Bank Marketing Case Study

Libraries used

install.packages("xfun") install.packages(c("rmarkdown","knitr","rstudioapi","tidyverse"), dependencies = TRUE) packageVersion("xfun") tinytex::install_tinytex()

```
library(MASS)
library(readr)
```

```
## Warning: package 'readr' was built under R version 4.4.2
```

```
library(ggplot2)
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.4.3
```

```
## Warning: package 'dplyr' was built under R version 4.4.2
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v stringr   1.5.1
## v forcats   1.0.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.4.3
```

```
## corrplot 0.95 loaded
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.4.2
```

```
## Loading required package: carData
##
## Attaching package: 'car'
```

```
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.4.2
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 4.4.2
```

```
## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
##
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 4.4.2
```

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

when loading the data initially i noticed that the data wasn't being separated correctly. So in order to correct that I had to

include (, sep = ";")

```r
df <- read.csv("C:/Users/jorda/Downloads/bank-additional-full.csv", sep = ";")

str(df)
```

```
## 'data.frame':    41188 obs. of  21 variables:
##  $ age           : int  56 57 37 40 56 45 59 41 24 25 ...
##  $ job           : chr  "housemaid" "services" "services" "admin." ...
##  $ marital       : chr  "married" "married" "married" "married" ...
##  $ education     : chr  "basic.4y" "high.school" "high.school" "basic.6y" ...
##  $ default       : chr  "no" "unknown" "no" "no" ...
##  $ housing       : chr  "no" "no" "yes" "no" ...
##  $ loan          : chr  "no" "no" "no" "no" ...
##  $ contact       : chr  "telephone" "telephone" "telephone" "telephone" ...
##  $ month         : chr  "may" "may" "may" "may" ...
##  $ day_of_week   : chr  "mon" "mon" "mon" "mon" ...
##  $ 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      : chr  "nonexistent" "nonexistent" "nonexistent" "nonexistent" ...
##  $ 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             : chr  "no" "no" "no" "no" ...
```

##looking at the dat it seems that I need to change characters to factors in order to run a logistic regression model

```r
df = df %>%
  mutate(across(where(is.character), as.factor))
str(df)
```

```
## '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 ...
```

Characters are no Factors

checking levels of my target variable

```r
levels(df$y)
```

```
## [1] "no"  "yes"
```

#two levels, I want to check how many "yes," and "no," observations there are in my dataset.

```r
summary(df$y)
```

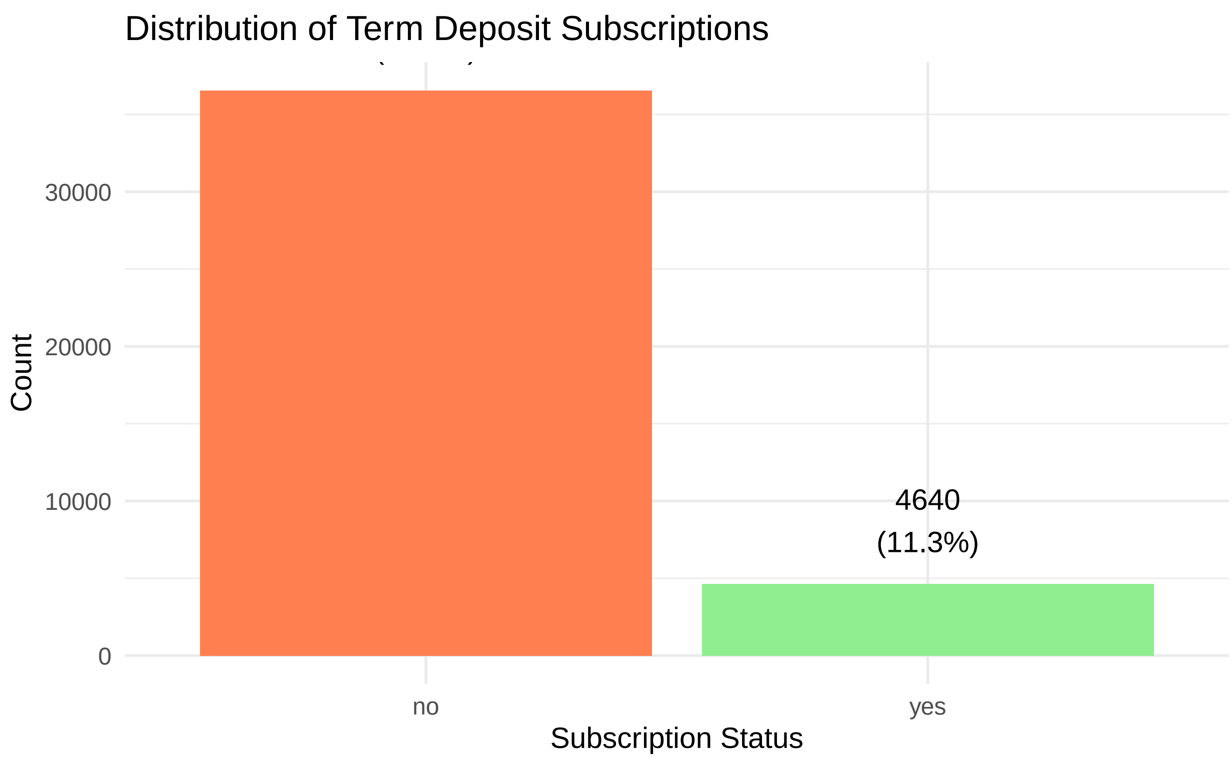
```
##    no   yes
## 36548  4640
```

```r
df %>%
  count(y) %>%
  mutate(percentage = round(n/sum(n)*100, 1)) %>%
  print()
```

```
##     y     n percentage
## 1  no 36548       88.7
## 2 yes  4640       11.3
```

Visualization 1: Target Variable Distribution

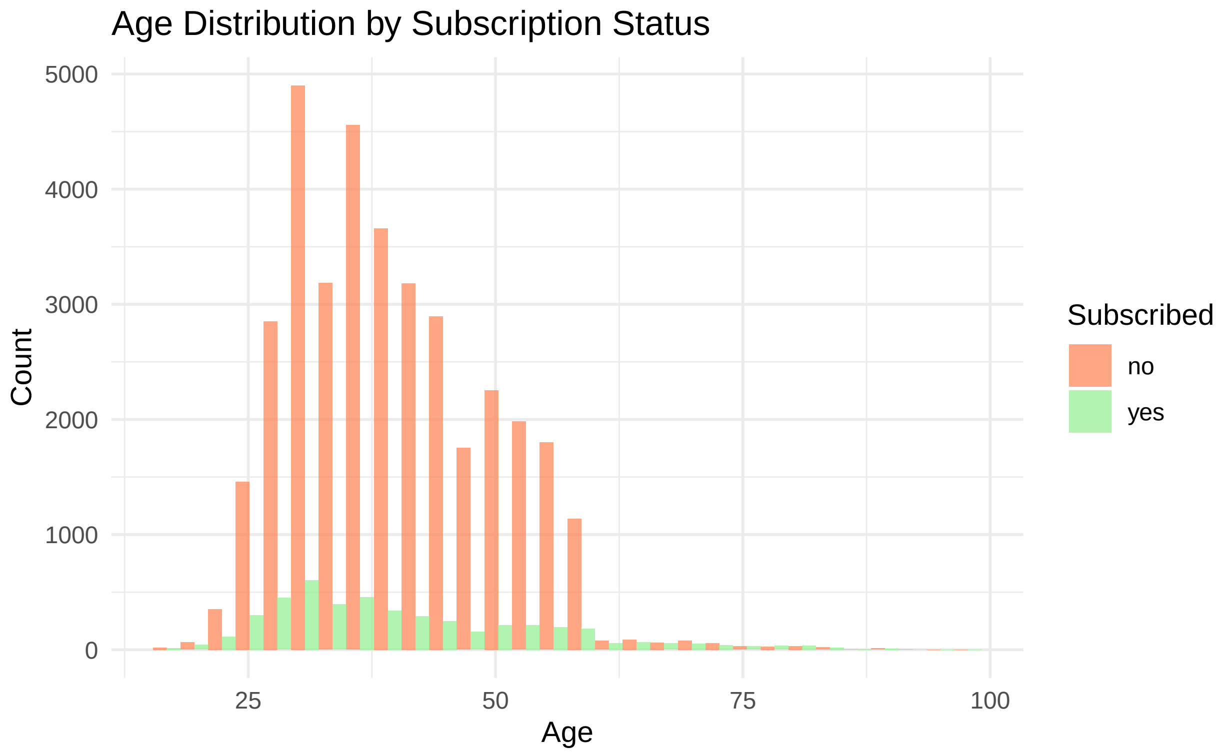
```r
colors_palette <- c("#E74C3C", "#3498DB", "#2ECC71", "#F39C12", "#9B59B6")
target_plot <- df %>%
  count(y) %>%
  mutate(percentage = n/sum(n) * 100) %>%
  ggplot(aes(x = y, y = n, fill = y)) +
  geom_col() +
  geom_text(aes(label = paste0(n, "\n(", round(percentage, 1), "%)")), vjust = -0.5) +
  scale_fill_manual(values = c("no" = "coral", "yes" = "lightgreen")) +
  labs(title = "Distribution of Term Deposit Subscriptions",
       x = "Subscription Status", y = "Count") +
  theme_minimal() +
  theme(legend.position = "none")
print(target_plot)
```



as seen in the summary above, the target variable contains far more "no's" than "yes"' in our dataset. I will address this at a later time but it's important to note now.

