Final Data Analysis on Bank Marketing

2023-04-17

Final Project : Data Analysis and Preliminary Analytics

Logan Van Dine, Sowmiya Kanmani Maruthavanan, Tara Dehdari

April 17, 2023

Preliminary Analysis of Portuguese Marketing Data

The data set was collected from the UCI Machine Learning Repository covering the period from May 2008 to November 2010. The data set pertains to the direct marketing campaigns conducted by a Portuguese banking institution. These campaigns were conducted by phone calls to potential clients to promote a bank term deposit product. In some cases, multiple contacts with the same client were necessary to determine if they are interested in subscribing to the deposit product or not.

The bank_marketing.csv file is imported using the read.csv() function where the fields are separated by the semicolon (;). This function reads the data and stores it in a data frame labeled "clientdata". The data set has 45211 rows and 17 variables before any pre-processing has been started.

The data set contains a mix of numerical, categorical, and binary variables. The age, balance, day, duration, campaign, pdays, and previous variables are numerical variables. Categorical data can be seen in the job, marital, education, contact, month, and poutcome variables. The binary variables of the data set can be seen in the default, housing, loan, and deposit fields.

Data Importing and Pre-Processing

```r
clientdata <- read.csv("bank_marketing.csv",header=T,sep=";")
#Returns no of rows and columns
dim(clientdata)
```

```
## [1] 45211    17
```

```r
#Returns the data type of the columns
str(clientdata)
```

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

```
#View(clientdata)
```

Missing Values The data set contains missing values in the age (1339), default (1306), and contact (1383) fields.

The descriptive statistics of age represent a skewed variable and an abnormal distribution. The mean will not be a valuable replacement to account for the missing values of this variable. Rather, the median age, 39, will be used to replace missing age values.

The default and contact variables are categorical data, not numerical. Replacing the missing value with the mode would not be appropriate in this case as we cannot make an informed decision based on other variables, rather it would be a random replacement. Without a replacement that would fit the data, it is best to remove the records that contain the missing values in these categorical variables.

```
colSums(is.na(clientdata) | clientdata == "")
```

```
##       age       job   marital education   default   balance   housing      loan 
##      1339         0         0         0      1306         0         0         0 
##   contact       day     month  duration  campaign     pdays  previous  poutcome 
##      1383         0         0         0         0         0         0         0 
##   deposit 
##         0
```

```
#Filtering rows which has non missing values in default and contact columns
clientdata <- clientdata %>% filter(default != "" & contact != "")
#Replace missing values in the "age" column with the median age of all non
#missing values in the same column
clientdata <- clientdata %>% mutate(across(age, ~replace_na(., median(., na.rm=TRUE))))
```

```
colSums(is.na(clientdata))
```

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

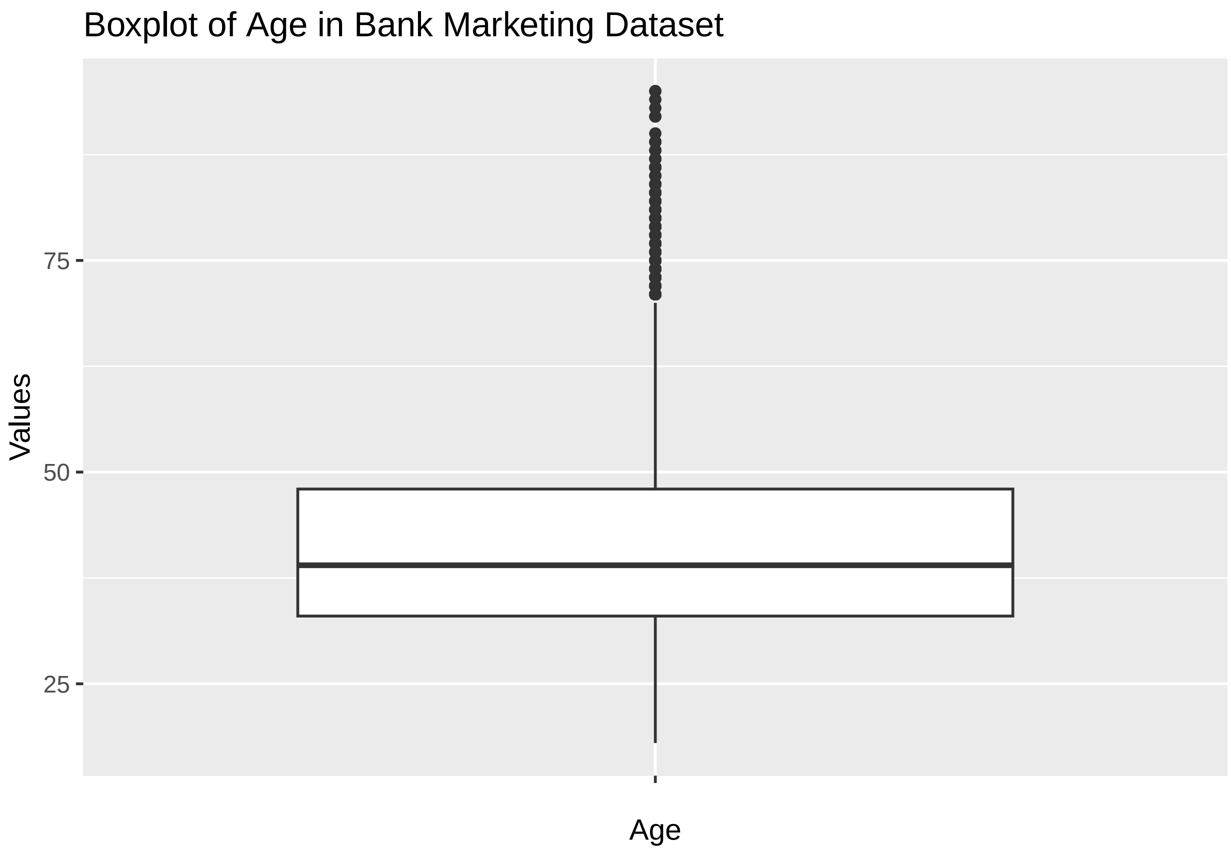
```
summary(clientdata)
```

```
##       age             job              marital           education    
##  Min.   :18.00   Length:42569       Length:42569       Length:42569      
##  1st Qu.:33.00   Class :character   Class :character   Class :character  
##  Median :39.00   Mode  :character   Mode  :character   Mode  :character  
##  Mean   :40.86                                                           
##  3rd Qu.:48.00                                                           
##  Max.   :95.00                                                           
##    default             balance        housing             loan          
##  Length:42569       Min.   : -8019   Length:42569       Length:42569      
##  Class :character   1st Qu.:    72   Class :character   Class :character  
##  Mode  :character   Median :   450   Mode  :character   Mode  :character  
##                     Mean   :  1364                                        
##                     3rd Qu.:  1430                                        
##                     Max.   :102127                                        
##    contact               day           month             duration     
##  Length:42569       Min.   : 1.00   Length:42569       Min.   :   0.0  
##  Class :character   1st Qu.: 8.00   Class :character   1st Qu.: 103.0  
##  Mode  :character   Median :16.00   Mode  :character   Median : 180.0  
##                     Mean   :15.81                      Mean   : 258.3  
##                     3rd Qu.:21.00                      3rd Qu.: 319.0  
##                     Max.   :31.00                      Max.   :3881.0  
##     campaign         pdays           previous          poutcome        
##  Min.   : 1.000   Min.   : -1.00   Min.   :  0.000   Length:42569      
##  1st Qu.: 1.000   1st Qu.: -1.00   1st Qu.:  0.000   Class :character  
##  Median : 2.000   Median : -1.00   Median :  0.000   Mode  :character  
##  Mean   : 2.764   Mean   : 40.27   Mean   :  0.581                     
##  3rd Qu.: 3.000   3rd Qu.: -1.00   3rd Qu.:  0.000                     
##  Max.   :63.000   Max.   :871.00   Max.   :275.000                     
##    deposit         
##  Length:42569      
##  Class :character  
##  Mode  :character  
##                    
##                    
##                    
```

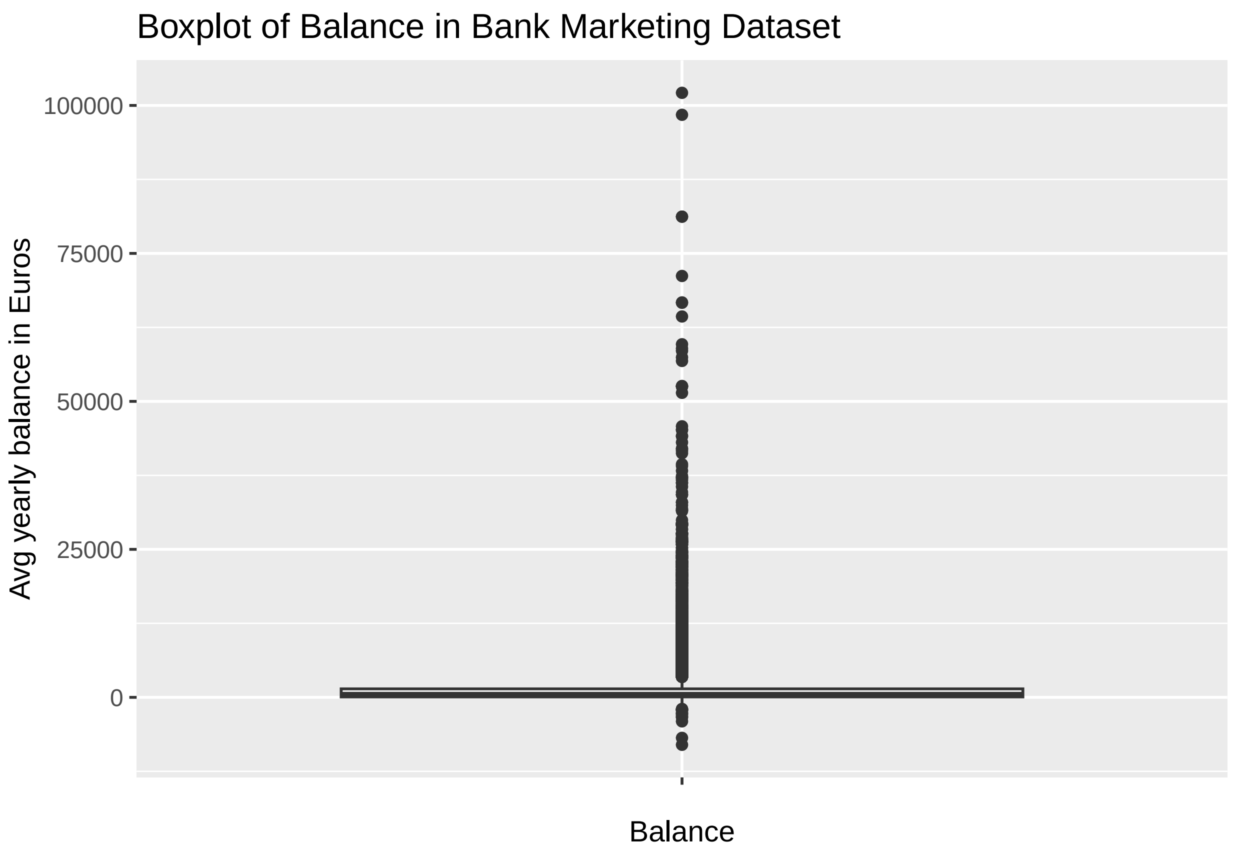
Identifying and Treating Outliers After removal of missing values, it is necessary to determine if any outliers are present in the data that could potentially skew final results of an analysis. Multiple box plots were created to identify the outliers for the seven numerical variables. In general, removing outliers from a large data set is a common practice in data pre-processing and analysis, especially when the outliers are believed to be due to errors or measurement issues.

Upon creation, it is found that six of the numerical values have outliers. The only variable proven to not have outliers present is day, which represent the last day of contact of the month. However, of the six variables with outliers, only one will have the outliers removed. The previous variable, measuring the number of contacts a client has had before the current campaign call, ranges between 0 and 58 with the exception of one record. The record with this value, 275, has been removed to smooth the distribution of the previous variable. All other variables that included outliers (age, balance, duration, campaign, and pdays) have been left untouched because the quantity of outliers is too high to remove. Removal of these outliers would significantly change the data overall.

