexistent          1.1
```

```
## 4         mon      151        1   999        0 nonexistent          1.1
## 5         mon      307        1   999        0 nonexistent          1.1
## 6         mon      198        1   999        0 nonexistent          1.1
##   cons.price.idx cons.conf.idx euribor3m nr.employed  y
## 1         93.994         -36.4     4.857        5191 no
## 2         93.994         -36.4     4.857        5191 no
## 3         93.994         -36.4     4.857        5191 no
## 4         93.994         -36.4     4.857        5191 no
## 5         93.994         -36.4     4.857        5191 no
## 6         93.994         -36.4     4.857        5191 no
```

We can see from looking at the first several rows of data that a lot of this data needs to be turned into factors. And, upon closer inspection of the data dictionary listed above, we can easily determine the categories in each feature. due to the large ranges associated with some of the numeric values, we may need to normalize and scale our numeric vectors for some of our models. Specifically, we will need to normalize our data for models that rely on distance

### 5.3 Summary of each Variable

```
##       age            job               marital            education
##  Min.   :17.00   Length:32950       Length:32950       Length:32950
##  1st Qu.:32.00   Class :character   Class :character   Class :character
##  Median :38.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   :40.03
##  3rd Qu.:47.00
##  Max.   :98.00
##    default            housing             loan             contact
##  Length:32950       Length:32950       Length:32950       Length:32950
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##     month           day_of_week          duration         campaign
##  Length:32950       Length:32950       Min.   :   0.0   Min.   : 1.000
##  Class :character   Class :character   1st Qu.: 102.0   1st Qu.: 1.000
##  Mode  :character   Mode  :character   Median : 179.0   Median : 2.000
##                                        Mean   : 257.6   Mean   : 2.583
##                                        3rd Qu.: 318.0   3rd Qu.: 3.000
##                                        Max.   :4918.0   Max.   :56.000
##      pdays          previous        poutcome          emp.var.rate
##  Min.   :  0.0   Min.   :0.0000   Length:32950       Min.   :-3.40000
##  1st Qu.:999.0   1st Qu.:0.0000   Class :character   1st Qu.:-1.80000
##  Median :999.0   Median :0.0000   Mode  :character   Median : 1.10000
##  Mean   :961.8   Mean   :0.1731                      Mean   : 0.08896
##  3rd Qu.:999.0   3rd Qu.:0.0000                      3rd Qu.: 1.40000
##  Max.   :999.0   Max.   :7.0000                      Max.   : 1.40000
##  cons.price.idx  cons.conf.idx      euribor3m      nr.employed     y
##  Min.   :92.20   Min.   :-50.80   Min.   :0.634   Min.   :4964   no :29238
##  1st Qu.:93.08   1st Qu.:-42.70   1st Qu.:1.344   1st Qu.:5099   yes: 3712
##  Median :93.75   Median :-41.80   Median :4.857   Median :5191
##  Mean   :93.58   Mean   :-40.49   Mean   :3.629   Mean   :5167
##  3rd Qu.:93.99   3rd Qu.:-36.40   3rd Qu.:4.961   3rd Qu.:5228
```

```
## Max.   :94.77   Max.   :-26.90   Max.   :5.045   Max.   :5228
```

One thing that jumps at me from the summary, is that some of the numeric variables in this data set have large ranges vs. others that have very small min-max range. Due to the variance in feature ranges with some of the numeric values, we may need to normalize and scale our numeric vectors for some of our models. Specifically, we will need to normalize our data for models that rely on distance

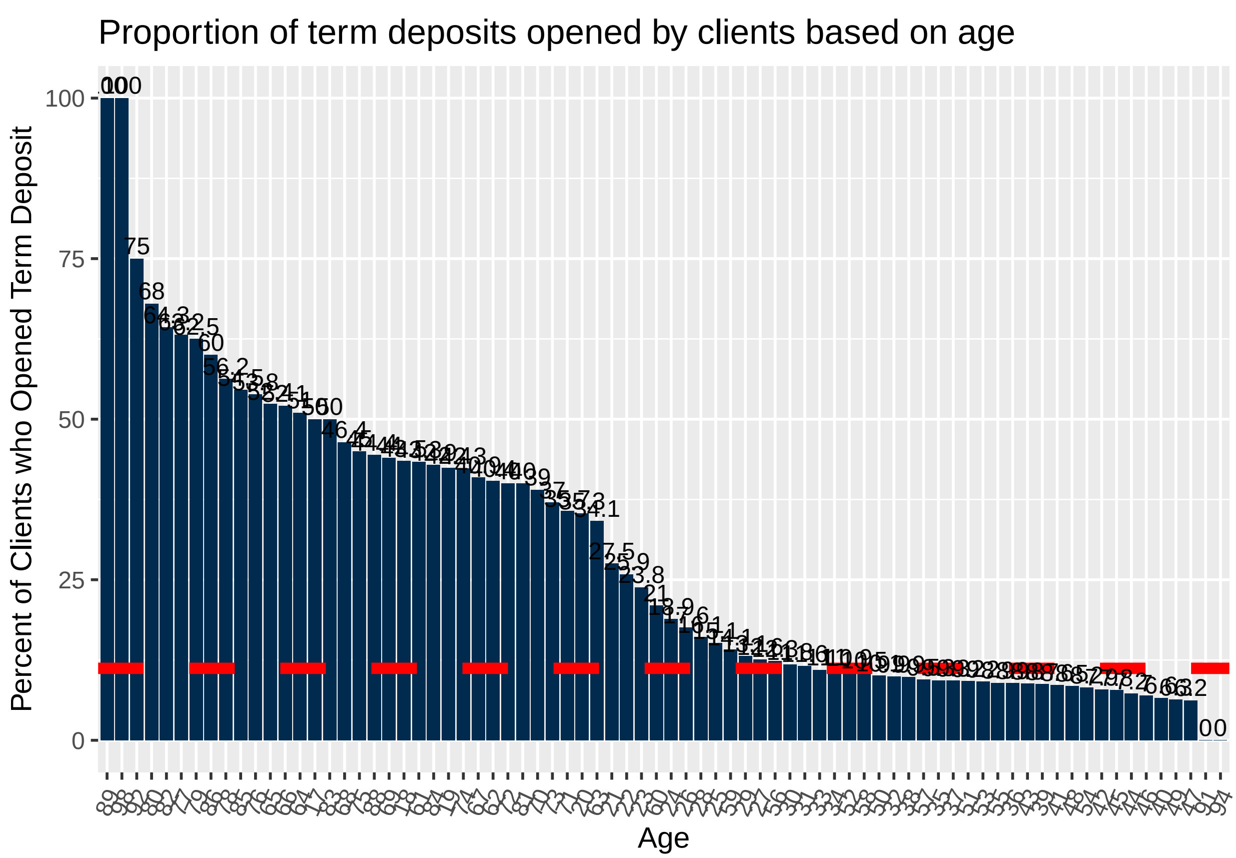
Both of the categorization and scaling of features will occur once we have determined the variables to use in our algorithms.

# 6 Feature Selection Analysis

## 6.1 Demographic Data Feature Analysis

### 6.1.1 Age vs Term Deposit

Age is actually considered a categorical variable because it is not continuous. In the following chart, we clearly outline the following: 1) The proportion of term deposit subscriptions for each age group, 2) the dashed red line represents the proportion of the total population that subscribed to a term deposit (as discussed above, the dashed red line is at 11.2%), 3) by comparing the proportion of each age group that subscribed to a term deposit to the proportion of the population that subscribed to the term deposit, we can clearly see whether specific ages effect whether or not the client will subscribe to a term deposit.



As you can see in the above chart, age clearly plays a role in whether a client will or will not subscribe to a term deposit. In this chart we can see that the top 14 age categories most likely to subscribe to a term deposit are above the age of 64. The next best category is the group at age 17. Additionally, this bar chart indicates that individuals with ages between 35 and 55 are less likely to subscribe to a term deposit.

```
##     age   no yes       perc total
## 1    17    2   2  50.000000     4
## 2    18   13  10  43.478261    23
## 3    19   19  14  42.424242    33
## 4    20   33  18  35.294118    51
## 5    21   58  22  27.500000    80
## 6    22   86  30  25.862069   116
## 7    23  144  45  23.809524   189
## 8    24  301  70  18.867925   371
## 9    25  404  72  15.126050   476
## 10   26  446  95  17.560074   541
## 11   27  596  86  12.609971   682
## 12   28  657 126  16.091954   783
## 13   29 1022 154  13.095238  1176
## 14   30 1222 164  11.832612  1386
## 15   31 1385 181  11.558110  1566
## 16   32 1325 146   9.925221  1471
## 17   33 1305 161  10.982265  1466
## 18   34 1227 151  10.957910  1378
## 19   35 1262 130   9.339080  1392
## 20   36 1312 128   8.888889  1440
## 21   37 1054 108   9.294320  1162
## 22   38 1020 112   9.893993  1132
## 23   39 1055 101   8.737024  1156
## 24   40  889  63   6.617647   952
## 25   41  924  87   8.605341  1011
## 26   42  851  73   7.900433   924
## 27   43  764  74   8.830549   838
## 28   44  742  58   7.250000   800
## 29   45  816  69   7.796610   885
## 30   46  763  57   6.951220   820
## 31   47  692  46   6.233062   738
## 32   48  724  67   8.470291   791
## 33   49  635  43   6.342183   678
## 34   50  624  70  10.086455   694
## 35   51  533  54   9.199319   587
## 36   52  548  67  10.894309   615
## 37   53  546  55   9.151414   601
## 38   54  493  44   8.193669   537
## 39   55  468  46   8.949416   514
## 40   56  504  71  12.347826   575
## 41   57  459  48   9.467456   507
## 42   58  426  50  10.504202   476
## 43   59  316  52  14.130435   368
## 44   60  177  47  20.982143   224
## 45   61   34  26  43.333333    60
## 46   62   31  21  40.384615    52
## 47   63   27  14  34.146341    41
## 48   64   24  25  51.020408    49
```

```
## 49  65   20  22  52.380952    42
## 50  66   23  25  52.083333    48
## 51  67   13   9  40.909091    22
## 52  68   15  13  46.428571    28
## 53  69   14  11  44.000000    25
## 54  70   25  16  39.024390    41
## 55  71   27  15  35.714286    42
## 56  72   18  12  40.000000    30
## 57  73   17  10  37.037037    27
## 58  74   15  11  42.307692    26
## 59  75   11   9  45.000000    20
## 60  76   12  14  53.846154    26
## 61  77    7  12  63.157895    19
## 62  78    7   9  56.250000    16
## 63  79    3   5  62.500000     8
## 64  80    8  17  68.000000    25
## 65  81    9   6  40.000000    15
## 66  82    5   9  64.285714    14
## 67  83    7   7  50.000000    14
## 68  84    4   3  42.857143     7
## 69  85    5   6  54.545455    11
## 70  86    2   3  60.000000     5
## 71  88   10   8  44.444444    18
## 72  89    0   2 100.000000     2
## 73  91    1   0   0.000000     1
## 74  92    1   3  75.000000     4
## 75  94    1   0   0.000000     1
## 76  98    0   2 100.000000     2
```

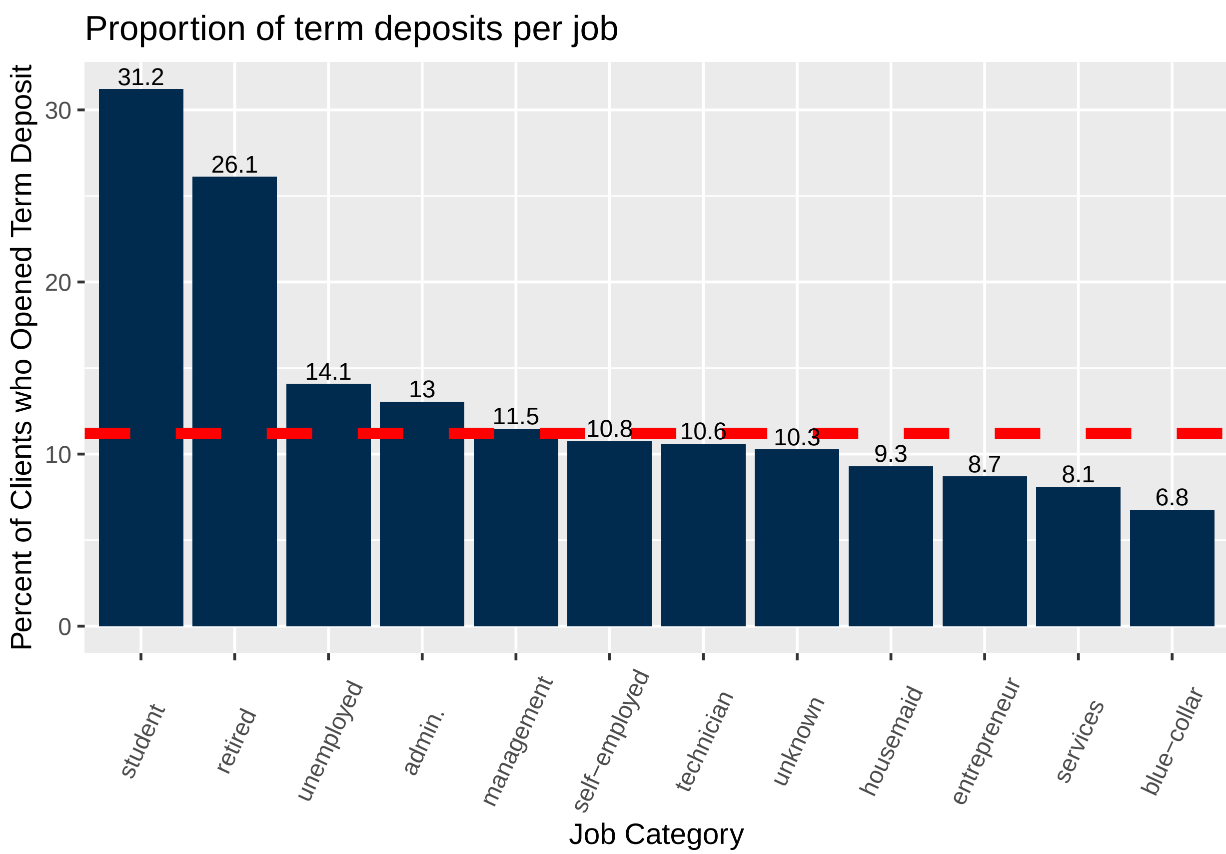
The custom contingency table further explains the bar charts findings, but also provides more insight into the total number of clients in each age group. While clients under the age of 30 and clients over the age of 59 are more likely to subscribe to a term deposit, the bulk of clients the bank reached out to are between the ages of 30 and 59. These clients are much less likely to open a term deposit.

**In summary, age is an extremely important variable that should be included in the final model.**

#### 6.1.2 Job Variable vs. Term Deposit

The following table and chart highlight a significant amount of variability among the likelihood a person with a specific job will subscribe to the data.

First, if you look at the following bar chart, we see the following. 1) the dashed redline represents the proportion of the total population that subscribed to a term deposit at approximately 11%. 2) For each segment of the population based on job title, we can see the percent likelihood an individual with that title will subscribe to a loan deposit. As you can see, students and retired individuals are more than 2x likely to sign up for a term deposit versus the population, while blue-collar and services jobs are half as likely to subscribe to a term deposit.




Second, if we compare the bar chart data to the custom contingency table below, we can see the total number of clients per category. This gives us a rough idea of how much each job segment plays a role in terms of total sign ups.

