# Data Due Diligence Project

Daniel Jackson

April 14th, 2024

## Table of Contents

Introduction.....1
Rmarkdown Code.....1
Summary of Data.....25
Tableau Visualizations.....27
Conclusion.....32

## Introduction

In this project, we will be looking at a data set that contains a sample of 5,000 customers of a telecommunications company. We will be acting as an incoming marketing analytics manager for the company. Our goal is to conduct a comprehensive assessment of the customer base. Within the Rmarkdown portion of this project, we will be using RStudio to read in, clean, and tidy the data. Throughout the cleansing of the data, we will create a subset of ten variables that we are interested in for further analysis. These variables will be composed of both qualitative and quantitative variables. The variables may also be transformed, or they may be a combination of multiple variables. We will go into detail of each of the ten variables that we choose and provide a detailed and clear justification of the feature engineering that we conduct. We will also identify the meaning and the type of each variable chosen.

After we create our subset of ten variables, we will then be using Tableau to provide several single variable plots of our data. We will also perform five two-column visuals to see if there are any relationships between the variables that we choose to compare.

The goal of this project is to provide a cleaned-up subset of our original data and then use data visualization tactics to see if there are any relationships between the variables that we choose. The layout of this project will be:

- Rmarkdown code
- Summary
- Visualizations

## Rmarkdown Code

### Libraries Used

```
library(dplyr)
library(tidyr)
library(caret)
library(DescTools)
```

```
## Warning: package 'DescTools' was built under R version 4.3.2
```

```
library(stringr)
library(openxlsx)
```

## Read in Data

```
# Read in data
cust_df = read.csv("customer_data.csv")

# Check dimension of data frame
dim(cust_df)
```

```
## [1] 5000   60
```

```
# 5000 observations
# 60 columns
```

## Clean Data

```
# Make all variable names lower case
colnames(cust_df) = tolower(colnames(cust_df))

# Check how many columns have NAs
colSums(is.na(cust_df))
```

```
##          customerid              region            townsize
gender
##                   0                   0                   0
0
##                 age      educationyears         jobcategory
unionmember
##                   0                   0                   0
0
##    employmentlength             retired            hhincome
debttoincomeratio
##                   0                   0                   0
0
##          creditdebt           otherdebt                   x
loandefault
##                   0                   0                5000
0
##       maritalstatus       householdsize          numberpets
numbercats
##                   0                   8                   6
7
##          numberdogs         numberbirds           homeowner
carsowned
##                   8                  34                  13
0
##        carownership            carbrand            carvalue
commutetime
```

```
##                   0                   0                   0
0
##   politicalpartymem               votes          creditcard
cardtenure
##                   0                   0                   0
0
##    carditemsmonthly      cardspendmonth     activelifestyle
phonecotenure
##                   0                   0                   0
0
##      voicelastmonth     voiceovertenure     equipmentrental
equipmentlastmonth
##                   0                   0                   0
0
## equipmentovertenure         callingcard        wirelessdata
datalastmonth
##                   0                   0                   0
0
##      dataovertenure           multiline                  vm
pager
##                   0                   0                   0
0
##             internet            callerid            callwait
callforward
##                   0                   0                   0
0
##     threewaycalling            ebilling     tvwatchinghours
ownspc
##                   0                   0                   0
0
##    ownsmobiledevice      ownsgamesystem             ownsfax
newssubscriber
##                   0                   0                   0
0

# Column x has 5000 missing NA values. Let's remove that column.
cust_df = cust_df[, -which(names(cust_df) == "x")]
```

## Variables Chosen

### Number of Pets

```
# We see that numberpets has 6 NAs, numbercats has 7, numberdogs has 8, and
# numberbirds has 34.
# Let us assume that an NA value in these columns equals zero.
cust_df = cust_df %>%
  mutate(numberpets = ifelse(is.na(numberpets), 0, numberpets),
         numbercats = ifelse(is.na(numbercats), 0, numbercats),
         numberdogs = ifelse(is.na(numberdogs), 0, numberdogs),
         numberbirds = ifelse(is.na(numberbirds), 0, numberbirds))
```

```
# We want to focus on just total number of pets in this analysis. So let us
make numberpets equal to the sum of numbercats + numberdogs + numberbirds
cust_df$numberpets = (cust_df$numbercats + cust_df$numberdogs +
cust_df$numberbirds)
# Now we can remove the numbercats, numberdogs, and numberbirds columns
cust_df = cust_df[, -which(names(cust_df) == "numbercats")]
cust_df = cust_df[, -which(names(cust_df) == "numberdogs")]
cust_df = cust_df[, -which(names(cust_df) == "numberbirds")]
```

## Household

```
# We also see that household size has 8 NA values.
# Let us find average values of household size based on if customers are
married or not.
cust_df %>%
  group_by(maritalstatus) %>%
  summarize(mean_household_size = mean(na.omit(householdsize)))
```

```
## # A tibble: 2 × 2
##   maritalstatus mean_household_size
##   <chr>                       <dbl>
## 1 Married                      3.11
## 2 Unmarried                    1.36
```

```
# We see that the average household size of customers that are married is
3.11 and the average household size of customers that are unmarried is 1.36.
# With this information, let us fill the NA values of household size using
these average values.
married_size = 3
unmarried_size = 1
cust_df = cust_df %>%
  mutate(householdsize = if_else(is.na(householdsize),
              if_else(maritalstatus == "Married", married_size,
unmarried_size),
              householdsize))
```

## Homeowner

```
# We see that homeowner has 13 NA values. The homeowner is a binary column
that says either 1 for homeowner or 0 for non-homeowner. Since this is a
binary column, let us replace the NA values with the mode for the column.
cust_df = cust_df %>%
  mutate(homeowner = ifelse(is.na(homeowner), Mode(homeowner, na.rm = TRUE),
                            homeowner))
Mode(cust_df$homeowner)
```

```
## [1] 1
## attr(,"freq")
## [1] 3153
```

## Job Category

```
# Looking at the jobcategory variable, we see a few empty strings.
# Check unique values
unique(cust_df$jobcategory)
```

```
## [1] "Professional" "Sales"        "Service"      "Labor"
"Agriculture"
## [6] ""             "Crafts"
```

```
# We see that there is an empty string in there. Let us make that empty
string a NA value
cust_df = cust_df %>%
  mutate(jobcategory = ifelse(jobcategory == "", NA, jobcategory))

# Now we see that there is 15 NA values for jobcategory.
# Let us look at highest count of jobcategory values for those that are in
the union and those that are not in the union to help impute values for NA
values in jobcategory
cust_df %>%
  group_by(unionmember, jobcategory) %>%
  summarize(count = n(), .groups = "drop")
```

```
## # A tibble: 14 × 3
##    unionmember jobcategory  count
##    <chr>       <chr>        <int>
##  1 No          Agriculture    198
##  2 No          Crafts         323
##  3 No          Labor          540
##  4 No          Professional  1164
##  5 No          Sales         1438
##  6 No          Service        569
##  7 No          <NA>            12
##  8 Yes         Agriculture     14
##  9 Yes         Crafts         129
## 10 Yes         Labor          146
## 11 Yes         Professional   216
## 12 Yes         Sales          197
## 13 Yes         Service         51
## 14 Yes         <NA>             3
```

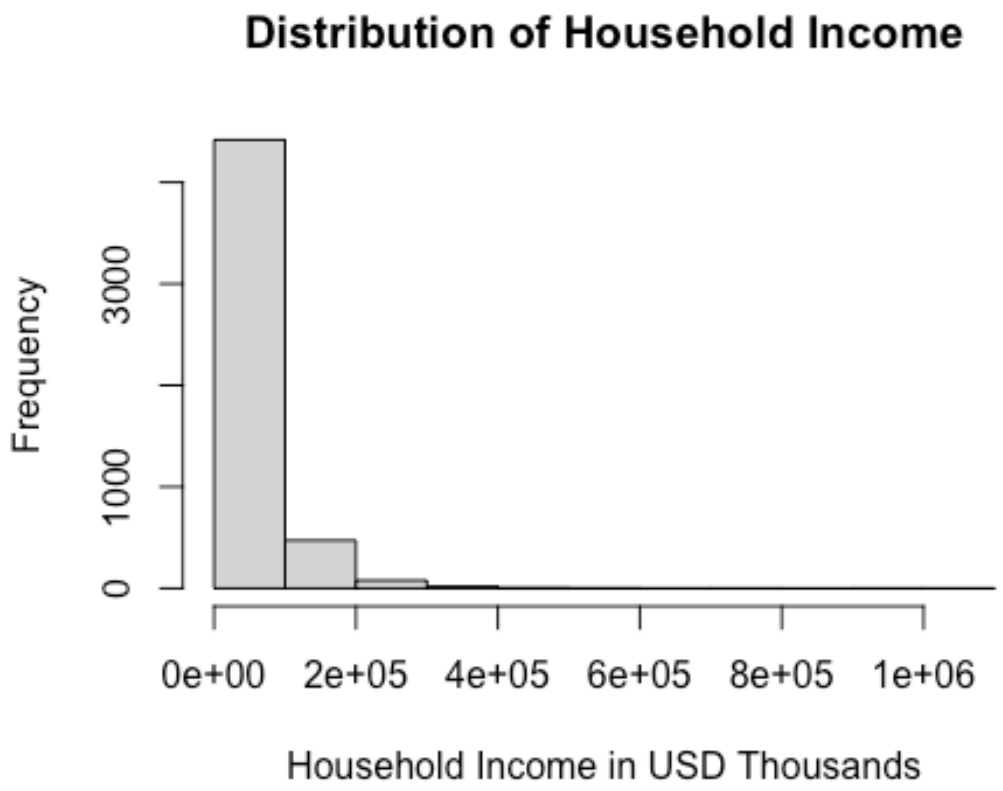
```
# For non-union members, sales has highest count in the jobcategory column.
# For union members, professional has highest count.
# Let us impute those for the missing NA values in jobcategory column.
cust_df = cust_df %>%
  mutate(jobcategory = if_else(is.na(jobcategory) & unionmember == "Yes",
                               "Professional",
         if_else(is.na(jobcategory) & unionmember ==
"No","Sales",jobcategory)))
```

## Region

```
# Let us change the region column from 1, 2, 3, 4, 5 to the respective region
names
unique(cust_df$region)

## [1] 1 5 3 4 2

cust_df = cust_df %>%
  mutate(region = ifelse(region == 1, "northeast",
                  ifelse(region == 2, "midwest",
                  ifelse(region == 3, "west",
                  ifelse(region == 4, "southwest",
                  ifelse(region == 5, "southeast", region))))))
# Check to see names were changed
unique(cust_df$region)

## [1] "northeast" "southeast" "west"      "southwest" "midwest"
```

## Household Income

```
# Let us look at the hhincome column. Let us first look at the distribution
of the variable and see if a transformation helps the distribution.
class(cust_df$hhincome)

## [1] "character"

# This is a character vector. We need to remove all non-numeric characters in
cust_df = cust_df %>%
  mutate(hhincome = as.numeric(str_replace_all(hhincome, "[^0-9.]", "")))

# Check hhincome class again
class(cust_df$hhincome)

## [1] "numeric"

# It is now numeric. Let us look at distribution:
hist(cust_df$hhincome,
     xlab = "Household Income in USD Thousands",
     ylab = "Frequency",
     main = "Distribution of Household Income")
```