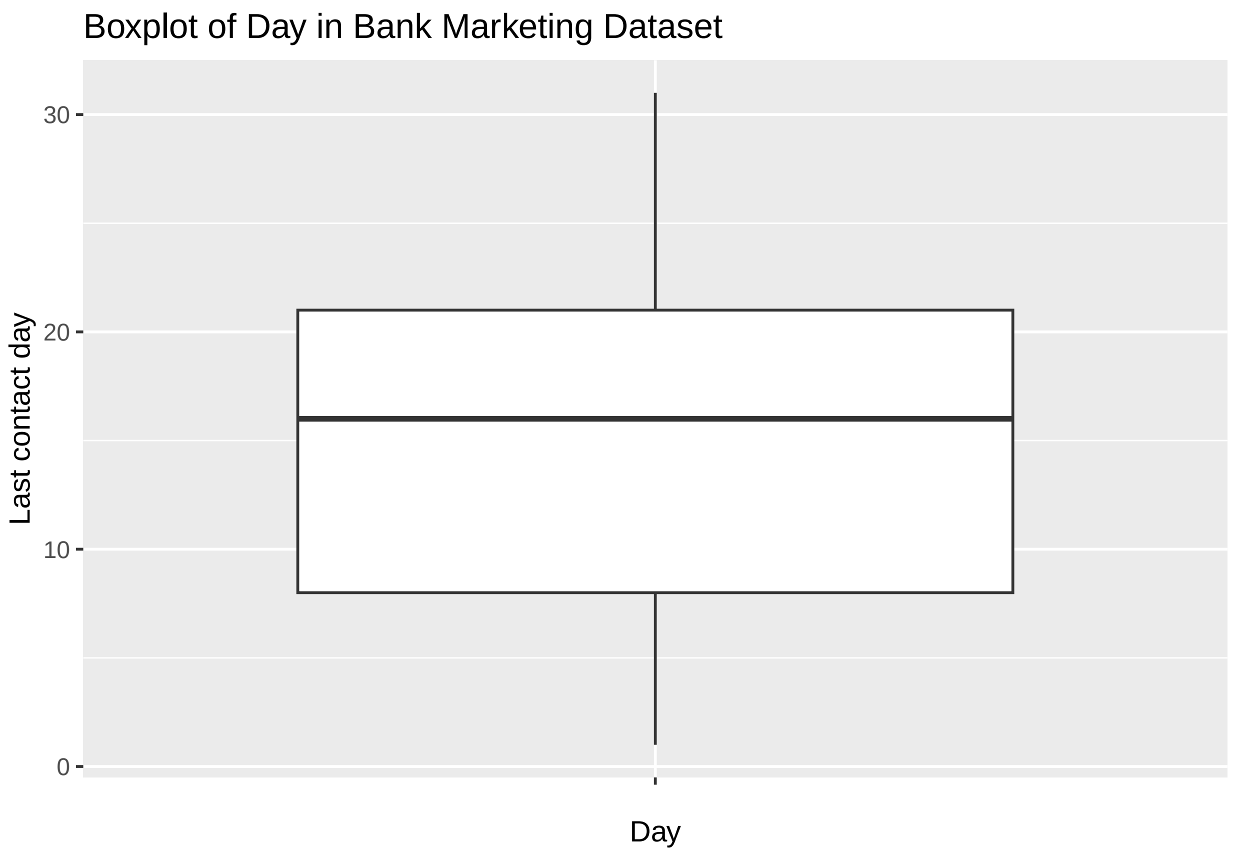
```
#identifying outliers
ggplot(data = clientdata, aes(x = "", y = age)) + geom_boxplot() +
  labs(title = "Boxplot of Age in Bank Marketing Dataset",
       x = "Age",
       y = "Values")
```



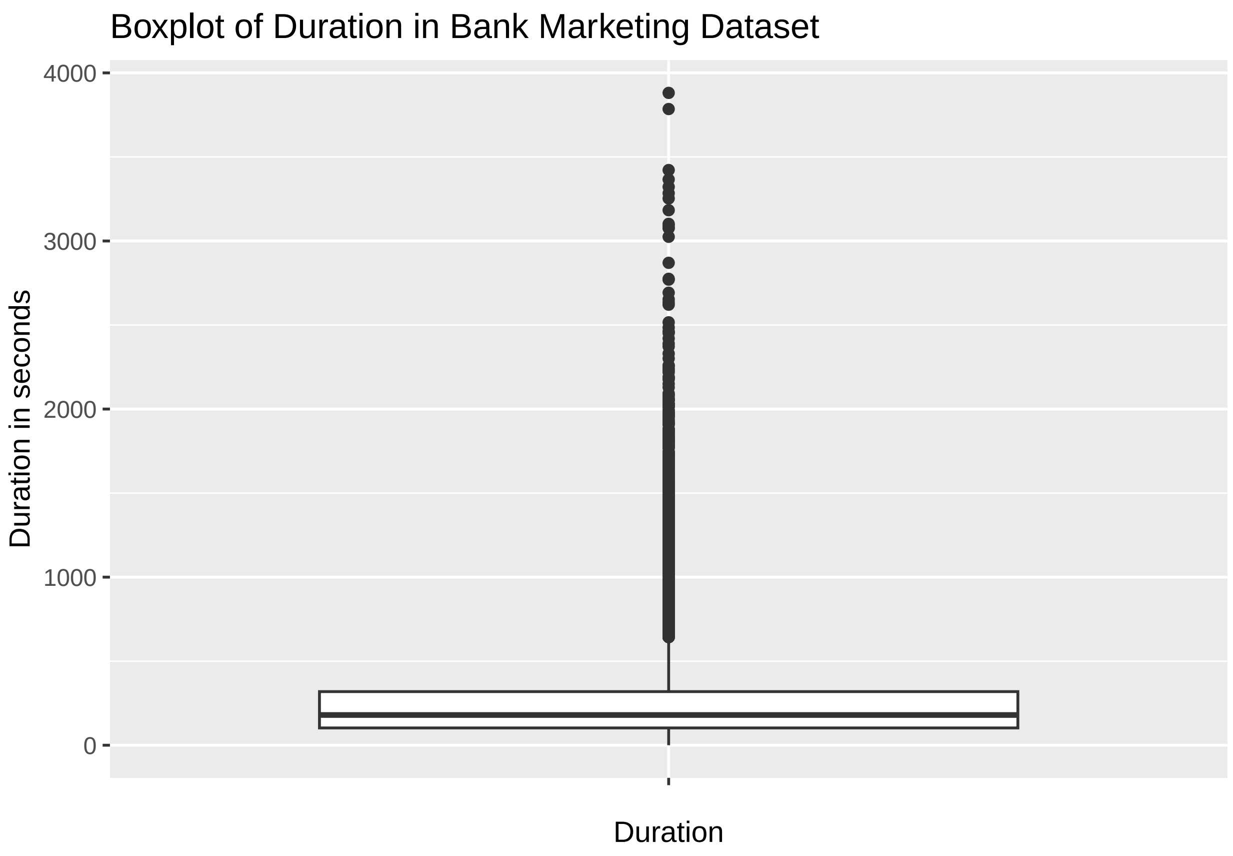
```
ggplot(data = clientdata, aes(x = "", y = balance)) + geom_boxplot() +
  labs(title = "Boxplot of Balance in Bank Marketing Dataset",
       x = "Balance",
       y = "Avg yearly balance in Euros")
```



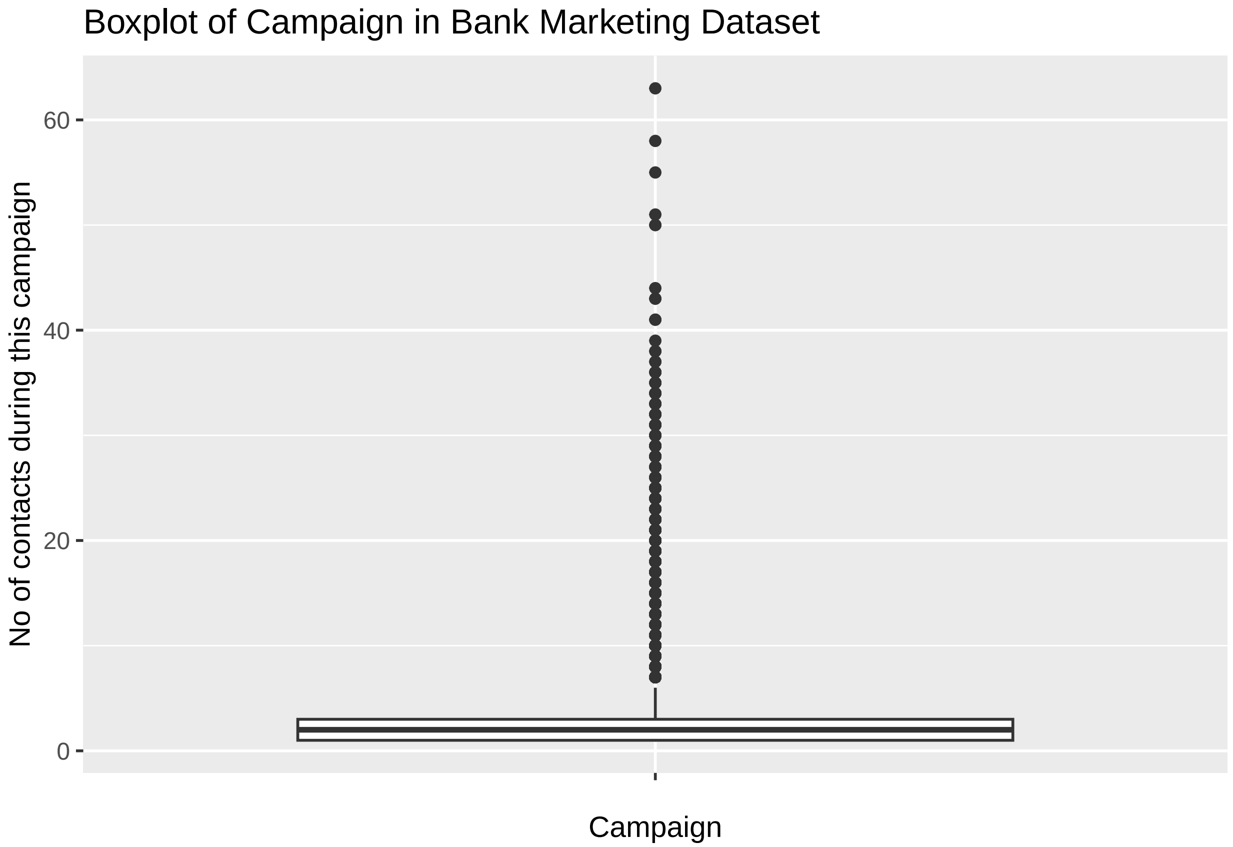
```
ggplot(data = clientdata, aes(x = "", y = day)) + geom_boxplot() +
  labs(title = "Boxplot of Day in Bank Marketing Dataset",
       x = "Day",
       y = "Last contact day")
```



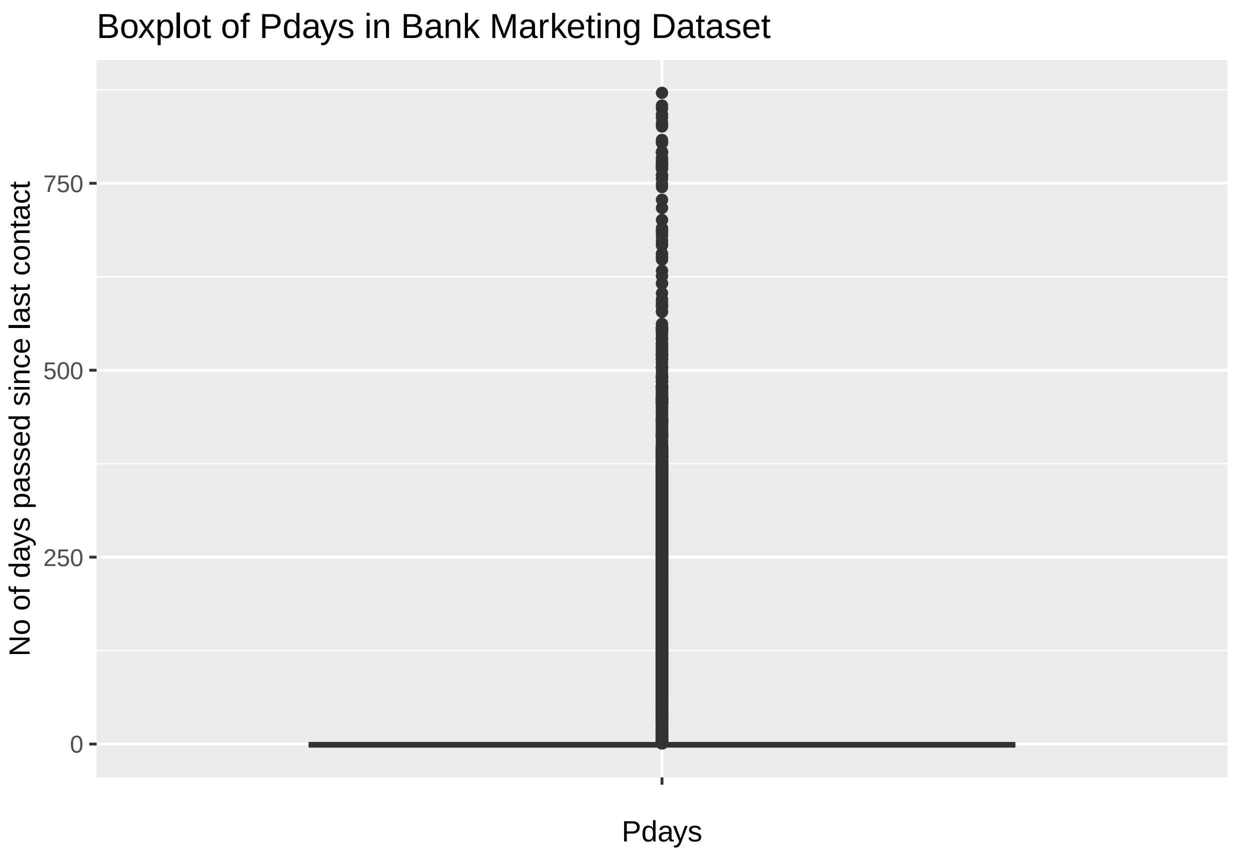
```
ggplot(data = clientdata, aes(x = "", y = duration)) + geom_boxplot() +
  labs(title = "Boxplot of Duration in Bank Marketing Dataset",
       x = "Duration",
       y = "Duration in seconds")
```