```
##             jobs   no  yes      perc total
## 1         admin. 7241 1086 13.041912  8327
## 2    blue-collar 6900  501  6.769355  7401
## 3   entrepreneur 1070  102  8.703072  1172
## 4      housemaid  761   78  9.296782   839
## 5     management 2052  266 11.475410  2318
## 6        retired 1033  365 26.108727  1398
## 7  self-employed 1021  123 10.751748  1144
## 8       services 2912  257  8.109814  3169
## 9        student  485  220 31.205674   705
## 10    technician 4837  574 10.608021  5411
## 11    unemployed  690  113 14.072229   803
## 12       unknown  236   27 10.266160   263
```

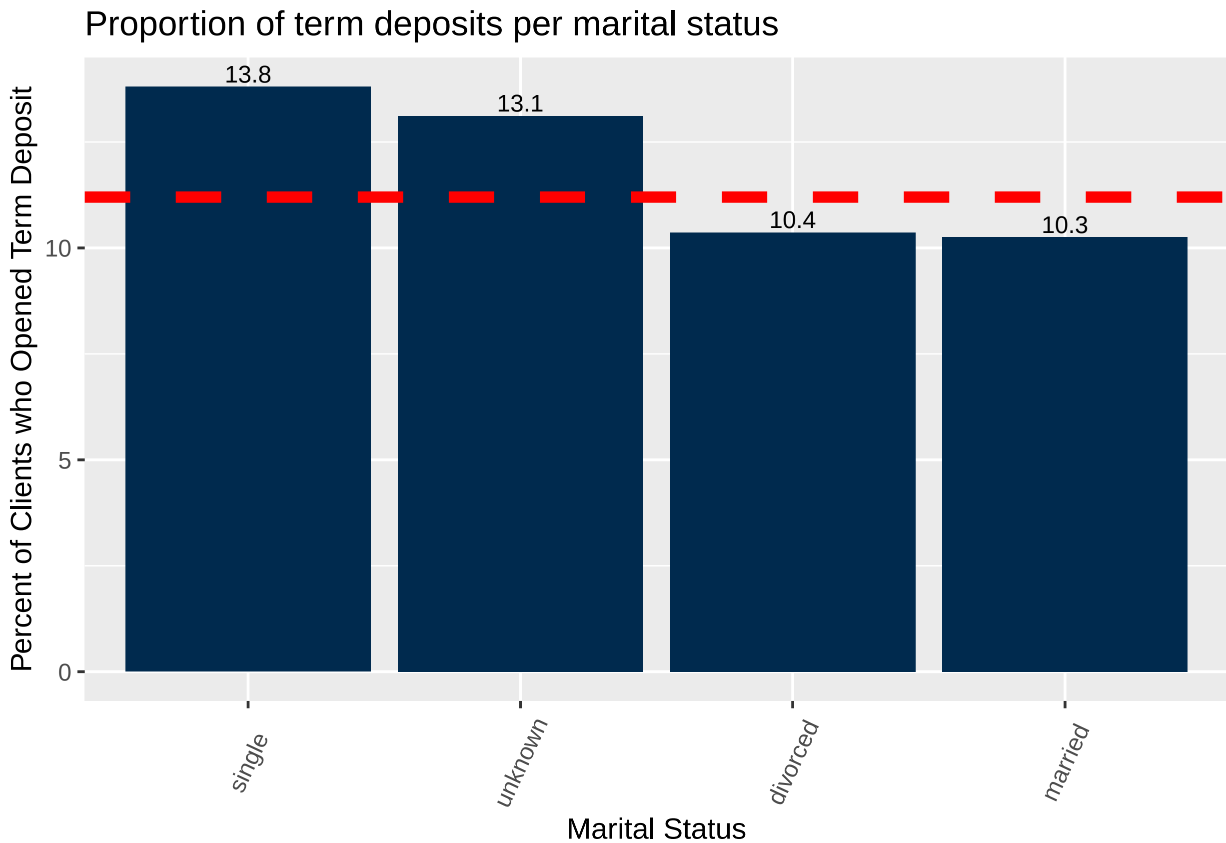
**In summary, Job title seems to be a good indication of whether a client subscribes or does not. This should be included in our model.**

#### 6.1.3 Marital Status vs. Term Deposit

Once again, we are looking at a bar chart and custom contingency table to determine how much variability occurs between marital status and our term deposit target variable.

The bar chart shows that single and unknown marital statuses are more likely than the population average (red dashed line at 11.2%) to open a term deposit than divorced and married individuals. However, both the contingency table and bar chart do not highlight a significant amount of variability stemming from this variable.

```
##     status    no  yes     perc total
## 1 divorced  3346  387 10.36700  3733
## 2  married 17902 2046 10.25667 19948
## 3   single  7937 1271 13.80321  9208
## 4  unknown    53    8 13.11475    61
```



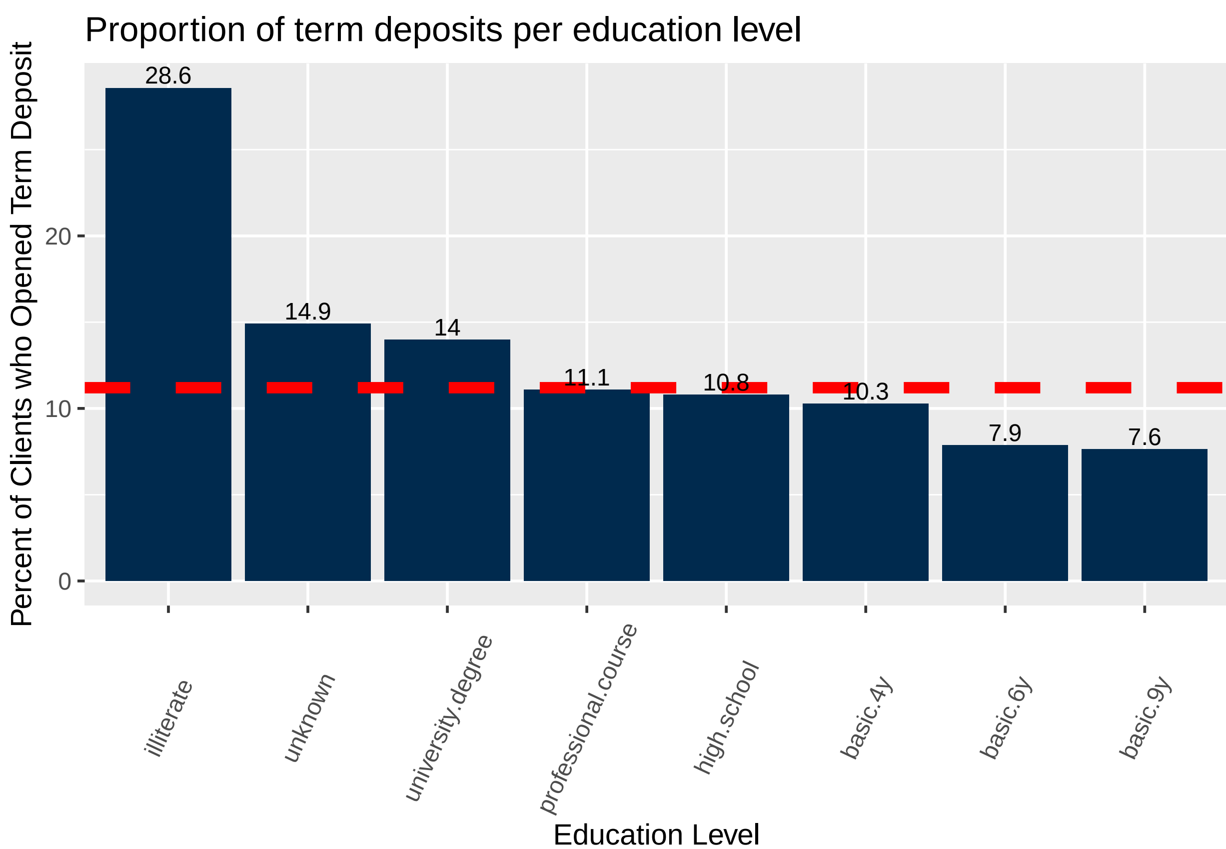

**I do not think we should include this variable in our model.**

#### 6.1.4 Education vs Term Deposit

Education does seem to play a larger role than marital status in discovering some of the variability of term deposit success rate. Specifically, individuals classified as illiterate and unknown educations are more likely to sign up for a term deposit. It is worth noting that illiterate classification (the, by far, most likely to subscribe) only accounts for 14 of the 32,950 clients in the bank database. Also noteworthy, people with 6 year and 9 year educations are significantly less likely to subscribe.

```
##                status   no  yes      perc total
## 1            basic.4y 3011  345 10.280095  3356
## 2            basic.6y 1700  145  7.859079  1845
```

```
## 3            basic.9y 4454  368  7.631688  4822
## 4         high.school 6784  820 10.783798  7604
## 5          illiterate   10    4 28.571429    14
## 6 professional.course 3734  465 11.074065  4199
## 7   university.degree 8397 1364 13.973978  9761
## 8             unknown 1148  201 14.899926  1349
```



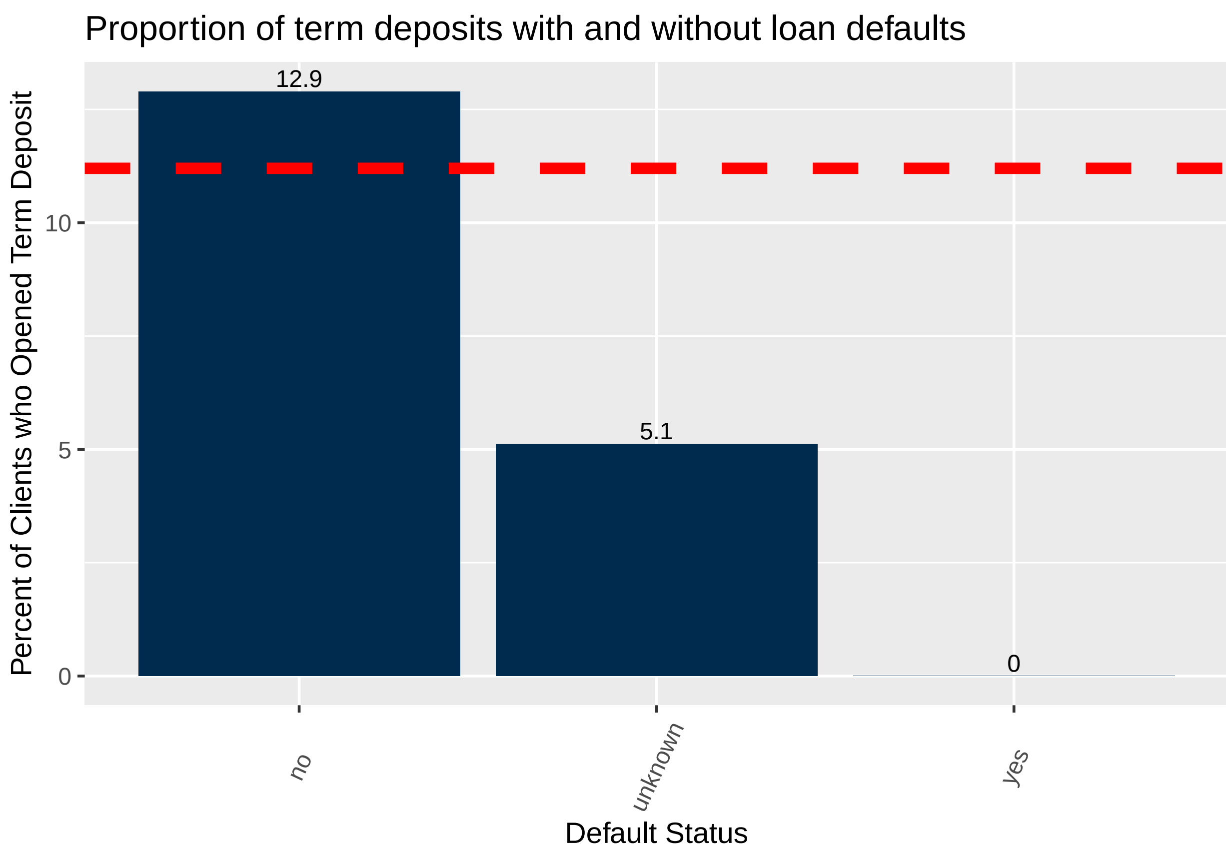

In total 6 year, 9 year, unknown, and illiterate education classifications account for 24.37% of the total population. While this variance isn't dramatic, it will probably help us classify term deposit subscriptions.

**We will include this variable in our classification model.**

#### 6.1.5 Default vs Term Deposit

My initial thoughts are that this will be an important variable. Since loan defaults refer to an individual's inability to pay back loans, it implies that the individual does not have enough funds to open new term deposits.

```
##    status    no  yes      perc total
## 1      no 22681 3358 12.896041 26039
## 2 unknown  6554  354  5.124493  6908
## 3     yes     3    0  0.000000     3
```

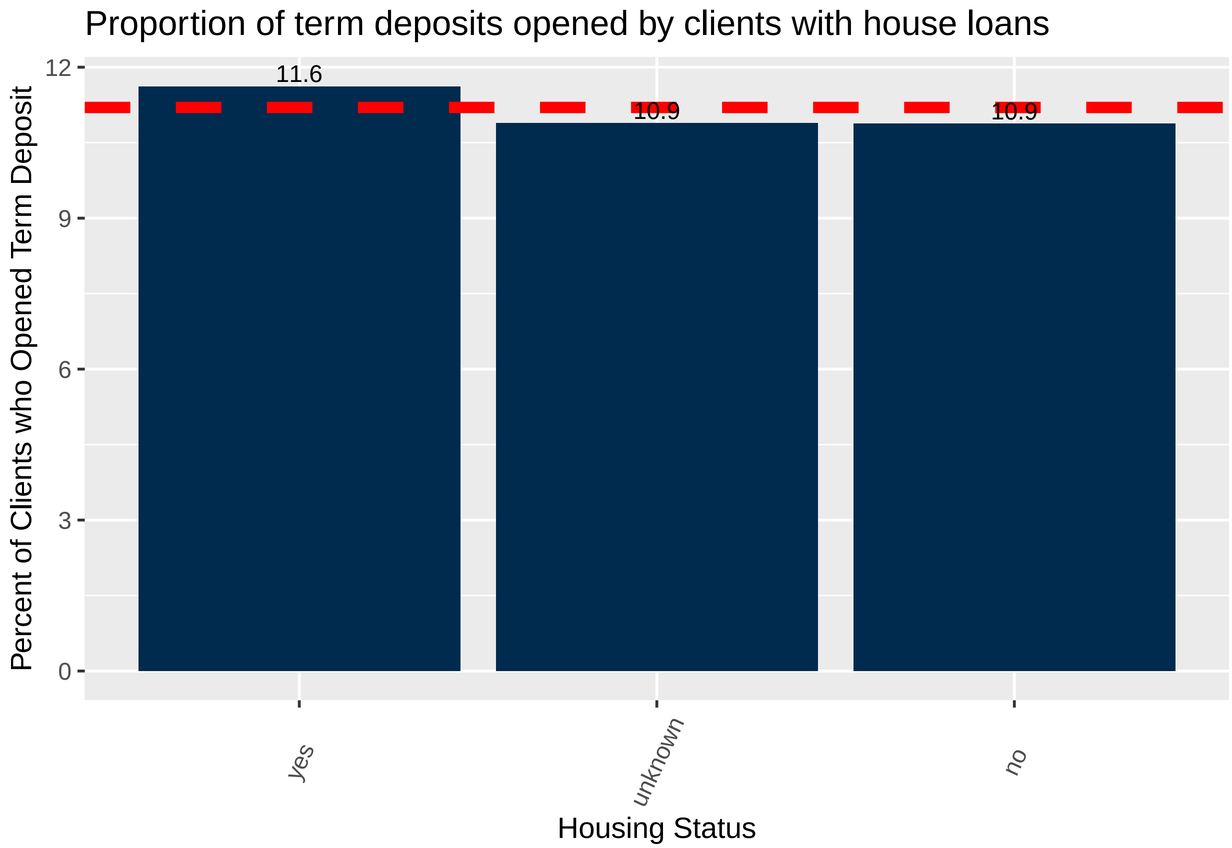



As predicted, clients without loan defaults were more likely to have the funds to open a term deposit with the bank. Clients with unknown status open term deposits at less than 50% the rate of the population. I was surprised that so few clients showed definitive loan defaults.

**This variable should definitely be added to the model.**

#### 6.1.6 Housing vs Term Deposit

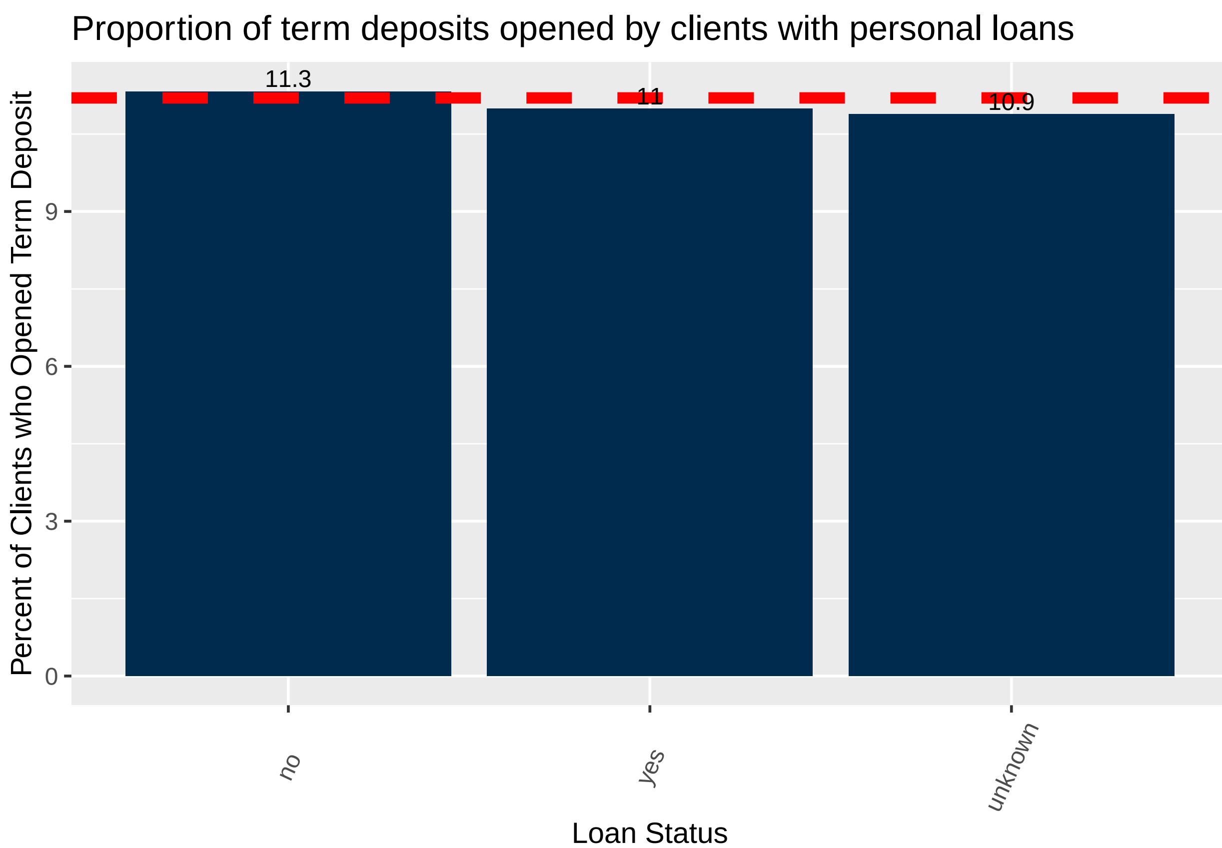
```
##    status    no  yes     perc total
## 1      no 13327 1627 10.88003 14954
## 2 unknown   704   86 10.88608   790
## 3     yes 15207 1999 11.61804 17206
```




**Do not include this variable. Very static among the different categories.**

#### 6.1.7 Personal Loan vs Term Deposit