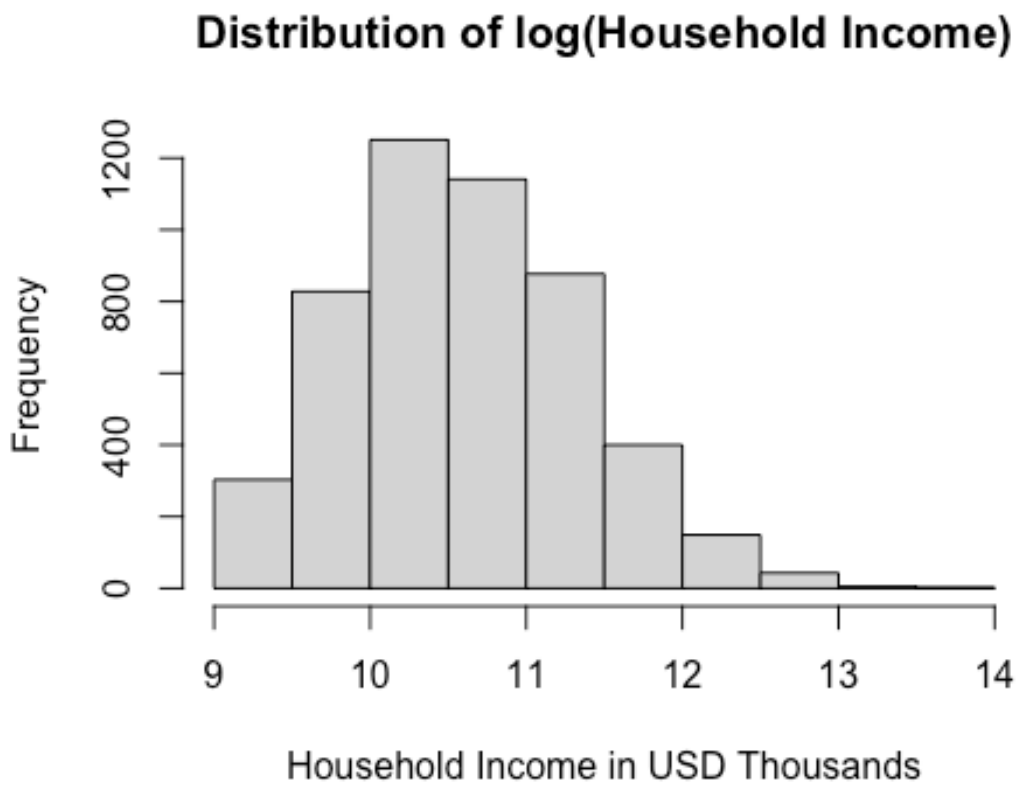

```
# Try log transformation to see how that affects distribution.
# Before we do that, let us see what the min/max value is:
min(cust_df$hhincome)
```

```
## [1] 9000
```

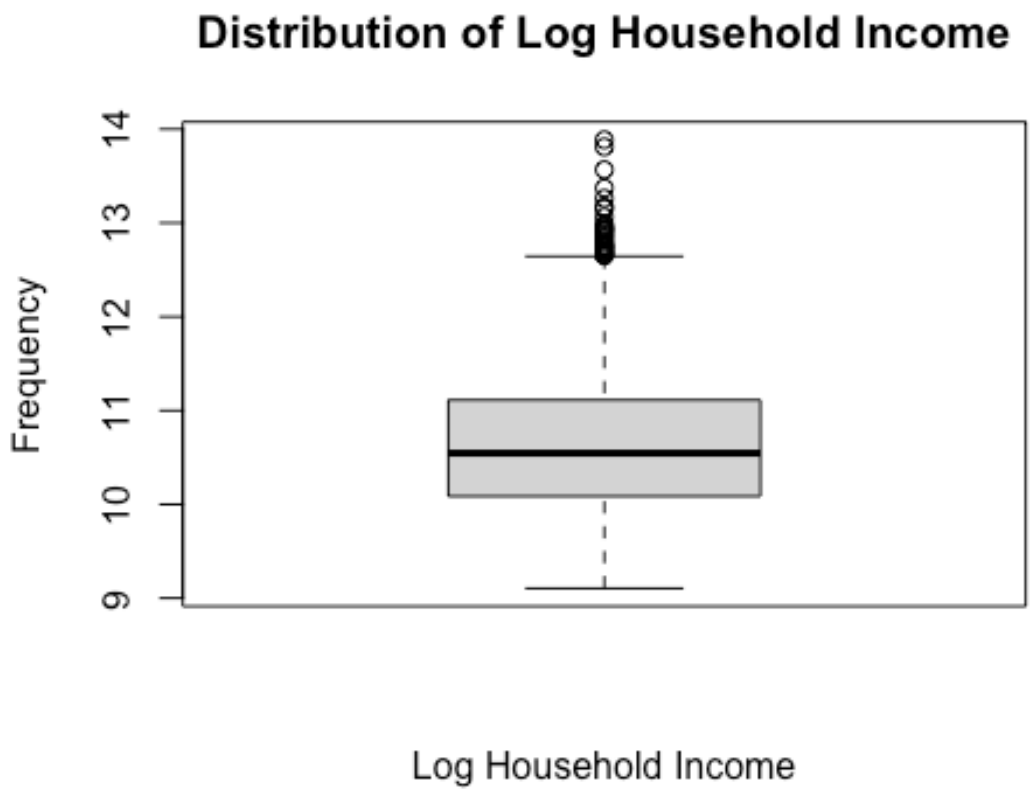
```
# Min is 9000
max(cust_df$hhincome)
```

```
## [1] 1073000
```

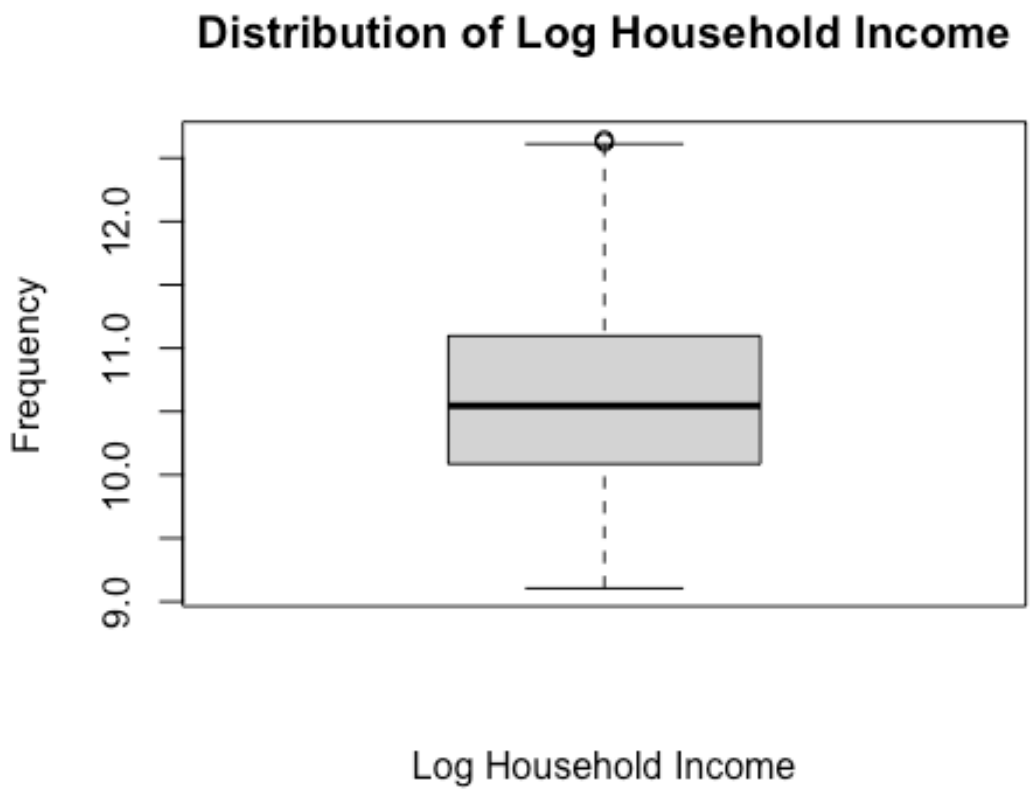
```
# Max is 1073000
hist(log(cust_df$hhincome),
     xlab = "Household Income in USD Thousands",
     ylab = "Frequency",
     main = "Distribution of log(Household Income)")
```




```r
# This made distribution more normal. Let us add a new column and call it
log_hh_inc
cust_df = cust_df %>%
  mutate(log_hh_income = log(hhincome))

# Let us look at boxplot to see if we have any outliers
boxplot(cust_df$log_hh_income,
        xlab = "Log Household Income",
        ylab = "Frequency",
        main = "Distribution of Log Household Income")
```




```r
# We see some outliers. Let us remove them
q = quantile(cust_df$log_hh_income, c(0.25, 0.75))
iqr = q[2] - q[1]

# Set the range for outliers
low_q = q[1] - 1.5 * iqr
upper_q = q[2] + 1.5 * iqr

# Remove outliers from the dataframe
cust_df = cust_df[cust_df$log_hh_income >= low_q
                  & cust_df$log_hh_income <= upper_q, ]

# Let us look at box plot again
boxplot(cust_df$log_hh_income,
        xlab = "Log Household Income",
        ylab = "Frequency",
        main = "Distribution of Log Household Income")
```