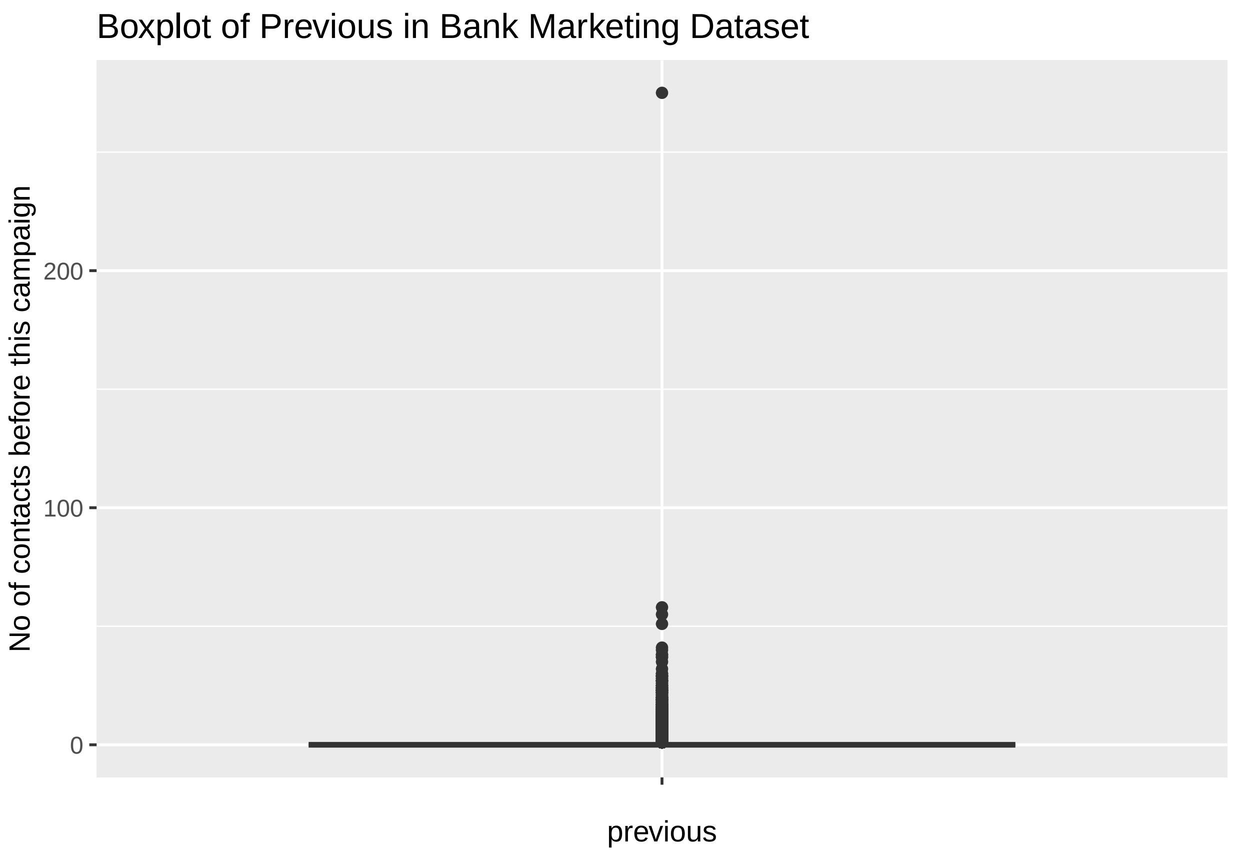
```r
ggplot(data = clientdata, aes(x = "", y = campaign)) + geom_boxplot() +
  labs(title = "Boxplot of Campaign in Bank Marketing Dataset",
       x = "Campaign",
       y = "No of contacts during this campaign")
```



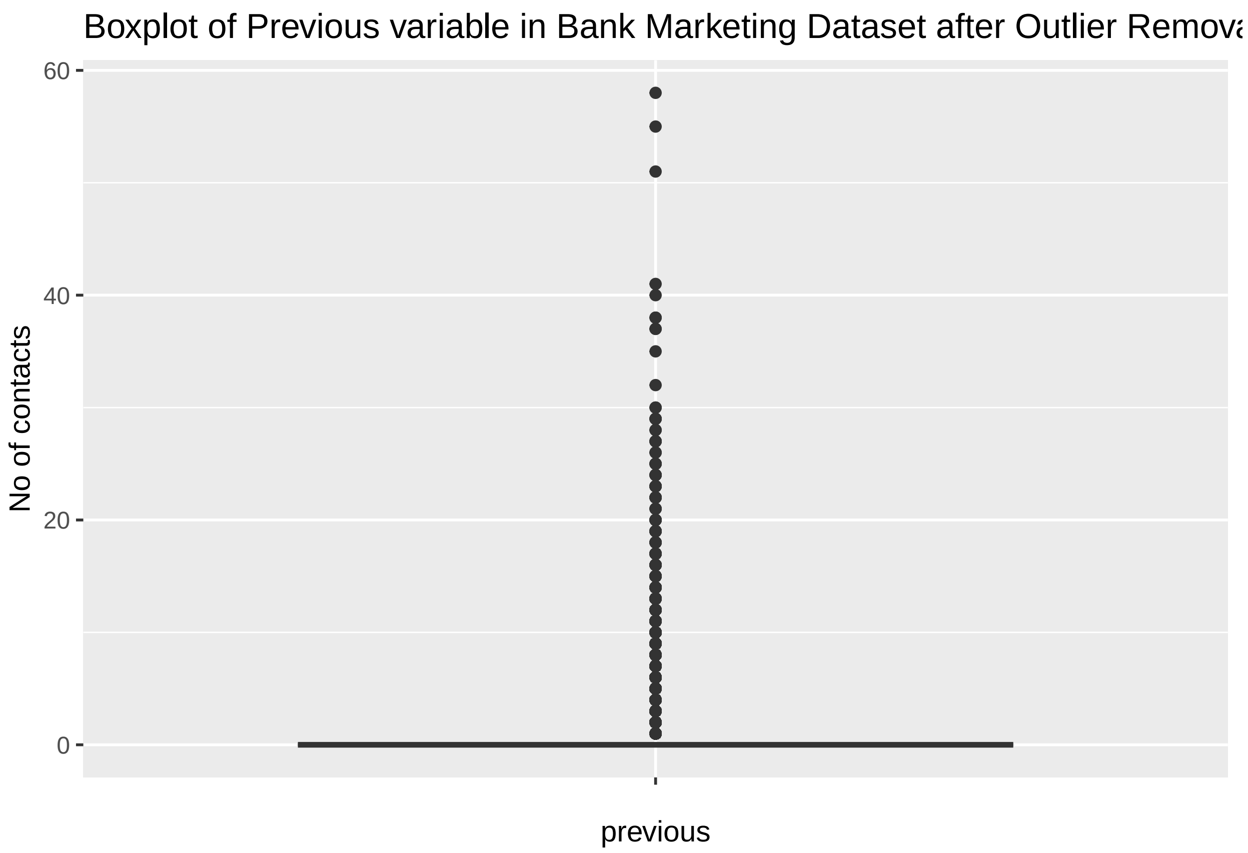
```r
ggplot(data = clientdata, aes(x = "", y = pdays)) + geom_boxplot() +
  labs(title = "Boxplot of Pdays in Bank Marketing Dataset",
       x = "Pdays",
       y = "No of days passed since last contact")
```



```
ggplot(data = clientdata, aes(x = "", y = previous)) + geom_boxplot() +
  labs(title = "Boxplot of Previous in Bank Marketing Dataset",
       x = "previous",
       y = "No of contacts before this campaign")
```



```
#Treating Outliers
clientdata <- clientdata %>% filter(previous != max(previous))
#Checking the box plot of Previous variable after removing one outlier
ggplot(data = clientdata, aes(x = "", y = previous)) + geom_boxplot() +
  labs(title = "Boxplot of Previous variable in Bank Marketing Dataset after Outlier Removal",
       x = "previous",
       y = "No of contacts")
```



```
summary(clientdata$previous)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0000  0.0000  0.0000  0.5745  0.0000 58.0000
```

Data Aggregation Now that the data set is complete and representing significant data, we move to aggregate the different categorical variables to maintain a better understanding of the population.

Below, we are visualizing the number of term deposits subscribed by the clients based on their job type. Based on the data provided, it appears that clients in management roles have subscribed to the highest number of term deposits, with a total of 1232 deposits, followed by technicians with 795 deposits and blue-collar workers with 670 deposits.

Clients in the services, student, unemployed, self-employed, and entrepreneur categories have subscribed to fewer term deposits, with the numbers ranging from 115 to 344.

This initial aggregate of job type does not take in to account that there are potentially more records with job type representative of management, therefor holding more weight. Normalization will assist in visualizing that the ratios of deposits made may be larger in other categories.

```
#Aggregating the data by job type
clientdata %>%
  group_by(job) %>%
  summarize(
    total_deposits = sum(deposit == 'yes')
    ) %>%
  arrange(desc(total_deposits))
```

```
## # A tibble: 12 x 2
##    job           total_deposits
##    <chr>                  <int>
##  1 management              1232
##  2 technician               795
##  3 blue-collar              670
##  4 admin.                   596
##  5 retired                  474
##  6 services                 344
##  7 student                  257
##  8 unemployed               189
##  9 self-employed            179
## 10 entrepreneur             115
## 11 housemaid                101
## 12 unknown                   33
```

The second aggregation shown below represents the total number of deposits subscribed for each month in the marketing campaign. Looking at the data, we can see that May had the highest number of deposits, with a total of 880. This suggests that the marketing campaign was particularly successful in May or that customers tend to have more income or savings during this time of year. Following up the aggregation of total deposits by month with total contacts by month, we can see that the most contacts to clients were made in May. This increase in contacts could be reason for the number of secured deposits in the month of May.

The second-highest number of deposits were subscribed in August, with a total of 658 deposits followed by July, with a total of 590 deposits. This indicates that the campaign was also successful in these months. Similar to the month of May, the number of contacts for August and July coincide with the level of secured deposits.

```
#Aggregating the data by month
clientdata %>%
  group_by(month) %>%
  summarize(
    total_deposits = sum(deposit == 'yes')
    ) %>%
  arrange(desc(total_deposits))
```

```
## # A tibble: 12 x 2
##    month total_deposits
##    <chr>          <int>
##  1 may              880
##  2 aug              658
##  3 jul              590
##  4 apr              532
##  5 jun              515
##  6 feb              409
##  7 nov              380
##  8 oct              307
##  9 sep              253
## 10 mar              228
## 11 jan              136
## 12 dec               97
```

```
clientdata %>%
  group_by(month) %>%
  summarize(
    total_contacts = sum(campaign)
    ) %>%
  arrange(desc(total_contacts))
```

```
## # A tibble: 12 x 2
##    month total_contacts
##    <chr>          <int>
##  1 may            31721
##  2 aug            23290
##  3 jul            22863
##  4 jun            15618
##  5 nov             7203
##  6 feb             5886
##  7 apr             5394
##  8 jan             2213
##  9 oct             1067
## 10 mar              994
## 11 sep              954
## 12 dec              444
```

As mentioned previously, a few clients that are being contacted with the current campaign have been contacted prior. The below table represents the total number of secured deposits based on the result of the previous campaign call (success, failure, unknown, other). The aggregation shows that when the previous outcome was unknown, the client was more likely to elect to make a deposit. The second most secured deposits is listed when the previous campaign was a success. We can infer that if a client was deemed a success in previous campaigns that they are a loyal customer and will continue to be secured deposits for the bank.

```
#Aggregating the data by poutcome
clientdata %>%
  group_by(poutcome) %>%
  summarize(
    total_deposits = sum(deposit == 'yes')
    ) %>%
  arrange(desc(total_deposits))
```

```
## # A tibble: 4 x 2
##   poutcome total_deposits
##   <chr>             <int>
## 1 unknown            3190
## 2 success             921
## 3 failure             589
## 4 other               285
```

Data Analysis and Visualization

Measures of Centrality The mean, median, and mode have been calculated in the summary data frame below of the seven numerical variables. These measures of centrality provide insight to the distribution of the numerical data and clue us in to any possibility of skewness that we may see in specific variables.

The variable that stands out against the others is balance. The balance in the “clientdata” data set is said to be a client’s average yearly balance in Euros. The mode of this variable is 0, implying that this client did not have a yearly balance which is highly unlikely and potentially makes the data unreliable to help predict the outcome of the dependent variable. Moving forward, this variable specifically will not be reliable in the relationship of balance to deposit result.

Similarly, the pdays, or the number of days that have passed since the client was last contacted, shows a mode of -1. The -1 represents that the client has not been contacted for a prior campaign meaning that most records included in the data set are new clients. Overall there are too many variations in the consistency of records that make is difficult to find patterns among relationships. This makes the understanding of loyal customers from previous, successful campaigns, harder to analyze when majority of clients are new to the Portuguese bank.

```r
#numerical measures of centrality --> mean, median, mode
#measures of centrality of age
mean_age <- round(mean(clientdata$age),2)
median_age <- round(median(clientdata$age),2)
mode_age <- clientdata %>%
  count(age) %>%
  filter(n == max(n)) %>%
  pull(age)

#measures of centrality of balance
mean_balance <- round(mean(clientdata$balance),2)
median_balance <- round(median(clientdata$balance),2)
mode_balance <- clientdata %>%
  count(balance) %>%
  filter(n == max(n)) %>%
  pull(balance)

#measures of centrality of day
mean_day <- round(mean(clientdata$day),2)
median_day <- round(median(clientdata$day),2)
mode_day <- clientdata %>%
  count(day) %>%
  filter(n == max(n)) %>%
  pull(day)

#measures of centrality of duration
mean_duration <- round(mean(clientdata$duration),2)
median_duration <- round(median(clientdata$duration),2)
mode_duration <- clientdata %>%
  count(duration) %>%
  filter(n == max(n)) %>%
  pull(duration)

#check if there are two modes
if (length(mode_duration) > 1) {
  mode_duration <- paste(mode_duration, collapse = ", ")
}

#measures of centrality of campaign
mean_campaign <- round(mean(clientdata$campaign),2)
median_campaign <- round(median(clientdata$campaign),2)
mode_campaign <- clientdata %>%
  count(campaign) %>%
  filter(n == max(n)) %>%
  pull(campaign)

#measures of centrality of pdays
mean_pdays <- round(mean(clientdata$pdays),2)
median_pdays <- round(median(clientdata$pdays),2)
mode_pdays <- clientdata %>%
  count(pdays) %>%
  filter(n == max(n)) %>%
  pull(pdays)

#measures of centrality of previous
mean_previous <- round(mean(clientdata$previous),2)
median_previous <- round(median(clientdata$previous),2)
mode_previous <- clientdata %>%
  count(previous) %>%
  filter(n == max(n)) %>%
  pull(previous)

summary_df <- data.frame(
  row.names = c("Age","Balance","Day","Duration","Campaign","pdays","Previous"),
  Mean = c(mean_age,
           mean_balance,
           mean_day,
           mean_duration,
           mean_campaign,
           mean_pdays,
           mean_previous),
  Median = c(median_age,
             median_balance,
             median_day,
             median_duration,
             median_campaign,
             median_pdays,
             median_previous),
  Mode = c(mode_age,
           mode_balance,
           mode_day,
           mode_duration,
           mode_campaign,
           mode_pdays,
           mode_previous)
)
summary_df
```

```
##             Mean Median    Mode
## Age        40.86     39      39
## Balance  1364.01    450       0
## Day        15.82     16      20
## Duration  258.32    180 90, 124
## Campaign    2.76      2       1
## pdays      40.26     -1      -1
## Previous    0.57      0       0
```