```
##     status    no  yes     perc total
## 1       no 24098 3078 11.32617 27176
## 2  unknown   704   86 10.88608   790
## 3      yes  4436  548 10.99518  4984
```



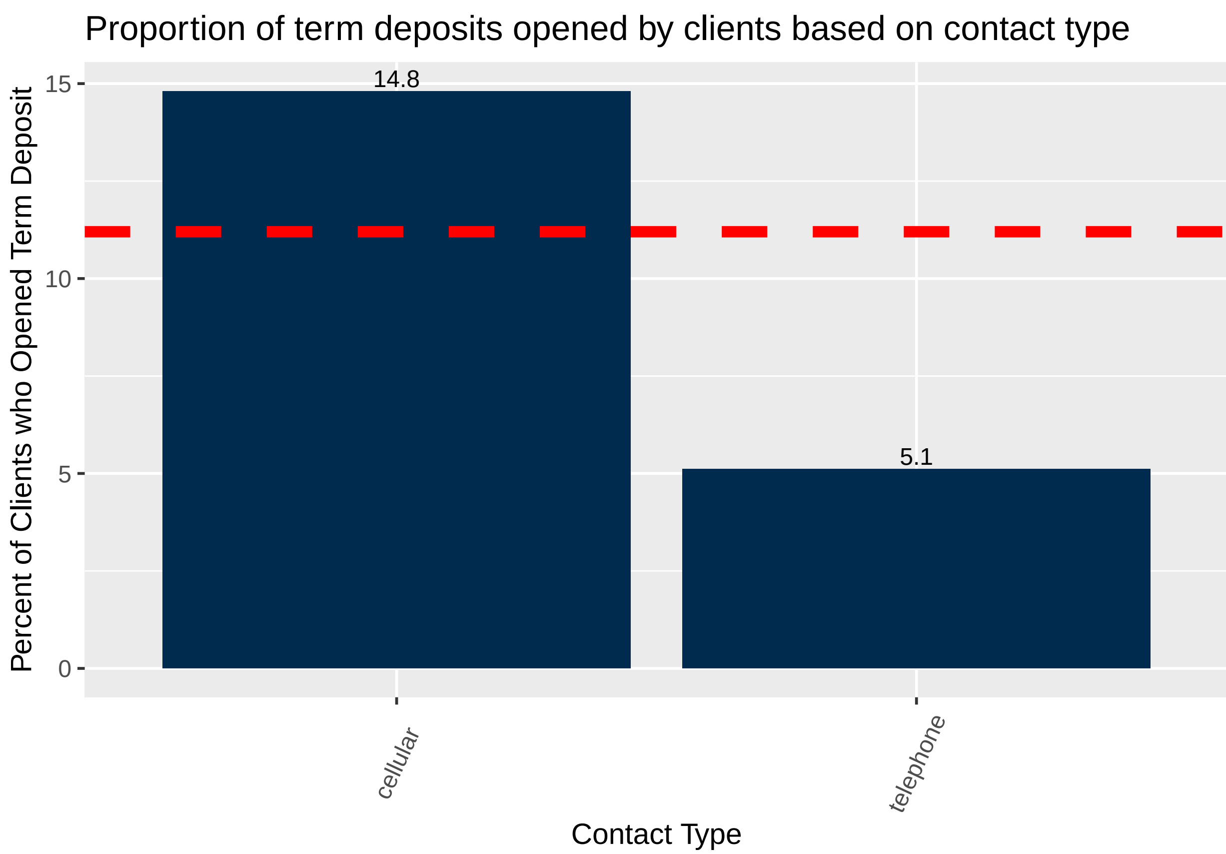

**Do not include this variable. Very static among the different categories of this feature.**

## 6.2 Campaign Data Feature Analysis

### 6.2.1 Contact Type vs Term Deposit

To my surprise, the type of audio device that the marketing team connected with does seem to play a role in determining whether or not an individual will subscribe to a term deposit. As you can see in the below chart and contingency table, clients reached on telephone only signed up to a term deposit 5.12% of the time. This is less than 50% of the population's average subscription rate.

```
##      status    no  yes      perc total
## 1  cellular 17813 3096 14.807021 20909
## 2 telephone 11425  616  5.115854 12041
```

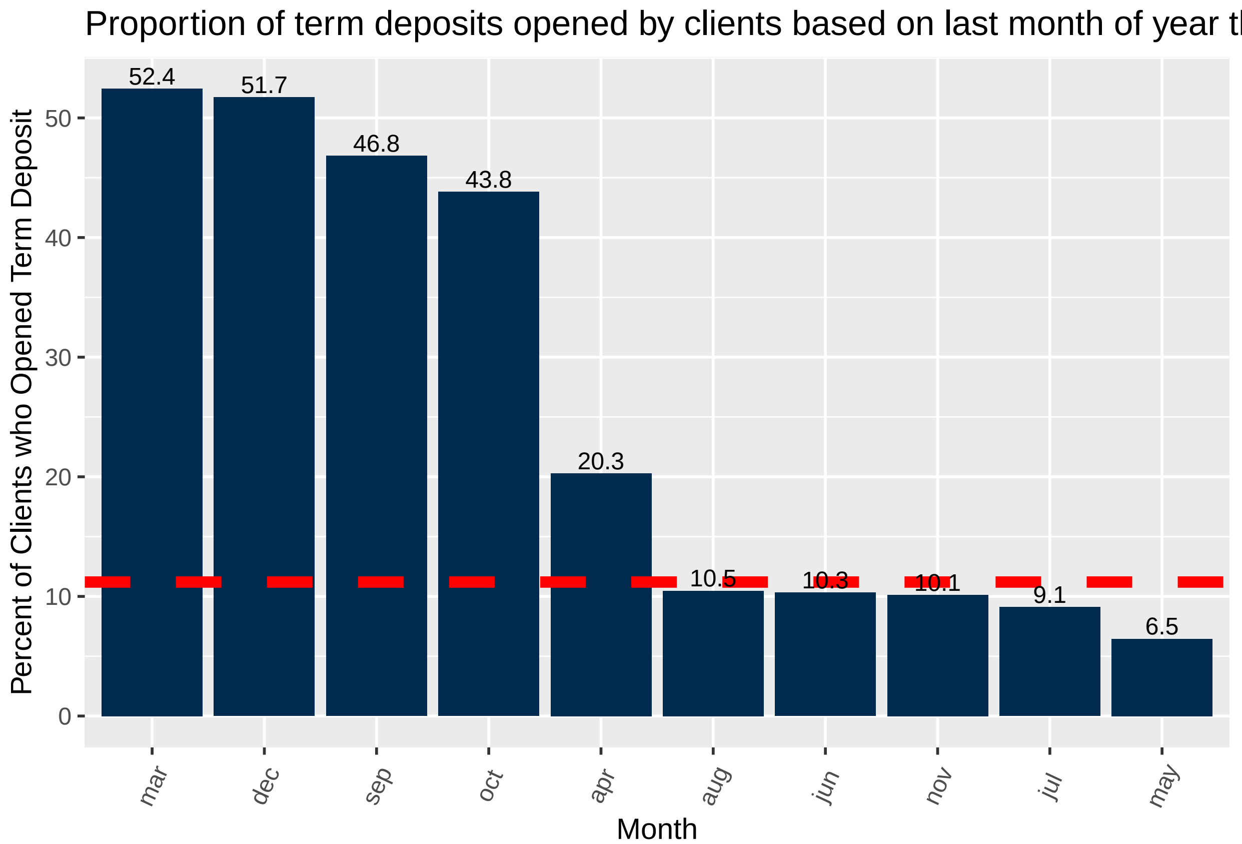



**Definitely include this variable in the model.**

#### 6.2.2 Last month reached vs Term Deposit

First, it is important to note that the marketing team did not reach out to any clients in the months of January and February.

```
##    status    no yes      perc total
## 1     apr  1670 425 20.286396  2095
## 2     aug  4476 523 10.462092  4999
## 3     dec    70  75 51.724138   145
## 4     jul  5198 520  9.094089  5718
## 5     jun  3804 438 10.325318  4242
## 6     mar   204 225 52.447552   429
## 7     may 10306 711  6.453663 11017
## 8     nov  2948 332 10.121951  3280
## 9     oct   318 248 43.816254   566
## 10    sep   244 215 46.840959   459
```

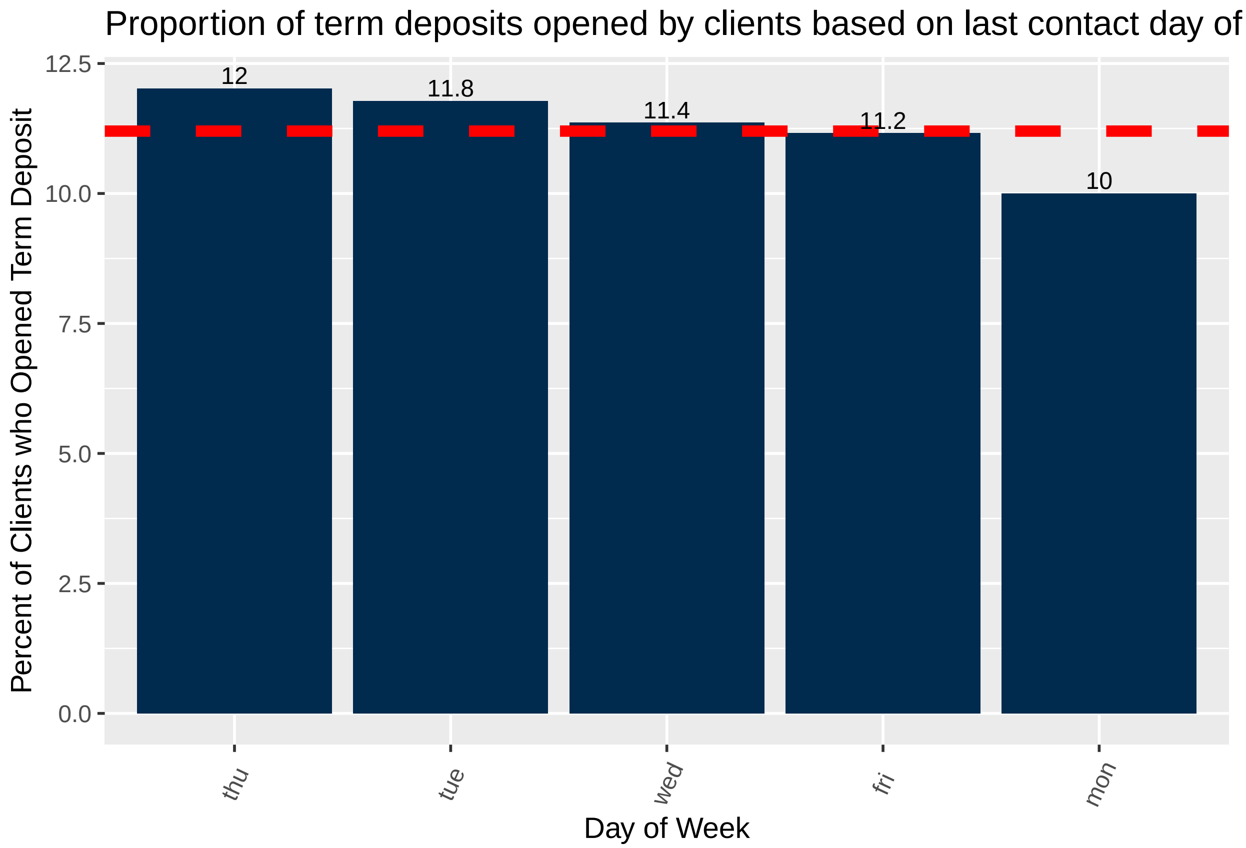



Not including those two months, the data shows a success rate of nearly 4x the number of subscriptions in March, December, September, and October as compared with the average rate of success from the population. Additionally, the late Spring and Summer months do not seem to be successful in generating term deposit subscriptions.

**Definitely include this variable in the model.**

#### 6.2.3 Last contact Day of the Week vs Term Deposit