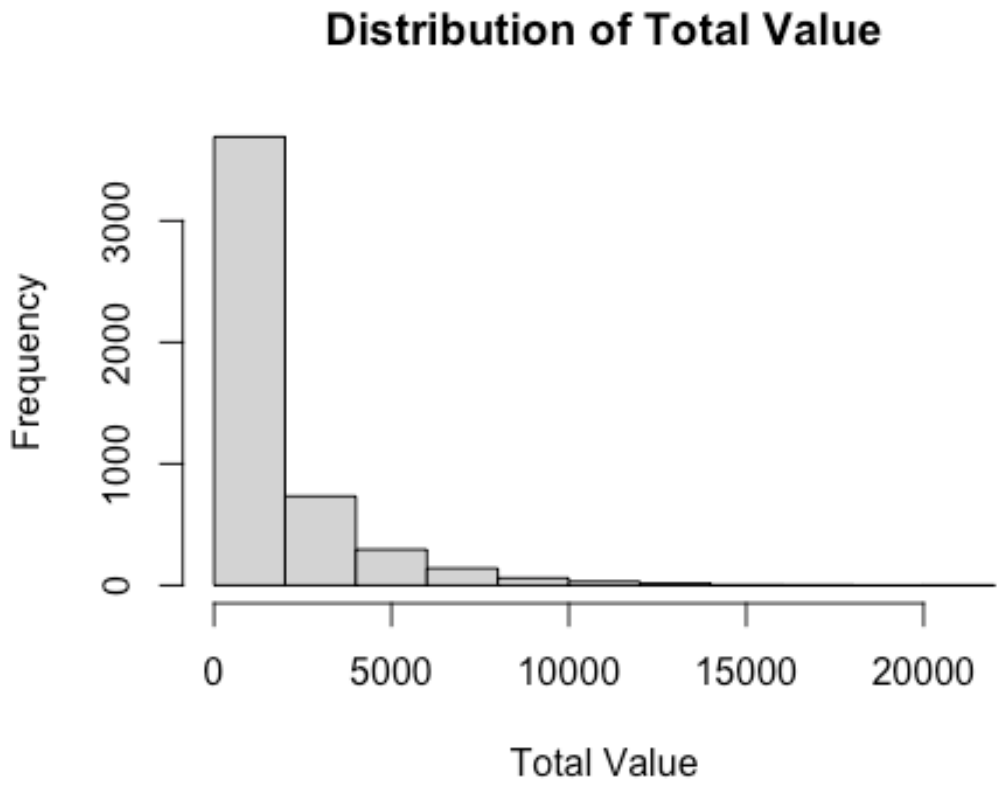

```
# We removed those outliers
```

## Total Value Over Tenure

```
# Let us look at the total value of each customer over their tenure by adding
equipmentovertenure, voiceovertenure, and dataovertenure variable together.
# Let us first look at the class of each variable
class(cust_df$equipmentovertenure)

## [1] "character"

# Character
class(cust_df$voiceovertenure)

## [1] "character"

# Character
class(cust_df$dataovertenure)

## [1] "character"

# Character

# Now we need to remove all non-numeric characters
cust_df = cust_df %>%
```

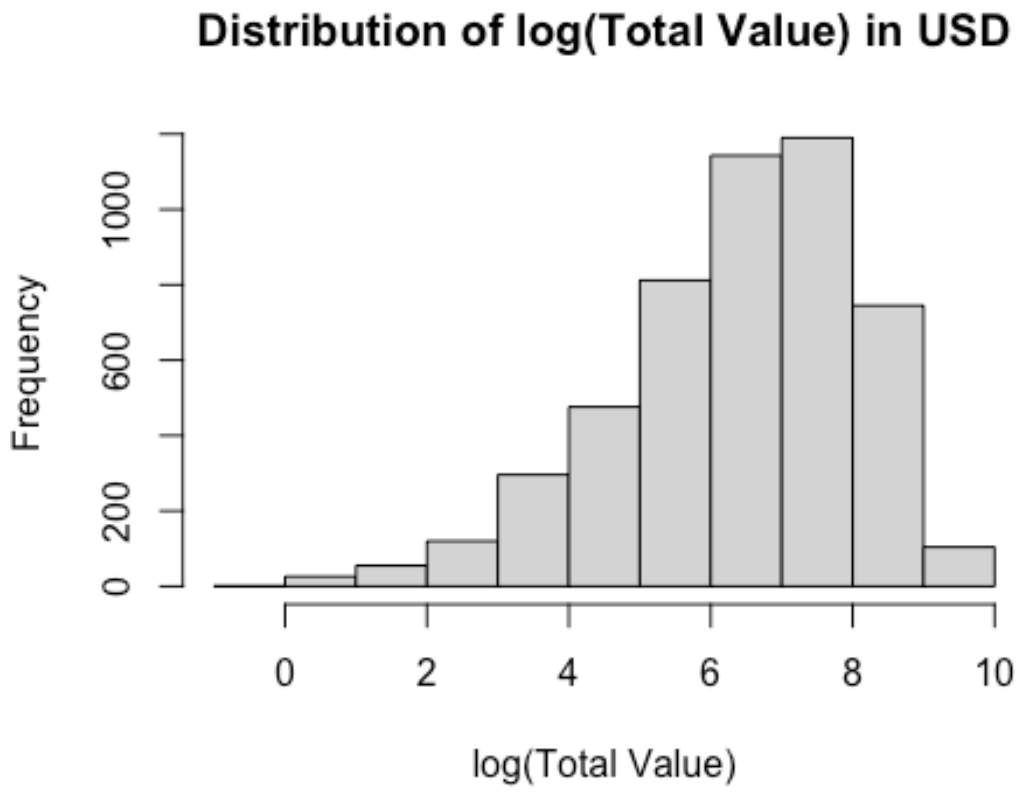
```
  mutate(equipmentovertenure =
as.numeric(str_replace_all(equipmentovertenure, "[^0-9.]", "")),
  voiceovertenure = as.numeric(str_replace_all(voiceovertenure, "[^0-9.]",
"")),
  dataovertenure = as.numeric(str_replace_all(dataovertenure, "[^0-9.]",
"")))

# Check for NA values in each column
colSums(is.na(cust_df[, c("equipmentovertenure", "voiceovertenure",
"dataovertenure")]))

## equipmentovertenure     voiceovertenure      dataovertenure
##                3279                   3                3642

# If there is a NA value in those columns, it means that the customer does
not use those services, therefore we can change all of those NA values to 0
in each column
cust_df = cust_df %>%
  mutate(equipmentovertenure = ifelse(is.na(equipmentovertenure), 0,
equipmentovertenure),
         voiceovertenure = ifelse(is.na(voiceovertenure), 0,
voiceovertenure),
         dataovertenure = ifelse(is.na(dataovertenure), 0, dataovertenure))

# Check for NA values
colSums(is.na(cust_df[, c("equipmentovertenure", "voiceovertenure",
"dataovertenure")]))

## equipmentovertenure     voiceovertenure      dataovertenure
##                   0                   0                   0

# No more NA values
# Confirm each class is numeric
class(cust_df$equipmentovertenure)

## [1] "numeric"

class(cust_df$voiceovertenure)

## [1] "numeric"

class(cust_df$dataovertenure)

## [1] "numeric"

# All variables are numeric

# Now we can create new variable called total_value which will represent
total spent over tenure for each customer
cust_df = cust_df %>%
  mutate(total_value = equipmentovertenure + voiceovertenure +
dataovertenure)
```

```r
# Let us look at distribution of total_value
hist(cust_df$total_value,
     xlab = "Total Value",
     ylab = "Frequency",
     main = "Distribution of Total Value")
```

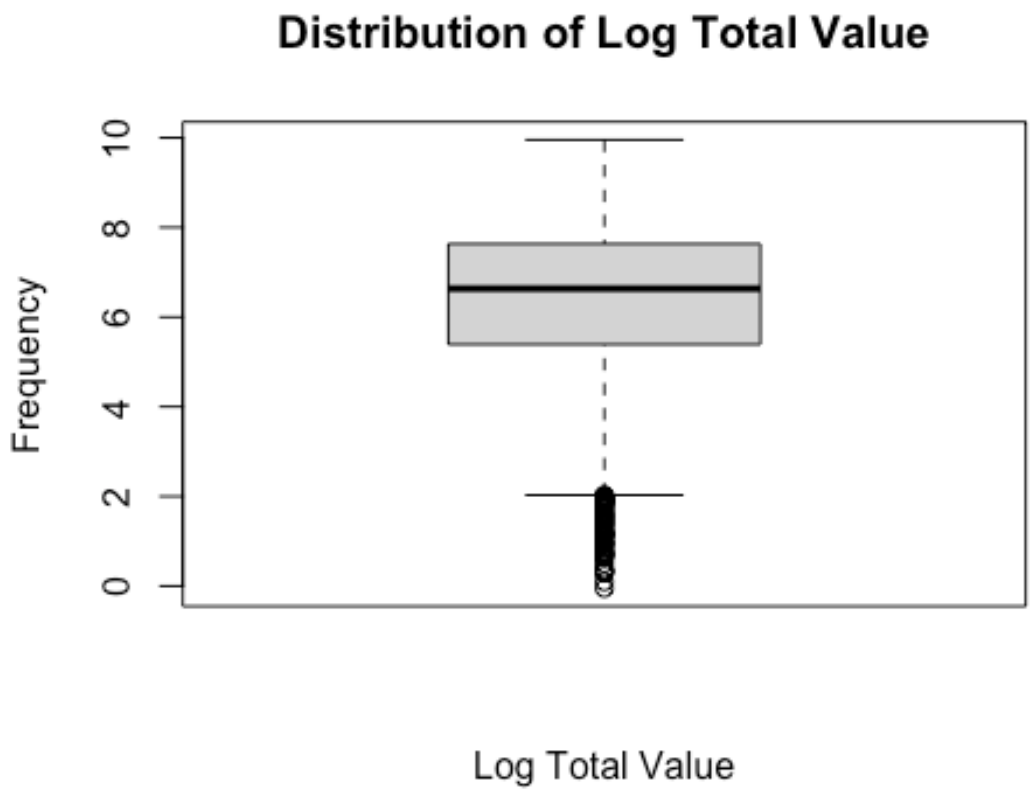



```r
# We see that there are three customers that have 0 value over their time as
a customer.
head(table(cust_df$total_value))

##
##    0 0.95  1.1 1.35  1.4 1.45
##    3    1    1    2    3    1

# We see that there are three customers that have 0 total value. Let us drop
those observations.
cust_df = cust_df[cust_df$total_value != 0, ]

# Now let us see if a log transformation helps the right skewedness of the
total value distribution
hist(log(cust_df$total_value),
     xlab = "log(Total Value)",
```

