Joseph Annand

DSE5004

Project #2 Segmentation and Profiling

Summary

Goals

The goals of the project are to explore a customer dataset for a telecommunications company and come up with meaningful interpretations of the data and how it may be leveraged for better business decisions. The first step in achieving this is modifying the data set to better align the data with analytical goals and yield more insight from the information already provided. This includes handling null values in various columns and deriving new features for analysis. Once completed, exploratory data analysis is performed on the data set to gain insights on specific features and characterize the customer base as a whole. Segmentation of the data set is performed through three different methods: rule-based, supervised learning, and unsupervised learning. Each segment created by each approach is characterized using various visualizations.

Data Pre-Processing

Prior to segmentation, the data set is processed to better align features with analytical goals by improving usefulness. Initially, the monetary features are parsed for numbers to convert these columns from character vectors to numeric vectors. This is done to make it possible to use these features for mathematical operations and analysis as continuous variables. Next, null values in various numeric features were handled. For the "VoiceOverTenure", "VoiceLastMonth", "CardSpendMonth", and "TVWatchingHours" features, all null values were replaced with the median values of their respective column. The median was chosen over using the average to reduce the affect of outliers on these imputed values, which would affect the distributions of values in each of these columns. For the "EquipmentOverTenure", "EquipmentLastMonth", "DataOverTenure", and "DataLastMonth" features, if the customer was on record renting equipment or using wireless data services, then the null values were replaced with the medians of the respective columns. Otherwise, it was assumed that the customer does not pay for these services and the null values were set to zero. Additionally, five new features were created based on already-provided data. "TotalOverTenure" is a sum of how much a customer has spent on services through their entire tenure as a customer with the phone company. "TotalByTenure" is the average amount each customer spends on telecommunications services per month. "TotalLastMonth" is the total amount each customer spent on telecommunications services last month. "TotalDevices" is the total number of devices that each customer owns. "TotalServices" is the total number of telecommunications services each customer uses.

Approaches

The data set was segmented using three different approaches: rule-based segmentation, supervised learning with logistic regression, and unsupervised learning using K-means clustering. For each segmentation approach, three features were used to determine the different segments by each approach: the number of months that the customer has been a customer with the telecommunications company ("PhoneCoTenure"), the total amount spent in the last month ("TotalLastMonth"), and the total spending on primary credit card in a month ("CardSpendMonth").

Findings/Recommendations

Each approach to segmentation yields fairly different findings. In the rule-based segmentation approach, we find high value customers are generally older, wealthier, own more devices, and utilize more telecommunication services. None of which is particularly surprising given that we expect customers that are more valuable will have more money to spend and be more interested in telecommunication services. What is surprising is that the results from supervised learning segmentation yielded very different results. The logistic regression model used to create segments based on customer value classified more than double the number of customers as "High" value

compared to the rules-based segmentation approach. In the future, a thorough model evaluation should be performed on the logistic regression. Also, perhaps, there are better predictors to be used that may narrow down what is considered “high” value. The unsupervised approach using K-means clustering produced three different clusters. Cluster 2 seemed particularly interesting as this cluster had the highest average amount spent on services in the past month, was older, had more customers making $200K or more, and utilized more telecommunication services on average. Our recommendation is that the telecommunications should focus on attention on customers that are older, have higher incomes, and own more devices.