```
##   status   no yes     perc total
## 1    fri 5621 706 11.15853  6327
## 2    mon 6063 674 10.00445  6737
## 3    thu 6097 833 12.02020  6930
## 4    tue 5684 759 11.78023  6443
## 5    wed 5773 740 11.36189  6513
```

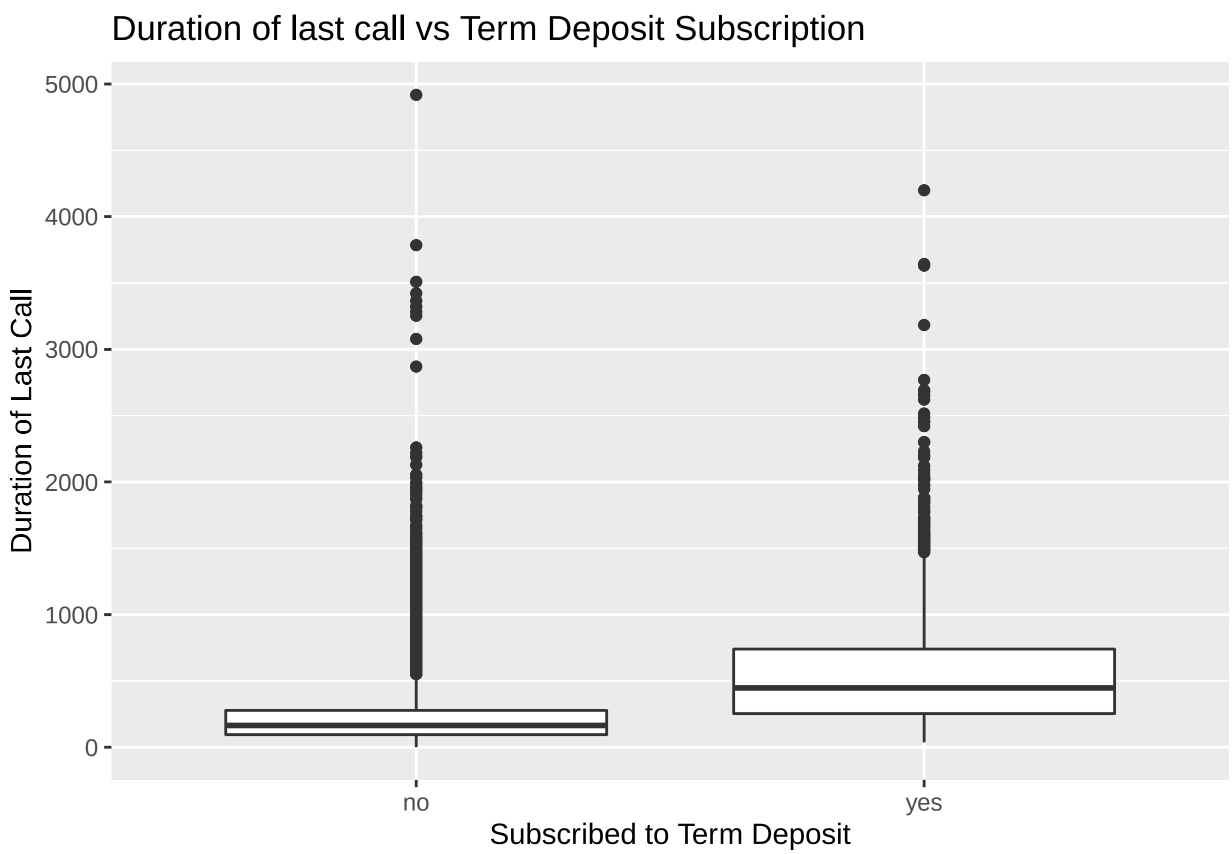



**This variable does not highlight much variability from the population subscription rate.**

#### 6.2.4 Duration of the Last Call vs. Term Deposits

Surprisingly, this is the first continuous variable we have seen in the data set. This variable references the length of the last phone call with a client and is measured in seconds. A box and whiskers plot is an effective way to compare continuous and categorical variables.

As you can see from the box and whisker plot below, there is not much overlap between the duration of calls for people who subscribed and people who did not subscribe. This highlights that individuals that did subscribe to the term deposit were much more likely to have a longer discussion with a member of the marketing team.




In fact, we can summarize the duration values associated with the box plot for the status of a term deposit. As you can see below, the first quartile for duration from clients that DID subscribe is 254 seconds. The third quartile for duration from clients that DID NOT subscribe was 278 seconds. This highlights the gap in duration between the two term deposit statuses. Essentially, a little more than 25% of the DID NOT Subscribe call durations overlap with a little less than 75% of the DID Subscribe call durations.

```
### calculated summary of duration when y == yes
summary(train_set$duration[train_set$y=="yes"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    37.0   254.0   447.5   550.3   740.0  4199.0
```

```
### total number of subscriptions for perspective
train_set %>%
  filter(y == "yes") %>%
  summarize(total = n())
```

```
##   total
## 1  3712
```

```
### calculated summary of duration when y == no
summary(train_set$duration[train_set$y=="no"])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0    95.0   164.0   220.5   278.0  4918.0
```

```r
### Total number of non-subscriptions for perspective
train_set %>%
  filter(y == "no") %>%
  summarize(total = n())
```

```
##   total
## 1 29238
```

**This is going to be a very important variable in our model.**

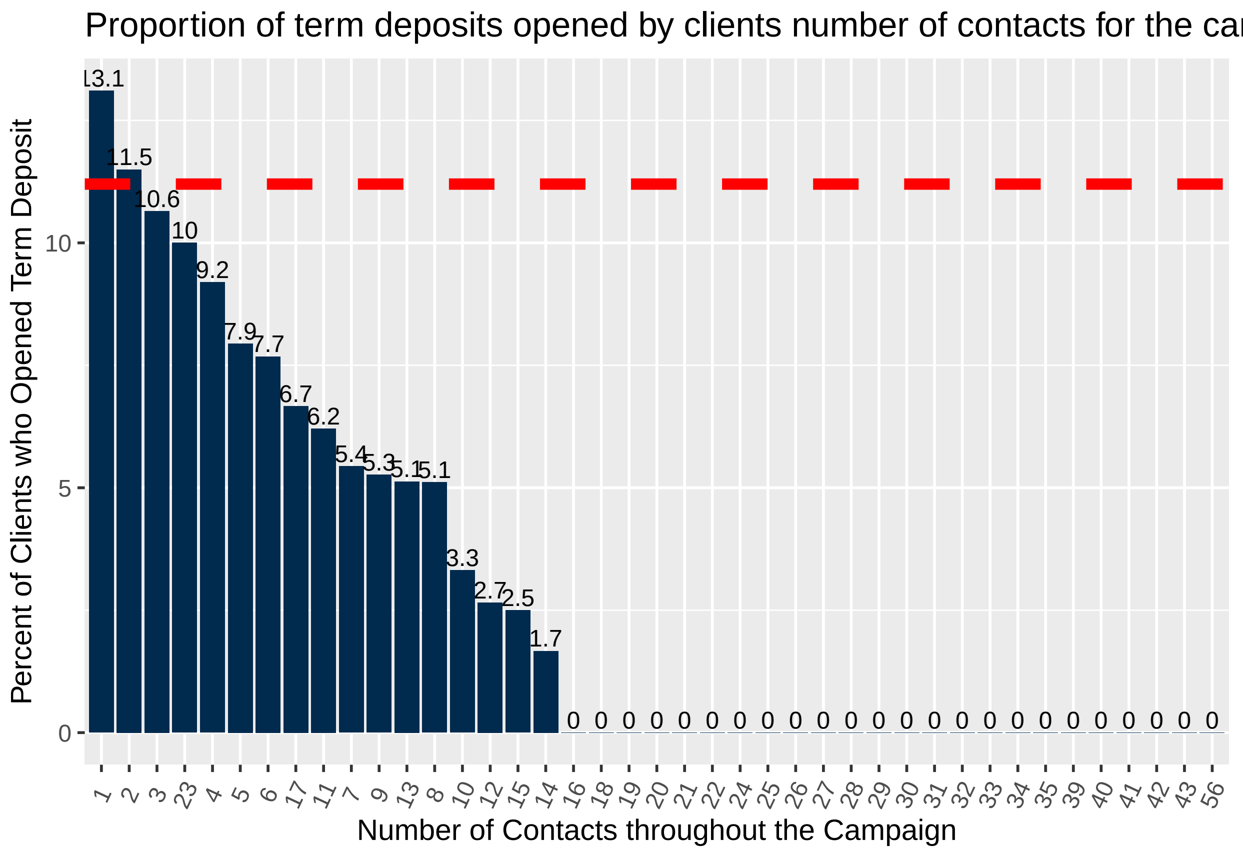
## 6.3 Additional Data Feature Analysis

### 6.3.1 Number of Times Contacted throughout Campaign vs Term Deposit

As the number of contacts throughout the campaign increased, the probability of the client subscribing to a term deposit decreases.