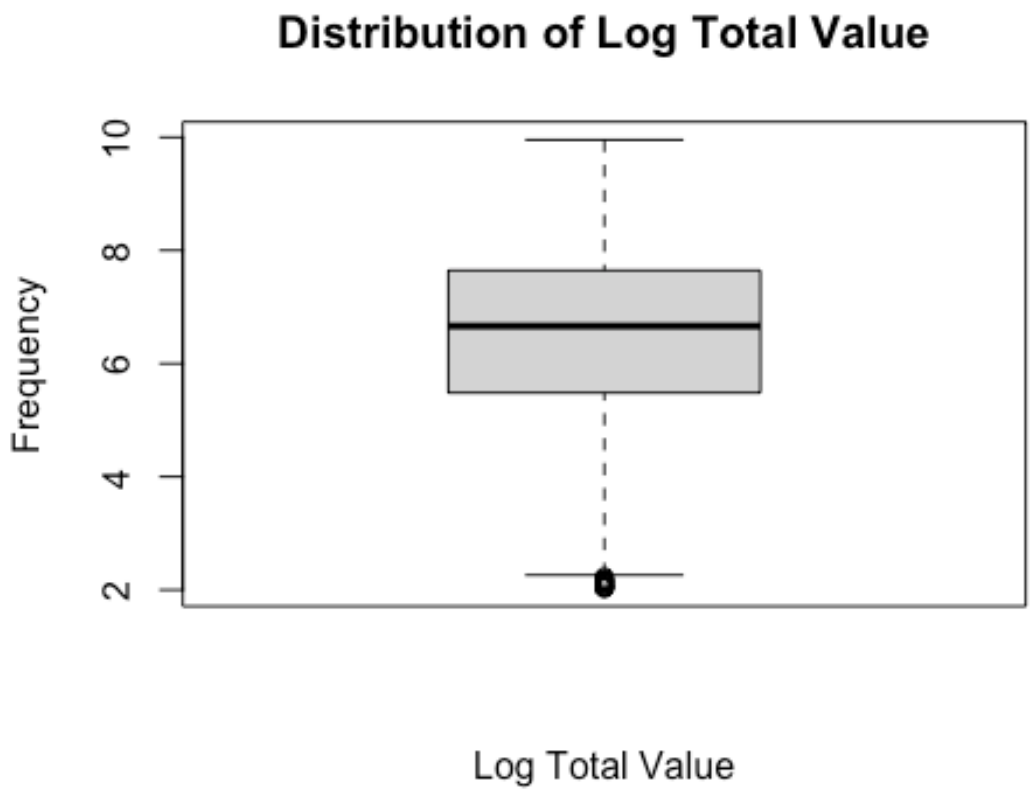
```
     ylab = "Frequency",
     main = "Distribution of log(Total Value) in USD")
```




```
# The log transformation makes the distribution slightly left skewed.
However, the log transformed distribution is much less skewed than the non-
transformed distribution.

# Let us create a new variable called log_tot_value
cust_df = cust_df %>%
  mutate(log_tot_value = log(total_value))

# Let us look at log transformed box plot
boxplot(cust_df$log_tot_value,
     xlab = "Log Total Value",
     ylab = "Frequency",
     main = "Distribution of Log Total Value")
```




```
# We see a lot of outliers.

# Let us remove those outliers by calculating the interquartile range of our
distribution and remove the points out side of that range
# Calculate the interquartile range
q = quantile(cust_df$log_tot_value, c(0.25, 0.75))
iqr = q[2] - q[1]

# Set the range for outliers
low_q = q[1] - 1.5 * iqr
upper_q = q[2] + 1.5 * iqr

# Remove outliers from the dataframe
cust_df = cust_df[cust_df$log_tot_value >= low_q
                  & cust_df$log_tot_value <= upper_q, ]

# Look at box plot again
boxplot(cust_df$log_tot_value,
     xlab = "Log Total Value",
     ylab = "Frequency",
     main = "Distribution of Log Total Value")
```




```
# This removed the outliers
```

## Total Debt

```
# Let us add creditdebt and otherdebt together and create total_debt column.
class(cust_df$creditdebt)
```

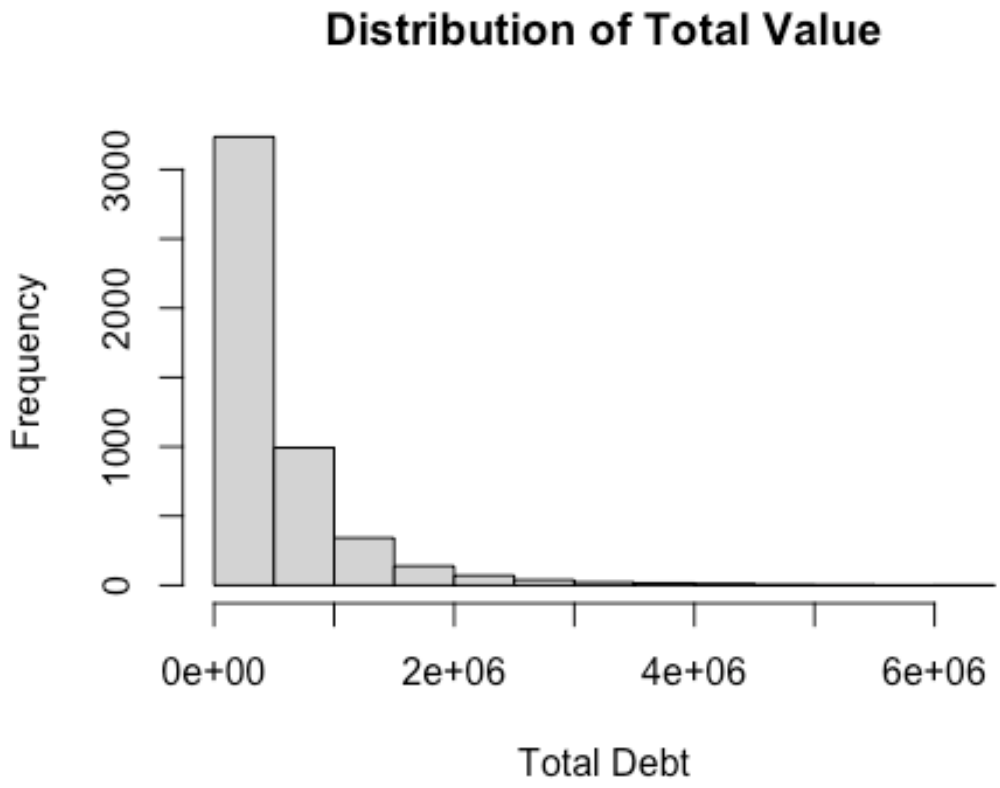
```
## [1] "numeric"
```

```
# numeric
class(cust_df$otherdebt)
```

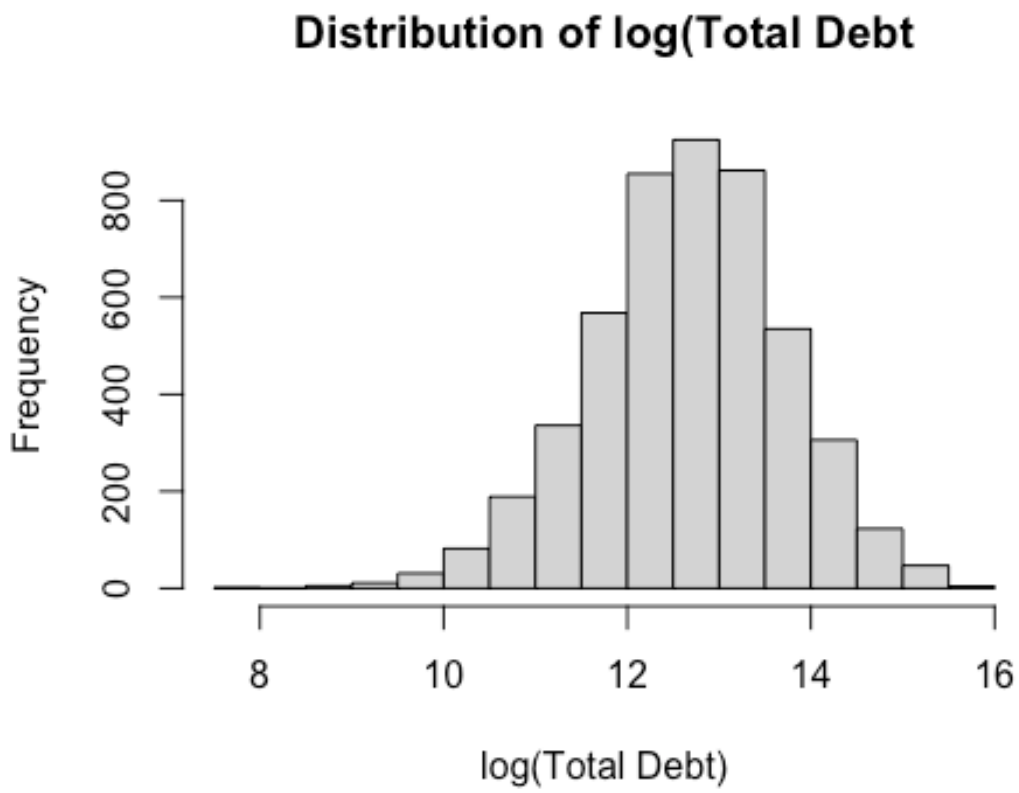
```
## [1] "numeric"
```

```
# numeric
# From our data dictionary, we know that each debt is expressed in $100,000.
Let us convert those debt numbers to represent debt in $100,000 and then add
them together.
cust_df = cust_df %>%
  mutate(creditdebt = (creditdebt * 100000),
         otherdebt = (otherdebt * 100000))
# Now let us create our total_debt column
cust_df = cust_df %>%
  mutate(total_debt = (creditdebt + otherdebt))
```

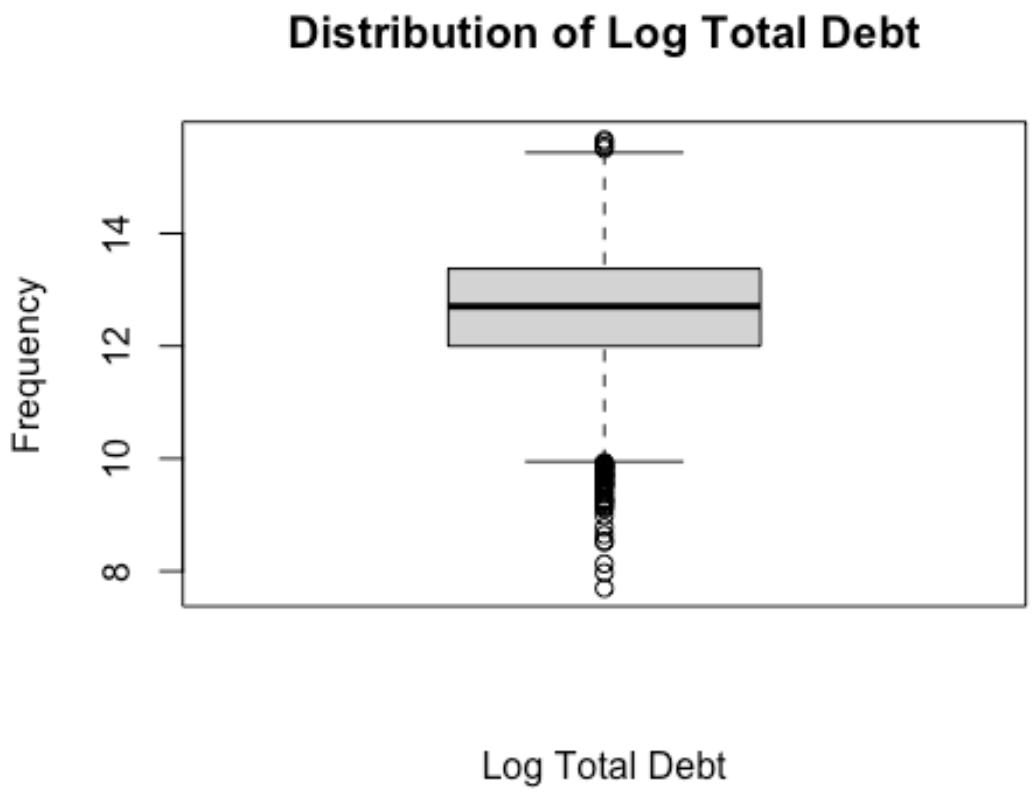
```
# Let us look at distribution of total_debt
hist(cust_df$total_debt,
     xlab = "Total Debt",
     ylab = "Frequency",
     main = "Distribution of Total Value")
```