The most useful measure of centrality for categorical variables is mode, showing the value that occurs most often in the "clientdata" data set. The below table shows each categorical variable and the corresponding value that is seen most throughout the data.

```
#categorical measures of centrality --> mode
#mode of job
mode_job <- clientdata %>%
  count(job) %>%
  filter(n == max(n)) %>%
  pull(job)

#mode of marital
mode_marital <- clientdata %>%
  count(marital) %>%
  filter(n == max(n)) %>%
  pull(marital)

#mode of education
mode_education <- clientdata %>%
  count(education) %>%
  filter(n == max(n)) %>%
  pull(education)

#mode of contact
mode_contact <- clientdata %>%
  count(contact) %>%
  filter(n == max(n)) %>%
  pull(contact)

#mode of month
mode_month <- clientdata %>%
  count(month) %>%
  filter(n == max(n)) %>%
  pull(month)

#mode of poutcome
mode_poutcome <- clientdata %>%
  count(poutcome) %>%
  filter(n == max(n)) %>%
  pull(poutcome)

mode_df <- data.frame(
  row.names = c("Jobs","Marital","Education","Contact","Month","poutcome"),
  Mode = c(mode_job,mode_marital,mode_education,mode_contact,mode_month,mode_poutcome)
)
mode_df
```

```
##                  Mode
## Jobs      blue-collar
## Marital       married
## Education   secondary
## Contact      cellular
## Month             may
## poutcome      unknown
```

In order to find measures of centrality for binary variables, frequency tables have been created below. It is seen that variables default, loan, and deposit experience more no responses. Whereas the housing variable records more yes responses. In other words, when questioned, more people have current housing loans than not.

```
#binary measures of centrality --> frequency tables
default_table <- table(clientdata$default)
housing_table <- table(clientdata$housing)
loan_table <- table(clientdata$loan)
deposit_table <- table(clientdata$deposit)

freq_df <- rbind(default_table,housing_table,loan_table,deposit_table)
row.names(freq_df) <-  c("Default","Housing","Loan","Deposit")
freq_df
```

```
##            no   yes
## Default 41795   773
## Housing 18941 23627
## Loan    35768  6800
## Deposit 37583  4985
```

Distribution Outside of the main measures of centrality, mean, median, and mode, different measurements can be made in order to make inferences about the distribution of the data. Below, we find the range, variance, standard deviation, and interquartile range of the numerical variables of the "clientdata".

It is observed that balance, duration, and pdays have a higher standard deviation meaning that there is a wide range of values in these variables in relation to the mean. This is an expected observation since each client is going to be in a different place financially and because duration is recorded in seconds. These two variables also represent the largest interquartile ranges in the below table, validating that the middle portion of data of these variables experiences more variability than that of the other five numerical variables.

```
#measures of distribution of age
var_age <- round(var(clientdata$age),2)
sd_age <- round(sd(clientdata$age),2)
IQR_age <- round(IQR(clientdata$age),2)
q1_age <- summary(clientdata$age)[2]
q3_age <- summary(clientdata$age)[5]
range_age <- summary(clientdata$age)[6] - summary(clientdata$age)[1]

#measures of distribution of balance
var_balance <- round(var(clientdata$balance),2)
sd_balance <- round(sd(clientdata$balance),2)
IQR_balance <- round(IQR(clientdata$balance),2)
q1_balance <- summary(clientdata$balance)[2]
q3_balance <- summary(clientdata$balance)[5]
range_balance <- summary(clientdata$balance)[6] - summary(clientdata$balance)[1]

#measures of distribution of day
var_day <- round(var(clientdata$day),2)
```

```r
sd_day <- round(sd(clientdata$day),2)
IQR_day <- round(IQR(clientdata$day),2)
q1_day <- summary(clientdata$day)[2]
q3_day <- summary(clientdata$day)[5]
range_day <- summary(clientdata$day)[6] - summary(clientdata$day)[1]

#measures of distribution of duration
var_duration <- round(var(clientdata$duration),2)
sd_duration <- round(sd(clientdata$duration),2)
IQR_duration <- round(IQR(clientdata$duration),2)
q1_duration <- summary(clientdata$duration)[2]
q3_duration <- summary(clientdata$duration)[5]
range_duration <- summary(clientdata$duration)[6] - summary(clientdata$duration)[1]

#measures of distribution of campaign
var_campaign <- round(var(clientdata$campaign),2)
sd_campaign <- round(sd(clientdata$campaign),2)
IQR_campaign <- round(IQR(clientdata$campaign),2)
q1_campaign <- summary(clientdata$campaign)[2]
q3_campaign <- summary(clientdata$campaign)[5]
range_campaign <- summary(clientdata$campaign)[6] - summary(clientdata$campaign)[1]

#measures of distribution of pdays
var_pdays <- round(var(clientdata$pdays),2)
sd_pdays <- round(sd(clientdata$pdays),2)
IQR_pdays <- round(IQR(clientdata$pdays),2)
q1_pdays <- summary(clientdata$pdays)[2]
q3_pdays <- summary(clientdata$pdays)[5]
range_pdays <- summary(clientdata$pdays)[6] - summary(clientdata$pdays)[1]

#measures of distribution of previous
var_previous <- round(var(clientdata$previous),2)
sd_previous <- round(sd(clientdata$previous),2)
IQR_previous <- round(IQR(clientdata$previous),2)
q1_previous <- summary(clientdata$previous)[2]
q3_previous <- summary(clientdata$previous)[5]
range_previous <- summary(clientdata$previous)[6] - summary(clientdata$previous)[1]
variability_df = data.frame(
  row.names = c("Age","Balance","Day","Duration","Campaign","pdays","Previous"),
  Variance = c(var_age,var_balance,var_day,var_duration,var_campaign,var_pdays,var_previous),
  Std_Dev = c(sd_age,sd_balance,sd_day,sd_duration,sd_campaign,sd_pdays,sd_previous),
  IQR = c(IQR_age,IQR_balance,IQR_day,IQR_duration,IQR_campaign,IQR_pdays,IQR_previous),
  First_Quartile = c(q1_age,q1_balance,q1_day,q1_duration,q1_campaign,q1_pdays,q1_previous),
  Third_Quartile = c(q3_age,q3_balance,q3_day,q3_duration,q3_campaign,q3_pdays,q3_previous),
  Range = c(range_age,range_balance,range_day,range_duration,range_campaign,range_pdays,range_previous)
)
variability_df
```

```
##             Variance Std_Dev  IQR First_Quartile Third_Quartile  Range
## Age           109.09   10.44   15             33             48     77
## Balance   9373994.73 3061.70 1358             72           1430 110146
## Day            69.25    8.32   13              8             21     30
## Duration    66147.57  257.19  216            103            319   3881
## Campaign       9.54    3.09    2              1              3     62
## pdays      10050.82  100.25    0             -1             -1    872
## Previous       3.66    1.91    0              0              0     58
```

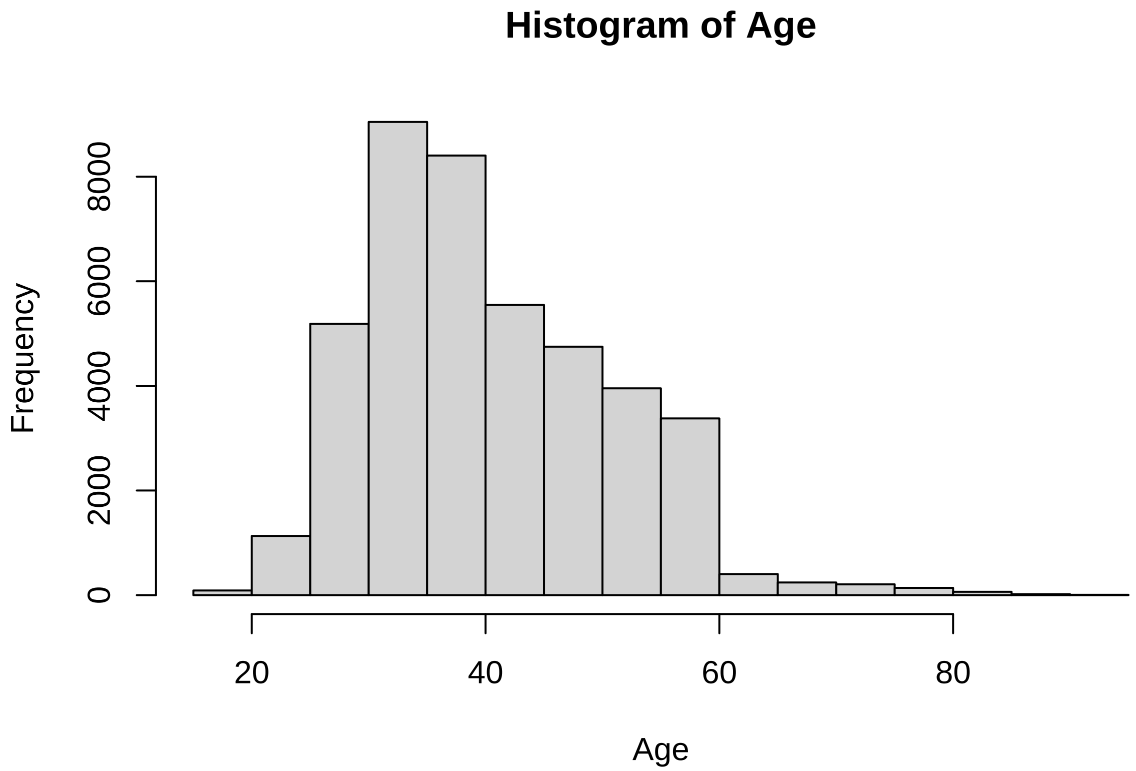
In order to have an idea of the distribution of categorical and binary variables, the below frequency tables have been created. The frequencies of each variable provides insight to why certain categories may be more present in the deposit variable.

```
categorical_vars <- c("job","marital","education","contact", "month", "poutcome")
for (var in categorical_vars){
  freq_table <- table(clientdata[[var]])
  print(paste("Frequency Table For", var))
  print(freq_table)
}
```

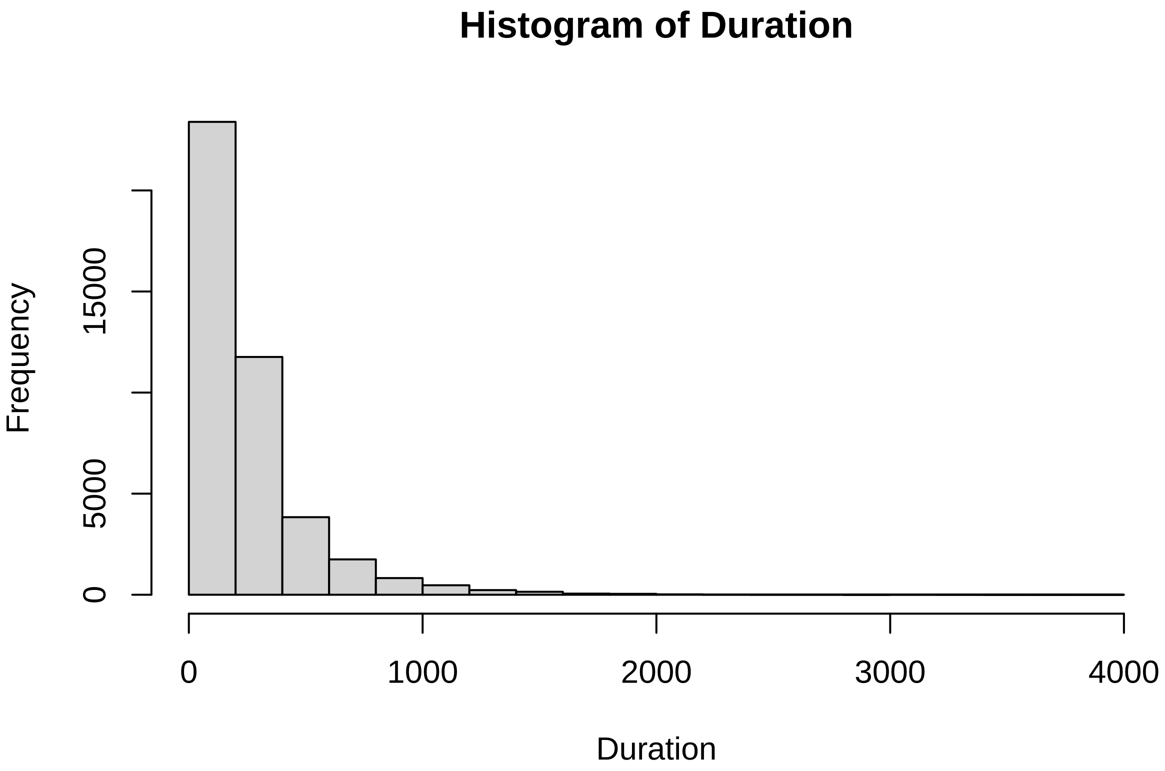
```
## [1] "Frequency Table For job"
## 
##         admin.   blue-collar  entrepreneur     housemaid    management 
##           4849          9144          1402          1160          8935 
##        retired self-employed      services       student    technician 
##           2124          1485          3914           890          7171 
##     unemployed       unknown 
##           1222           272 
## [1] "Frequency Table For marital"
## 
## divorced  married   single 
##     4901    25650    12017 
## [1] "Frequency Table For education"
## 
##   primary secondary  tertiary   unknown 
##      6429     21845     12579      1715 
## [1] "Frequency Table For contact"
## 
##  cellular telephone   unknown 
##     27608      2702     12258 
## [1] "Frequency Table For month"
## 
##   apr   aug   dec   feb   jan   jul   jun   mar   may   nov   oct   sep 
##  2757  5941   202  2465  1318  6490  5022   445 12951  3740   695   542 
## [1] "Frequency Table For poutcome"
## 
## failure   other success unknown 
##    4640    1711    1431   34786
```

The histograms below further analyze the distribution of individual variables. Similar to frequency tables, the histograms provide a visualization of the frequency numerical variables and the variability across the variables.