```
##    status    no  yes      perc total
## 1       1 12235 1845 13.103693 14080
## 2       2  7459  969 11.497390  8428
## 3       3  3818  455 10.648256  4273
## 4       4  1945  197  9.197012  2142
## 5       5  1182  102  7.943925  1284
## 6       6   721   60  7.682458   781
## 7       7   487   28  5.436893   515
## 8       8   297   16  5.111821   313
## 9       9   216   12  5.263158   228
## 10     10   175    6  3.314917   181
## 11     11   136    9  6.206897   145
## 12     12   110    3  2.654867   113
## 13     13    74    4  5.128205    78
## 14     14    59    1  1.666667    60
## 15     15    39    1  2.500000    40
## 16     16    43    0  0.000000    43
## 17     17    42    3  6.666667    45
## 18     18    28    0  0.000000    28
## 19     19    19    0  0.000000    19
## 20     20    24    0  0.000000    24
## 21     21    20    0  0.000000    20
## 22     22    14    0  0.000000    14
## 23     23     9    1 10.000000    10
## 24     24    12    0  0.000000    12
## 25     25     8    0  0.000000     8
## 26     26     7    0  0.000000     7
## 27     27     8    0  0.000000     8
## 28     28     8    0  0.000000     8
## 29     29     8    0  0.000000     8
## 30     30     5    0  0.000000     5
## 31     31     7    0  0.000000     7
## 32     32     3    0  0.000000     3
## 33     33     4    0  0.000000     4
## 34     34     3    0  0.000000     3
```

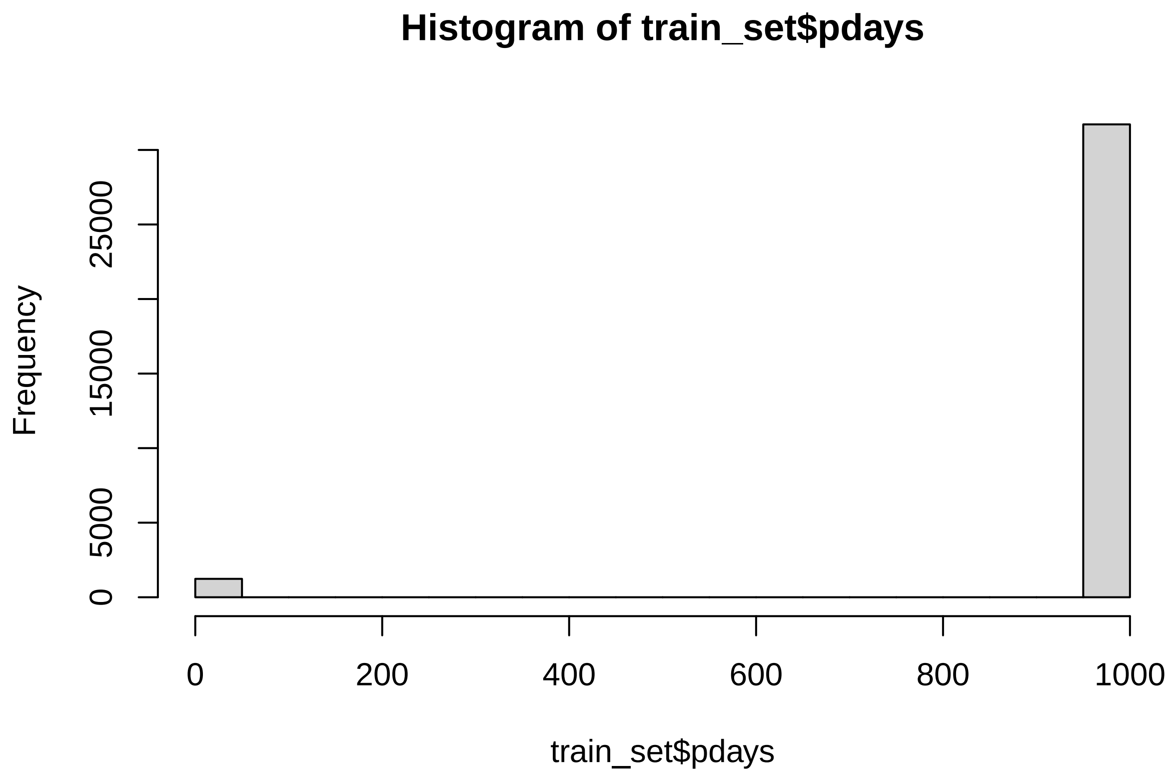
```
## 35      35      4     0   0.000000      4
## 36      39      1     0   0.000000      1
## 37      40      2     0   0.000000      2
## 38      41      1     0   0.000000      1
## 39      42      2     0   0.000000      2
## 40      43      2     0   0.000000      2
## 41      56      1     0   0.000000      1
```




It is definitely worth including this variable, but we can consolidate some of the categories in to new groups. This will be addressed in the feature engineering component of the report.

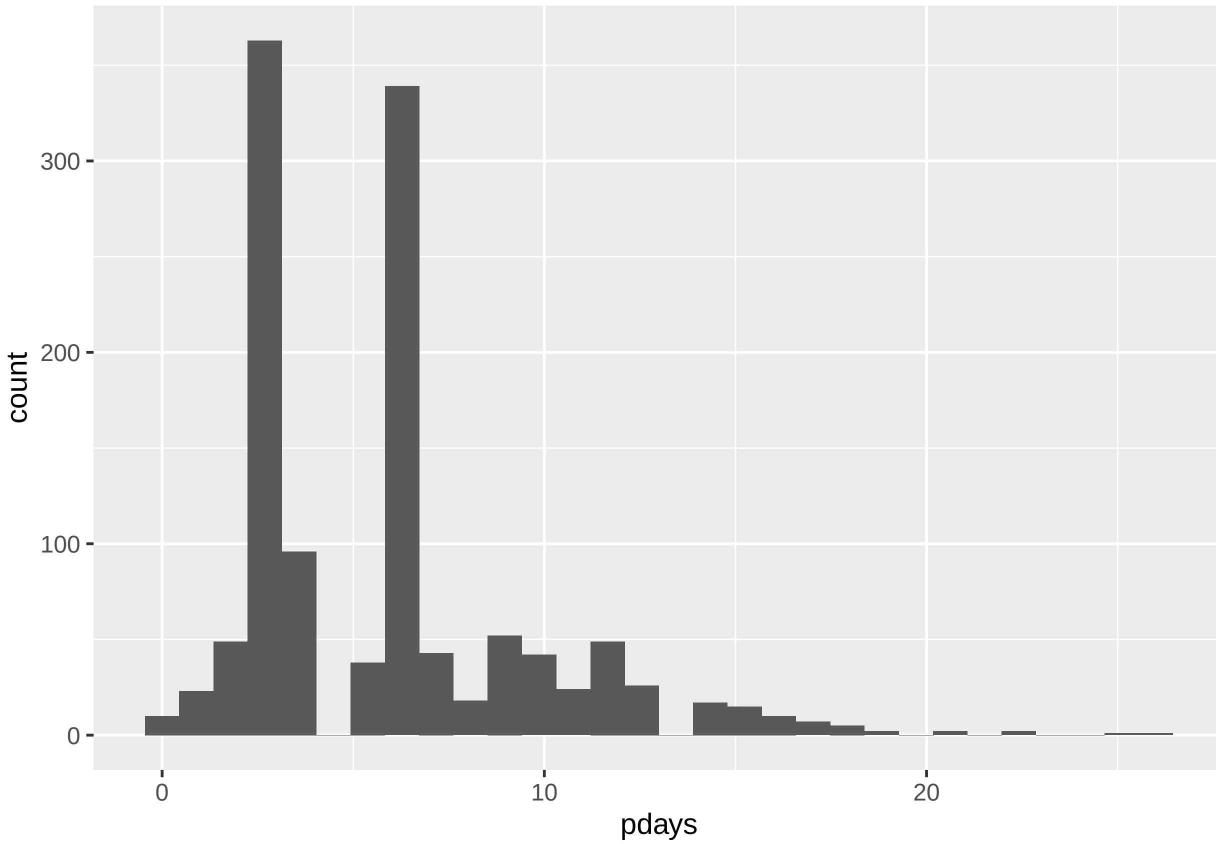
#### 6.3.2 Number of days passed since the client was last contacted vs Term Deposit

The first step in this analysis is to understand how the data is distributed for this variable. A simple histogram shows two modes and a large gap between the number of days. The first mode is from 0 – 50 days since the previous contact. The second mode is at 999, which is described as a value for the client never having been contacted.



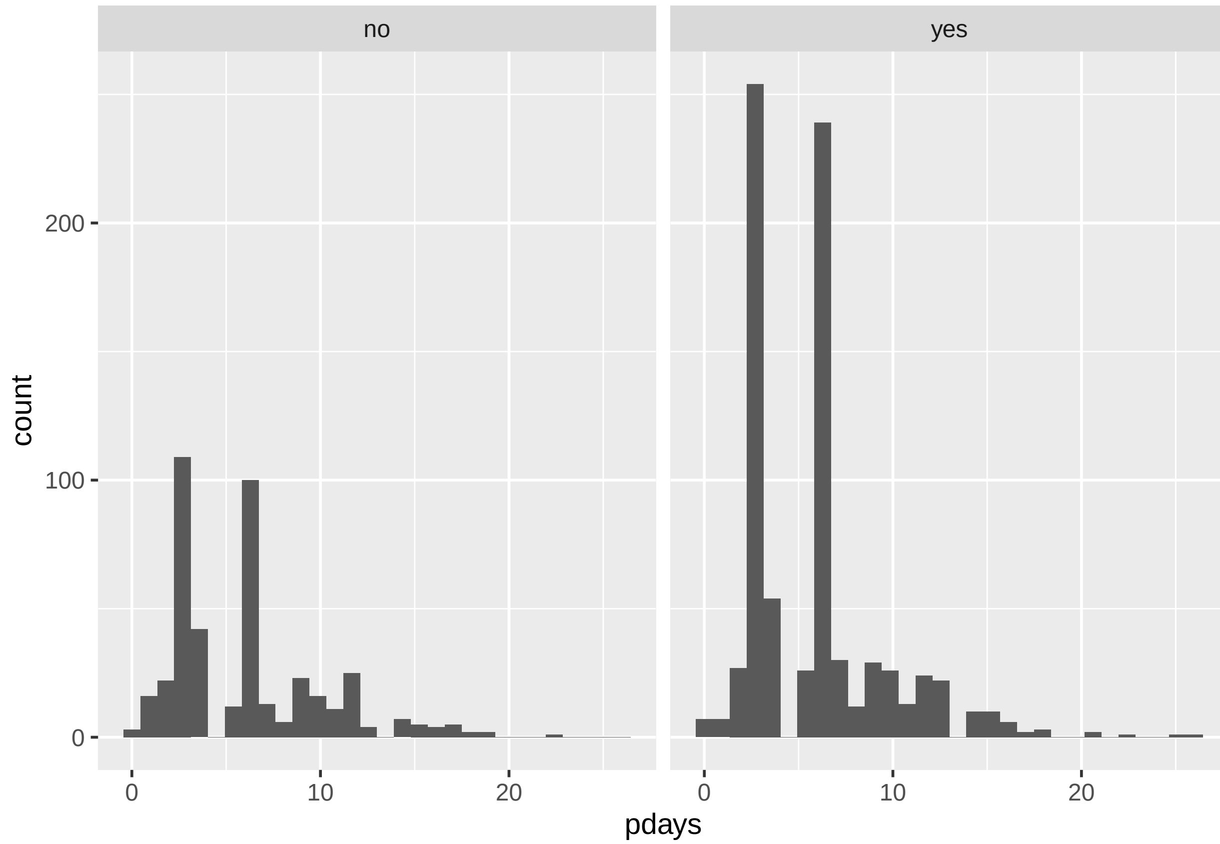

Now lets look at the data which does not include clients who were never contacted. As you can see of the clients that were contacted, it seems that the maximum amount of days since contact is around 30.

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



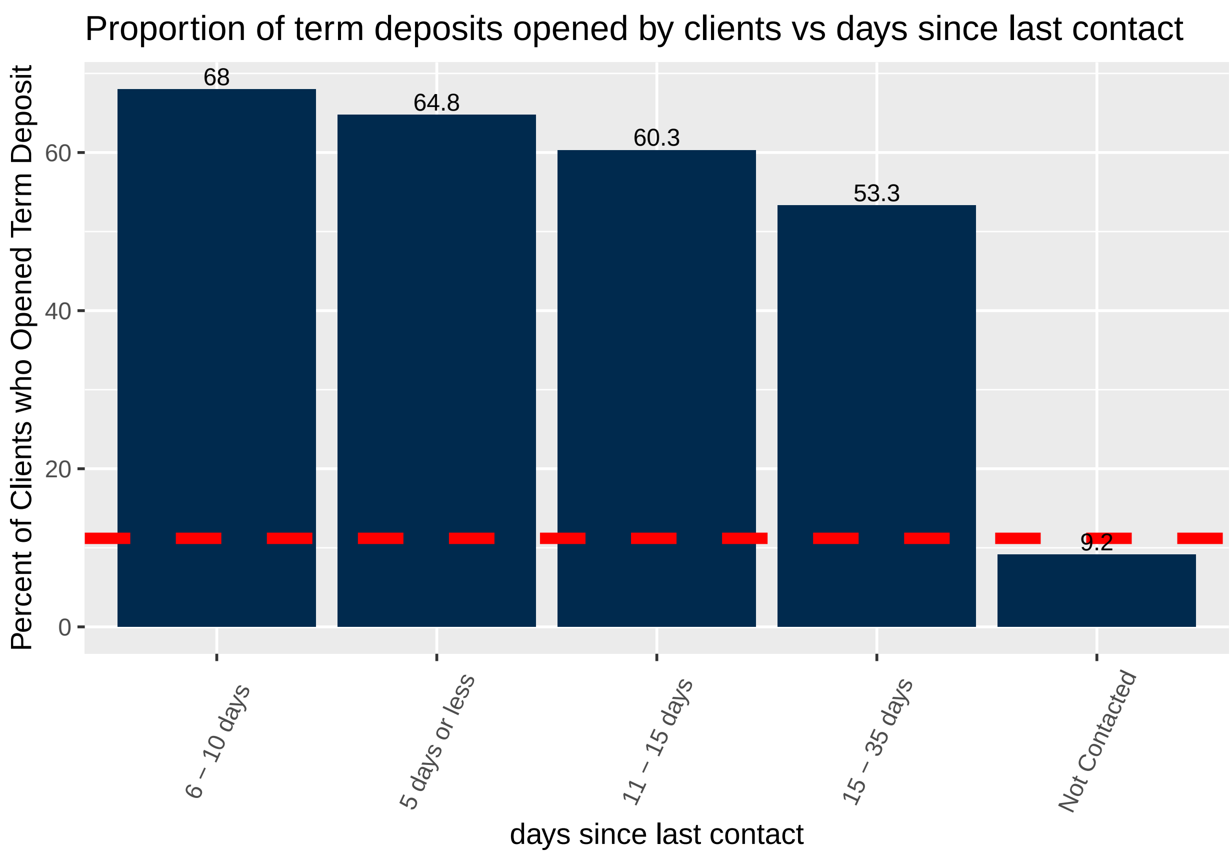

Additionally, we can facet this chart by our outcome variable. The below histogram highlights that individuals have been contacted by another campaign are more likely to subscribe to a term deposit.

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



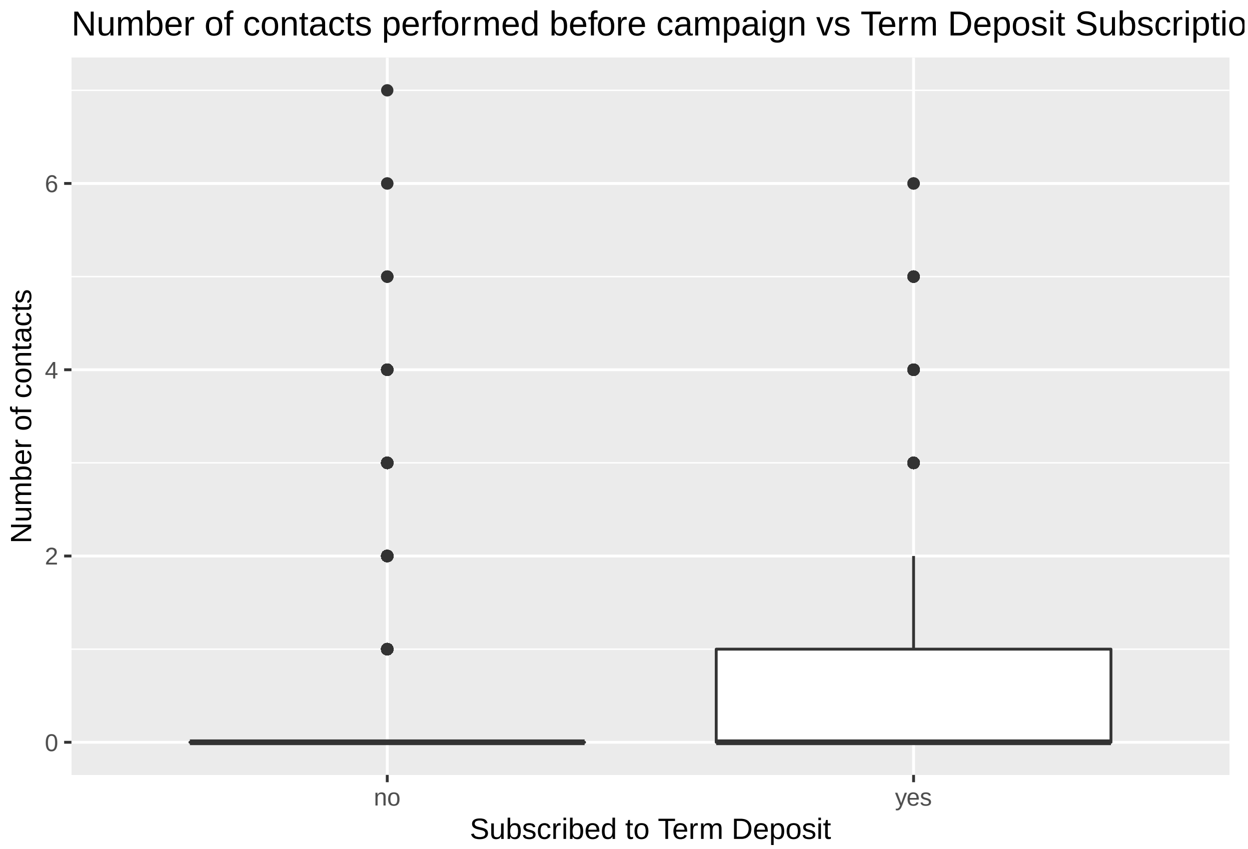

Finally, we can categorize the pdays variable into smaller segments and obtain the proportion of clients per segment that are likely to subscribe to the term deposit. For the sake of the data exploration, I have created a unique variable to store this data. I will make the final changes should this variable be considered in the final algorithm.