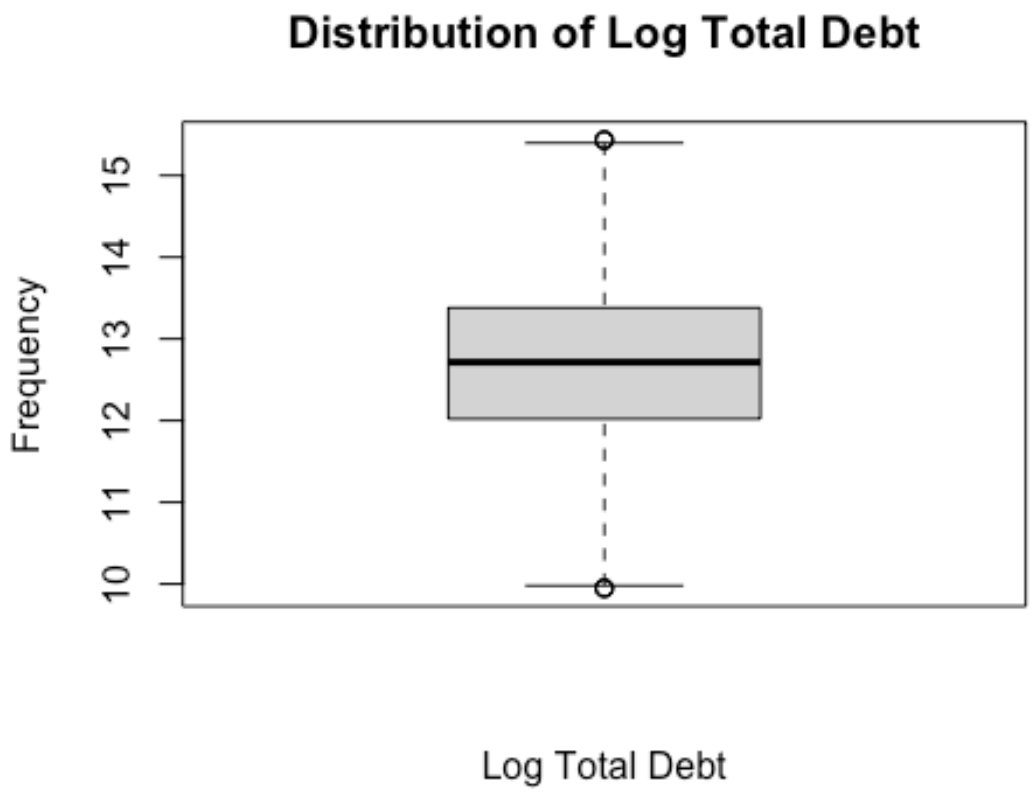

```
# Distribution is skewed heavily to the right. Let us try log transformation
head(table(cust_df$total_debt))

##
##    0 2200 2900 3400 5000 5100
##    1    1    1    1    1    1

# We see that one customer has 0 debt. We cannot log transform that value so
let us remove it
cust_df = cust_df[cust_df$total_debt != 0, ]

# Now let us look at log transformation
hist(log(cust_df$total_debt),
     xlab = "log(Total Debt)",
     ylab = "Frequency",
     main = "Distribution of log(Total Debt")
```




```
# Much more normal distribution. Let us log transform the vector in our data
set
cust_df = cust_df %>%
  mutate(log_tot_debt = log(total_debt))

# Now let us look at outliers in log_tot_debt
boxplot(cust_df$log_tot_debt,
        xlab = "Log Total Debt",
        ylab = "Frequency",
        main = "Distribution of Log Total Debt")
```




```r
# There are a lot of outliers. Let us remove the outliers.
q = quantile(cust_df$log_tot_debt, c(0.25, 0.75))
iqr = q[2] - q[1]

# Set the range for outliers
low_q = q[1] - 1.5 * iqr
upper_q = q[2] + 1.5 * iqr

# Remove outliers from the dataframe
cust_df = cust_df[cust_df$log_tot_debt >= low_q
                  & cust_df$log_tot_debt <= upper_q, ]
# Look at boxplot again
boxplot(cust_df$log_tot_debt,
        xlab = "Log Total Debt",
        ylab = "Frequency",
        main = "Distribution of Log Total Debt")
```




```
# This removed our outliers.
```

## Car Brand

```
# Let us look at carbrand values
class(cust_df$carbrand)

## [1] "character"

# character
unique(cust_df$carbrand)

## [1] "Domestic" "Foreign"  "-1"

# We see that there is a -1 value that shows up. Let us remove the -1 and
have it say "None"
cust_df = cust_df %>%
  mutate(carbrand = ifelse(carbrand == "-1", "None", carbrand))
unique(cust_df$carbrand)

## [1] "Domestic" "Foreign"  "None"

# Check to see total counts of each observation
table(cust_df$carbrand)
```

```
##
## Domestic  Foreign     None
##     2217     2132      483
```

## Car Value

```
# Let us look at carvalue variable. Let us look at class of variable.
class(cust_df$carvalue)
```

```
## [1] "character"
```

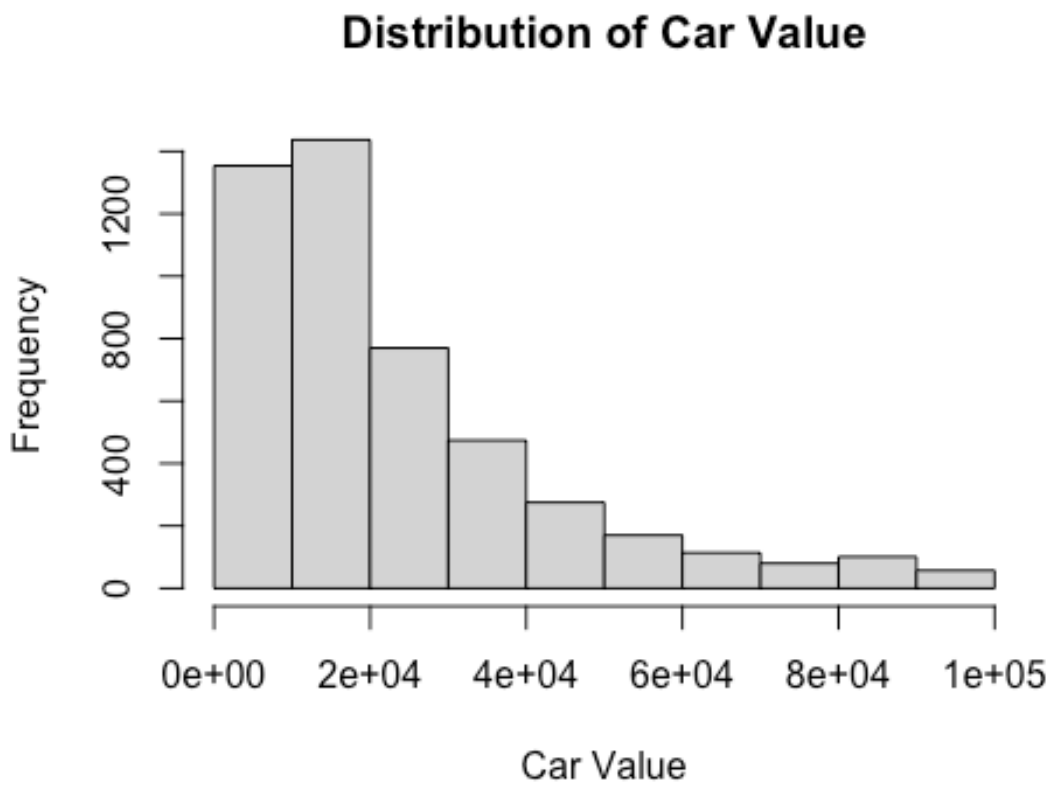
```
# Character
# We see that for those that do not own a vehicle, the value for carvalue is
$(1,000.00). Let us convert those to 0 strings.
cust_df = cust_df %>%
  mutate(carvalue = ifelse(carbrand == "None", "0", carvalue))
# Now let us remove non-numeric characters and convert carvalue to a numeric
vector
cust_df = cust_df %>%
  mutate(carvalue = as.numeric(str_replace_all(carvalue, "[^0-9.]", "")))
class(cust_df$carvalue)
```

```
## [1] "numeric"
```

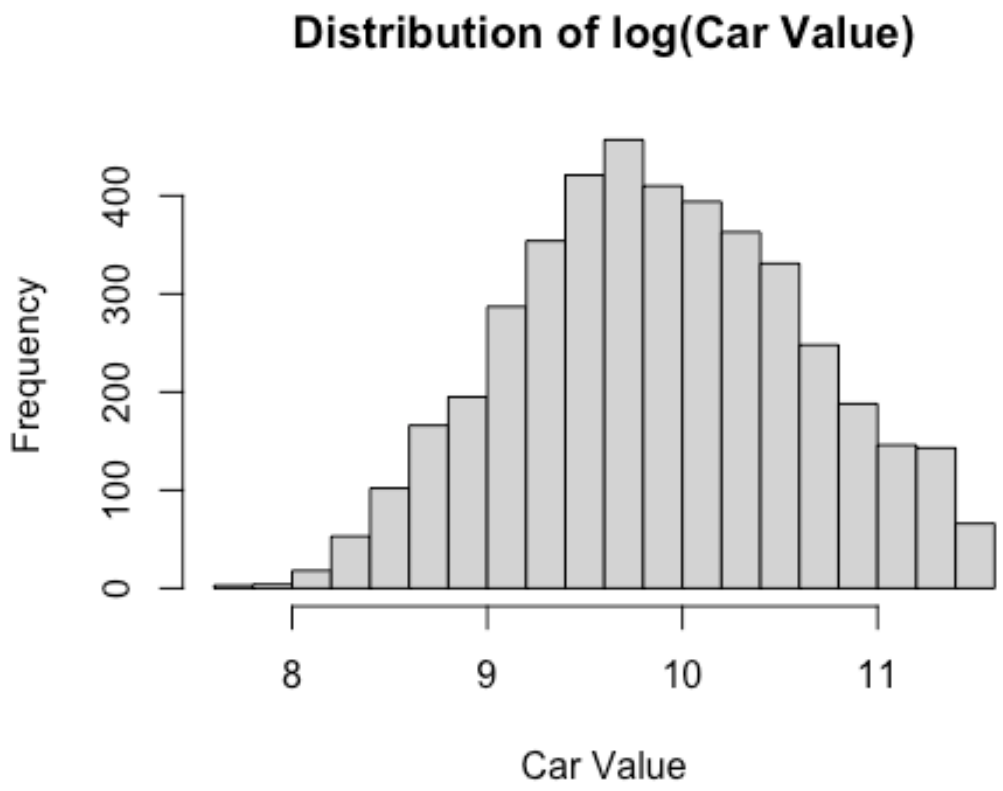
```
# Numeric
min(cust_df$carvalue)
```

```
## [1] 0
```