Technical Section

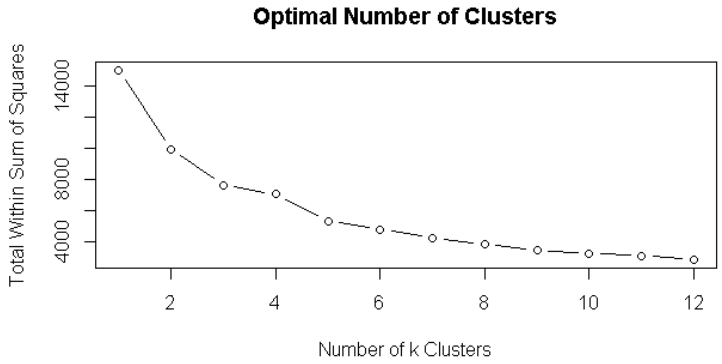
The data set was segmented using three different methods: rule-based segmentation, supervised learning with a logistic regression model, and unsupervised learning with K-mean clustering. The first method, rule-based segmentation, was performed initially by segmenting customers based on three features: the number of months that the customer has been a customer with the telecommunications company (“PhoneCoTenure”), the total amount spent in the last month (“TotalLastMonth”), and the total spending on primary credit card in a month (“CardSpendMonth”). For “PhoneCoTenure”, if the customer had been a customer for longer than 40 months, they were classified as “Long” to indicate a long tenure as a customer. If the customer’s tenure was shorter than 40 months, they were classified as “Short.” The break point for classification based on the tenure feature was chosen to be 40 months based on a histogram of the tenures of all customers; 40 months was slightly greater than the median and determined to be a significant point where the peak of the distribution dropped to mark a good separation point between newer customers and loyal customers.

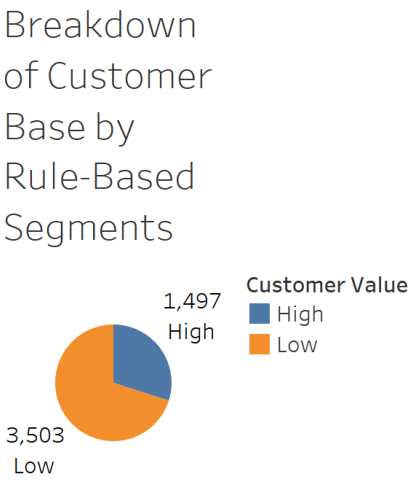
Customers were classified as either “Low” or “High” based on the values for “TotalLastMonth” and “CardSpendMonth” features. Customers who paid more than $70 for telecommunications services were classified as “High” to separate customers who recently spent above average amounts on telecommunication services. Similarly, customers who spent more than $5000 on their primary credit card were classified as “High” to indicate that they spend more than average and may have more disposable income for additional or more expensive services. Other customers spending below $5000 were classified as “Low”.

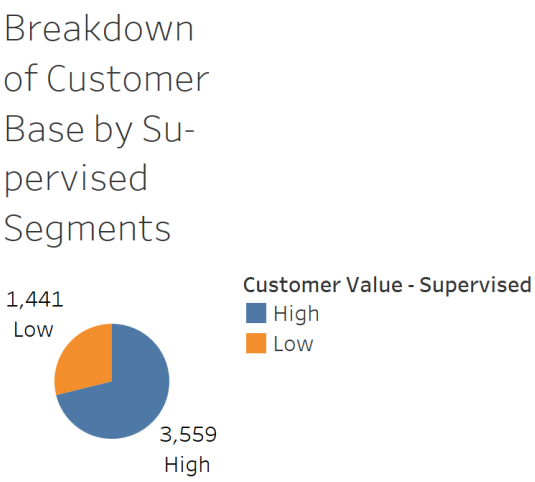
After classifications for each feature were assigned, eight segments were created for every possible combination of the different classifications. Classifications of interest were determined to be “High” in card spending and last month total for services and “Long” in phone company tenure. Customers that were placed in segments that met two of those classifications of interest were placed in a “High” customer value segment and all others were placed in a “Low Value” segment.