```
##           status    no  yes      perc total
## 1   11 - 15 days    52   79 60.305344   131
## 2   15 - 35 days    14   16 53.333333    30
## 3 5 days or less   204  375 64.766839   579
## 4    6 - 10 days   158  336 68.016194   494
## 5  Not Contacted 28810 2906  9.162568 31716
```




**Summary – while a vast majority of clients were never contacted by another campaign. We can see that those who have been contacted are much more likely to subscribe to a term deposit. This variable should definitely be included in our model.**

#### 6.3.3 Number of contacts prior to this campaign vs. Term Deposit

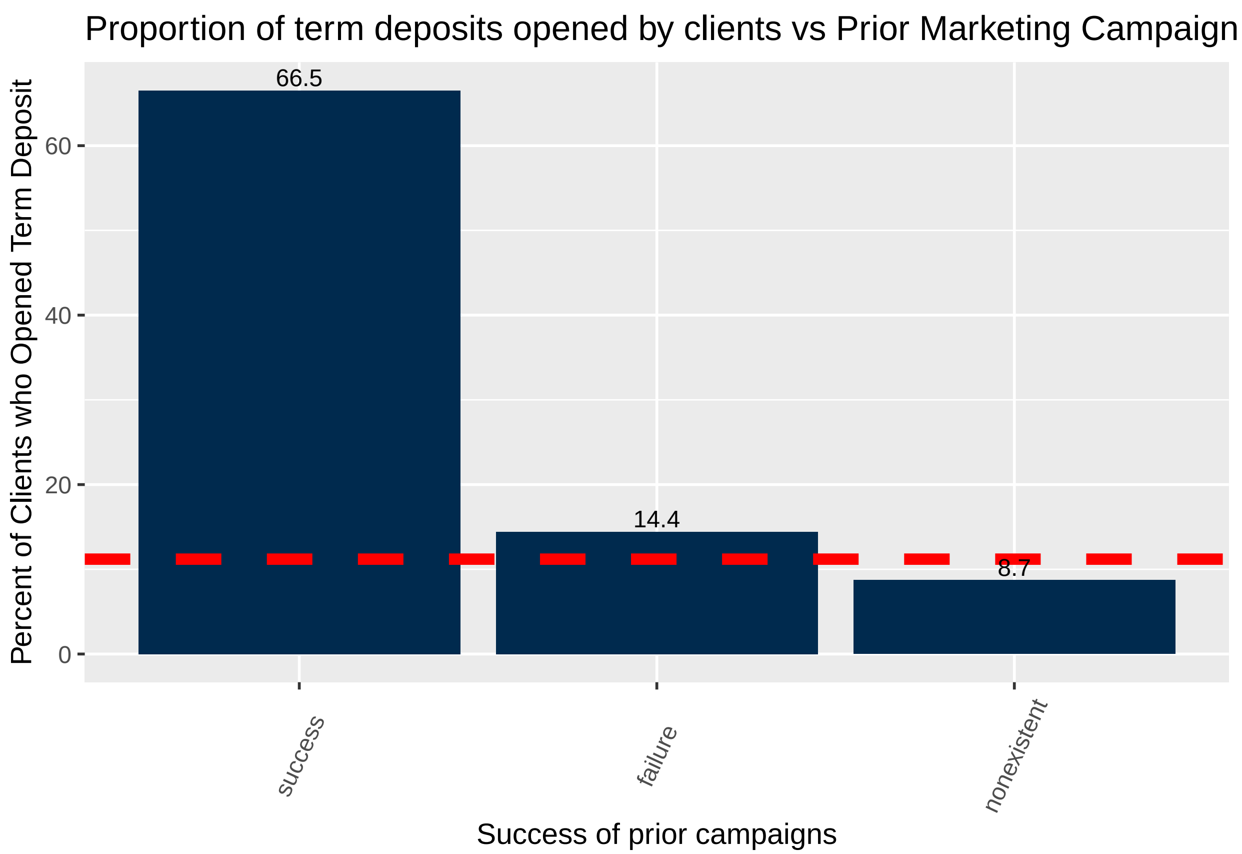



**Since the median values are both 0 and little variance in t his chart, I assume this variable does not impact term deposits.**

#### 6.3.4 Comparing the success of previous campaigns per client vs Term Deposit

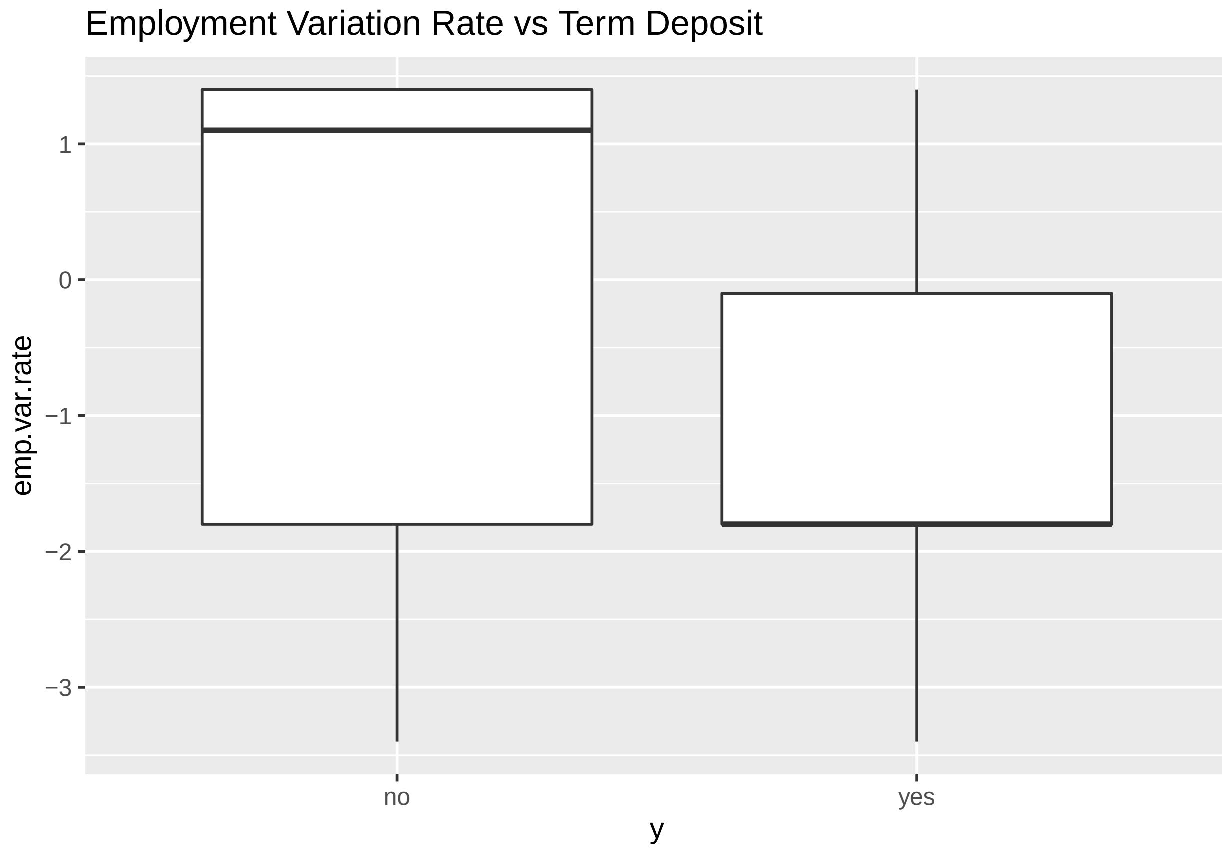
The following chart shows a significant correlation between clients with prior marketing campaign success and opening a term deposit. This variable should be included in the model.

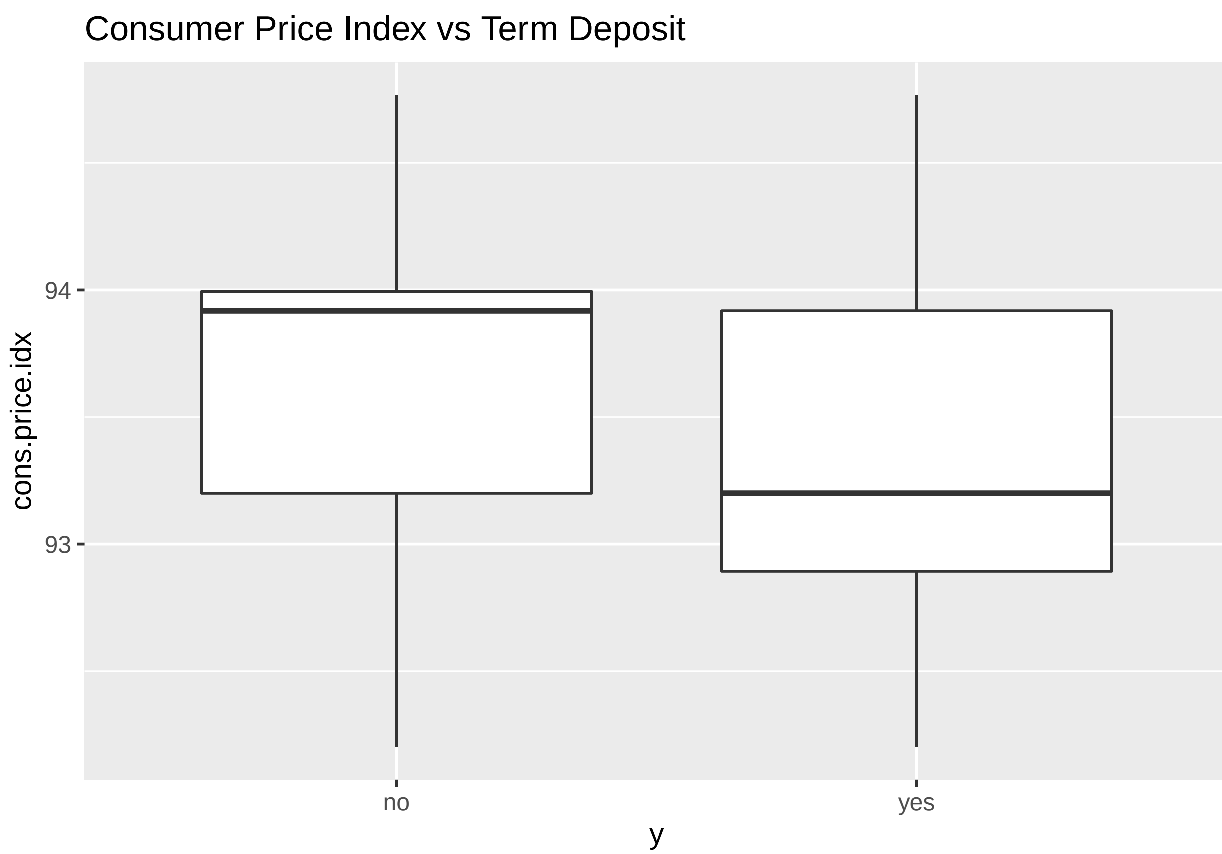
```
##        status    no  yes      perc total
## 1     failure  2884  486 14.421365  3370
## 2 nonexistent 25979 2481  8.717498 28460
## 3     success   375  745 66.517857  1120
```

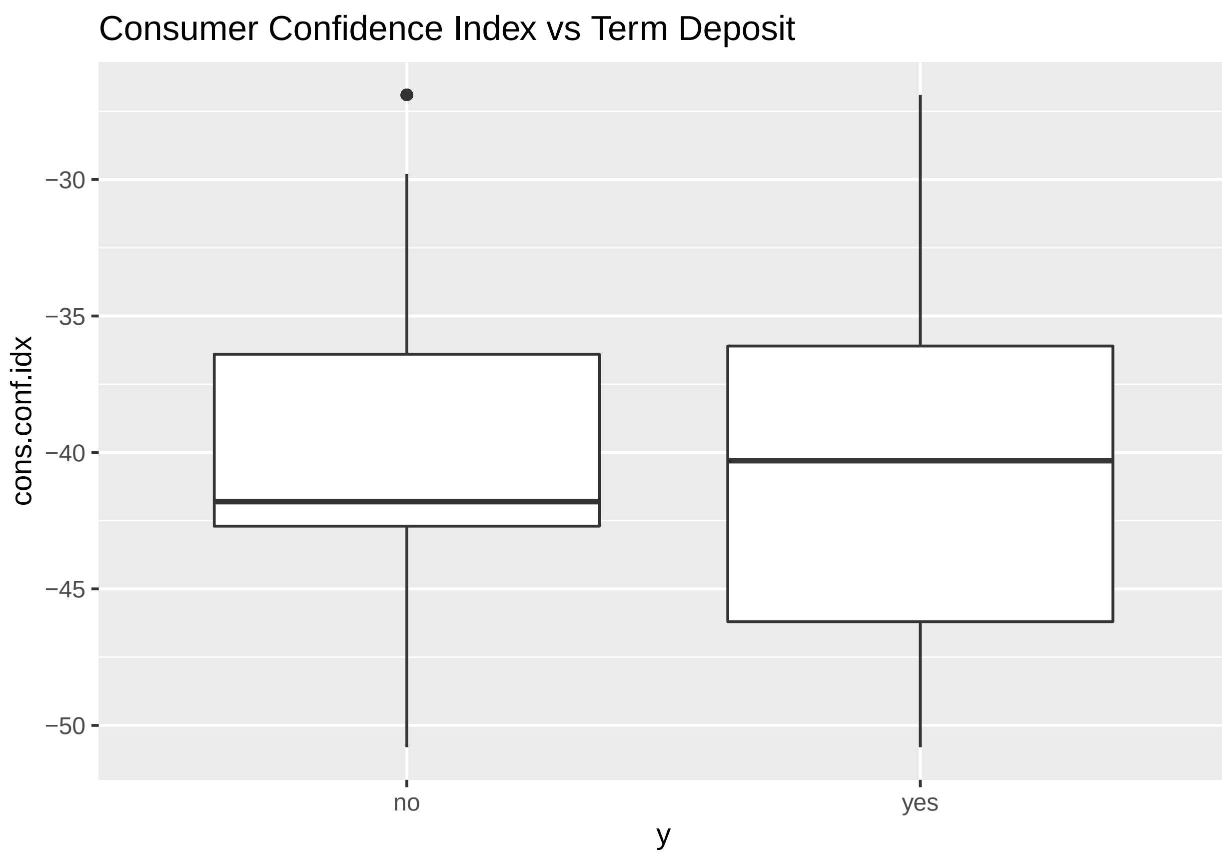


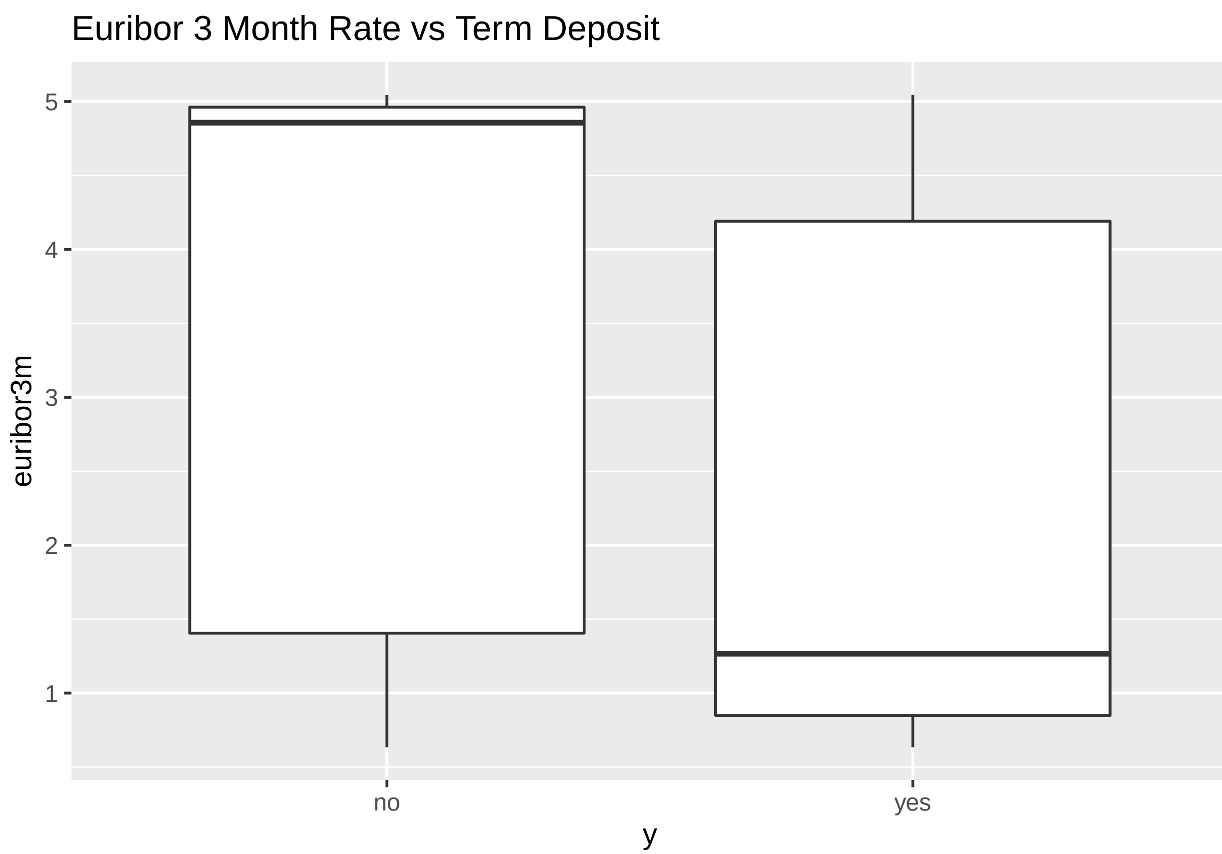

### 6.4 Socioeconomic Data Feature Analysis

As you can see from the following box and whisker plots, there is not much variation in the data among the socioeconomic data features and term deposits.


Employment Variation Rate vs Term Deposit
emp.var.rate
1
0
−1
−2
−3
no
yes
y


Consumer Price Index vs Term Deposit
94
93
cons.price.idx
no
yes
y


Consumer Confidence Index vs Term Deposit
−30
−35
−40
−45
−50
cons.conf.idx
no
yes
y


Euribor 3 Month Rate vs Term Deposit
euribor3m
5
4
3
2
1
no
yes
y


**One variable that jumps out to me is the Euribor 3 Month Rate. The median values are at opposite ends and may be considered in our model.**

# 7 Feature Selection & Engineering

## 7.1 Selecting the Important Variables

Based on the prior exploratory analysis, I was able to remove 9 explanatory variables from both the training and test data sets.

Here are the remaining variables and what needs to be done to ready for algorithm training:

1) Age – currently an integer variable which we will use as a categorical variable – also need to classify all ages over 86 as one variable to ensure that this variable accounts for all age groups.
2) Job – must convert to factor
3) Education – must convert to factor
4) Default – must convert to factor
5) Contact – must convert to factor
6) Month – must convert to factor
7) Duration – integer value in seconds. For some models will need to apply scaling and normalization
8) Campaign – integer value. For some models will need to apply scaling and normalization
9) Pdays – currently an integer that needs to be converted to a factor variable with levels displayed in exploratory analysis
10) Poutcome – convert to factor

11) Euribor3m – integer value. For some models will need to apply scaling and normalization.

These changes should be applied on both the train and test data sets.

```
## Train Set
train_set_mini <- train_set %>%
     select(age, job, education, default, contact, month, duration, campaign, pdays, poutcome, euribor3m
## Test Set
test_set_mini <- test_set %>%
  select(age, job, education, default, contact, month, duration, campaign, pdays, poutcome, euribor3m, y
```

## 7.2 Feature Engineering

### 7.2.1 Converting to Categorical Data

Here is the code for converting each of our character vectors to the appropriate factor.