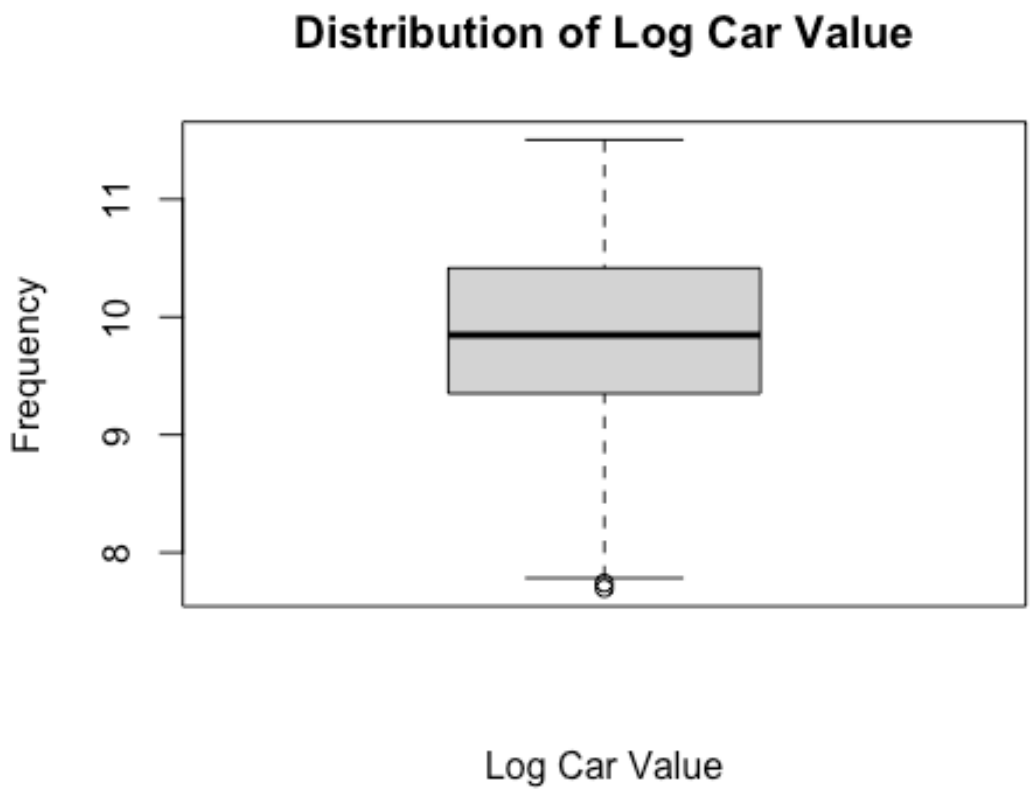
```
# 0

# Let us look at distribution of car value
hist(cust_df$carvalue,
     xlab = "Car Value",
     ylab = "Frequency",
     main = "Distribution of Car Value")
```



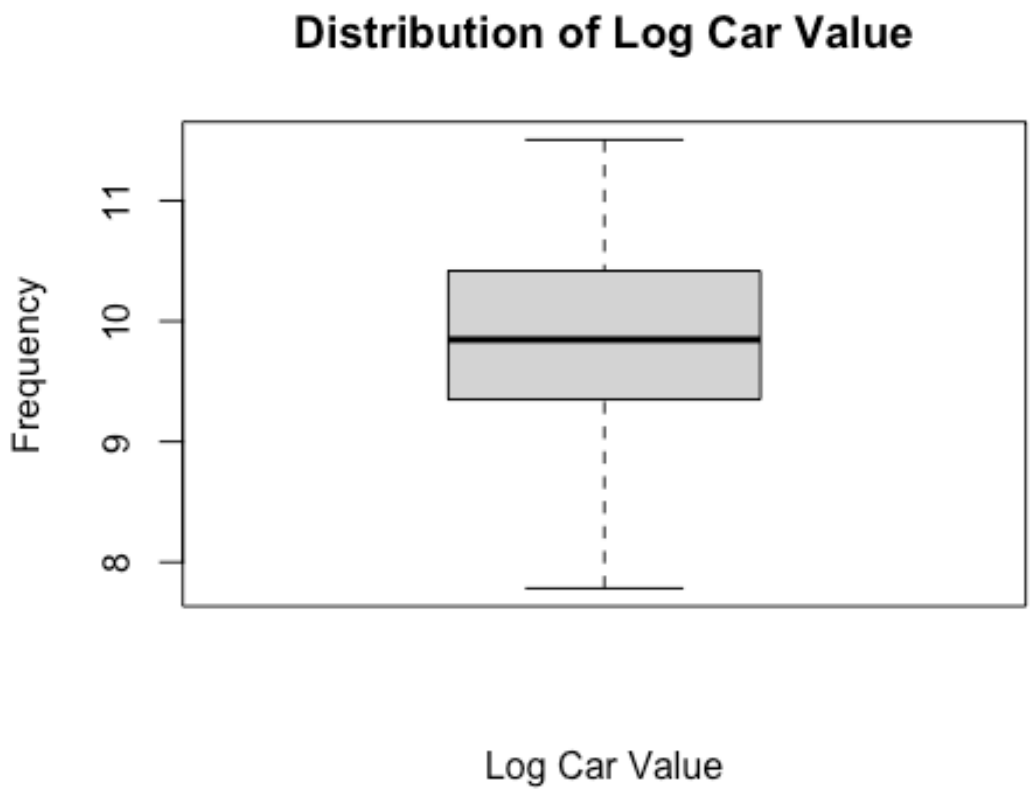

```
# Distribution is right skewed.
# Let us remove observations where carvalue equals 0
# Remove observations where carvalue equals zero
cust_df = cust_df[cust_df$carvalue != 0, ]

# Now let us try log transformation
hist(log(cust_df$carvalue),
     xlab = "Car Value",
     ylab = "Frequency",
     main = "Distribution of log(Car Value)")
```




```
# This makes the distribution much more normal. Let us create a new variable
that takes log of carvalue
cust_df = cust_df %>%
  mutate(log_car_val = log(carvalue))

# Let us look at boxplot of log_car_val
boxplot(cust_df$log_car_val,
        xlab = "Log Car Value",
        ylab = "Frequency",
        main = "Distribution of Log Car Value")
```




```
# We only see a few outliers. Let us remove them
q = quantile(cust_df$log_car_val, c(0.25, 0.75))
iqr = q[2] - q[1]

# Set the range for outliers
low_q = q[1] - 1.5 * iqr
upper_q = q[2] + 1.5 * iqr

# Remove outliers from the dataframe
cust_df = cust_df[cust_df$log_car_val >= low_q
                  & cust_df$log_car_val <= upper_q, ]

# Check for outliers again
boxplot(cust_df$log_car_val,
        xlab = "Log Car Value",
        ylab = "Frequency",
        main = "Distribution of Log Car Value")
```




```
# Outliers are removed
```

## Create Subset of Data

```
sub_columns = c("region", "numberpets", "householdsize", "homeowner",
"jobcategory", "carbrand", "log_car_val", "log_hh_income", "log_tot_debt",
"log_tot_value")
sub_cust_df = cust_df[,sub_columns]

dim(sub_cust_df)

## [1] 4347   10

# 4347 observations
# 10 variables

# Export subset to Excel file
write.xlsx(sub_cust_df, "customer_data_subset.xlsx", rowNames = TRUE)
```

## Summary of Data

Our first step was to read in our data set into RStudio. We see that our data set contains 5,000 observations and 60 variables. To make the analysis easier for coding purposes, we changed all the variables to lower case. We then looked to see how many columns in the data set had NA values. We saw that there was a variable named "x" that had NA values for every observation. We went ahead and just removed that entire column.

The first variable that we chose to include in our subset was number of pets, which tells us how many pets each customer has. The number of pets variable had 6 NA values. The number of cats, number of dogs and number of birds variables had 7, 8 and 34 NA values respectively. We assumed that the NA values in these columns equaled 0. After mutating all the NA values to 0, we just added the number of cats, the number of dogs and the number of birds observations together and imputed those summations into the number of pets column. This gave us a more accurate idea of the total number of pets each customer owns.

The second variable that chose for our subset was household size, which tells us how many people live in the house. We saw that household size had 8 NA values. To fill those NA values, we found the average household size based on if customers are married or unmarried. We found that the average household size for married couples is about 3 while the average household size for unmarried couples is about 1. We then imputed those into the NA values based on the customer's marital status.

The third variable that we chose for our subset was homeowner. The homeowner variable is a binary column that says either 1 for homeowner or 0 for non-homeowner. Since this is a binary column, we decided to replace the NA values for the mode for the column. The mode was 1 meaning that more customers own a home compared to those who do not.

The fourth variable that we chose for our subset was job category, which tells us what each customer does for work. When looking at the job category variable, we found multiple empty strings. We confirmed this by checking the unique values for job category. Those unique values were: Professional, Sales, Labor, Agriculture, Service, Crafts, and an empty string. After mutating those empty strings to NA, we counted that there were 15 NA values in the column. We then looked at the highest count of job category values for those that are in the union and for those that are not in the union. For customers in the union, the "Professional" job category was the highest count. For non-union customers, the "Sales" job category had the highest count. Therefore, we imputed "Professional" for all NA values for union customers, and we imputed "Sales" for all non-union customers.

The fifth variable that we chose for our subset was region. The region variable had 5 unique values: 1, 2, 3, 4 and 5. Each number represented a region that the customer lives in. For visual purposes, we wanted to convert those numbers to their respective regions. Using the data dictionary that was provided, we were able to see what region each number represented:

- 1: Northeast
- 2: Midwest
- 3: West
- 4: Southwest
- 5: Southeast

We then converted each number to the actual name of the region.

The sixth variable that we chose for our subset was household income, which tells us what the household income is for each customer. We found that the household income variable was a character vector, meaning each observation was represented by a string of characters. Since we wanted this variable to be numeric, we removed all non-numeric characters and then converted the column into a numeric column. We then looked at the distribution of the household income. The distribution was right skewed. Therefore, we tried a log transformation on the data, which made the distribution more normal. We then created a new variable to take the log of household income. We looked at a boxplot of the transformed variable to see if there were any outliers. We then removed those outliers from the data set as a whole.

The seventh variable that we chose for our subset was one that was created from multiple variables in the data set. We wanted to look at the total value of each customer over their tenure with the company. We looked at three variables: equipment over tenure, voice over tenure and data over tenure. These three variables gave us the dollar amount that each customer paid over their lifetime with the company. After converting the three variables to numeric variables, we added them together to find a total dollar value for each customer. We checked the distribution of the total value for each customer and saw that it was heavily skewed right. We then used a log transformation on the variable. This made the distribution slightly skewed left. However, this left skew was much less skewed than the right skewedness in the non-transformed distribution. After we did the log transformation, we then analyzed a boxplot and removed any outliers present.

The eighth variable that we chose for our subset was one that we also created from other variables in the data set. We wanted to look at the total debt for each customer by adding the credit debt and other debt variables together. Using the data dictionary, we saw that both columns were expressed in $100,000 USD. Therefore, we multiplied each variable by 100000 and added them together to create a total debt variable. We saw that the distribution was skewed heavily to the right. We removed any customers that had zero debt (which was only one) and did a log transformation. This made our distribution much more normal. We then removed any outliers present.

The ninth variable that we chose for our subset was car brand, which tells us if a customer's vehicle is either domestic or foreign. Along with the domestic and foreign classifications in the variable, there was also a "-1" classification. This was for any customer that did not have a vehicle. We converted any "-1" classification to be represented by "None".