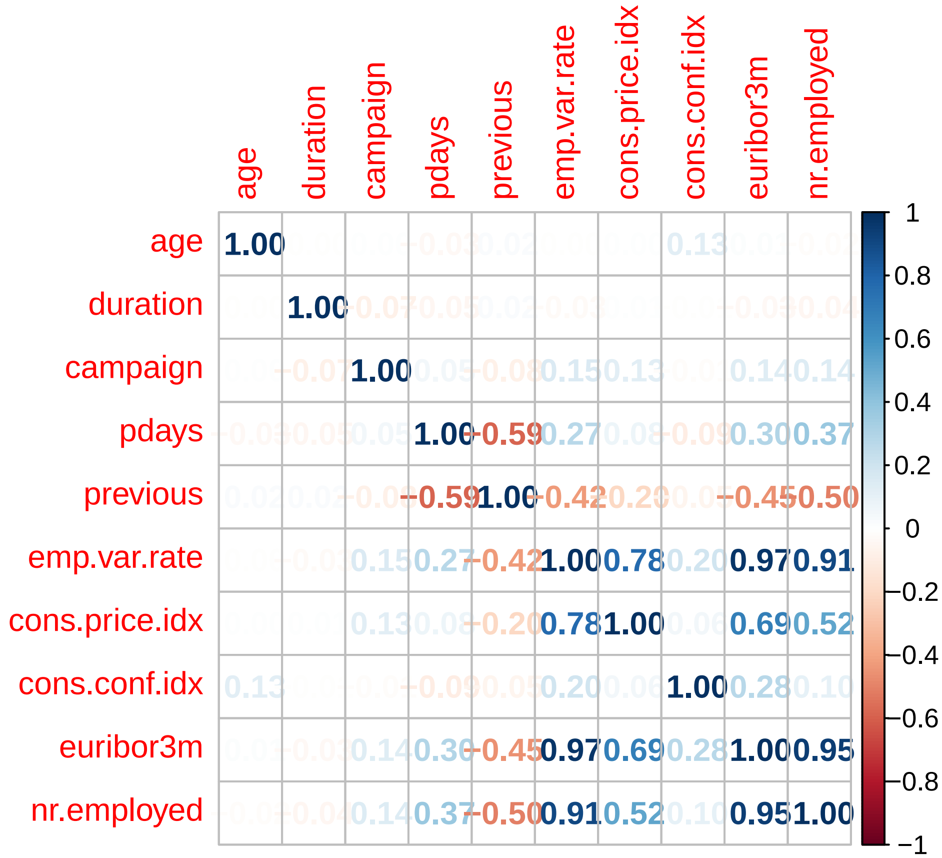
Visualization 2: Age Distribution by Outcome

```r
age_plot <- ggplot(df, aes(x = age, fill = y)) +
  geom_histogram(bins = 30, position = "dodge", alpha = 0.7) +
  scale_fill_manual(values = c("no" = "coral", "yes" = "lightgreen")) +
  labs(title = "Age Distribution by Subscription Status",
       x = "Age", y = "Count", fill = "Subscribed") +
  theme_minimal()
print(age_plot)
```



I want to check for multicollinearity in my numerical data.

```
df_num = dplyr::select_if(df, is.numeric)
M = cor(df_num)
corrplot(M, method = 'number')
```



#there appears to be high multicollinearity with the following variables: #euribor3m: euribor 3 month rate - daily indicator (numeric) - daily short-term interest rate # emp.var.rate: employment variation rate - quarterly indicator (numeric) - measures change in employment quartely #nr.employed: number of employees - quarterly indicator (numeric) - Captures quartely size of the work force #as a group we decided to remove emp.var.rate and nr.employed since they essentially measure the same thing #per instructions we are removing duration variable and the default variable

```
df = dplyr::select(df, - emp.var.rate)
df = dplyr::select(df, - nr.employed)
df = dplyr::select(df, - duration)
df = dplyr::select(df, - default)
```

Visualization 3: Job Type Success Rates

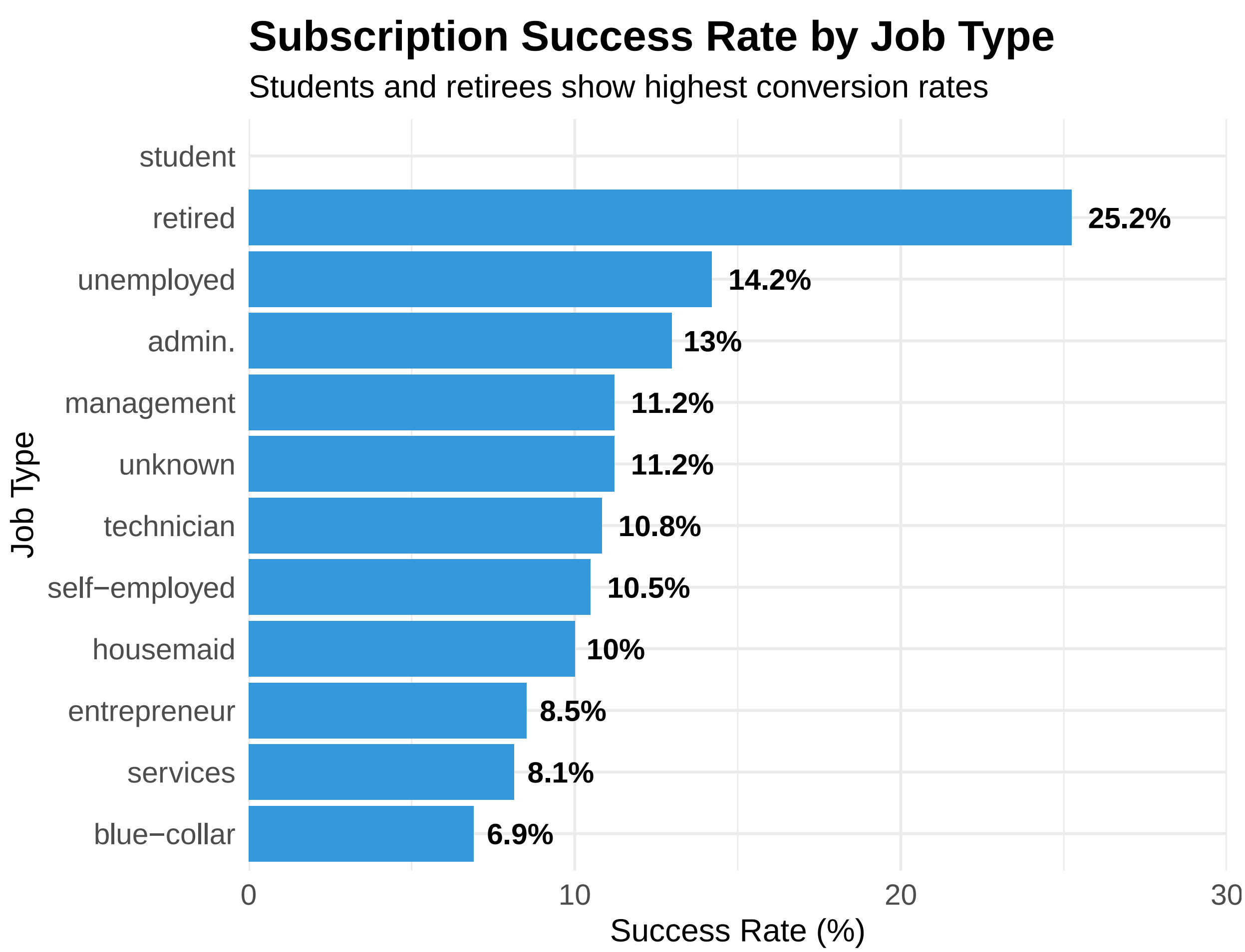
```
job_success <- df %>%
  group_by(job) %>%
  summarise(
    total = n(),
    subscribed = sum(y == "yes"),
    success_rate = (subscribed/total) * 100
  ) %>%
  arrange(desc(success_rate))

job_plot <- ggplot(job_success, aes(x = reorder(job, success_rate), y = success_rate)) +
  geom_col(fill = colors_palette[2]) +
```

```r
  geom_text(aes(label = paste0(round(success_rate, 1), "%")),
            hjust = -0.2, size = 4, fontface = "bold") +
  coord_flip() +
  labs(title = "Subscription Success Rate by Job Type",
       subtitle = "Students and retirees show highest conversion rates",
       x = "Job Type",
       y = "Success Rate (%)") +
  scale_y_continuous(limits = c(0, 30), expand = c(0, 0)) +
  theme_minimal() +
  theme(plot.title = element_text(size = 16, face = "bold"),
        plot.subtitle = element_text(size = 12),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 11))
print(job_plot)
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (‘geom_col()‘).
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (‘geom_text()‘).
```



#The pdays variable is interesting since this is measuring the number of days since last contact # looking at the unique values in pdays this seems to measure the days from 1 - 27, with 999 indicating client was not previously contacted.

```r
unique(df$pdays)
```

```
##  [1] 999   6   4   3   5   1   0  10   7   8   9  11   2  12  13  14  15  16  21
## [20]  17  18  22  25  26  19  27  20
```

```r
contacted <- df$pdays[df$pdays != 999] #checking max number of never contacted observations
summary(contacted)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   3.000   6.000   6.015   7.000  27.000
```

```r
levels(as.factor(df$pdays))
```

```
##  [1] "0"   "1"   "2"   "3"   "4"   "5"   "6"   "7"   "8"   "9"   "10"  "11"
## [13] "12"  "13"  "14"  "15"  "16"  "17"  "18"  "19"  "20"  "21"  "22"  "25"
## [25] "26"  "27"  "999"
```

I will also look into putting age variable into buckets and campaign for the number of times a client was contacted

```r
unique(df$age)
```

```
##  [1] 56 57 37 40 45 59 41 24 25 29 35 54 46 50 39 30 55 49 34 52 58 32 38 44 42
## [26] 60 53 47 51 48 33 31 43 36 28 27 26 22 23 20 21 61 19 18 70 66 76 67 73 88
## [51] 95 77 68 75 63 80 62 65 72 82 64 71 69 78 85 79 83 81 74 17 87 91 86 98 94
## [76] 84 92 89
```

```r
levels(as.factor(df$campaign))
```

```
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13" "14" "15"
## [16] "16" "17" "18" "19" "20" "21" "22" "23" "24" "25" "26" "27" "28" "29" "30"
## [31] "31" "32" "33" "34" "35" "37" "39" "40" "41" "42" "43" "56"
```