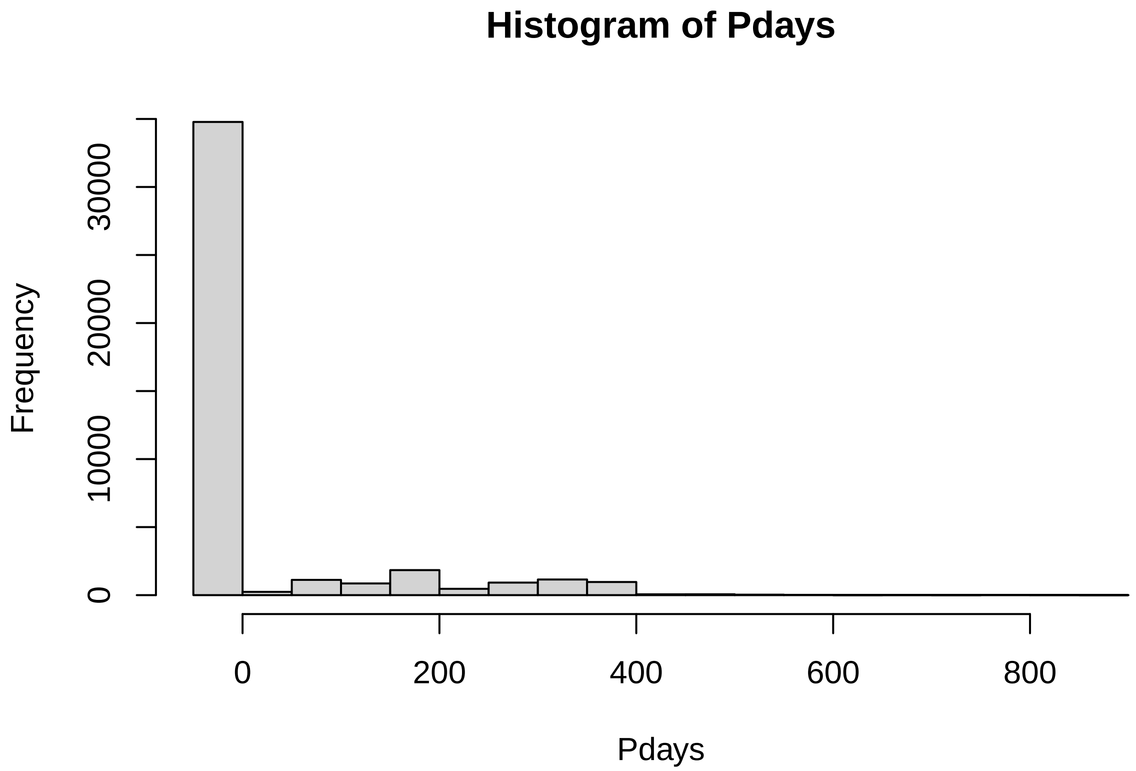
```
#histogram of age
hist(clientdata$age, main = "Histogram of Age", xlab = "Age", ylab = "Frequency")
```



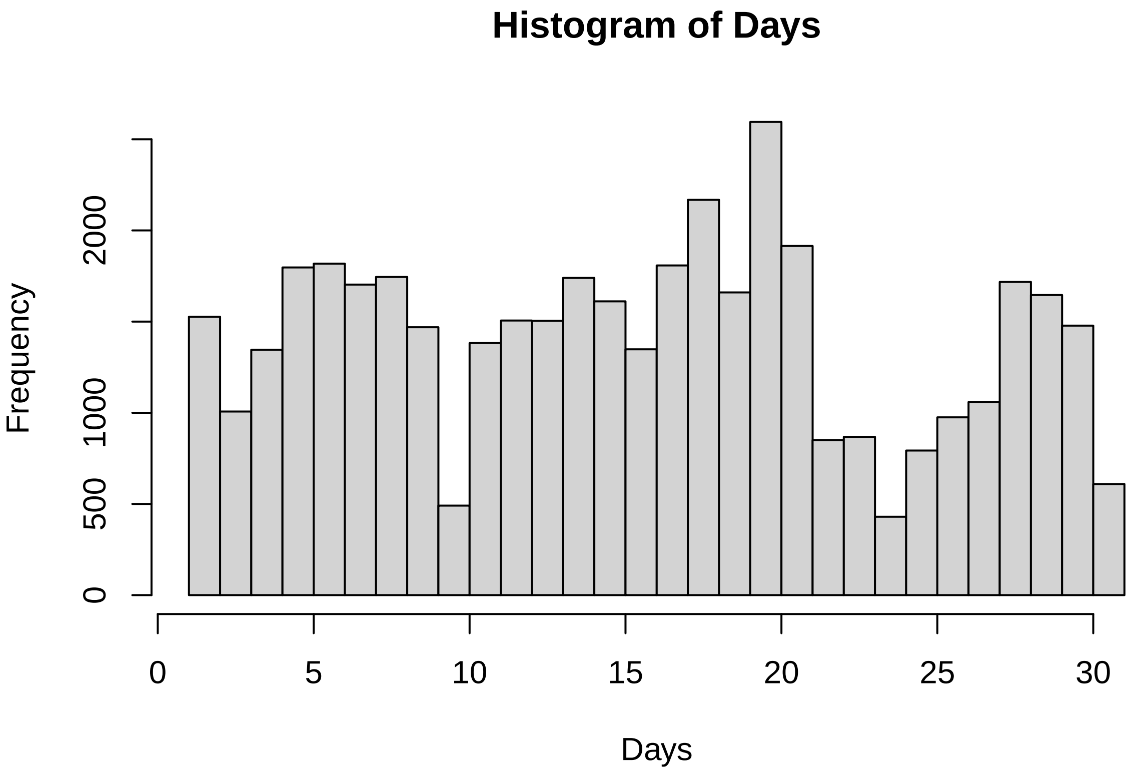
```
#histogram of duration
hist(clientdata$duration, main = "Histogram of Duration", xlab = "Duration", ylab = "Frequency")
```



```r
#histogram of pdays
hist(clientdata$pdays, main = "Histogram of Pdays", xlab = "Pdays", ylab = "Frequency")
```



```
#histogram of days
hist(clientdata$day, main="Histogram of Days", breaks = 40, xlab= "Days", ylab="Frequency")
```



Correlations

Numerical Correlations To determine what variables have a relationship and a potential significance to the outcome of the deposit variable, a correlation matrix has been created for all seven numerical variables. To visualize the correlation between these variables and the binary variable, deposit, deposit has been translated to the value 1 for response "yes", and 0 for "no".

A strong positive and/or negative relationship between variables is represented by a coefficient close to +/-1. In the matrix below of numeric variables, there does not appear to be any strong relationships, negative or positive. Specifically, looking at the coefficients of deposit vs. all other variables, no significance is recorded.

```
#correlation matrix
cols <- clientdata[, c("age",
                       "balance",
                       "duration",
                       "campaign",
                       "pdays",
                       "previous",
                       "day",
                       "deposit")]

# convert deposit to 1,0
cols$deposit <- ifelse(cols$deposit == "yes", 1, 0)
#correlation matrix
```

```
corr_matrix <- cor(cols, use="pairwise.complete.obs")
options(scipen = 999) # disable scientific notation
print(corr_matrix, digits = 6) # specify the number of decimal places
```

```
##                     age     balance     duration    campaign        pdays
## age       1.0000000000  0.09652217 -0.004808921  0.00634282 -0.023417892
## balance   0.0965221685  1.00000000  0.018325861 -0.01285158  0.005042925
## duration -0.0048089208  0.01832586  1.000000000 -0.08414691 -0.000809568
## campaign  0.0063428192 -0.01285158 -0.084146909  1.00000000 -0.088637126
## pdays    -0.0234178922  0.00504292 -0.000809568 -0.08863713  1.000000000
## previous -0.0000577061  0.02217550  0.000600737 -0.03867572  0.540523992
## day      -0.0092850018  0.00495808 -0.029508782  0.16362836 -0.094202052
## deposit   0.0211766287  0.05203644  0.396054950 -0.07282014  0.103454358
##               previous         day    deposit
## age      -0.0000577061 -0.00928500  0.0211766
## balance   0.0221754976  0.00495808  0.0520364
## duration  0.0006007369 -0.02950878  0.3960549
## campaign -0.0386757248  0.16362836 -0.0728201
## pdays     0.5405239917 -0.09420205  0.1034544
## previous  1.0000000000 -0.05874543  0.1129454
## day      -0.0587454338  1.00000000 -0.0301474
## deposit   0.1129454455 -0.03014737  1.0000000
```

Tetrachoric Correlation With no apparent significance in the numerical correlation matrix, we move to create a correlation between categorical variables in order to discover any relationship for the deposit variable that will help us predict the outcome of a marketing campaign.

A tetrachoric correlation, a correlation of binary and categorical variables, is performed on deposit and age group, job, marital, and education. The tetrachoric correlation shows similar results to that of the numerical correlation matrix. There are not any categorical variables that have a strong positive or negative relationship with the outcome of the deposit variable.

```
# Creating age groups
clientdata$age_group <- cut(clientdata$age, breaks = c(15, 29, 39, 49, 59, 69, 79, 95),
                            labels = c("18-29",
                                       "30-39",
                                       "40-49",
                                       "50-59",
                                       "60-69",
                                       "70-79",
                                       "80-95"))

# Converting age to a factor
clientdata$age_group <- factor(clientdata$age_group,
                               levels = c("18-29",
                                          "30-39",
                                          "40-49",
                                          "50-59",
                                          "60-69",
                                          "70-79",
                                          "80-95"))

# Save the modified data back to same file
```

```r
write.csv(clientdata, "clientdata.csv", row.names = FALSE)

# categorical columns
cat_data <- clientdata[, c("deposit",
                           "age_group",
                           "marital",
                           "job",
                           "education",
                           "default",
                           "housing",
                           "loan",
                           "poutcome")]

# convert deposit to 1,0
cat_data$deposit <- ifelse(cat_data$deposit == "yes", 1, 0)
#view(cat_data)

#convert categorical variables to factors
cat_data$age_group <- factor(cat_data$age_group)
cat_data$marital <- factor(cat_data$marital)
cat_data$job <- factor(cat_data$job)
cat_data$education <- factor(cat_data$education)
cat_data$default <- factor(cat_data$default)
cat_data$housing <- factor(cat_data$housing)
cat_data$loan <- factor(cat_data$loan)
cat_data$poutcome <- factor(cat_data$poutcome)
corr_matrix_cat <- hetcor(as.matrix(cat_data), use = "pairwise.complete.obs")
corr_matrix_cat$correlations
```

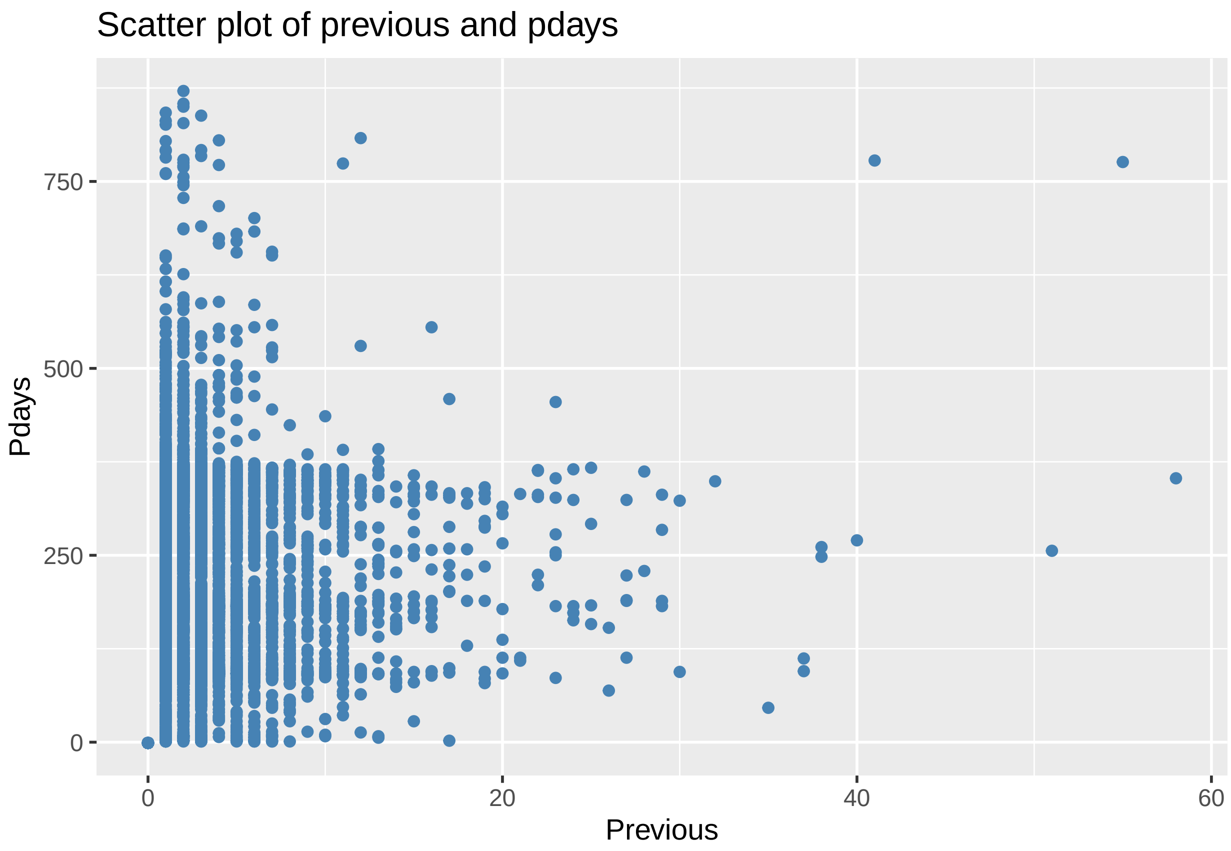
```
##                deposit    age_group     marital         job   education
## deposit    1.000000000  0.005336988  0.08102663  0.05812812  0.11705285
## age_group  0.005336988  1.000000000 -0.47645315  0.01127200 -0.11364708
## marital    0.081026631 -0.476453147  1.00000000  0.05076655  0.13513019
## job        0.058128119  0.011272000  0.05076655  1.00000000  0.17517312
## education  0.117052850 -0.113647078  0.13513019  0.17517312  1.00000000
## default   -0.120926963 -0.054783543 -0.02938109 -0.01303370 -0.03401202
## housing   -0.278320224 -0.207109454 -0.02222982 -0.16628768 -0.12229475
## loan      -0.189149867 -0.022276729 -0.08348077 -0.05989160 -0.07675838
## poutcome  -0.247738001  0.002619595 -0.03264672  0.02063477 -0.04380116
##               default     housing        loan     poutcome
## deposit   -0.12092696 -0.27832022 -0.18914987 -0.247738001
## age_group -0.05478354 -0.20710945 -0.02227673  0.002619595
## marital   -0.02938109 -0.02222982 -0.08348077 -0.032646721
## job       -0.01303370 -0.16628768 -0.05989160  0.020634767
## education -0.03401202 -0.12229475 -0.07675838 -0.043801159
## default    1.00000000 -0.02265491  0.26928451  0.204441026
## housing   -0.02265491  1.00000000  0.08011348 -0.151041283
## loan       0.26928451  0.08011348  1.00000000  0.053233487
## poutcome   0.20444103 -0.15104128  0.05323349  1.000000000
```

After creating the two correlation matrices for both numerical and categorical variables and finding no significant relationship with the deposit variable it is difficult to single in on select independent variables. Moving forward, the dependent variable, deposit, will be included in visualizations with multiple other variables to attempt to determine a relationship beyond the correlation coefficient. At this point in the analysis, the assumption is that all other variables will be independent variables in the prediction of the deposit outcome.