```
## basic conversion to categorical data
train_set_mini$job <- as.factor(train_set_mini$job)
train_set_mini$education <- as.factor(train_set_mini$education)
train_set_mini$default <- as.factor(train_set_mini$default)
train_set_mini$contact <- as.factor(train_set_mini$contact)
train_set_mini$month <- as.factor(train_set_mini$month)
train_set_mini$poutcome <- as.factor(train_set_mini$poutcome)

test_set_mini$job <- as.factor(test_set_mini$job)
test_set_mini$education <- as.factor(test_set_mini$education)
test_set_mini$default <- as.factor(test_set_mini$default)
test_set_mini$contact <- as.factor(test_set_mini$contact)
test_set_mini$month <- as.factor(test_set_mini$month)
test_set_mini$poutcome <- as.factor(test_set_mini$poutcome)
```

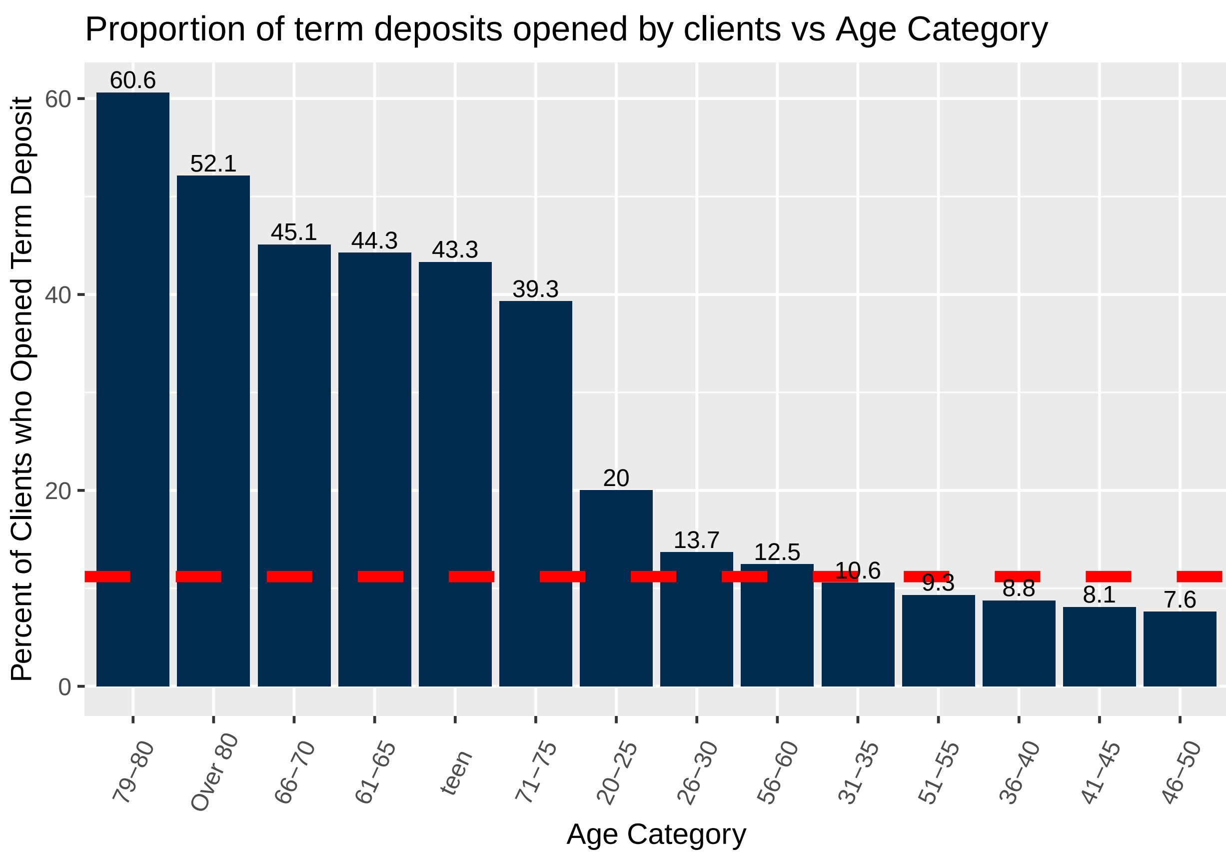
### 7.2.2 Engineering Age Category

Three problems occurred due to specifying age as a category for every possible age in the training set:

1) Some of the older clients in the 80+ category were not represented in the test set which would cause errors because our model had not seen those values before.
2) There are much fewer older and younger clients than clients between 30 and 50, which could cause the training data to be over fit to specific ages. For example, only one 91 year old was in the training set who did not subscribe vs one 94 year old who did subscribe to the term deposit – categorizing by unique age would most likely overweight the likelihood that a 91 year old would not subscribe and a 94 year old will subscribe even though there isn't much difference in age.
3) By reducing the number of age categories from each unique age to age groups, I am able to reduce the number of factor levels evaluated in each model. By reducing from 76 categories for age to 14 age categories, the models should run faster.

```
##      status   no yes      perc total
## 1     20-25 1026 257 20.031177  1283
## 2     26-30 3943 625 13.682137  4568
```

```
## 3     31-35 6504 769 10.573353  7273
## 4     36-40 5330 512  8.764122  5842
## 5     41-45 4097 361  8.097802  4458
## 6     46-50 3438 283  7.605482  3721
## 7     51-55 2588 266  9.320252  2854
## 8     56-60 1882 268 12.465116  2150
## 9     61-65  136 108 44.262295   244
## 10    66-70   90  74 45.121951   164
## 11    71-75   88  57 39.310345   145
## 12    79-80   37  57 60.638298    94
## 13 Over 80   45  49 52.127660    94
## 14     teen   34  26 43.333333    60
```




Additionally, when comparing the new age category variable to the unique ages we still see a significant amount of variability explained for each age segment while reducing the likelihood of overfitting.

#### 7.2.3 Engineering pdays

Here is the code to convert pdays into a categorical variable:

```
### PDAYS
train_set_mini <- train_set_mini %>%
  mutate(pdays_category =
      ifelse(pdays <= 5, "5 days or less",
```

```
      ifelse(pdays >5 & pdays <= 10, "6 - 10 days",
       ifelse(pdays >10 & pdays <= 15, "11 - 15 days",
        ifelse(pdays >15 & pdays <= 35, "15 - 35 days", "Not Contacted")))))

### apply to test set
test_set_mini <- test_set_mini %>%
  mutate(pdays_category =
    ifelse(pdays <= 5, "5 days or less",
     ifelse(pdays >5 & pdays <= 10, "6 - 10 days",
      ifelse(pdays >10 & pdays <= 15, "11 - 15 days",
       ifelse(pdays >15 & pdays <= 35, "15 - 35 days", "Not Contacted")))))
```

#### 7.2.4 Convert age category and pdays category to factors and remove age and pdays from both train and test mini data sets

```
## Convert age and pdays categories to factors on both train and test mini data sets
train_set_mini$age_cat <- as.factor(train_set_mini$age_cat)
test_set_mini$age_cat <- as.factor(test_set_mini$age_cat)
train_set_mini$pdays_category <- as.factor(train_set_mini$pdays_category)
test_set_mini$pdays_category <- as.factor(test_set_mini$pdays_category)

## Remove age and pdays from both train and test data sets
train_set_mini <- train_set_mini[,-c(1,9)]
test_set_mini <- test_set_mini[,-c(1,9)]
```

### 7.3 Normalizing numeric features

Since some of the algorithms I plan to use are distance-based and apply gradient descent as an optimization technique, it is essential that I scale the numeric features so that each feature is on a similar scale.

#### 7.3.1 Normalization

The approach I plan to use for feature scaling is min-max normalization. Here is the equation for min-max normalization:

$$X_{new} = (X \breve{} X_{min})/(X_{max} \breve{} X_{min})$$

This formula will scale each feature to be a value between 0 and 1. The three features requiring normalization

1) Duration
2) Campaign
3) Euribor3m

Here is the normalization code:

```
## Normalization of Numeric Features
min_duration <- min(train_set_mini$duration)
max_duration <- max(train_set_mini$duration)
min_campaign <- min(train_set_mini$campaign)
max_campaign <- max(train_set_mini$campaign)
min_euribor3m <- min(train_set_mini$euribor3m)
```

```r
max_euribor3m <- max(train_set_mini$euribor3m)

train_set_mini <- train_set_mini %>%
  mutate(duration_norm = (duration - min_duration)/(max_duration - min_duration),
         campaign_norm = (campaign - min_campaign)/(max_campaign - min_campaign),
         euribor3m_norm = (euribor3m - min_euribor3m)/(max_euribor3m - min_euribor3m))

test_set_mini <- test_set_mini %>%
  mutate(duration_norm = (duration - min_duration)/(max_duration - min_duration),
         campaign_norm = (campaign - min_campaign)/(max_campaign - min_campaign),
         euribor3m_norm = (euribor3m - min_euribor3m)/(max_euribor3m - min_euribor3m))

train_set_mini <- train_set_mini[,-c(6,7,9)]
test_set_mini <- test_set_mini[,-c(6,7,9)]
```

# 8 Model Evaluation

## 8.1 Logistic Regression

### 8.1.1 Training the Logistic Regression

Since I've already gone through the process of feature selection, engineering, and normalization, the training data is ready to teach our logistic regression. Here is the code:

```r
## Logistic Regression
logistic.train <- glm(y ~ ., data = train_set_mini, family = "binomial")
```

Additionally, we can get greater insight on the quality of our predictor variables by summarizing the logistic regression model.

```r
summary(logistic.train)
```

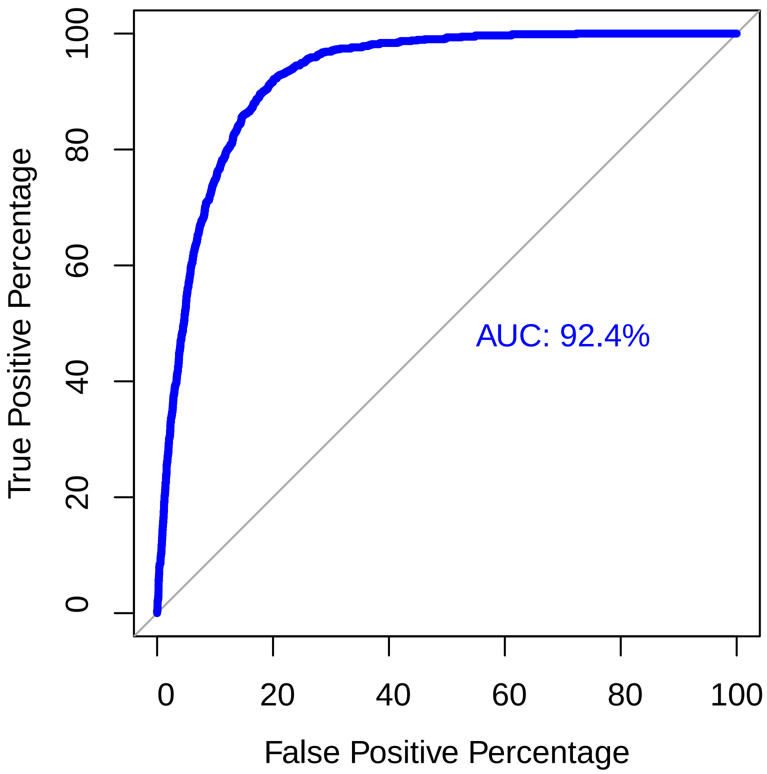
```
##
## Call:
## glm(formula = y ~ ., family = "binomial", data = train_set_mini)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -6.1044  -0.3146  -0.1904  -0.1198   3.3472
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -1.56675    0.32028  -4.892 9.99e-07 ***
## jobblue-collar                -0.29882    0.08938  -3.343 0.000828 ***
## jobentrepreneur               -0.16119    0.13567  -1.188 0.234803
## jobhousemaid                  -0.21565    0.17117  -1.260 0.207737
## jobmanagement                 -0.08126    0.09490  -0.856 0.391847
## jobretired                    -0.03957    0.13867  -0.285 0.775349
## jobself-employed              -0.12940    0.12846  -1.007 0.313806
## jobservices                   -0.19037    0.09616  -1.980 0.047728 *
## jobstudent                     0.05975    0.13287   0.450 0.652958
```

```
## jobtechnician                 -0.06690    0.07889  -0.848 0.396455    
## jobunemployed                 -0.00407    0.14329  -0.028 0.977339    
## jobunknown                    -0.23869    0.28137  -0.848 0.396250    
## educationbasic.6y              0.17264    0.13689   1.261 0.207240    
## educationbasic.9y              0.01878    0.10792   0.174 0.861877    
## educationhigh.school           0.05327    0.10433   0.511 0.609633    
## educationilliterate            1.76960    0.81256   2.178 0.029419 *  
## educationprofessional.course   0.15087    0.11498   1.312 0.189491    
## educationuniversity.degree     0.26499    0.10402   2.547 0.010853 *  
## educationunknown               0.30642    0.13430   2.282 0.022509 *  
## defaultunknown                -0.29592    0.07579  -3.904 9.45e-05 ***
## defaultyes                    -7.28129  113.48849  -0.064 0.948844    
## contacttelephone              -0.05047    0.06793  -0.743 0.457521    
## monthaug                       0.57423    0.09370   6.129 8.86e-10 ***
## monthdec                       0.70349    0.21032   3.345 0.000823 ***
## monthjul                       0.70396    0.09873   7.130 1.00e-12 ***
## monthjun                       0.64631    0.09797   6.597 4.19e-11 ***
## monthmar                       1.55714    0.12980  11.997  < 2e-16 ***
## monthmay                      -0.61196    0.08166  -7.494 6.70e-14 ***
## monthnov                       0.32635    0.10305   3.167 0.001541 ** 
## monthoct                       0.79483    0.12330   6.446 1.15e-10 ***
## monthsep                       0.64836    0.13046   4.970 6.71e-07 ***
## poutcomenonexistent            0.45074    0.07025   6.416 1.40e-10 ***
## poutcomesuccess                0.77915    0.23963   3.251 0.001148 ** 
## age_cat26-30                  -0.13850    0.11151  -1.242 0.214222    
## age_cat31-35                  -0.25965    0.11189  -2.321 0.020306 *  
## age_cat36-40                  -0.41890    0.11721  -3.574 0.000352 ***
## age_cat41-45                  -0.44883    0.12406  -3.618 0.000297 ***
## age_cat46-50                  -0.42722    0.12917  -3.307 0.000942 ***
## age_cat51-55                  -0.23278    0.13195  -1.764 0.077702 .  
## age_cat56-60                  -0.01746    0.13919  -0.125 0.900189    
## age_cat61-65                   0.22684    0.19964   1.136 0.255847    
## age_cat66-70                   0.20742    0.24077   0.862 0.388955    
## age_cat71-75                   0.10200    0.25441   0.401 0.688474    
## age_cat79-80                   0.76111    0.29128   2.613 0.008976 ** 
## age_catOver 80                 0.59517    0.29561   2.013 0.044072 *  
## age_catteen                    0.08854    0.34836   0.254 0.799361    
## pdays_category15 - 35 days    -0.43574    0.46851  -0.930 0.352334    
## pdays_category5 days or less   0.14707    0.24251   0.606 0.544206    
## pdays_category6 - 10 days      0.16259    0.24224   0.671 0.502114    
## pdays_categoryNot Contacted   -1.11220    0.28571  -3.893 9.91e-05 ***
## duration_norm                 23.14648    0.41123  56.286  < 2e-16 ***
## campaign_norm                 -3.56449    0.73281  -4.864 1.15e-06 ***
## euribor3m_norm                -2.79124    0.08409 -33.195  < 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: 23199  on 32949  degrees of freedom
## Residual deviance: 13736  on 32897  degrees of freedom
## AIC: 13842
## 
## Number of Fisher Scoring iterations: 10
```

It genuinely seems that the predictor variables included in this model are appropriate for the algorithm. In fact each of the categorical variables and all three of the numeric variables have low p-values and have high slopes associated with the variable.

#### 8.1.2 Evaluating AUC