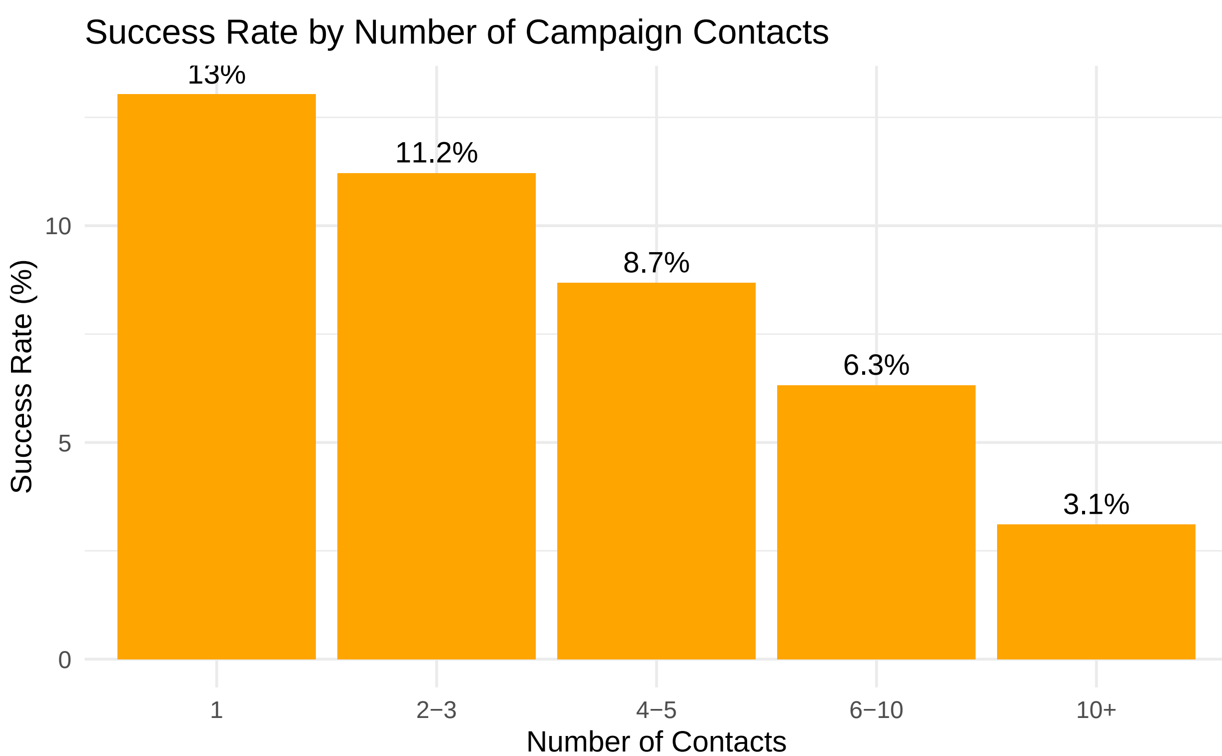
```r
summary(as.factor(df$campaign))
```

```
##     1     2     3     4     5     6     7     8     9    10    11    12    13
## 17642 10570  5341  2651  1599   979   629   400   283   225   177   125    92
##    14    15    16    17    18    19    20    21    22    23    24    25    26
##    69    51    51    58    33    26    30    24    17    16    15     8     8
##    27    28    29    30    31    32    33    34    35    37    39    40    41
##    11     8    10     7     7     4     4     3     5     1     1     2     1
##    42    43    56
##     2     2     1
```

Visualization 4: Campaign Frequency Impact

```
campaign_impact <- df %>%
  mutate(campaign_group = cut(campaign,
                              breaks = c(0, 1, 3, 5, 10, Inf),
                              labels = c("1", "2-3", "4-5", "6-10", "10+"))) %>%
  group_by(campaign_group) %>%
  summarise(
    count = n(),
    success_rate = mean(y == "yes") * 100
  )

campaign_plot <- ggplot(campaign_impact, aes(x = campaign_group, y = success_rate)) +
  geom_col(fill = "orange") +
  geom_text(aes(label = paste0(round(success_rate, 1), "%")), vjust = -0.5, size = 4) +
  labs(title = "Success Rate by Number of Campaign Contacts",
       x = "Number of Contacts", y = "Success Rate (%)") +
  theme_minimal()
print(campaign_plot)
```



#will create buckets for pdays in order to gather deeper insights into when they best time is to reach out to clients #to do this I will create a new variable and then remove the original pdays since we not use the numerical value in our modeling

```
df = df %>%
  mutate(pdays_bucket = case_when(
    pdays == 999 ~ "Never Contacted",
    pdays <= 7 ~ "1 Week",
```

```r
    pdays >7 & pdays <= 14 ~ "2 Weeks",
    pdays >14 ~ "3 Weeks or more",
    TRUE ~ "Other"
  ))
```

```r
df$pdays_bucket = as.factor(df$pdays_bucket) #seeting new column pdays_bucket to be factor
levels(df$pdays_bucket)
```

```
## [1] "1 Week"          "2 Weeks"         "3 Weeks or more" "Never Contacted"
```

#dropping original pdays column

```r
df = df %>% select(-pdays)
str(df)
```

```
## 'data.frame':    41188 obs. of  17 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 ...
##  $ 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 ...
##  $ campaign      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ 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 ...
##  $ 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 ...
##  $ y             : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ pdays_bucket  : Factor w/ 4 levels "1 Week","2 Weeks",..: 4 4 4 4 4 4 4 4 4 4 ...
```

#creating age bucket, setting as factor and then dropping original age variable

```r
df = df %>%
  mutate(age_bucket = case_when(
    age >= 18 & age <= 24 ~ "Young Adult",
    age >= 25 & age <= 35 ~ "Adult",
    age >= 36 & age <= 49 ~ "Older Adult",
    age >=50 ~ "Senior",
    TRUE ~ "Other"
  ))
```

```r
df$age_bucket = as.factor(df$age_bucket)
levels(df$age_bucket)
```

```
## [1] "Adult"       "Older Adult" "Other"       "Senior"      "Young Adult"
```

```r
df = df %>% select(-age)
str(df)
```

```
## 'data.frame':    41188 obs. of  17 variables:
##  $ 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 ...
##  $ 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 ...
##  $ campaign      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ 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 ...
##  $ 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 ...
##  $ y             : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ pdays_bucket  : Factor w/ 4 levels "1 Week","2 Weeks",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ age_bucket    : Factor w/ 5 levels "Adult","Older Adult",..: 4 4 2 2 4 2 4 2 5 1 ...
```

#creating campaign bucket, setting as factor and then dropping original age variable

```r
df = df %>%
  mutate(campaign_bucket = case_when(
    campaign <= 10 ~ "10 or less contacts",
    campaign >= 11 & campaign <= 20 ~ "11-20 contacts",
    campaign >= 21  & campaign <= 30 ~ "21-30 contacts",
    campaign >= 31 & campaign <= 40 ~ "31-40 contacts",
    campaign >=40 ~ "40+",
    TRUE ~ "Other"
  ))
```

```r
df$campaign_bucket = as.factor(df$campaign_bucket)
levels(df$campaign_bucket)
```

```
## [1] "10 or less contacts" "11-20 contacts"      "21-30 contacts"
## [4] "31-40 contacts"      "40+"
```

```r
df = df %>% select(-campaign)
str(df)
```

```
## 'data.frame':    41188 obs. of  17 variables:
##  $ 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 ...
##  $ 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 ...
```

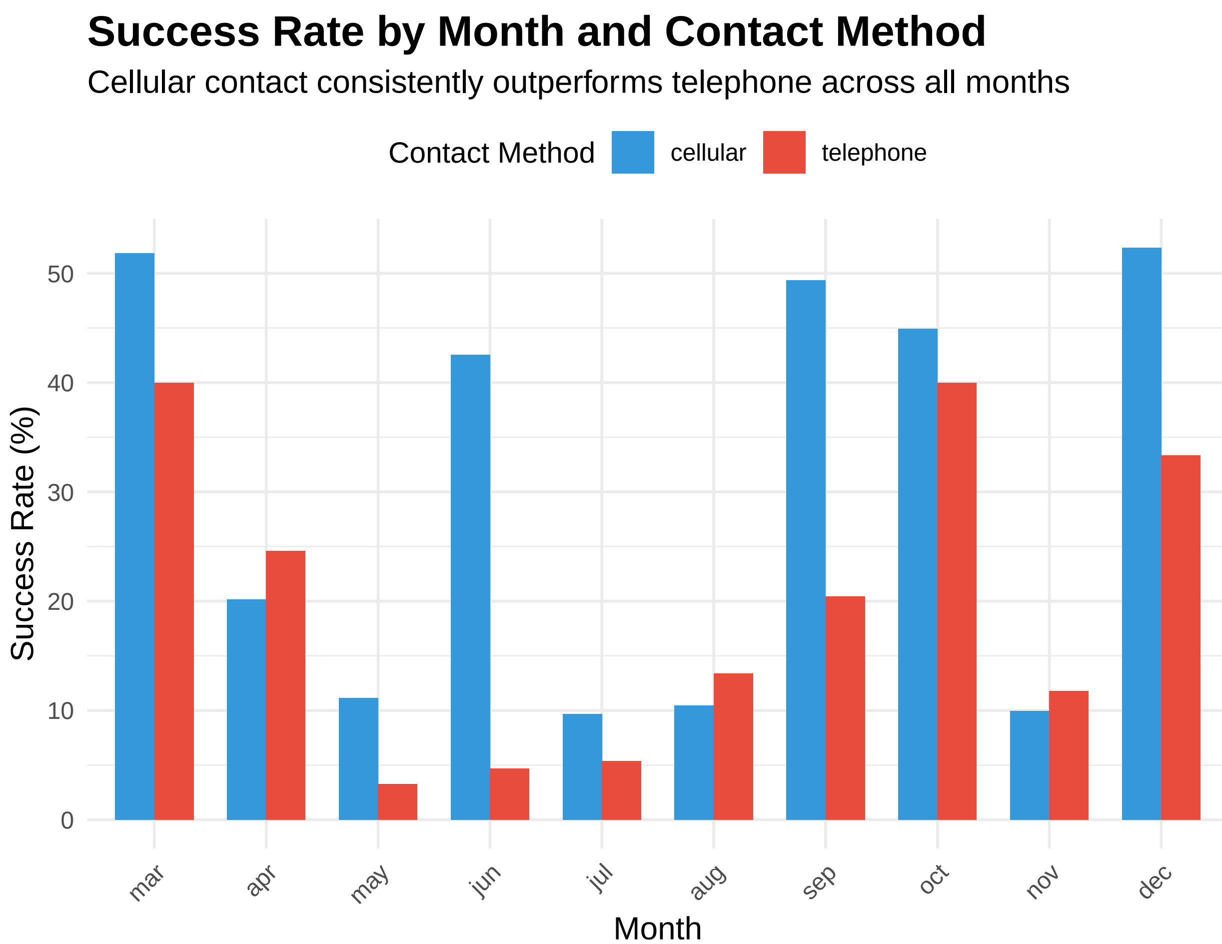
```
## $ day_of_week    : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2 2 2 2 2 2 2 2 ...
## $ previous       : int  0 0 0 0 0 0 0 0 0 0 ...
## $ poutcome       : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 2 2 2 2 2 ...
## $ cons.price.idx : num  94 94 94 94 94 ...
## $ cons.conf.idx  : 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 ...
## $ y              : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
## $ pdays_bucket   : Factor w/ 4 levels "1 Week","2 Weeks",..: 4 4 4 4 4 4 4 4 4 4 ...
## $ age_bucket     : Factor w/ 5 levels "Adult","Older Adult",..: 4 4 2 2 4 2 4 2 5 1 ...
## $ campaign_bucket: Factor w/ 5 levels "10 or less contacts",..: 1 1 1 1 1 1 1 1 1 1 ...
```