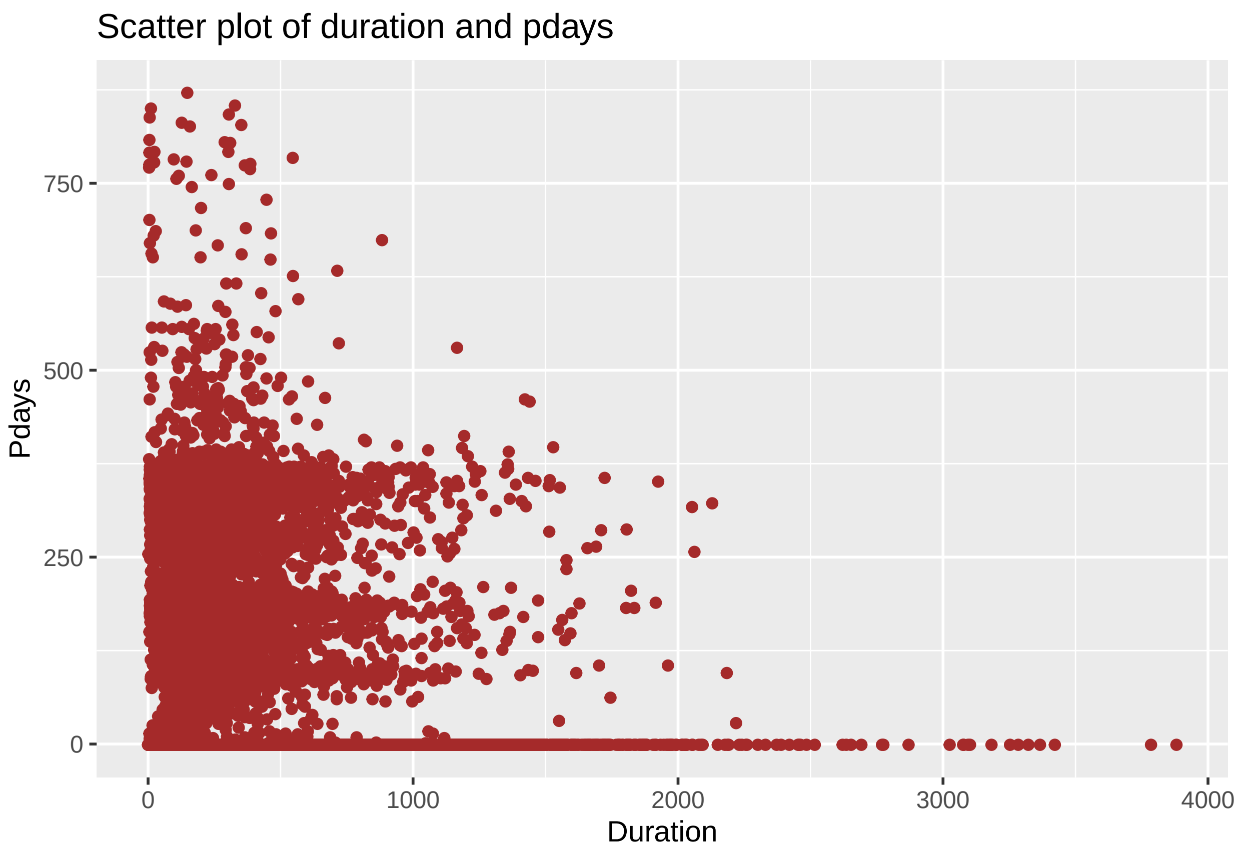
Exploratory Visualizations Knowing that the correlation matrices do not support the visualization of relationships among variables, different types of visualizations have been created below to further understand the variables. Below, scatterplots have been created among different independent variables to determine any relations.

The plots do not exemplify any positive or negative relationship, as we expected after the correlation matrices results. However, the relationship between duration and both the pdays and previous variables is more closely comparable to an exponential relationship.

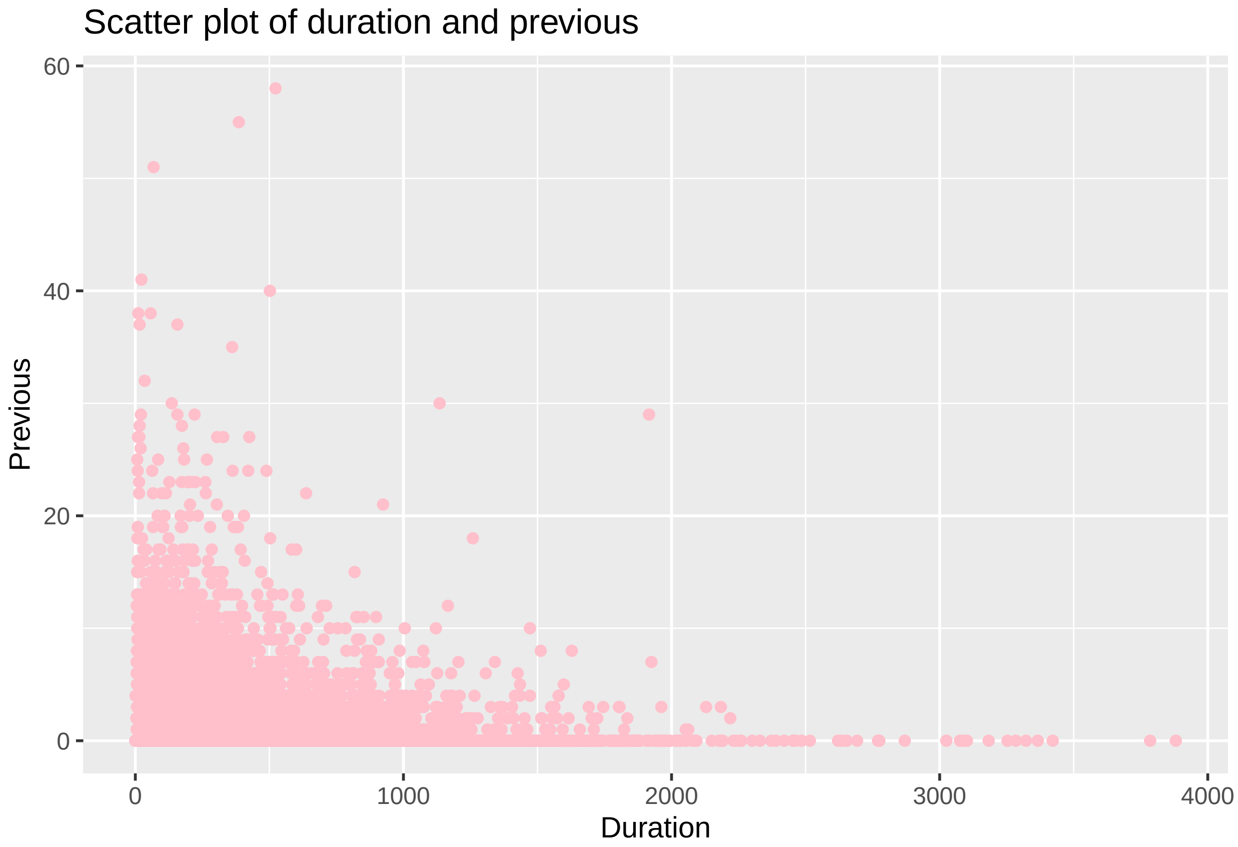
```
ggplot(clientdata, aes(x = previous, y = pdays)) +
  geom_point(color = "steelblue") +
  labs(title = "Scatter plot of previous and pdays", x = "Previous", y = "Pdays")
```



```
ggplot(clientdata, aes(x = duration, y = pdays)) +
  geom_point(color = "brown") +
  labs(title = "Scatter plot of duration and pdays", x = "Duration", y = "Pdays")
```



```
ggplot(clientdata, aes(x = duration, y = previous)) +
  geom_point(color = "pink") +
  labs(title = "Scatter plot of duration and previous", x = "Duration", y = "Previous")
```



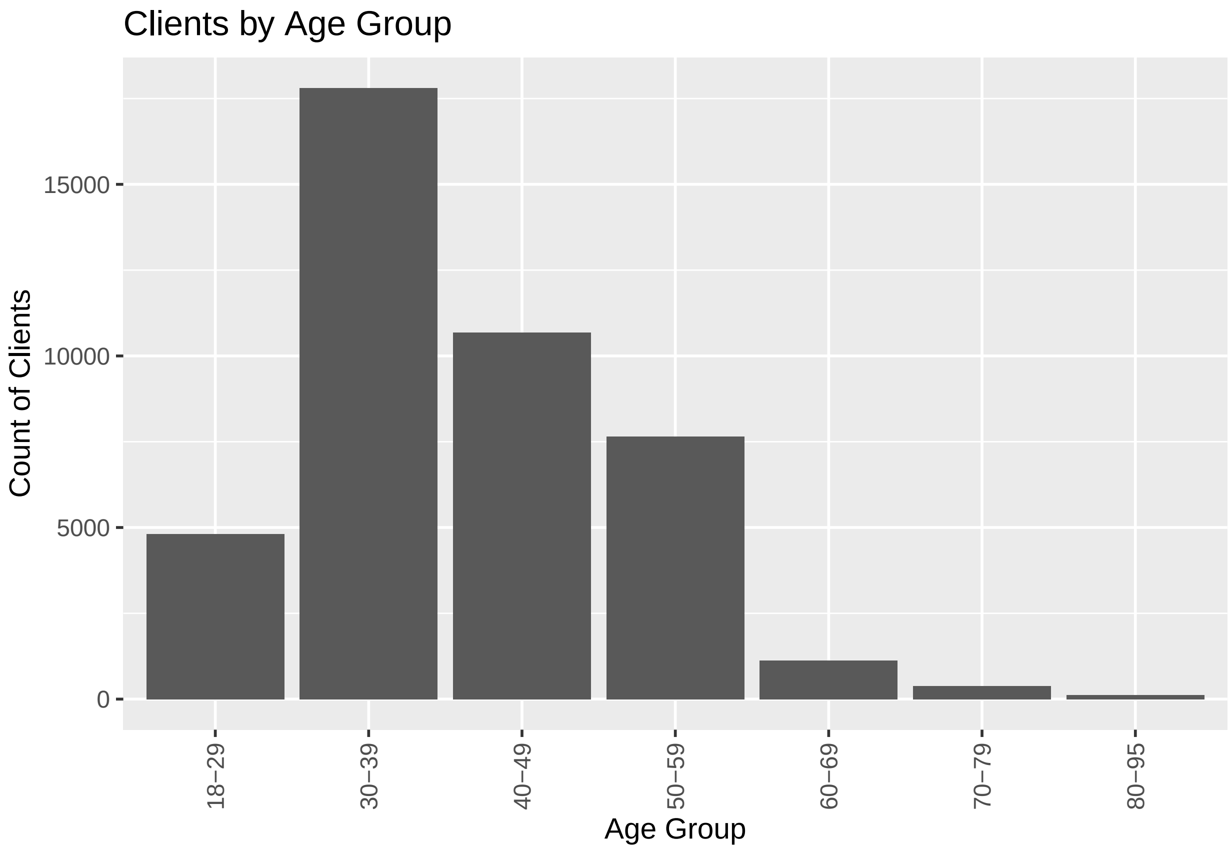
Normalizations and Visualizations Since both the numerical and categorical correlation matrices did not represent strong positive or negative relationships, we are moving to visualize the dependent variable, deposit, against independent variables that we believe may have a significance. In order to properly compare the visualizations against one another, normalization was needed in the independent variables to hold an even weight by counts of observations. The independent variable was normalized in all cluster bar charts to ensure that if a group in that variable had more occurrences in the data set it wouldn't affect the results. For example, there are more ages from 30-39 so if a normalization wasn't done there would be more occurrences of 30-39 for deposits but with a normalization we can observe by ration whether a age group does more or less deposits.