The tenth and final variable that we chose for our subset was car value, which tells us the value of each customer's vehicle. For those that do not own a vehicle, there was a "$(1,000.00)" string. We then mutated the column to change the "$(1,000.00)" string to "0" if the customer's car brand was equal to "None". We then removed non-numeric characters and converted the car value variable to a numeric column. We then looked at the distribution of the variable and found that it is right skewed. Since we wanted to try a log transformation, we decided to remove any customer that does not have a vehicle and only focus on those that do have one. After doing so, we found that a log transformation did make the distribution much more normal. We then removed any outliers present.

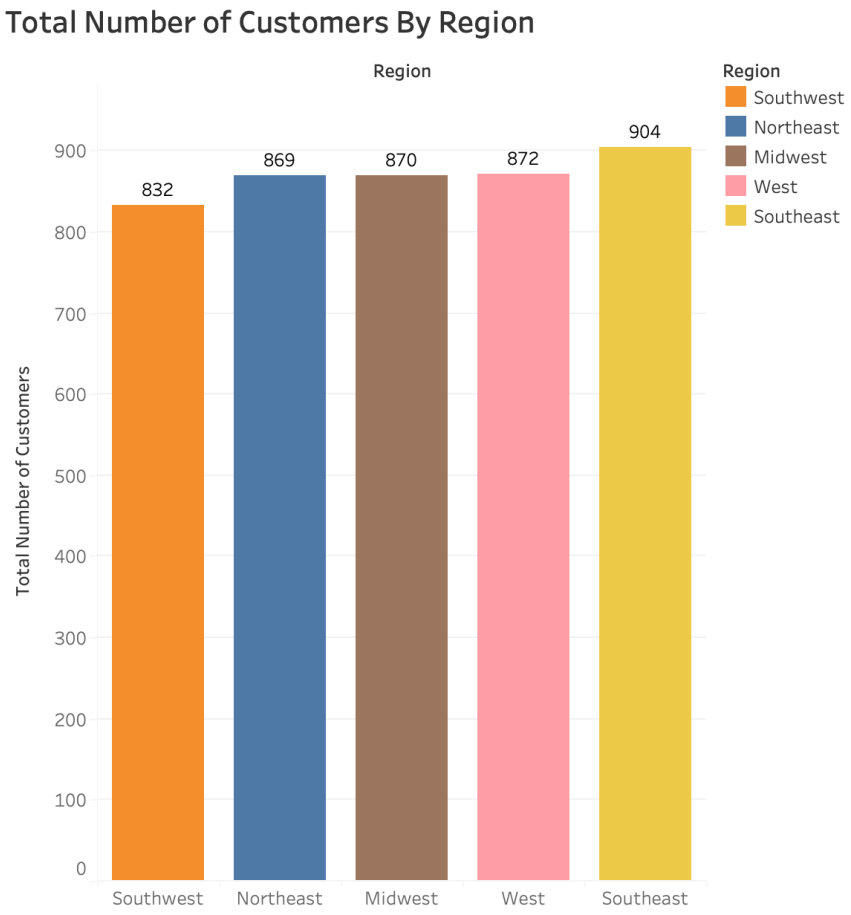
Our cleaned-up subset of the original data has 4,347 observations and 10 variables. Below you will find a table that shows each of the 10 variables that we have created and the characteristics of each variable. The first 4 variables are the qualitative variables, and the last 6 variables are the quantitative ones.

| **Variable** | **Data Type** | **Variable Type** |
|---|---|---|
| Homeowner | Qualitative | Binary |
| Job Category | Qualitative | Nominal |
| Region | Qualitative | Nominal |
| Car Brand | Qualitative | Nominal |

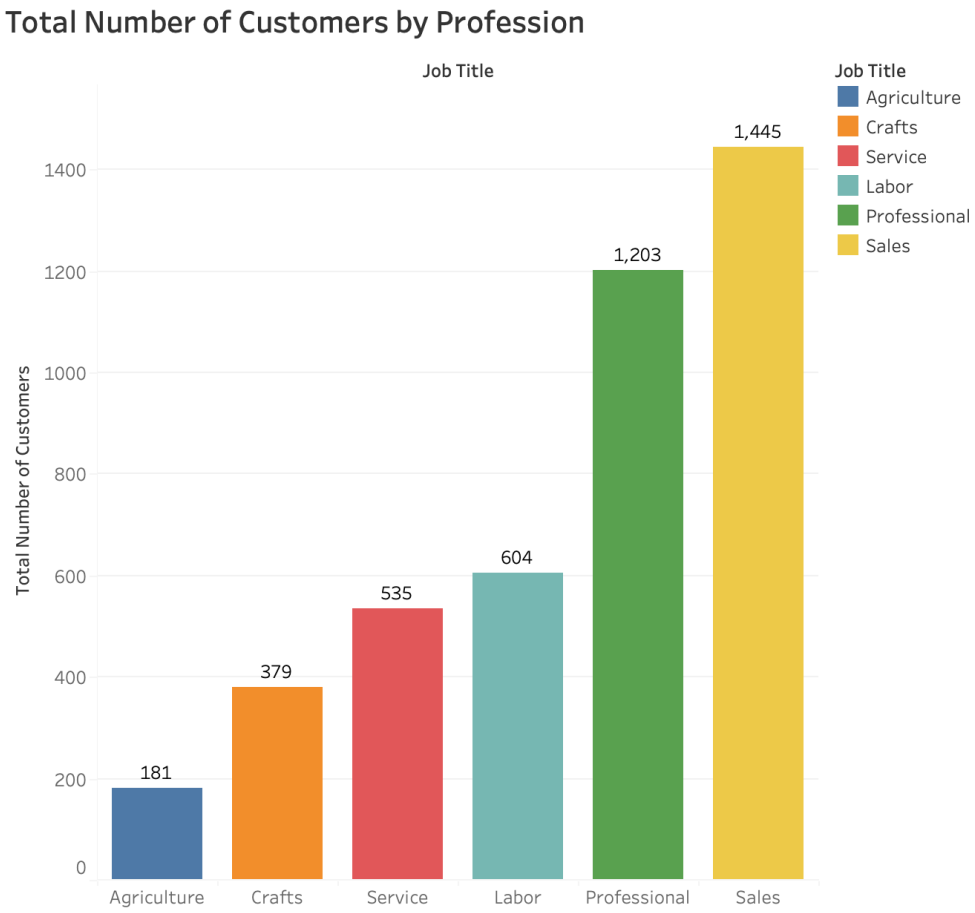
| Variable | Data Type | Variable Type |
|---|---|---|
| Number of Pets | Quantitative | Discrete |
| Household Size | Quantitative | Discrete |
| Log Household Income | Quantitative | Continuous |
| Log Total Value | Quantitative | Continuous |
| Log Total Debt | Quantitative | Continuous |
| Log Car Value | Quantitative | Continuous |

Now we will look at some visualizations using Tableau.

## Tableau Visualizations

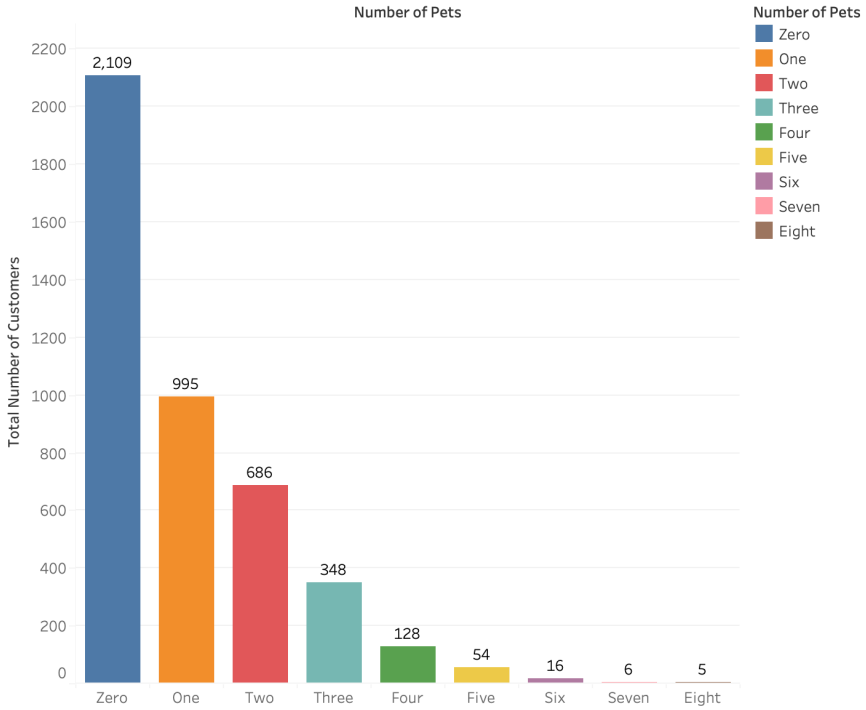



This plot shows the total number of customers by each region. You can see that the total number of customers in our data subset are evenly distributed throughout each region with most of the customers residing in the southeast region.



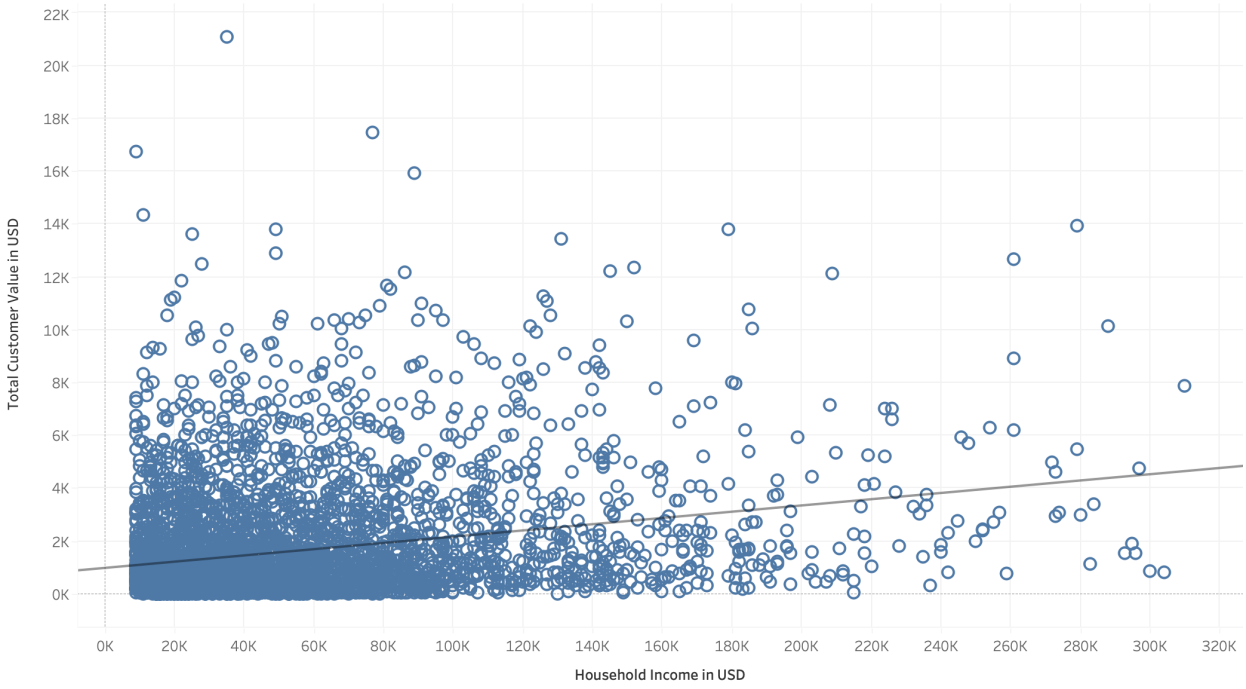

This plot shows the total number of customers by profession. We see that most customers work as “professionals” or work in “sales”.