Visualization 5: Contact Method and Month Analysis

```r
contact_month <- df %>%
  group_by(month, contact) %>%
  summarise(
    success_rate = mean(y == "yes") * 100,
    .groups = 'drop'
  )

month_order <- c("jan", "feb", "mar", "apr", "may", "jun",
                 "jul", "aug", "sep", "oct", "nov", "dec")
contact_month$month <- factor(contact_month$month, levels = month_order)

contact_plot <- ggplot(contact_month, aes(x = month, y = success_rate, fill = contact)) +
  geom_col(position = "dodge", width = 0.7) +
  scale_fill_manual(values = c("cellular" = colors_palette[2],
                               "telephone" = colors_palette[1])) +
  labs(title = "Success Rate by Month and Contact Method",
       subtitle = "Cellular contact consistently outperforms telephone across all months",
       x = "Month",
       y = "Success Rate (%)",
       fill = "Contact Method") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(size = 16, face = "bold"),
        plot.subtitle = element_text(size = 12),
        legend.position = "top",
        axis.title = element_text(size = 12))
print(contact_plot)
```



#will now check for any missing values

```
df = subset(df, !is.na(df$previous))
df = subset(df, !is.na(df$cons.price.idx))
df = subset(df, !is.na(df$cons.conf.idx))
df = subset(df, !is.na(df$euribor3m))

df = subset(df, !is.nan(df$pdays_bucket))
df = subset(df, !is.nan(df$age_bucket))
df = subset(df, !is.nan(df$campaign_bucket))
df = subset(df, !is.nan(df$job))
df = subset(df, !is.nan(df$marital))
df = subset(df, !is.nan(df$education))
df = subset(df, !is.nan(df$housing))
df = subset(df, !is.nan(df$loan))
df = subset(df, !is.nan(df$contact))
df = subset(df, !is.nan(df$month))
df = subset(df, !is.nan(df$day_of_week))
df = subset(df, !is.nan(df$poutcome))
df = subset(df, !is.nan(df$y))
```

#there didnt appear to be any missing observations in dataset

#splitting training/test

```r
set.seed(42)
tr_ind = sample(nrow(df), 0.8*nrow(df), replace = F)
dftrain = df[tr_ind,]
dftest = df[-tr_ind]
```

#building logistic model

```r
m1.log = glm(y ~., data = dftrain, family = binomial)
summary(m1.log)
```

```
##
## Call:
## glm(formula = y ~ ., family = binomial, data = dftrain)
##
## Coefficients: (1 not defined because of singularities)
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -45.872018   4.457424 -10.291  < 2e-16 ***
## jobblue-collar                -0.180782   0.076387  -2.367  0.01795 *
## jobentrepreneur               -0.027296   0.118893  -0.230  0.81841
## jobhousemaid                  -0.024734   0.139912  -0.177  0.85968
## jobmanagement                 -0.047968   0.083136  -0.577  0.56395
## jobretired                     0.164368   0.099631   1.650  0.09899 .
## jobself-employed              -0.089191   0.113117  -0.788  0.43041
## jobservices                   -0.169344   0.083606  -2.025  0.04282 *
## jobstudent                     0.082421   0.120435   0.684  0.49375
## jobtechnician                 -0.049254   0.069333  -0.710  0.47746
## jobunemployed                 -0.081138   0.124511  -0.652  0.51462
## jobunknown                    -0.055406   0.230364  -0.241  0.80993
## maritalmarried                -0.015508   0.066154  -0.234  0.81466
## maritalsingle                  0.023289   0.075251   0.309  0.75695
## maritalunknown                 0.478705   0.400624   1.195  0.23213
## educationbasic.6y              0.143349   0.114864   1.248  0.21203
## educationbasic.9y              0.008467   0.091225   0.093  0.92605
## educationhigh.school           0.043642   0.088540   0.493  0.62208
## educationilliterate            0.857812   0.746618   1.149  0.25058
## educationprofessional.course   0.094800   0.097978   0.968  0.33326
## educationuniversity.degree     0.138323   0.088428   1.564  0.11776
## educationunknown               0.069385   0.119038   0.583  0.55997
## housingunknown                -0.085814   0.134404  -0.638  0.52316
## housingyes                    -0.021866   0.040139  -0.545  0.58592
## loanunknown                          NA         NA      NA       NA
## loanyes                       -0.028032   0.055631  -0.504  0.61433
## contacttelephone              -0.526339   0.067202  -7.832 4.79e-15 ***
## monthaug                      -0.112803   0.101581  -1.110  0.26680
## monthdec                       0.448675   0.196495   2.283  0.02241 *
## monthjul                       0.166398   0.092647   1.796  0.07249 .
## monthjun                       0.118213   0.090511   1.306  0.19153
## monthmar                       1.004704   0.122358   8.211  < 2e-16 ***
## monthmay                      -0.604369   0.073249  -8.251  < 2e-16 ***
## monthnov                      -0.063766   0.096907  -0.658  0.51053
## monthoct                       0.160438   0.124097   1.293  0.19606
## monthsep                      -0.063071   0.132818  -0.475  0.63488
## day_of_weekmon                -0.195388   0.064520  -3.028  0.00246 **
```

```
## day_of_weekthu                  0.080414   0.061899   1.299  0.19391
## day_of_weektue                  0.089130   0.063834   1.396  0.16263
## day_of_weekwed                  0.170567   0.063402   2.690  0.00714 **
## previous                       -0.112885   0.062978  -1.792  0.07306 .
## poutcomenonexistent             0.423292   0.096669   4.379 1.19e-05 ***
## poutcomesuccess                 0.669113   0.232768   2.875  0.00405 **
## cons.price.idx                  0.518306   0.049024  10.573  < 2e-16 ***
## cons.conf.idx                   0.044182   0.005172   8.543  < 2e-16 ***
## euribor3m                      -0.564399   0.017863 -31.597  < 2e-16 ***
## pdays_bucket2 Weeks            -0.229327   0.174276  -1.316  0.18821
## pdays_bucket3 Weeks or more    -0.242971   0.339488  -0.716  0.47418
## pdays_bucketNever Contacted    -1.296179   0.249041  -5.205 1.94e-07 ***
## age_bucketOlder Adult          -0.161167   0.049460  -3.259  0.00112 **
## age_bucketOther                -0.459358   1.047084  -0.439  0.66088
## age_bucketSenior                0.038604   0.064184   0.601  0.54753
## age_bucketYoung Adult           0.186150   0.111931   1.663  0.09630 .
## campaign_bucket11-20 contacts  -0.648504   0.232030  -2.795  0.00519 **
## campaign_bucket21-30 contacts  -1.900483   1.007944  -1.886  0.05936 .
## campaign_bucket31-40 contacts -10.665448 108.051063  -0.099  0.92137
## campaign_bucket40+            -10.240800 237.220100  -0.043  0.96557
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 23162  on 32949  degrees of freedom
## Residual deviance: 18331  on 32894  degrees of freedom
## AIC: 18443
##
## Number of Fisher Scoring iterations: 12
```

#there seems to be some N/As in my initial run in the loan variable which tracks whether or not client has a personal loan. Looks like the error is being specifically caused by the unkown observations. I know that housing variable, which tracks whether or not client has a housing loan also has "unknown," observations, so first I will check how many unknowns are in the dataset and then decide whether to remove those or not.

```r
summary(df$loan)
```

```
##      no unknown     yes
##   33950     990    6248
```

```r
summary(df$housing)
```

```
##      no unknown     yes
##   18622     990   21576
```

#Interesting, that there is exaclty 990 "unknown," observations in both housing and loan variables. I will remove these observations from my dataset.

#removing from loan variable first

```r
df = df %>%
  filter(loan != "unknown")
df$loan = droplevels(df$loan)
levels(df$loan)
```

```
## [1] "no"  "yes"
```

```r
table(df$loan)
```

```
##
##    no   yes
## 33950  6248
```

#removing from housing variable unkown observations

```r
df = df %>%
  filter(housing != "unknown")
df$housing = droplevels(df$housing)
levels(df$housing)
```

```
## [1] "no"  "yes"
```

```r
table(df$housing)
```

```
##
##    no   yes
## 18622 21576
```

#now will rebuild my training split

```r
set.seed(42)
tr_ind = sample(nrow(df), 0.8*nrow(df), replace = F)
dftrain = df[tr_ind,]
dftest = df[-tr_ind,]
```