The first independent variable to be visualized, age, was aggregated into an age range so that visually we can see the different ages. The first graph shows how there is more 30-39 year olds in our age groups so a normalization was a necessity so that the second graph wouldn't be biased.

```
# Group data by age group and deposit, and count number of observations
agecount <- clientdata %>%
  group_by(age_group) %>%
  summarize(count = n()) %>%
  ungroup()

# Create bar chart of age groups
ggplot(agecount, aes(x = age_group, y = count)) +
  geom_bar(stat = "identity") +
  labs(title = "Clients by Age Group",
       x = "Age Group",
```

```r
       y = "Count of Clients") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))
```

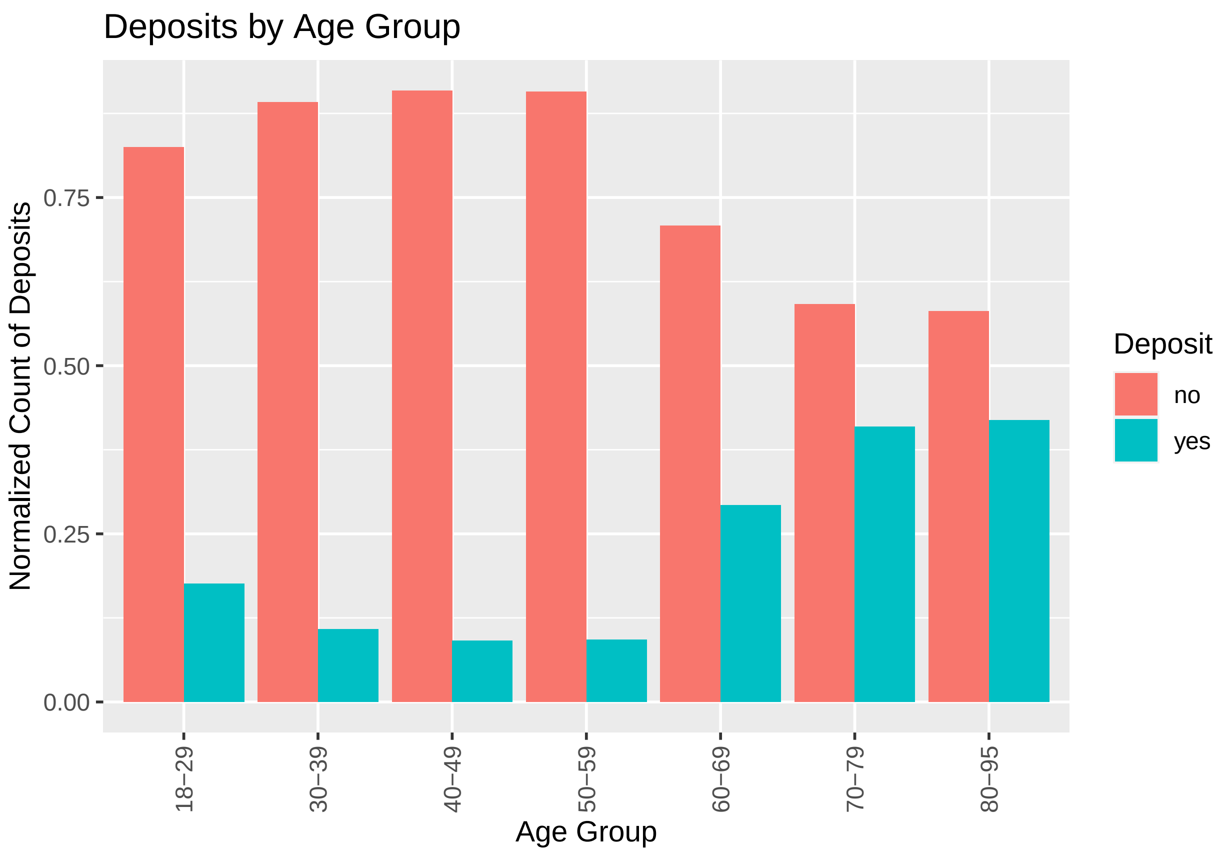


```r
# group data by age and deposit and calculate the count
age_depo_count <- clientdata %>%
  group_by(age_group, deposit) %>%
  summarize(count = n()) %>%
  ungroup()

# join two datasets by age group and deposit
age_depo_count_norm <- left_join(age_depo_count, agecount, by = "age_group")
#view(age_depo_count_norm)

# calculate the normalized count of deposits
age_depo_count_norm$count_norm <- age_depo_count_norm$count.x / age_depo_count_norm$count.y

# clustered bar chart deposit vs, age
ggplot(age_depo_count_norm, aes(x = age_group, y = count_norm, fill = deposit)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Deposits by Age Group",
       x = "Age Group",
       y = "Normalized Count of Deposits",
       fill = "Deposit") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))
```



After normalization, we can see that all age groups are subject to recording more no's than secured deposits. However, the age group 80-95 seems to record the highest value of secured deposits based on the normalized values. Since there is a wide spread between the recorded answers, it is best if age group is not included as an independent variable in the final model as most recorded a no for deposit within the range, not independent records.

Similar normalization techniques of the independent variable a clustered bar chart has been created to compare the job type to the deposit type of the client. This bar chart will show which job type has the most deposits recorded and separate these recordings out to "yes" and "no".

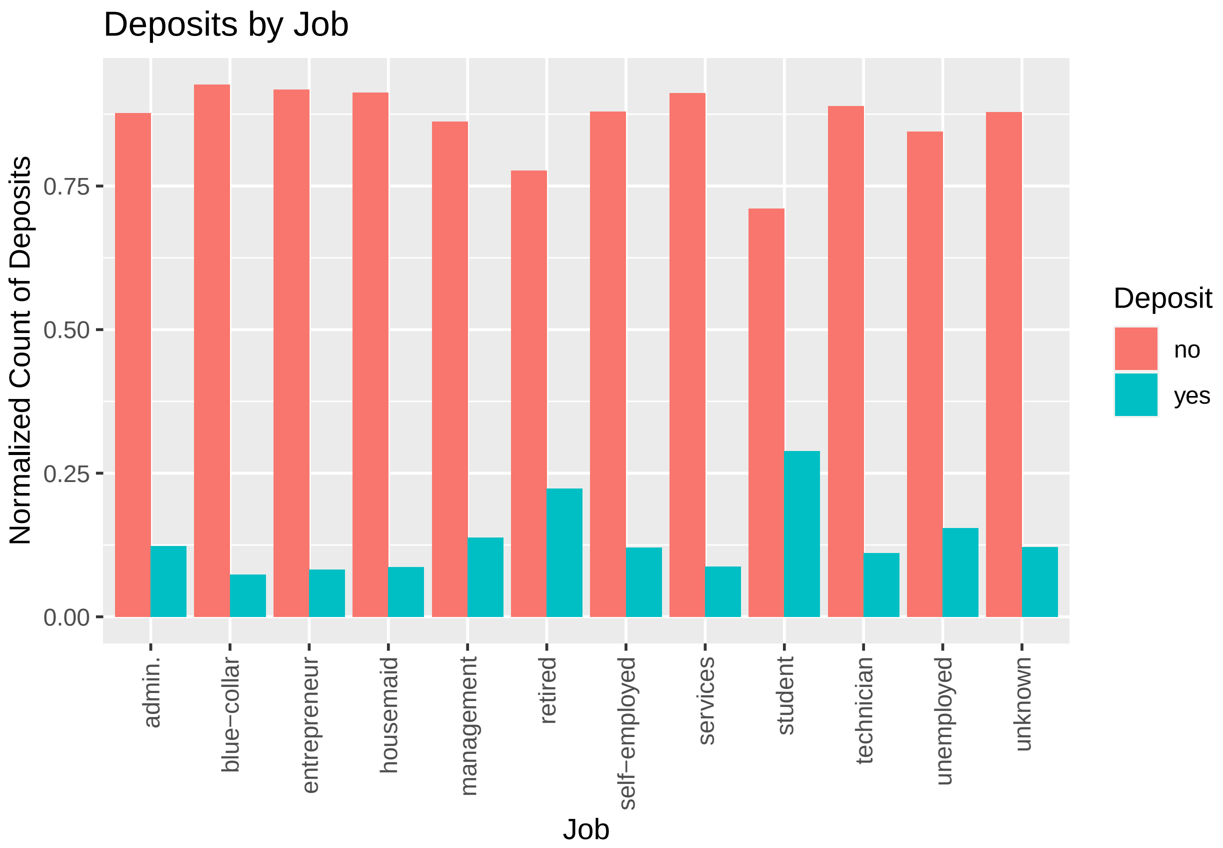
```r
# Group data by  job and deposit, and count number of observations
jobcount <- clientdata %>%
  group_by(job) %>%
  summarize(count1 = n()) %>%
  ungroup()
#view(jobcount)

# group data by job and deposit and calculate the count
job_depo_count <- clientdata %>%
  group_by(job, deposit) %>%
  summarize(count = n()) %>%
  ungroup()
#view(job_depo_count)

# join two datasets by job and deposit
job_depo_count_norm <- left_join(job_depo_count, jobcount, by = c("job"))
#view(job_depo_count_norm)
```

```r
# calculate the normalized count of deposits
job_depo_count_norm$count_norm <- job_depo_count_norm$count / job_depo_count_norm$count1

# clustered bar chart deposit vs job type
ggplot(job_depo_count_norm, aes(x = job, y = count_norm, fill = deposit)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Deposits by Job",
       x = "Job",
       y = "Normalized Count of Deposits",
       fill = "Deposit") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))
```



Similar to "Deposits by Age Group" the above bar chart experiences many more no's recorded. However, the chart allows us to visualize that of the job types, students recorded the most secured deposits. This is interesting because logically, one would assume that students are not going to be placing large deposits financially.

Below, we have normalized the deposit variable again to compare it to the marital status of the clients.

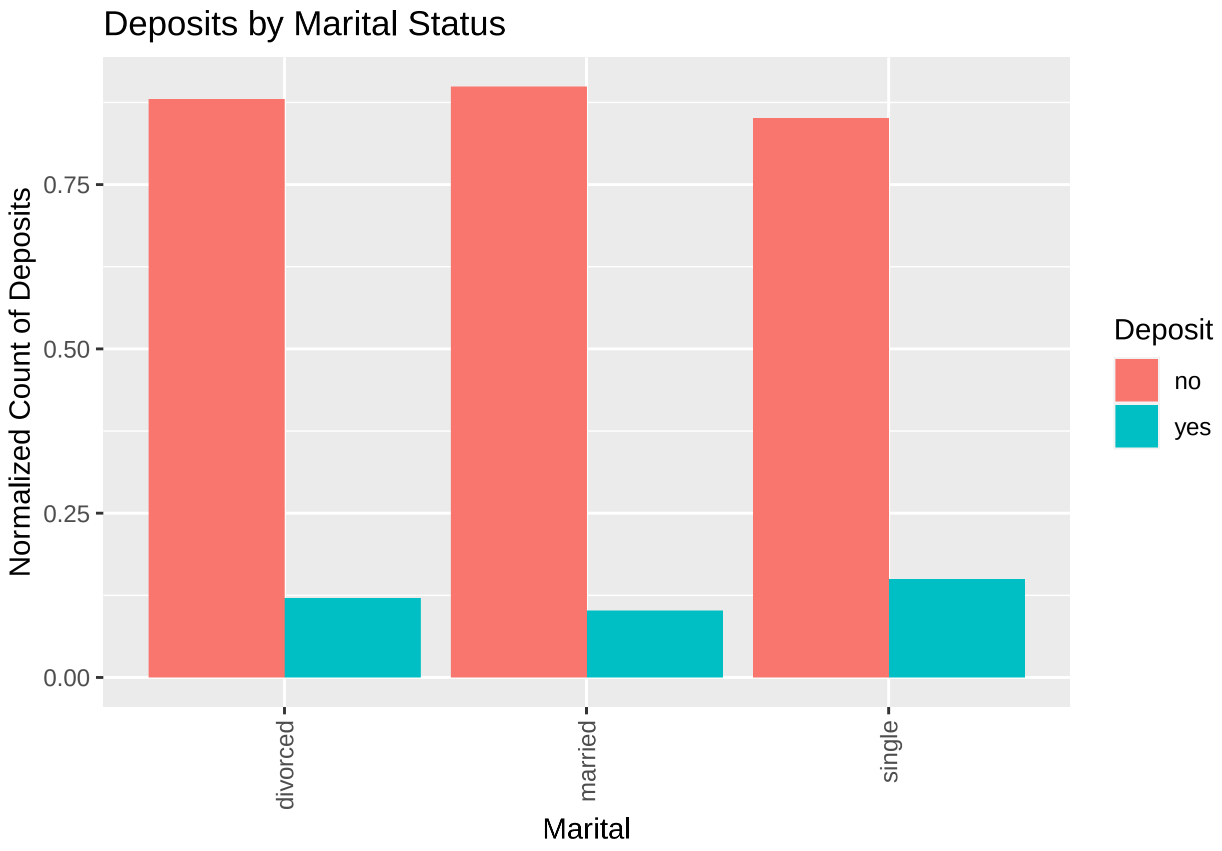
```r
# Group data by marital and deposit, and count number of observations
maritalcount <- clientdata %>%
  group_by(marital) %>%
  summarize(count1 = n()) %>%
  ungroup()
#view(maritalcount)
```

```r
# group data by marital and deposit and calculate the count
marital_depo_count <- clientdata %>%
  group_by(marital, deposit) %>%
  summarize(count = n()) %>%
  ungroup()
#view(marital_depo_count)

# join two datasets by marital and deposit
marital_depo_count_norm <- left_join(marital_depo_count, maritalcount, by = c("marital"))
#view(marital_depo_count_norm)

# calculate the normalized count of deposits
marital_depo_count_norm$count_norm <-
  marital_depo_count_norm$count / marital_depo_count_norm$count1

# clustered bar chart deposit vs marital status
ggplot(marital_depo_count_norm, aes(x = marital, y = count_norm, fill = deposit)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Deposits by Marital Status",
       x = "Marital",
       y = "Normalized Count of Deposits",
       fill = "Deposit") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))
```



The “Deposits by Marital Status” clustered bar chart represents that of the deposits that recorded a yes, single clients accounted for more. Once again, the chart is taken over by recorded “no’s”.

The last variable suspected to have an affect on the deposit variable is the previous outcome of an earlier campaign call made to the same client, poutcome. Again, the deposit variable has been normalized.