```
## Setting levels: control = no, case = yes

## Setting direction: controls < cases
```




```
##
## Call:
## roc.formula(formula = logistic.predict$y ~ logistic.predict$y_hat,     plot = TRUE, legacy.axes = TRU
##
## Data: logistic.predict$y_hat in 7310 controls (logistic.predict$y no) < 928 cases (logistic.predict$y
## Area under the curve: 92.41%
```

The logistic regression model provided an AUC of 92.4%. This makes it a very good model, especially in terms of balancing Sensitivity and Specificity.

### 8.2 KNN Algorithm

The KNN algorithm uses distance to classify variables into groups. Since it's primary use is for classification, I thought it would be an appropriate algorithm for this data set.

The KNN algorithm requires categorical variables to be stored as numerical values. So the following code converts each factor to a numeric. In addition, I've already scaled the numeric variables.

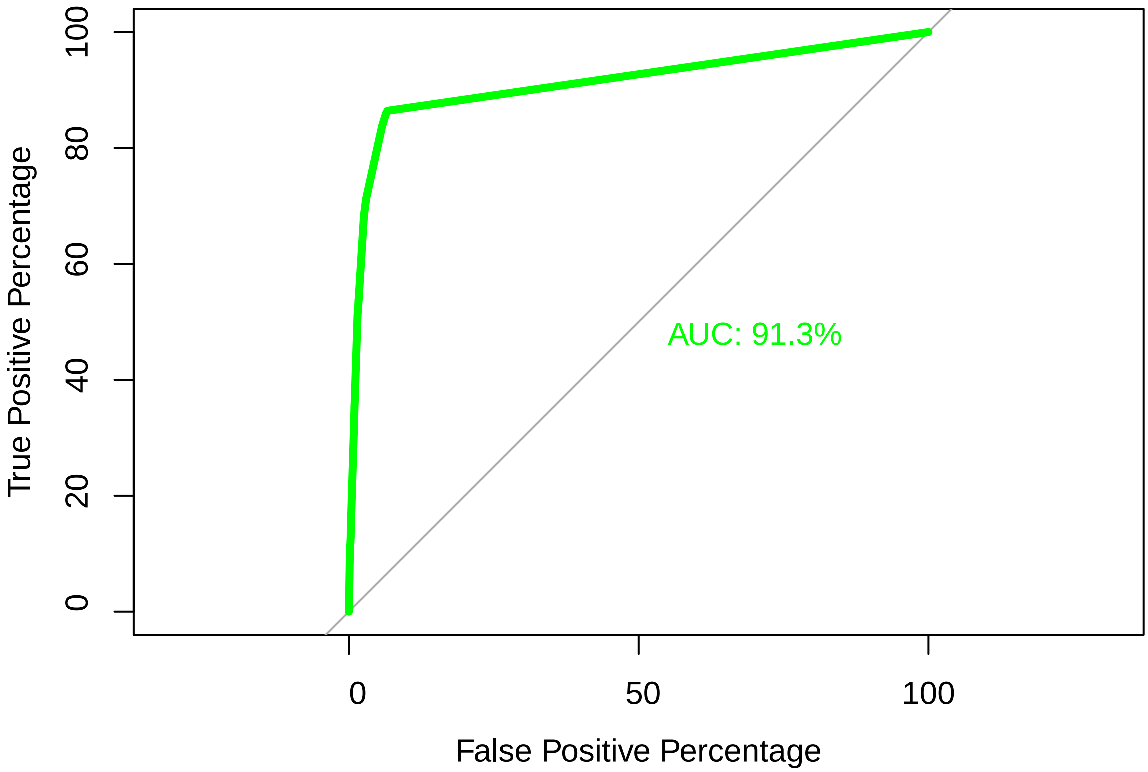
```
### first we must convert all of our factor variables to numeric variables for the knn model
knn_train_set_mini <- train_set_mini
knn_test_set_mini <- test_set_mini

knn_train_set_mini$job <- as.numeric(knn_train_set_mini$job)
knn_train_set_mini$education <- as.numeric(knn_train_set_mini$education)
knn_train_set_mini$default <- as.numeric(knn_train_set_mini$default)
knn_train_set_mini$contact <- as.numeric(knn_train_set_mini$contact)
knn_train_set_mini$month <- as.numeric(knn_train_set_mini$month)
knn_train_set_mini$poutcome <- as.numeric(knn_train_set_mini$poutcome)
knn_train_set_mini$y <- as.numeric(knn_train_set_mini$y)
knn_train_set_mini$age_cat <- as.numeric(knn_train_set_mini$age_cat)
knn_train_set_mini$pdays_category <- as.numeric(knn_train_set_mini$pdays_category)

knn_test_set_mini$job <- as.numeric(knn_test_set_mini$job)
knn_test_set_mini$education <- as.numeric(knn_test_set_mini$education)
knn_test_set_mini$default <- as.numeric(knn_test_set_mini$default)
knn_test_set_mini$contact <- as.numeric(knn_test_set_mini$contact)
knn_test_set_mini$month <- as.numeric(knn_test_set_mini$month)
knn_test_set_mini$poutcome <- as.numeric(knn_test_set_mini$poutcome)
knn_test_set_mini$y <- as.numeric(knn_test_set_mini$y)
knn_test_set_mini$age_cat <- as.numeric(knn_test_set_mini$age_cat)
knn_test_set_mini$pdays_category <- as.numeric(knn_test_set_mini$pdays_category)
```

Finally, we convert the knn model predictions to probabilities that can be interpreted by the roc function.

```
### Run and Plot AUC for KNN Algorithm
knn_model <- knn(train = knn_train_set_mini, test = knn_test_set_mini, cl = knn_train_set_mini$y, k = 1
knn_prob <- attr(knn_model, "prob")
knn_prob <- 2*ifelse(knn_model == "-1", 1-knn_prob, knn_prob) - 1
```

### 8.2.1 Evaluating AUC

```
## Setting levels: control = 1, case = 2

## Setting direction: controls > cases
```

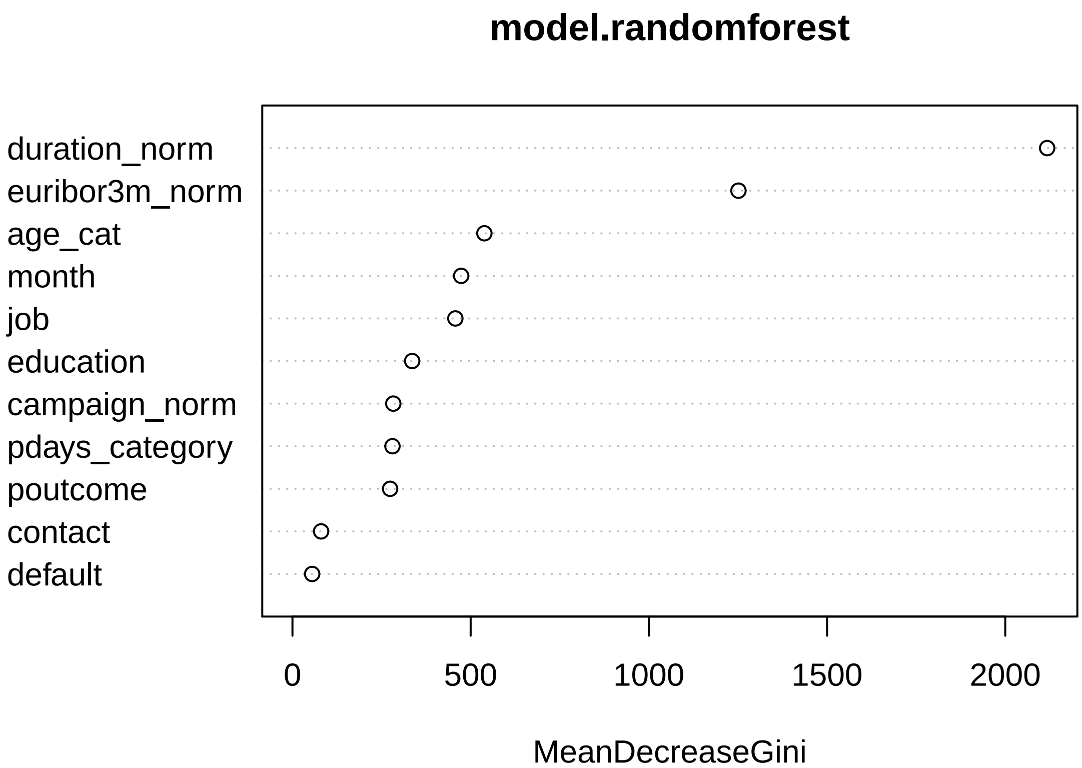



```
##
## Call:
## roc.formula(formula = knn_test_set_mini$y ~ knn_prob, plot = TRUE,      legacy.axes = TRUE, percent =
##
## Data: knn_prob in 7310 controls (knn_test_set_mini$y 1) > 928 cases (knn_test_set_mini$y 2).
## Area under the curve: 91.26%
```

The AUC for the KNN clustering algorithm is 91.3%. While very good, it performs slightly worse than the logistic regression algorithm.

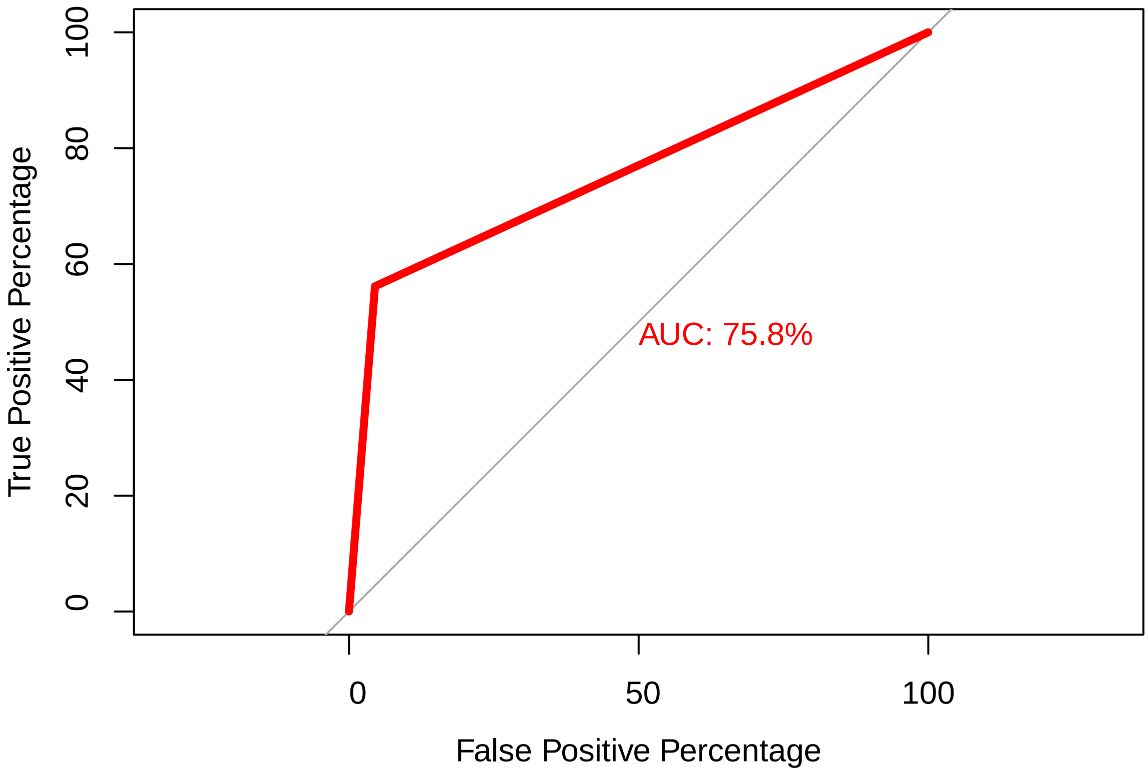
### 8.3 Random Forest

One of the best features of the random forest model is that it provides a unique list of variable importance to the model. Here is the plot of the results:




Here is the AUC for the Random Forest model. As you can see the Random Forest has an AUC of 75.4% and is the worst performing algorithm. We can tune the parameters of the Random Forest to improve this score, but logistic regression has a great AUC and I am happy with that algorithm.

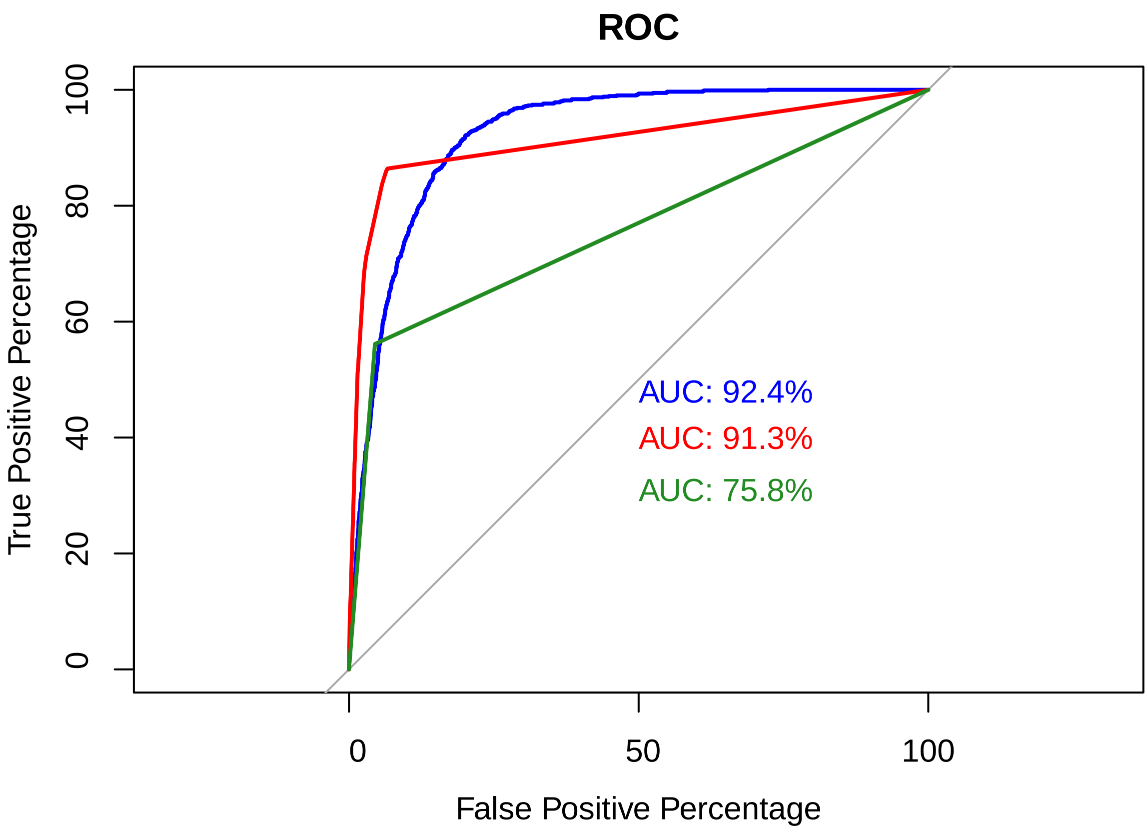
```
## Setting levels: control = no, case = yes

## Setting direction: controls < cases
```




```
##
## Call:
## roc.formula(formula = rf_test_set_mini$y ~ rf.predict, plot = TRUE,     legacy.axes = TRUE, percent =
##
## Data: rf.predict in 7310 controls (rf_test_set_mini$y no) < 928 cases (rf_test_set_mini$y yes).
## Area under the curve: 75.83%
```

## 8.4 Comparing each of the AUCs on one graph.

I've drawn and printed the AUC for each of the models I've run. Here are the results.




Note: Blue = Logistic Regression, Red = KNN Clustering, Green = Random Forest

While this isn't anything new, it is always good to visualize the AUC as compared to the various models. As you can see, the logistic regression provides us with the greatest AUC and should be selected as the final model.

## 9 Conclusion: Logistic Regression for the Win

Oftentimes, feature engineering and variable selection are the most important traits of a machine learning algorithm. They are always the most impactful components of less complex models like logistic regression. This is the case for the bank marketing data set.

***What the marketing team knows now***

First, the marketing team has a way of classifying their clients for a marketing campaign and will help them focus their efforts on the right audience. This will provide the team with better results moving forward while allowing them to reduce resources allocated to the campaign (if they call a more targeted list of clients with fewer campaign participants, they don't need as many resources on the campaign).

Second, we learned a lot about the demographic, campaign, additional attributes, and socioeconomic features that play a role in determining whether or not a client will subscribe to a term deposit. Here are some of the facts:

- **Call Length Matters** - The most important attribute associated with effectively predicting term deposit subscriptions was the duration of the last call with the client. Clients that spent several minutes on the phone with a bank representative were much more likely to subscribe.

- **Age Matters** – Opening a term deposit is much more prevalent among young people. While clients aged, 30 – 50 are much less likely to open a term deposit. This is probably due to the amount of expenses that these groups have in relation to other age groups.

- **Seasonality to this campaign** – Clients were much less likely to open term deposits during the summer months. Maybe this is due to vacation expenses, but the winter and early spring months had a far higher term deposit success rate.

- **Unknown Default Status prevents Term Deposits** – Overall, individuals with unknown default statuses and known defaults (even though known defaults is very rare) are much less likely to open a term deposit.

- **Clients engaged in previous marketing campaigns are more likely to sign up for new campaigns**

- **Lastly, when Euribor rates are lower, clients are likely to open a term deposit.**