#running logistic model again on training data

```r
m1.log2 = glm(y ~., data = dftrain, family = binomial)
summary(m1.log2)
```

```
##
## Call:
## glm(formula = y ~ ., family = binomial, data = dftrain)
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -42.422472   4.485588  -9.458  < 2e-16 ***
## jobblue-collar                -0.183707   0.077454  -2.372 0.017701 *
## jobentrepreneur               -0.078092   0.120663  -0.647 0.517508
```

```
## jobhousemaid                   -0.114576   0.143775  -0.797 0.425505
## jobmanagement                  -0.029561   0.084356  -0.350 0.726017
## jobretired                      0.197079   0.099892   1.973 0.048504 *
## jobself-employed               -0.096215   0.115656  -0.832 0.405461
## jobservices                    -0.151661   0.085628  -1.771 0.076533 .
## jobstudent                      0.129396   0.122443   1.057 0.290608
## jobtechnician                  -0.119485   0.070760  -1.689 0.091297 .
## jobunemployed                  -0.034145   0.126181  -0.271 0.786699
## jobunknown                     -0.246764   0.231917  -1.064 0.287320
## maritalmarried                  0.024037   0.067815   0.354 0.723008
## maritalsingle                   0.100680   0.076928   1.309 0.190616
## maritalunknown                  0.552614   0.381780   1.447 0.147766
## educationbasic.6y               0.206593   0.115594   1.787 0.073900 .
## educationbasic.9y               0.006109   0.092377   0.066 0.947272
## educationhigh.school            0.058034   0.089501   0.648 0.516715
## educationilliterate             0.973875   0.792373   1.229 0.219049
## educationprofessional.course    0.112873   0.098640   1.144 0.252505
## educationuniversity.degree      0.159315   0.089702   1.776 0.075725 .
## educationunknown                0.182649   0.118908   1.536 0.124524
## housingyes                     -0.034600   0.040264  -0.859 0.390160
## loanyes                         0.002121   0.054963   0.039 0.969216
## contacttelephone               -0.523594   0.068492  -7.645 2.10e-14 ***
## monthaug                       -0.094694   0.102897  -0.920 0.357428
## monthdec                        0.355885   0.197668   1.800 0.071795 .
## monthjul                        0.233804   0.093935   2.489 0.012810 *
## monthjun                        0.154277   0.091590   1.684 0.092096 .
## monthmar                        1.092297   0.124931   8.743  < 2e-16 ***
## monthmay                       -0.619132   0.074637  -8.295  < 2e-16 ***
## monthnov                       -0.080340   0.098442  -0.816 0.414435
## monthoct                        0.199895   0.124312   1.608 0.107833
## monthsep                       -0.077795   0.134741  -0.577 0.563690
## day_of_weekmon                 -0.227338   0.065408  -3.476 0.000510 ***
## day_of_weekthu                  0.044297   0.063018   0.703 0.482101
## day_of_weektue                  0.075259   0.064498   1.167 0.243279
## day_of_weekwed                  0.169483   0.063946   2.650 0.008040 **
## previous                       -0.055507   0.062674  -0.886 0.375805
## poutcomenonexistent             0.471679   0.097150   4.855 1.20e-06 ***
## poutcomesuccess                 0.799778   0.234572   3.410 0.000651 ***
## cons.price.idx                  0.478256   0.049349   9.691  < 2e-16 ***
## cons.conf.idx                   0.045191   0.005227   8.645  < 2e-16 ***
## euribor3m                      -0.557897   0.018058 -30.895  < 2e-16 ***
## pdays_bucket2 Weeks            -0.094081   0.177020  -0.531 0.595094
## pdays_bucket3 Weeks or more    -0.544380   0.309143  -1.761 0.078250 .
## pdays_bucketNever Contacted    -1.110168   0.250789  -4.427 9.57e-06 ***
## age_bucketOlder Adult          -0.136323   0.050320  -2.709 0.006746 **
## age_bucketOther                -1.202815   1.373325  -0.876 0.381116
## age_bucketSenior                0.033375   0.065521   0.509 0.610482
## age_bucketYoung Adult           0.145214   0.113341   1.281 0.200119
## campaign_bucket11-20 contacts  -0.336101   0.209758  -1.602 0.109083
## campaign_bucket21-30 contacts -12.767592 145.606133  -0.088 0.930126
## campaign_bucket31-40 contacts -12.652415 299.807132  -0.042 0.966338
## campaign_bucket40+            -12.465411 711.217068  -0.018 0.986016
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22584  on 32157  degrees of freedom
## Residual deviance: 17793  on 32103  degrees of freedom
## AIC: 17903
##
## Number of Fisher Scoring iterations: 14
```

#Based on my initial logistical regression model, the following variables are shown to be statistically significant predictors of whether or not a client will subscribe to a term deposit. #jobblue-collar
#jobretired
#contacttelephone #monthjul #monthmar #monthmay #day_of_weekmon
#day_of_weekwed
#poutcomenonexistent
#poutcomesuccess
#cons.price.idx
#cons.conf.idx #euribor3m #pdays_bucketNever Contacted
#age_bucketOlder Adult

#using VIF funtion to check for multicollinearity

```r
vif(m1.log2)
```

```
##                      GVIF Df GVIF^(1/(2*Df))
## job              5.746693 11        1.082727
## marital          1.439107  3        1.062549
## education        3.196912  7        1.086556
## housing          1.010455  1        1.005214
## loan             1.004657  1        1.002326
## contact          1.900466  1        1.378574
## month            5.457298  9        1.098862
## day_of_week      1.043908  4        1.005386
## previous         4.638597  1        2.153740
## poutcome        28.203378  2        2.304492
## cons.price.idx   2.585490  1        1.607946
## cons.conf.idx    2.322444  1        1.523957
## euribor3m        2.768519  1        1.663887
## pdays_bucket    12.658073  3        1.526609
## age_bucket       2.342032  4        1.112242
## campaign_bucket  1.012210  4        1.001518
```

#making predictions for logistic model

```r
predprob = predict.glm(m1.log2, newdata = dftest, type = "response")
predclass_log = ifelse(predprob >=.08, "yes", "no" )
caret::confusionMatrix(as.factor(predclass_log), as.factor(dftest$y), positive = "yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   no  yes
##        no  5181  273
```

```
##        yes 1935  651
##
##                Accuracy : 0.7254
##                  95% CI : (0.7155, 0.7351)
##     No Information Rate : 0.8851
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2427
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.70455
##             Specificity : 0.72808
##          Pos Pred Value : 0.25174
##          Neg Pred Value : 0.94994
##              Prevalence : 0.11493
##          Detection Rate : 0.08097
##    Detection Prevalence : 0.32164
##       Balanced Accuracy : 0.71631
##
##        'Positive' Class : yes
##
```

#to account for the imbalanced dataset I set my decision threshold to .08 since almost 90% of the dataset consists of observations that resulted in client saying "no" to making a term deposit. At this threshold I achieved my best results listed below.

Accuracy : 0.7254

#Sensitivity : 0.70455
#Specificity : 0.72808

Visualization 6: ROC Curve for Initial Model

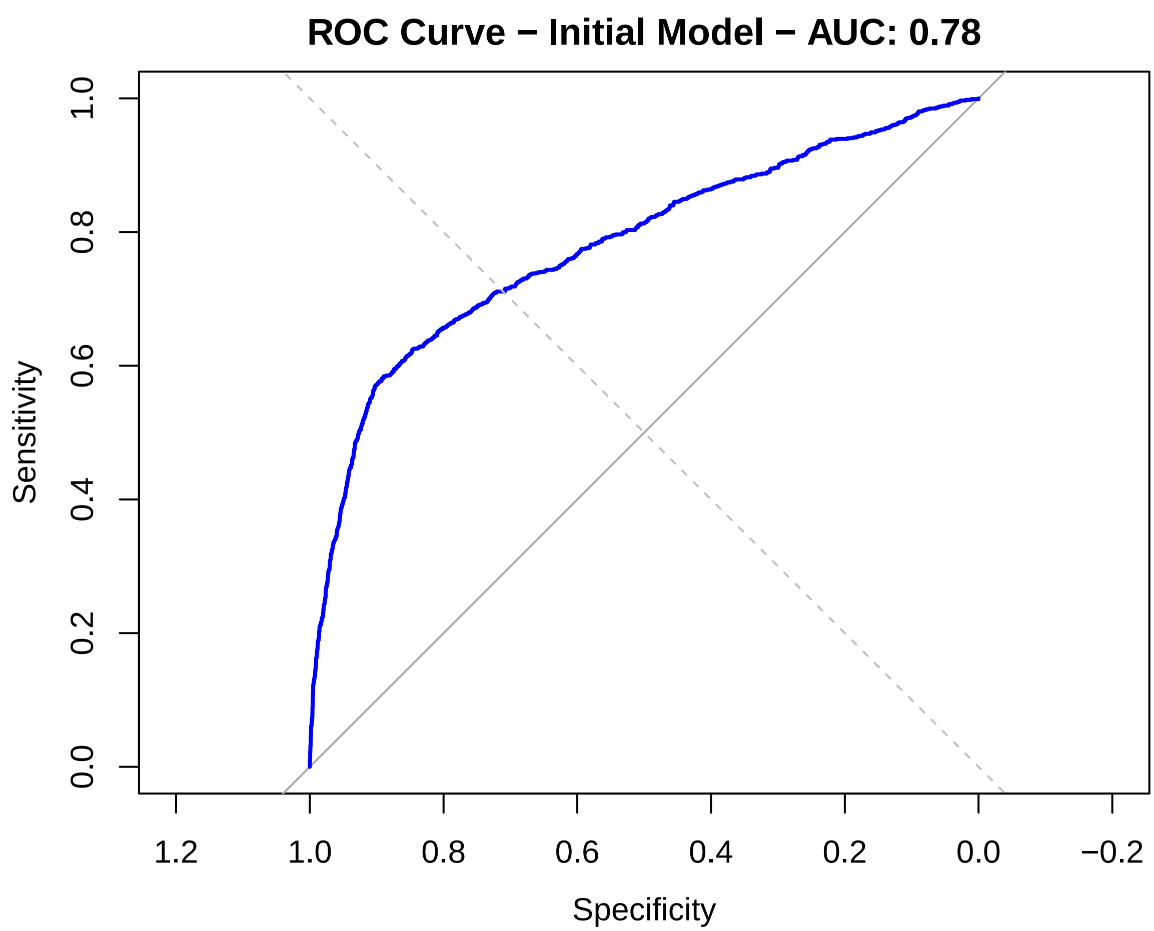
```
roc_obj1 <- roc(dftest$y, predprob)
```

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

```
## Setting direction: controls < cases
```

```
auc_value1 <- auc(roc_obj1)

plot(roc_obj1,
     main = paste("ROC Curve - Initial Model - AUC:", round(auc_value1, 3)),
     col = "blue", lwd = 2)
abline(a = 0, b = 1, lty = 2, col = "gray")
```