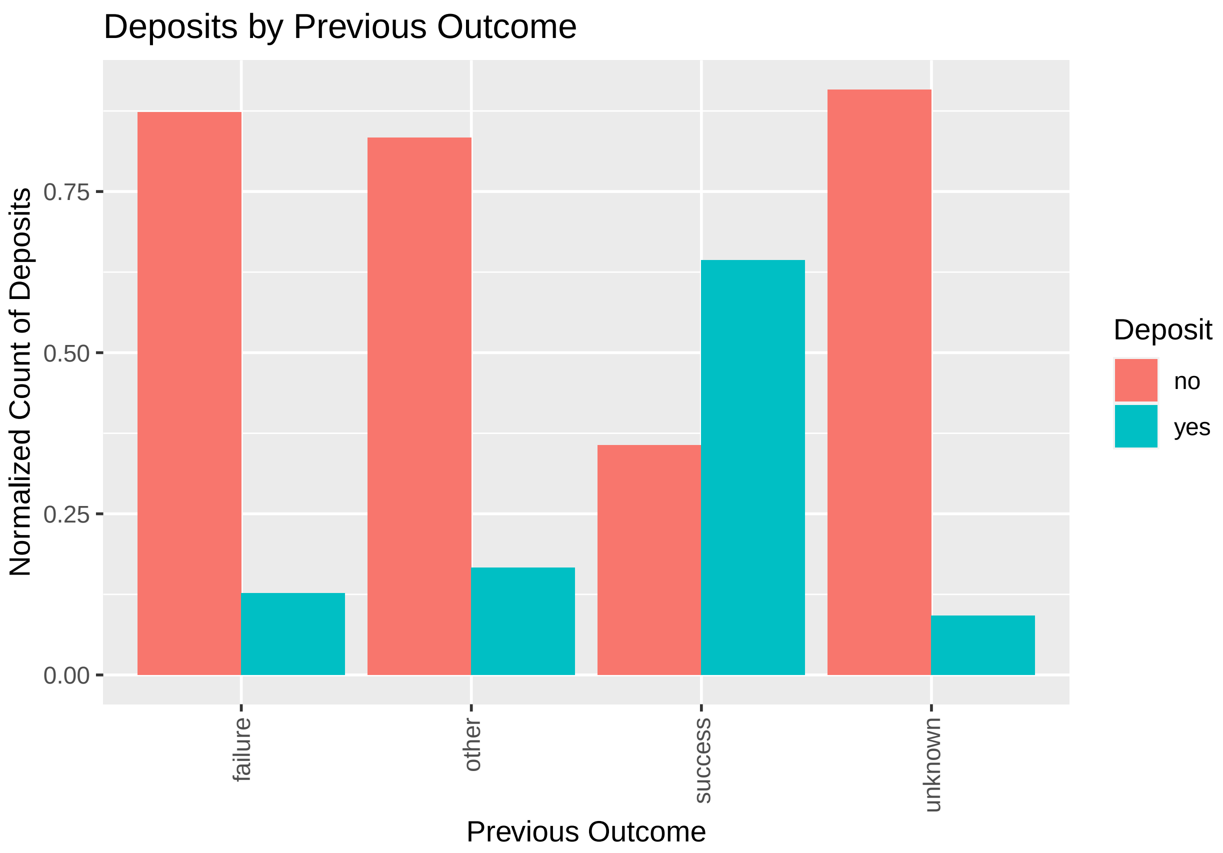
```r
# Group data by previous outcome and deposit, and count number of observations
previousoutcomecount <- clientdata %>%
  group_by(poutcome) %>%
  summarize(count1 = n()) %>%
  ungroup()
#view(maritalcount)

# group data by poutcome and deposit and calculate the count
poutcome_depo_count <- clientdata %>%
  group_by(poutcome, deposit) %>%
  summarize(count = n()) %>%
  ungroup()
#view(poutcome_depo_count)

# join two datasets by poutcome and deposit
poutcome_depo_count_norm <- left_join(poutcome_depo_count, previousoutcomecount, by = c("poutcome"))
#view(poutcome_depo_count_norm)

# calculate the normalized count of deposits
poutcome_depo_count_norm$count_norm <-
  poutcome_depo_count_norm$count / poutcome_depo_count_norm$count1

# clustered bar chart deposit vs previous outcome
ggplot(poutcome_depo_count_norm, aes(x = poutcome, y = count_norm, fill = deposit)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Deposits by Previous Outcome",
       x = "Previous Outcome",
       y = "Normalized Count of Deposits",
       fill = "Deposit") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))
```



The “Deposits by Previous Outcome” bar chart records the only variable that has been visualized to have more “yes” records than “no”. This variable is going to be one that is significant to include in the final model in predicting the deposit variable. The bank recorded more successful deposits when the client has previously made a deposit in an earlier campaign. This further supports the idea that the bank has loyal customers that are continuing to do business with the Portuguese bank.

Data Analytics

Supervised Learning - Logistic Regression After visualizing the dependent variable, deposit, against independent variables that we felt would have a relationship, although small, we are going to move forward with creating an overall model to then be transitioned to better fit. The initial model below will include all independent variables from the “clientdata” data set after normalization. A supervised learning model, logistic regression, will be used to to help predict our binary, dependent variable. Supervised learning will allow us to predict the deposit outcome based on the outcomes of previous records from the “clientdata” data set.

```
#creating categorical variables into factors
clientdata$job <- as.factor(clientdata$job)
clientdata$marital <- as.factor(clientdata$marital)
clientdata$education <- as.factor(clientdata$education)
clientdata$default <- as.factor(clientdata$default)
clientdata$housing <- as.factor(clientdata$housing)
clientdata$loan <- as.factor(clientdata$loan)
```

```r
clientdata$contact <- as.factor(clientdata$contact)
clientdata$poutcome <- as.factor(clientdata$poutcome)
clientdata$deposit <- factor(clientdata$deposit, levels = c("no", "yes"))

#splitting data
set.seed(123)
trainIndex <- createDataPartition(clientdata$deposit, p = 0.7, list = FALSE)
train <- clientdata[trainIndex, ]
test <- clientdata[-trainIndex, ]

#building the logistic regression model
model <- glm(deposit ~ age +
               job +
               marital +
               education +
               default +
               balance +
               housing +
               loan +
               contact +
               duration +
               campaign +
               pdays +
               previous +
               poutcome,
             data = train,
             family = "binomial")
summary(model)
```

Initial Logistic Regression

```
##
## Call:
## glm(formula = deposit ~ age + job + marital + education + default +
##     balance + housing + loan + contact + duration + campaign +
##     pdays + previous + poutcome, family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.7983  -0.4053  -0.2698  -0.1584   3.1482
##
## Coefficients:
##                        Estimate   Std. Error z value             Pr(>|z|)
## (Intercept)        -2.632681929  0.204516479 -12.873 < 0.0000000000000002 ***
## age                 0.001537941  0.002626513   0.586             0.558181
## jobblue-collar     -0.470150896  0.087429608  -5.377  0.00000007553595668 ***
## jobentrepreneur    -0.655042299  0.155587902  -4.210  0.00002552455405112 ***
## jobhousemaid       -0.624137324  0.161276698  -3.870             0.000109 ***
## jobmanagement      -0.293887915  0.087199295  -3.370             0.000751 ***
## jobretired          0.336408168  0.112865504   2.981             0.002877 **
## jobself-employed   -0.349545865  0.132220331  -2.644             0.008201 **
## jobservices        -0.283734038  0.099237253  -2.859             0.004248 **
## jobstudent          0.490838886  0.128625182   3.816             0.000136 ***
## jobtechnician      -0.341198264  0.081771035  -4.173  0.00003011361636133 ***
## jobunemployed      -0.307638340  0.135569269  -2.269             0.023254 *
## jobunknown         -0.341307858  0.279075117  -1.223             0.221331
## maritalmarried     -0.171226107  0.070446640  -2.431             0.015075 *
## maritalsingle       0.109865227  0.080023335   1.373             0.169779
## educationsecondary  0.133514484  0.077969399   1.712             0.086824 .
## educationtertiary   0.377963975  0.090298346   4.186  0.00002842582733767 ***
## educationunknown    0.264679163  0.124364392   2.128             0.033316 *
## defaultyes         -0.361839466  0.218021743  -1.660             0.096985 .
## balance             0.000017468  0.000005803   3.010             0.002612 **
## housingyes         -0.781305960  0.048947093 -15.962 < 0.0000000000000002 ***
## loanyes            -0.563932029  0.072189312  -7.812  0.00000000000000564 ***
## contacttelephone   -0.176012190  0.090204430  -1.951             0.051026 .
## contactunknown     -1.208158374  0.071635584 -16.865 < 0.0000000000000002 ***
## duration            0.004092086  0.000077434  52.846 < 0.0000000000000002 ***
## campaign           -0.108310815  0.012090625  -8.958 < 0.0000000000000002 ***
## pdays               0.000408543  0.000371450   1.100             0.271393
## previous            0.029350840  0.011031799   2.661             0.007801 **
## poutcomeother       0.288834226  0.107233046   2.694             0.007070 **
## poutcomesuccess     2.304779127  0.097323276  23.682 < 0.0000000000000002 ***
## poutcomeunknown    -0.072850512  0.114848311  -0.634             0.525872
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 21523  on 29798  degrees of freedom
## Residual deviance: 14911  on 29768  degrees of freedom
## AIC: 14973
##
## Number of Fisher Scoring iterations: 6
```

```
fitted.results <- predict(model, newdata=subset(test, type = 'response'))
predicted.results <- factor(ifelse(fitted.results > 0.5, "yes", "no"),
                            levels = levels(test$deposit))

predicted.results <- mean(predicted.results != test$deposit)

print(paste('accuracy', 1-predicted.results))
```

```
## [1] "accuracy 0.898660819171431"
```

When creating a logistic model including all independent variables, we can see that the model breaks the variables out in to their respective labels (ex: “jobblue-collar”). This makes it very difficult to determine which variables as a whole have a large effect on the outcome of deposit, the dependent variable. As we can see the p-value for all variables are exceptionally low indicating that they would be good predictors of the dependent variable. The “***” column shows that many of the variables are valuable predictors of deposit. This is interesting considering the both the numerical and categorical correlation matrices did not show relationships. Overall, the accuracy score of the above logistic regression is computed at 0.8987 which suggests that the model is performing decently in the prediction of the dependent variable.

As expected after exploratory visualizations in prior analysis, the previous outcome of prior campaigns being a success is a good indication to predicting the deposit variable. Similarly, certain job types are seen to be helpful indicators as well. Duration, campaign, and whether or not a client has a housing loan also appear to be significant predictors. With this knowledge, we move to create a second logistic regression that narrows down to these variables with the goal of experiencing a higher accuracy rate.

Final Logistic Model The below logistic regression model represents the independent variables job type and previous outcome and the ability of these variables to predict the dependent variable, deposit.

```r
final_model <- glm(deposit ~ job +
                     housing +
                     duration  +
                     campaign +
                     poutcome,
                   data = train,
                   family = binomial())
summary(final_model)
```

```
##
## Call:
## glm(formula = deposit ~ job + housing + duration + campaign +
##     poutcome, family = binomial(), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.8838  -0.4014  -0.2881  -0.2014   3.1312
##
## Coefficients:
##                      Estimate  Std. Error z value             Pr(>|z|)
## (Intercept)      -2.24412517  0.08969895 -25.018 < 0.0000000000000002 ***
## jobblue-collar   -0.63965330  0.08249799  -7.754  0.00000000000000894 ***
## jobentrepreneur  -0.64945661  0.15178300  -4.279  0.00001878619001596 ***
## jobhousemaid     -0.71958539  0.15519888  -4.637  0.00000354294262734 ***
## jobmanagement    -0.02962072  0.07426622  -0.399             0.690007
## jobretired        0.28769903  0.09628550   2.988             0.002808 **
## jobself-employed -0.23787633  0.12737160  -1.868             0.061821 .
## jobservices      -0.35360209  0.09765157  -3.621             0.000293 ***
## jobstudent        0.78210838  0.12098139   6.465  0.00000000010150003 ***
## jobtechnician    -0.24359767  0.07983657  -3.051             0.002279 **
## jobunemployed    -0.21413721  0.13284927  -1.612             0.106988
## jobunknown       -0.38168406  0.27106917  -1.408             0.159111
## housingyes       -0.95608403  0.04654176 -20.542 < 0.0000000000000002 ***
## duration          0.00401197  0.00007521  53.346 < 0.0000000000000002 ***
## campaign         -0.10321848  0.01186136  -8.702 < 0.0000000000000002 ***
## poutcomeother     0.29390495  0.10640327   2.762             0.005742 **
## poutcomesuccess   2.27766178  0.09364076  24.323 < 0.0000000000000002 ***
## poutcomeunknown  -0.55860319  0.06559239  -8.516 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 21523  on 29798  degrees of freedom
## Residual deviance: 15417  on 29781  degrees of freedom
## AIC: 15453
##
```

```
## Number of Fisher Scoring iterations: 6
```

```
fitted.results1 <- predict(final_model, newdata=subset(test, type = 'response'))
predicted.results1 <- factor(ifelse(fitted.results1 > 0.5, "yes", "no"),
                             levels = levels(test$deposit))
confusion_matrix <- confusionMatrix(predicted.results1, test$deposit)
accuracy <- mean(predicted.results1 == test$deposit)


predicted.results1 <- mean(predicted.results1 != test$deposit)

predicted.results1 <- as.numeric(predicted.results1)
results_df <- data.frame(Predicted = predicted.results1,
                         Actual = as.numeric(test$deposit))

# print confusion matrix
print(confusion_matrix$table)
```

```
##            Reference
## Prediction    no   yes
##        no  11111  1128
##        yes   163   367
```

```
# print accuracy
accuracy <- confusion_matrix$overall['Accuracy']
print(paste('accuracy', accuracy))
```

```
## [1] "accuracy 0.898895763176443"
```

```
# print other evaluation metrics
print(confusion_matrix$byClass)
```

```
##          Sensitivity          Specificity       Pos Pred Value
##            0.9855420            0.2454849            0.9078356
##       Neg Pred Value            Precision               Recall
##            0.6924528            0.9078356            0.9855420
##                   F1           Prevalence       Detection Rate
##            0.9450942            0.8829196            0.8701543
## Detection Prevalence    Balanced Accuracy
##            0.9584932            0.6155135
```

Once narrowed down to significant independent variables, the accuracy of the prediction increases to 0.8989. This is a strong accuracy value that represents a well-performing model. In looking at each variable's individual performance in the model, the only characteristics that are not considered statistically significant are specific job types i.e. management, self-employed, unemployed, and unknown.

Again, it is interesting to see that the above independent variables will have a statistical significance in predicting the client's deposit behavior after not visually seeing a relationship with the deposit variable. However, the Positive Predictive Value seen in the logistic model, 0.9078, can provide the Portuguese bank with confidence when a client's deposit behavior is a predicted yes. Similarly, a high precision value offers confidence that the proportion of predicted positives, or predicted "yes" cases, are all actual positives, or yes recordings.

Upon request of further analysis, it may be beneficial for the Portuguese bank to begin analyzing more historical, financial information of their current clients. As expected, independent variables that are the recordings of a previous campaign (poutcome, campaign, and duration) have the lowest p-values among the variables. This further supports that current clients of the Portuguese bank are likely to continue to maintain deposits with the company.