Total Number of Customers by Number of Pets

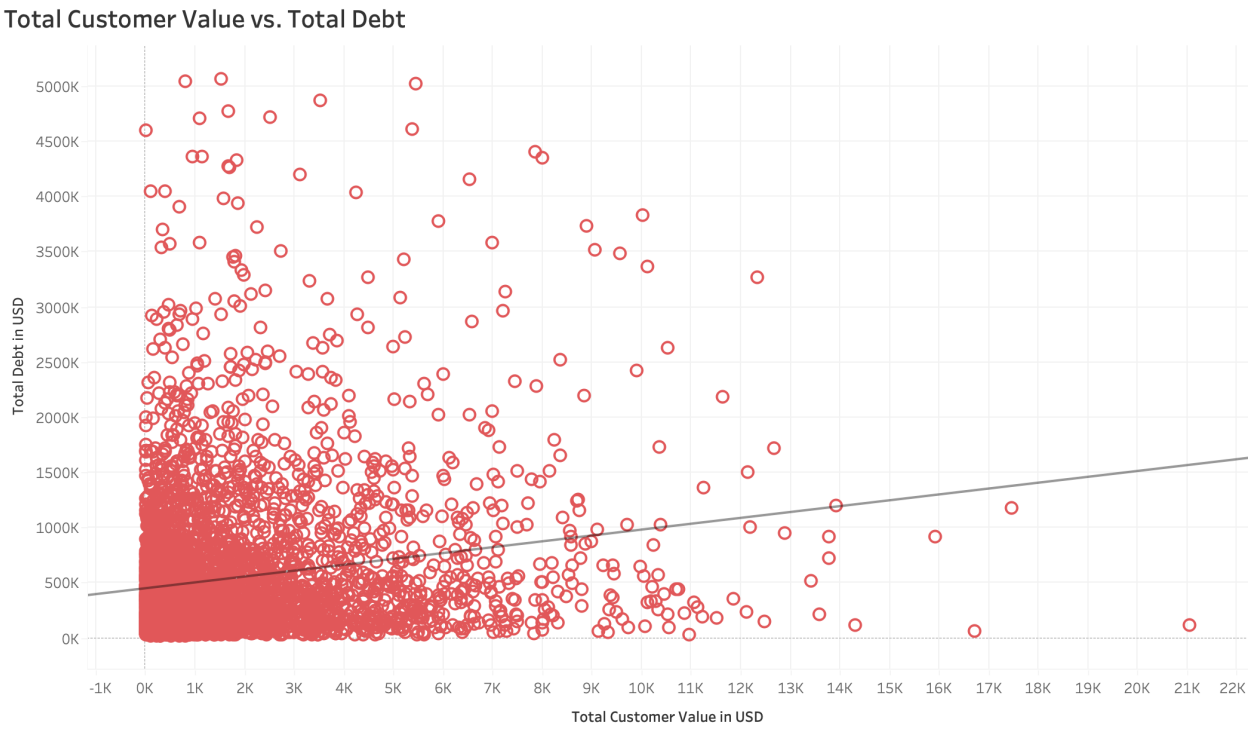



This plot shows the total number of customers based on the number of pets that they have. We can see that over 2100 customers have zero pets and a little under 1000 customers have zero pets.

Household Income vs. Total Customer Value



This plot shows that on average, as household income increases, the total customer value increases. We fit a regression line to show that there is a positive relationship between the two variables. It may not be the most linear relationship, but you can see that positive correlation between the two.

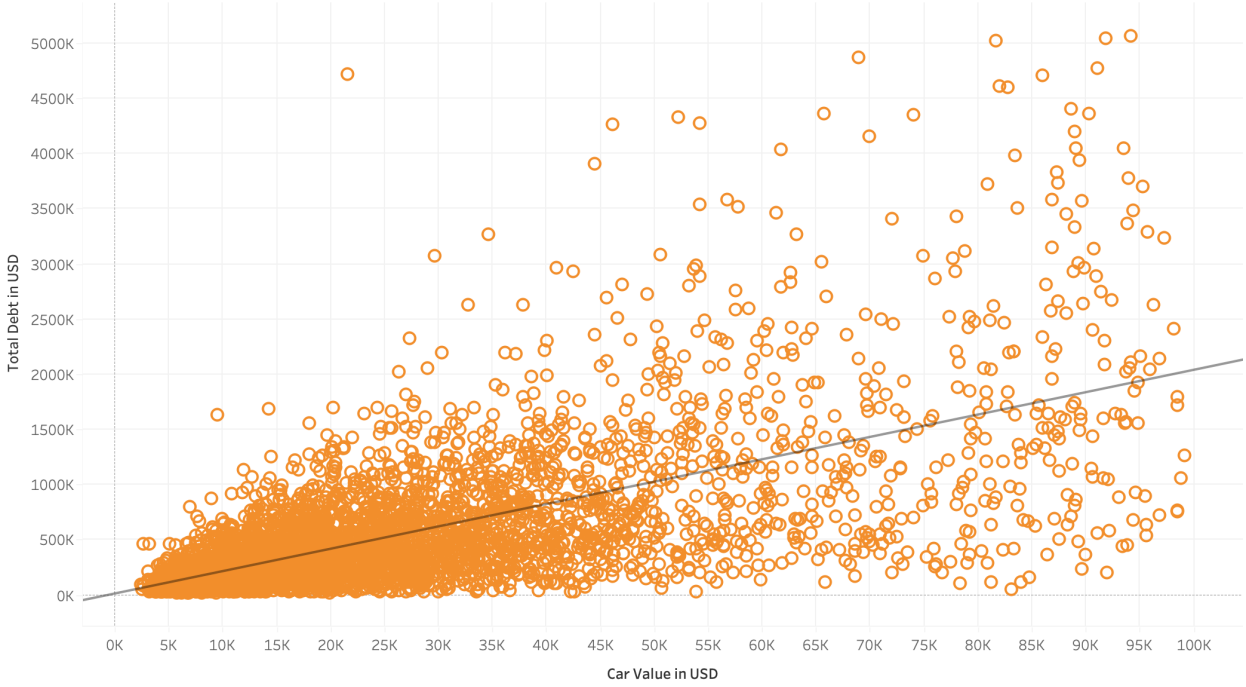



This plot shows that on average, as total customer value increases, total debt also increases. Like the previous plot that we looked at with household income and customer value, this plot shows a positive correlation. Once again, this may not be the most linear relationship.

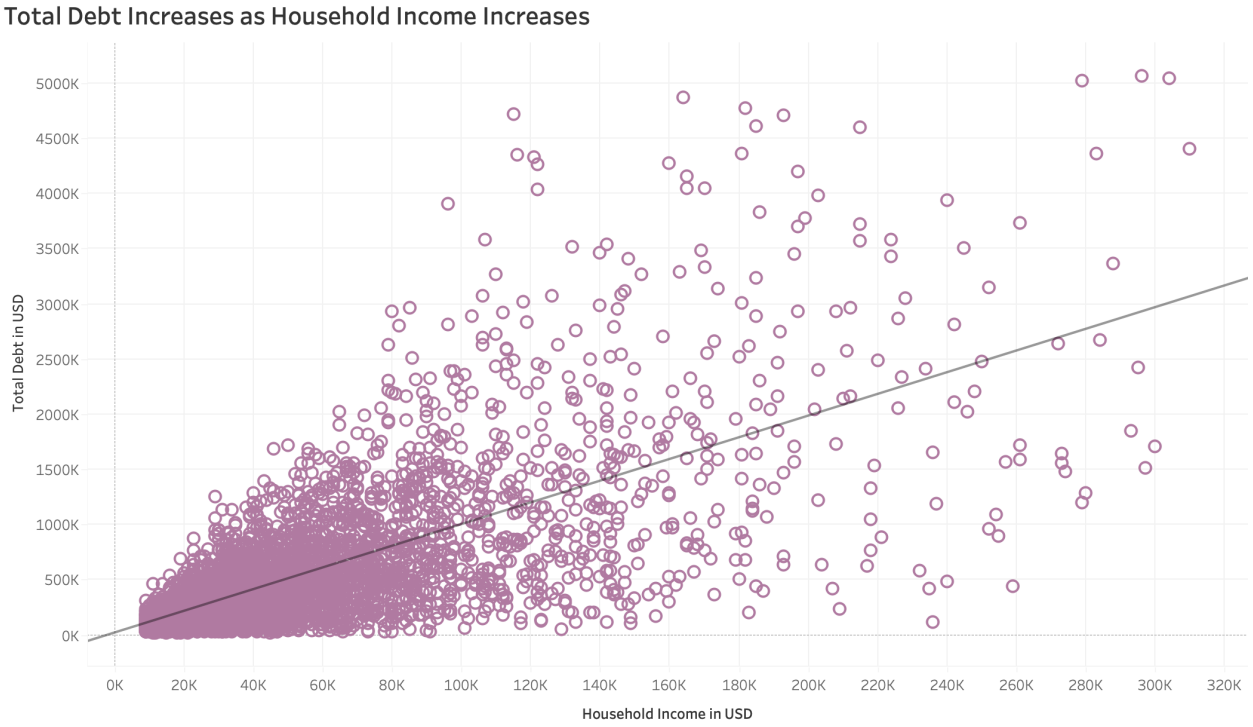
Car Value Increases as Household Income Increases


This plot shows the strong linear relationship between household income and car value. On average, we see that as household income increases, car value also increases. This makes sense. Relatively speaking, customers that make more money can afford more expensive vehicles.

Total Debt Increases as Car Value Increases



This plot shows a positive relationship between car value and total debt. As car value increases, you can see that total debt also increases. If a customer is purchasing a more expensive vehicle, they are most likely taking on more debt, assuming that they are financing the vehicle.




This plot shows a positive relationship between household income and total debt. Ultimately, the more money that customers make, the more debt they are willing to take on. There is a strong positive correlation between these two variables.

## Conclusion

In this exploratory data analysis project, we were able to generate a subset of 4,347 observations from the 5,000 observations in the original data set. We chose and created 10 variables to represent those observations. Using some visuals that we created in Tableau we were able to see the tendencies of the customers based on qualitative variables. We were also able to see some positive correlations between the continuous quantitative variables that we selected as well. Some quantitative variables had stronger relationships than others.

Now that we have created this subset and have concluded our exploratory data analysis, we can now move on to the segmentation and profiling portion of our analysis. We will be using this subset during the modeling portion to extract more information and gain more advantageous insights on marketing segmentation.