#I will now to a backwards stepwise to see if this will improve my model

```r
m2.log = step(m1.log2, direction = "backward")
```

```
## Start:  AIC=17903.49
## y ~ job + marital + education + housing + loan + contact + month +
##     day_of_week + previous + poutcome + cons.price.idx + cons.conf.idx +
##     euribor3m + pdays_bucket + age_bucket + campaign_bucket
##
##                   Df Deviance   AIC
## - education        7    17804 17900
## - loan             1    17794 17902
## - marital          3    17798 17902
## - housing          1    17794 17902
## - previous         1    17794 17902
## - job             11    17815 17903
## <none>                  17794 17904
## - age_bucket       4    17808 17910
## - campaign_bucket  4    17811 17913
## - pdays_bucket     3    17814 17918
## - poutcome         2    17824 17930
## - day_of_week      4    17837 17939
## - contact          1    17855 17963
```

```
## - cons.conf.idx    1    17868 17976
## - cons.price.idx   1    17887 17995
## - month            9    18115 18207
## - euribor3m        1    18664 18772
##
## Step:  AIC=17899.73
## y ~ job + marital + housing + loan + contact + month + day_of_week +
##     previous + poutcome + cons.price.idx + cons.conf.idx + euribor3m +
##     pdays_bucket + age_bucket + campaign_bucket
##
##                  Df Deviance   AIC
## - loan            1    17804 17898
## - previous        1    17804 17898
## - housing         1    17804 17898
## - marital         3    17809 17899
## <none>                 17804 17900
## - age_bucket      4    17817 17905
## - job            11    17833 17907
## - campaign_bucket  4    17821 17909
## - pdays_bucket    3    17824 17914
## - poutcome        2    17834 17926
## - day_of_week     4    17847 17935
## - contact         1    17866 17960
## - cons.conf.idx   1    17880 17974
## - cons.price.idx  1    17898 17992
## - month           9    18129 18207
## - euribor3m       1    18680 18774
##
## Step:  AIC=17897.73
## y ~ job + marital + housing + contact + month + day_of_week +
##     previous + poutcome + cons.price.idx + cons.conf.idx + euribor3m +
##     pdays_bucket + age_bucket + campaign_bucket
##
##                  Df Deviance   AIC
## - previous        1    17804 17896
## - housing         1    17804 17896
## - marital         3    17809 17897
## <none>                 17804 17898
## - age_bucket      4    17817 17903
## - job            11    17833 17905
## - campaign_bucket  4    17821 17907
## - pdays_bucket    3    17824 17912
## - poutcome        2    17834 17924
## - day_of_week     4    17847 17933
## - contact         1    17866 17958
## - cons.conf.idx   1    17880 17972
## - cons.price.idx  1    17898 17990
## - month           9    18129 18205
## - euribor3m       1    18680 18772
##
## Step:  AIC=17896.42
## y ~ job + marital + housing + contact + month + day_of_week +
##     poutcome + cons.price.idx + cons.conf.idx + euribor3m + pdays_bucket +
##     age_bucket + campaign_bucket
```

```
##
##                  Df Deviance   AIC
## - housing         1    17805 17895
## - marital         3    17809 17895
## <none>                 17804 17896
## - age_bucket      4    17818 17902
## - job            11    17834 17904
## - campaign_bucket  4    17822 17906
## - pdays_bucket    3    17825 17911
## - day_of_week     4    17848 17932
## - contact         1    17866 17956
## - cons.conf.idx   1    17880 17970
## - poutcome        2    17893 17981
## - cons.price.idx  1    17900 17990
## - month           9    18132 18206
## - euribor3m       1    18702 18792
##
## Step:  AIC=17895.15
## y ~ job + marital + contact + month + day_of_week + poutcome +
##     cons.price.idx + cons.conf.idx + euribor3m + pdays_bucket +
##     age_bucket + campaign_bucket
##
##                  Df Deviance   AIC
## - marital         3    17810 17894
## <none>                 17805 17895
## - age_bucket      4    17819 17901
## - job            11    17834 17902
## - campaign_bucket  4    17822 17904
## - pdays_bucket    3    17826 17910
## - day_of_week     4    17848 17930
## - contact         1    17867 17955
## - cons.conf.idx   1    17881 17969
## - poutcome        2    17893 17979
## - cons.price.idx  1    17901 17989
## - month           9    18132 18204
## - euribor3m       1    18702 18790
##
## Step:  AIC=17894.11
## y ~ job + contact + month + day_of_week + poutcome + cons.price.idx +
##     cons.conf.idx + euribor3m + pdays_bucket + age_bucket + campaign_bucket
##
##                  Df Deviance   AIC
## <none>                 17810 17894
## - campaign_bucket  4    17827 17903
## - job            11    17841 17903
## - age_bucket      4    17827 17903
## - pdays_bucket    3    17830 17908
## - day_of_week     4    17853 17929
## - contact         1    17872 17954
## - cons.conf.idx   1    17886 17968
## - poutcome        2    17899 17979
## - cons.price.idx  1    17906 17988
## - month           9    18140 18206
## - euribor3m       1    18714 18796
```

```r
summary(m2.log)
```

```
##
## Call:
## glm(formula = y ~ job + contact + month + day_of_week + poutcome +
##     cons.price.idx + cons.conf.idx + euribor3m + pdays_bucket +
##     age_bucket + campaign_bucket, family = binomial, data = dftrain)
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -41.408681   4.366688  -9.483  < 2e-16 ***
## jobblue-collar                -0.250252   0.062937  -3.976 7.00e-05 ***
## jobentrepreneur               -0.096158   0.119512  -0.805 0.421057
## jobhousemaid                  -0.176908   0.138162  -1.280 0.200391
## jobmanagement                 -0.017267   0.083015  -0.208 0.835227
## jobretired                     0.147195   0.094938   1.550 0.121036
## jobself-employed              -0.092356   0.114607  -0.806 0.420326
## jobservices                   -0.201212   0.081306  -2.475 0.013333 *
## jobstudent                     0.125072   0.118755   1.053 0.292253
## jobtechnician                 -0.124345   0.062965  -1.975 0.048289 *
## jobunemployed                 -0.074526   0.124356  -0.599 0.548973
## jobunknown                    -0.234219   0.228534  -1.025 0.305421
## contacttelephone              -0.522062   0.068357  -7.637 2.22e-14 ***
## monthaug                      -0.089160   0.102270  -0.872 0.383311
## monthdec                       0.346416   0.197239   1.756 0.079032 .
## monthjul                       0.237504   0.093758   2.533 0.011304 *
## monthjun                       0.163830   0.091442   1.792 0.073191 .
## monthmar                       1.098941   0.124587   8.821  < 2e-16 ***
## monthmay                      -0.626189   0.074377  -8.419  < 2e-16 ***
## monthnov                      -0.084759   0.098121  -0.864 0.387688
## monthoct                       0.199663   0.124190   1.608 0.107896
## monthsep                      -0.076812   0.134584  -0.571 0.568179
## day_of_weekmon                -0.230556   0.065324  -3.529 0.000416 ***
## day_of_weekthu                 0.043790   0.062921   0.696 0.486462
## day_of_weektue                 0.072686   0.064407   1.129 0.259096
## day_of_weekwed                 0.164820   0.063883   2.580 0.009879 **
## poutcomenonexistent            0.538565   0.064541   8.345  < 2e-16 ***
## poutcomesuccess                0.869021   0.224059   3.879 0.000105 ***
## cons.price.idx                 0.467725   0.047682   9.809  < 2e-16 ***
## cons.conf.idx                  0.045452   0.005218   8.710  < 2e-16 ***
## euribor3m                     -0.557939   0.017765 -31.407  < 2e-16 ***
## pdays_bucket2 Weeks           -0.080733   0.175483  -0.460 0.645473
## pdays_bucket3 Weeks or more   -0.494980   0.306158  -1.617 0.105933
## pdays_bucketNever Contacted   -1.024328   0.232740  -4.401 1.08e-05 ***
## age_bucketOlder Adult         -0.164180   0.047902  -3.427 0.000609 ***
## age_bucketOther               -1.248532   1.350633  -0.924 0.355275
## age_bucketSenior              -0.015955   0.061050  -0.261 0.793828
## age_bucketYoung Adult          0.146741   0.112445   1.305 0.191891
## campaign_bucket11-20 contacts -0.326298   0.209598  -1.557 0.119524
## campaign_bucket21-30 contacts -12.769426 145.870957  -0.088 0.930243
## campaign_bucket31-40 contacts -12.655479 300.097768  -0.042 0.966362
## campaign_bucket40+            -12.498559 713.386690  -0.018 0.986022
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22584  on 32157  degrees of freedom
## Residual deviance: 17810  on 32116  degrees of freedom
## AIC: 17894
##
## Number of Fisher Scoring iterations: 14
```

#The variables listed below were statistically significant using backwards stepwise. #jobblue-collar -0.250252 0.062937 -3.976 7.00e-05 ***#jobservices -0.201212 0.081306 -2.475 0.013333*** **#jobtechnician -0.124345 0.062965 -1.975 0.048289 * #contacttelephone -0.522062 0.068357 -7.637 2.22e-14** *#monthjul 0.237504 0.093758 2.533 0.011304* #monthmar 1.098941 0.124587 8.821 < 2e-16 ***#monthmay -0.626189 0.074377 -8.419 < 2e-16*** #day_of_weekmon -0.230556 0.065324 -3.529 0.000416 ***#day_of_weekwed 0.164820 0.063883 2.580 0.009879*** *#poutcomenonexistent 0.538565 0.064541 8.345 < 2e-16* **#poutcomesuccess 0.869021 0.224059 3.879 0.000105** *#cons.price.idx 0.467725 0.047682 9.809 < 2e-16* **#cons.conf.idx 0.045452 0.005218 8.710 < 2e-16** *#euribor3m -0.557939 0.017765 -31.407 < 2e-16* **#pdays_bucketNever Contacted -1.024328 0.232740 -4.401 1.08e-05** *#age_bucketOlder Adult -0.164180 0.047902 -3.427 0.000609* **

#checking for multicollinearity

```r
vif(m2.log)
```

```
##                      GVIF Df GVIF^(1/(2*Df))
## job              2.128046 11        1.034923
## contact          1.894161  1        1.376285
## month            5.246849  9        1.096464
## day_of_week      1.040074  4        1.004924
## poutcome        11.941379  2        1.858933
## cons.price.idx   2.415762  1        1.554272
## cons.conf.idx    2.316719  1        1.522077
## euribor3m        2.681408  1        1.637500
## pdays_bucket    10.864901  3        1.488233
## age_bucket       1.909412  4        1.084208
## campaign_bucket  1.011680  4        1.001453
```

#No multicollinearity

#Will check and see what features are being utilized in my model and then will filter dataset and run logistic regression again.

```r
all.vars(formula(m2.log))
```

```
##  [1] "y"                "job"              "contact"          "month"
##  [5] "day_of_week"      "poutcome"         "cons.price.idx"   "cons.conf.idx"
##  [9] "euribor3m"        "pdays_bucket"     "age_bucket"       "campaign_bucket"
```

```r
df2 = df %>%
  select("y","job","contact","month","day_of_week", "poutcome", "cons.price.idx","cons.conf.idx", "euri
```

#splitting new dataset

```
set.seed(42)
tr_ind2 = sample(nrow(df2), 0.8*nrow(df2), replace = F)
dftrain2 = df2[tr_ind2,]
dftest2 = df2[-tr_ind2,]
```

```
predprob2 = predict.glm(m2.log, newdata = dftest2, type = "response")
predclass_log2 = ifelse(predprob >=.078, "yes", "no" )
caret::confusionMatrix(as.factor(predclass_log2), as.factor(dftest2$y), positive = "yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   no  yes
##        no  5103  267
##        yes 2013  657
##
##                Accuracy : 0.7164
##                  95% CI : (0.7064, 0.7263)
##     No Information Rate : 0.8851
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.235
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.71104
##             Specificity : 0.71712
##          Pos Pred Value : 0.24607
##          Neg Pred Value : 0.95028
##              Prevalence : 0.11493
##          Detection Rate : 0.08172
##    Detection Prevalence : 0.33209
##       Balanced Accuracy : 0.71408
##
##        'Positive' Class : yes
##
```

#Backwards stepwise did not help improve overall accuracy or sensitivity. However, when adjusting decision threshold to .078 accuracy dropped from .7254 to 0.7164 but sensitivity which predicts 1 (yes) increased slightly from 0.70455 to 0.71104.

#will now do a stepwise that is both forward and backward.

```
m3.log = step(m1.log2, direction = "both")
```

```
## Start:  AIC=17903.49
## y ~ job + marital + education + housing + loan + contact + month +
##     day_of_week + previous + poutcome + cons.price.idx + cons.conf.idx +
##     euribor3m + pdays_bucket + age_bucket + campaign_bucket
##
##                     Df Deviance   AIC
```

```
## - education        7    17804 17900
## - loan             1    17794 17902
## - marital          3    17798 17902
## - housing          1    17794 17902
## - previous         1    17794 17902
## - job             11    17815 17903
## <none>                  17794 17904
## - age_bucket       4    17808 17910
## - campaign_bucket  4    17811 17913
## - pdays_bucket     3    17814 17918
## - poutcome         2    17824 17930
## - day_of_week      4    17837 17939
## - contact          1    17855 17963
## - cons.conf.idx    1    17868 17976
## - cons.price.idx   1    17887 17995
## - month            9    18115 18207
## - euribor3m        1    18664 18772
##
## Step:  AIC=17899.73
## y ~ job + marital + housing + loan + contact + month + day_of_week +
##     previous + poutcome + cons.price.idx + cons.conf.idx + euribor3m +
##     pdays_bucket + age_bucket + campaign_bucket
##
##                   Df Deviance   AIC
## - loan             1    17804 17898
## - previous         1    17804 17898
## - housing          1    17804 17898
## - marital          3    17809 17899
## <none>                  17804 17900
## + education        7    17794 17904
## - age_bucket       4    17817 17905
## - job             11    17833 17907
## - campaign_bucket  4    17821 17909
## - pdays_bucket     3    17824 17914
## - poutcome         2    17834 17926
## - day_of_week      4    17847 17935
## - contact          1    17866 17960
## - cons.conf.idx    1    17880 17974
## - cons.price.idx   1    17898 17992
## - month            9    18129 18207
## - euribor3m        1    18680 18774
##
## Step:  AIC=17897.73
## y ~ job + marital + housing + contact + month + day_of_week +
##     previous + poutcome + cons.price.idx + cons.conf.idx + euribor3m +
##     pdays_bucket + age_bucket + campaign_bucket
##
##                   Df Deviance   AIC
## - previous         1    17804 17896
## - housing          1    17804 17896
## - marital          3    17809 17897
## <none>                  17804 17898
## + loan             1    17804 17900
## + education        7    17794 17902
```

```
## - age_bucket       4    17817 17903
## - job             11    17833 17905
## - campaign_bucket  4    17821 17907
## - pdays_bucket     3    17824 17912
## - poutcome         2    17834 17924
## - day_of_week      4    17847 17933
## - contact          1    17866 17958
## - cons.conf.idx    1    17880 17972
## - cons.price.idx   1    17898 17990
## - month            9    18129 18205
## - euribor3m        1    18680 18772
##
## Step:  AIC=17896.42
## y ~ job + marital + housing + contact + month + day_of_week +
##     poutcome + cons.price.idx + cons.conf.idx + euribor3m + pdays_bucket +
##     age_bucket + campaign_bucket
##
##                   Df Deviance   AIC
## - housing          1    17805 17895
## - marital          3    17809 17895
## <none>                  17804 17896
## + previous         1    17804 17898
## + loan             1    17804 17898
## + education        7    17794 17900
## - age_bucket       4    17818 17902
## - job             11    17834 17904
## - campaign_bucket  4    17822 17906
## - pdays_bucket     3    17825 17911
## - day_of_week      4    17848 17932
## - contact          1    17866 17956
## - cons.conf.idx    1    17880 17970
## - poutcome         2    17893 17981
## - cons.price.idx   1    17900 17990
## - month            9    18132 18206
## - euribor3m        1    18702 18792
##
## Step:  AIC=17895.15
## y ~ job + marital + contact + month + day_of_week + poutcome +
##     cons.price.idx + cons.conf.idx + euribor3m + pdays_bucket +
##     age_bucket + campaign_bucket
##
##                   Df Deviance   AIC
## - marital          3    17810 17894
## <none>                  17805 17895
## + housing          1    17804 17896
## + previous         1    17804 17896
## + loan             1    17805 17897
## + education        7    17795 17899
## - age_bucket       4    17819 17901
## - job             11    17834 17902
## - campaign_bucket  4    17822 17904
## - pdays_bucket     3    17826 17910
## - day_of_week      4    17848 17930
## - contact          1    17867 17955
```

```
## - cons.conf.idx    1    17881 17969
## - poutcome         2    17893 17979
## - cons.price.idx   1    17901 17989
## - month            9    18132 18204
## - euribor3m        1    18702 18790
##
## Step:  AIC=17894.11
## y ~ job + contact + month + day_of_week + poutcome + cons.price.idx +
##     cons.conf.idx + euribor3m + pdays_bucket + age_bucket + campaign_bucket
##
##                   Df Deviance   AIC
## <none>                 17810 17894
## + marital          3    17805 17895
## + housing          1    17809 17895
## + previous         1    17809 17895
## + loan             1    17810 17896
## + education        7    17799 17897
## - campaign_bucket  4    17827 17903
## - job             11    17841 17903
## - age_bucket       4    17827 17903
## - pdays_bucket     3    17830 17908
## - day_of_week      4    17853 17929
## - contact          1    17872 17954
## - cons.conf.idx    1    17886 17968
## - poutcome         2    17899 17979
## - cons.price.idx   1    17906 17988
## - month            9    18140 18206
## - euribor3m        1    18714 18796
```

```r
summary(m3.log)
```

```
##
## Call:
## glm(formula = y ~ job + contact + month + day_of_week + poutcome +
##     cons.price.idx + cons.conf.idx + euribor3m + pdays_bucket +
##     age_bucket + campaign_bucket, family = binomial, data = dftrain)
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -41.408681   4.366688  -9.483  < 2e-16 ***
## jobblue-collar                -0.250252   0.062937  -3.976 7.00e-05 ***
## jobentrepreneur               -0.096158   0.119512  -0.805 0.421057
## jobhousemaid                  -0.176908   0.138162  -1.280 0.200391
## jobmanagement                 -0.017267   0.083015  -0.208 0.835227
## jobretired                     0.147195   0.094938   1.550 0.121036
## jobself-employed              -0.092356   0.114607  -0.806 0.420326
## jobservices                   -0.201212   0.081306  -2.475 0.013333 *
## jobstudent                     0.125072   0.118755   1.053 0.292253
## jobtechnician                 -0.124345   0.062965  -1.975 0.048289 *
## jobunemployed                 -0.074526   0.124356  -0.599 0.548973
## jobunknown                    -0.234219   0.228534  -1.025 0.305421
## contacttelephone              -0.522062   0.068357  -7.637 2.22e-14 ***
## monthaug                      -0.089160   0.102270  -0.872 0.383311
## monthdec                       0.346416   0.197239   1.756 0.079032 .
```

```
## monthjul                        0.237504   0.093758   2.533 0.011304 *
## monthjun                        0.163830   0.091442   1.792 0.073191 .
## monthmar                        1.098941   0.124587   8.821  < 2e-16 ***
## monthmay                       -0.626189   0.074377  -8.419  < 2e-16 ***
## monthnov                       -0.084759   0.098121  -0.864 0.387688
## monthoct                        0.199663   0.124190   1.608 0.107896
## monthsep                       -0.076812   0.134584  -0.571 0.568179
## day_of_weekmon                 -0.230556   0.065324  -3.529 0.000416 ***
## day_of_weekthu                  0.043790   0.062921   0.696 0.486462
## day_of_weektue                  0.072686   0.064407   1.129 0.259096
## day_of_weekwed                  0.164820   0.063883   2.580 0.009879 **
## poutcomenonexistent             0.538565   0.064541   8.345  < 2e-16 ***
## poutcomesuccess                 0.869021   0.224059   3.879 0.000105 ***
## cons.price.idx                  0.467725   0.047682   9.809  < 2e-16 ***
## cons.conf.idx                   0.045452   0.005218   8.710  < 2e-16 ***
## euribor3m                      -0.557939   0.017765 -31.407  < 2e-16 ***
## pdays_bucket2 Weeks            -0.080733   0.175483  -0.460 0.645473
## pdays_bucket3 Weeks or more    -0.494980   0.306158  -1.617 0.105933
## pdays_bucketNever Contacted    -1.024328   0.232740  -4.401 1.08e-05 ***
## age_bucketOlder Adult          -0.164180   0.047902  -3.427 0.000609 ***
## age_bucketOther                -1.248532   1.350633  -0.924 0.355275
## age_bucketSenior               -0.015955   0.061050  -0.261 0.793828
## age_bucketYoung Adult           0.146741   0.112445   1.305 0.191891
## campaign_bucket11-20 contacts  -0.326298   0.209598  -1.557 0.119524
## campaign_bucket21-30 contacts -12.769426 145.870957  -0.088 0.930243
## campaign_bucket31-40 contacts -12.655479 300.097768  -0.042 0.966362
## campaign_bucket40+            -12.498559 713.386690  -0.018 0.986022
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22584  on 32157  degrees of freedom
## Residual deviance: 17810  on 32116  degrees of freedom
## AIC: 17894
##
## Number of Fisher Scoring iterations: 14
```

```r
vif(m3.log)
```

```
##                     GVIF Df GVIF^(1/(2*Df))
## job             2.128046 11        1.034923
## contact         1.894161  1        1.376285
## month           5.246849  9        1.096464
## day_of_week     1.040074  4        1.004924
## poutcome       11.941379  2        1.858933
## cons.price.idx  2.415762  1        1.554272
## cons.conf.idx   2.316719  1        1.522077
## euribor3m       2.681408  1        1.637500
## pdays_bucket   10.864901  3        1.488233
## age_bucket      1.909412  4        1.084208
## campaign_bucket 1.011680  4        1.001453
```

```r
all.vars(formula(m3.log))
```

```
##  [1] "y"               "job"             "contact"         "month"
##  [5] "day_of_week"     "poutcome"        "cons.price.idx"  "cons.conf.idx"
##  [9] "euribor3m"       "pdays_bucket"    "age_bucket"      "campaign_bucket"
```

```r
df3 = df %>%
  select("y","job","contact","month","day_of_week", "poutcome", "cons.price.idx","cons.conf.idx", "euri
```

#ended up with the same variables

```r
set.seed(42)
tr_ind3 = sample(nrow(df3), 0.8*nrow(df3), replace = F)
dftrain3 = df3[tr_ind2,]
dftest3 = df3[-tr_ind2,]
```

```r
predprob3 = predict.glm(m3.log, newdata = dftest3, type = "response")
predclass_log3 = ifelse(predprob >=.078, "yes", "no" )
caret::confusionMatrix(as.factor(predclass_log3), as.factor(dftest3$y), positive = "yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   no  yes
##        no  5103  267
##        yes 2013  657
##
##                Accuracy : 0.7164
##                  95% CI : (0.7064, 0.7263)
##     No Information Rate : 0.8851
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.235
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.71104
##             Specificity : 0.71712
##          Pos Pred Value : 0.24607
##          Neg Pred Value : 0.95028
##              Prevalence : 0.11493
##          Detection Rate : 0.08172
##    Detection Prevalence : 0.33209
##       Balanced Accuracy : 0.71408
##
##        'Positive' Class : yes
##
```

#same results. I'm honeslty lost at what else we could do to improve accuracy and sensitivity

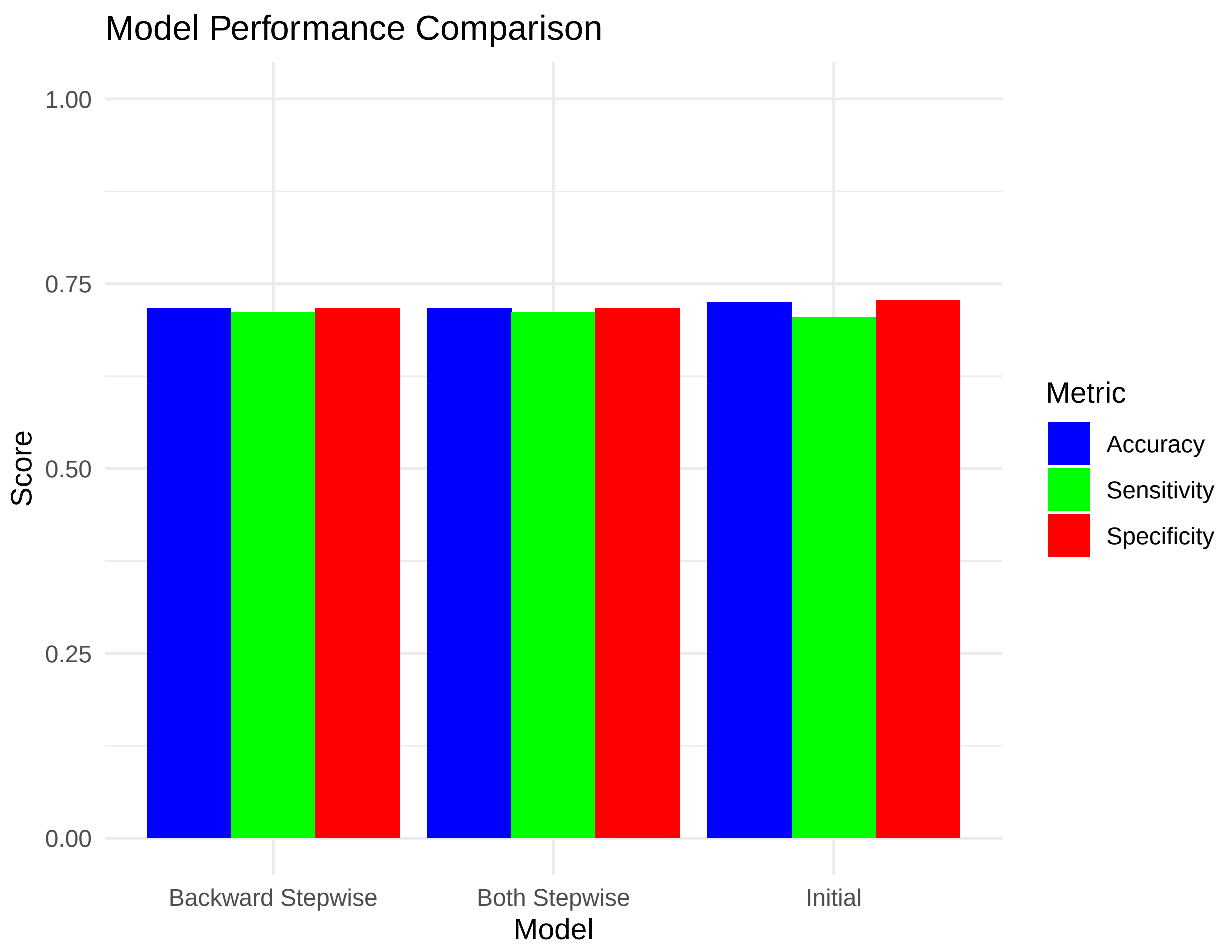
Visualization 7: Comparing Model Performance

```r
# Create a comparison dataframe
model_comparison <- data.frame(
  Model = c("Initial", "Backward Stepwise", "Both Stepwise"),
  Accuracy = c(0.7254, 0.7164, 0.7164),
  Sensitivity = c(0.70455, 0.71104, 0.71104),
  Specificity = c(0.72808, 0.7164, 0.7164)
)

# Reshape for plotting
model_long <- model_comparison %>%
  pivot_longer(cols = c(Accuracy, Sensitivity, Specificity),
               names_to = "Metric",
               values_to = "Value")

comparison_plot <- ggplot(model_long, aes(x = Model, y = Value, fill = Metric)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_fill_manual(values = c("Accuracy" = "blue",
                               "Sensitivity" = "green",
                               "Specificity" = "red")) +
  labs(title = "Model Performance Comparison",
       y = "Score", x = "Model") +
  theme_minimal() +
  ylim(0, 1)

print(comparison_plot)
```

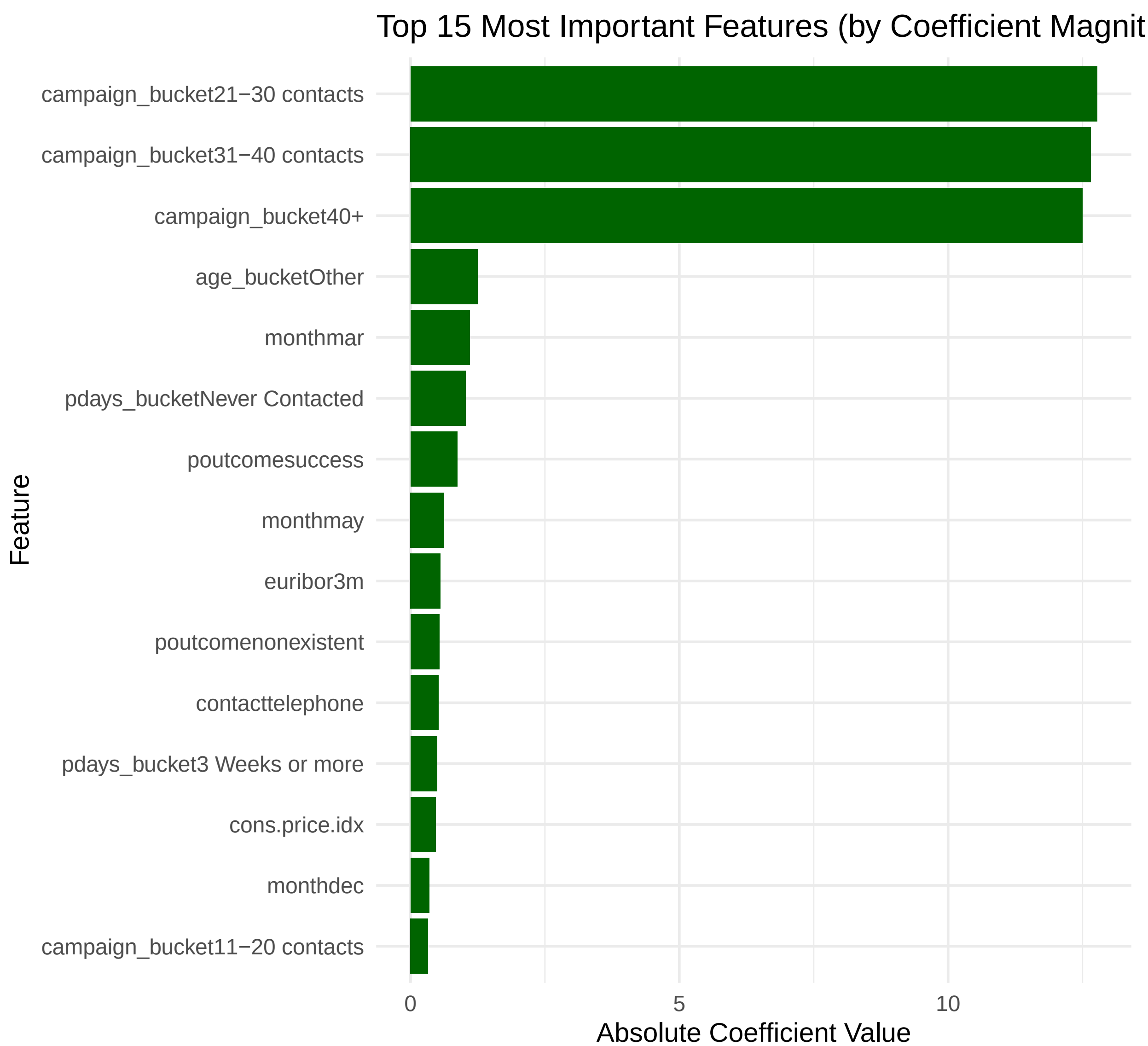


Visualization 8: Feature Importance from Coefficients

```
# Extract coefficients from the stepwise model
coef_df <- data.frame(
  Variable = names(coef(m2.log))[-1],  # Remove intercept
  Coefficient = abs(coef(m2.log))[-1]   # Take absolute values
) %>%
  arrange(desc(Coefficient)) %>%
  head(15)  # Top 15 features

importance_plot <- ggplot(coef_df, aes(x = reorder(Variable, Coefficient),
                                       y = Coefficient)) +
  geom_col(fill = "darkgreen") +
  coord_flip() +
  labs(title = "Top 15 Most Important Features (by Coefficient Magnitude)",
       x = "Feature", y = "Absolute Coefficient Value") +
  theme_minimal()

print(importance_plot)
```


Top 15 Most Important Features (by Coefficient Magnit
Feature
campaign_bucket21−30 contacts
campaign_bucket31−40 contacts
campaign_bucket40+
age_bucketOther
monthmar
pdays_bucketNever Contacted
poutcomesuccess
monthmay
euribor3m
poutcomenonexistent
contacttelephone
pdays_bucket3 Weeks or more
cons.price.idx
monthdec
campaign_bucket11−20 contacts
0
5
10
Absolute Coefficient Value