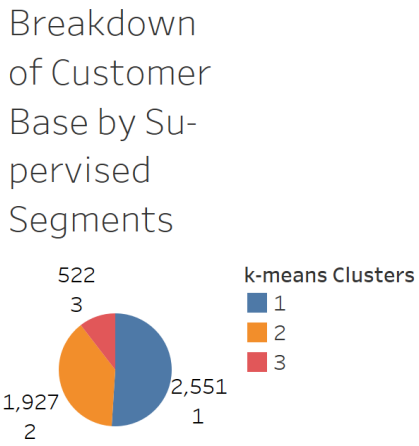
The second method for segmentation used was supervised learning using logistic regression. Logistic regression was chosen because the goal was two classify customers as “High” or “Low” value as was done with the rule-based segmentation approach. The logistic regression model was created using the customer values segments from the rule-based segmentation as the target variable, while the three features used in the rule-based segmentation approach were set as the predictor variables. Once the model was created, it was used to predict the probability for each customer of belonging to the “High” customer value segment. Customers with probabilities above 0.50 were assigned to the “High" customer value segment while customers with probabilities below 0.5 were assigned to the “Low" customer value segment.

The third method used for segmenting the customer data set was unsupervised learning with K-means clustering. In this approach, the same three features were used for segmentation. Prior to clustering, each feature was standardized. Then, the total within sum of squares was plotted as a function of number of k clusters from one to twelve clusters to determine the best number of clusters for segmenting the customer data set. The optimal number was determined to be 3. K-means clustering was applied to the data set to classify each customer in one of three clusters.

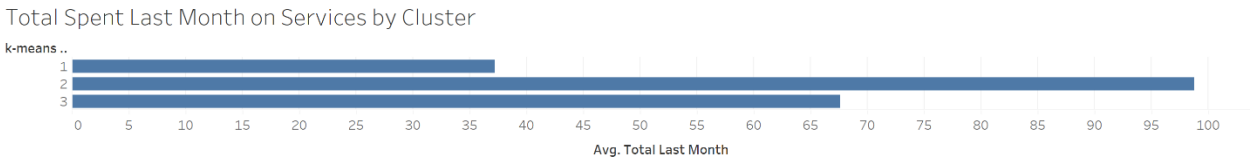








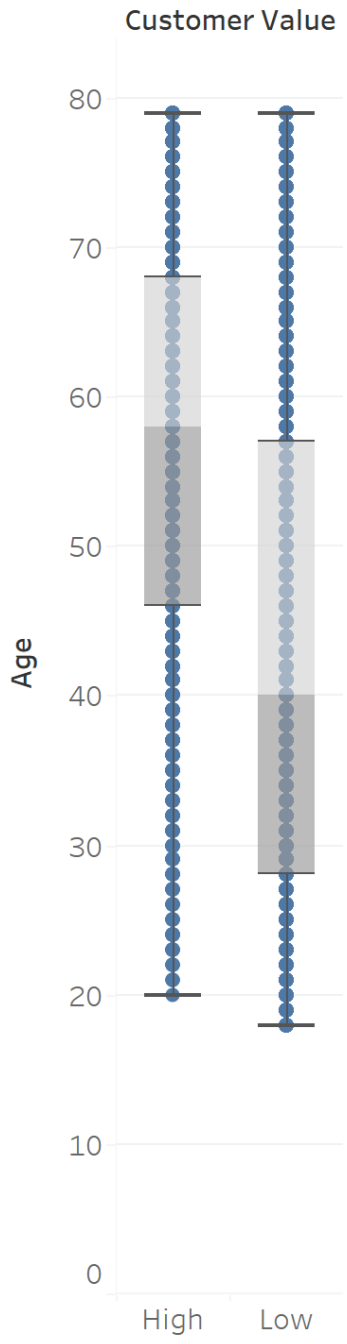
The rule-based segmentation approach assigns 1,497 customers, about 30%, to the “High” value segment while the logistic regression approach assigns 3,559 customers, 71%, to the “High” value segment. K-means clustering separates the customer data set into three segments. Segment 1 contains 2,551 customers, about 51% of customers, Segment 2 contains 1,927 customers, about 39% of customers, and Segment 3 contains 522 customers, about 10% of customers.



Average of Total Last Month for each k-means Clusters.

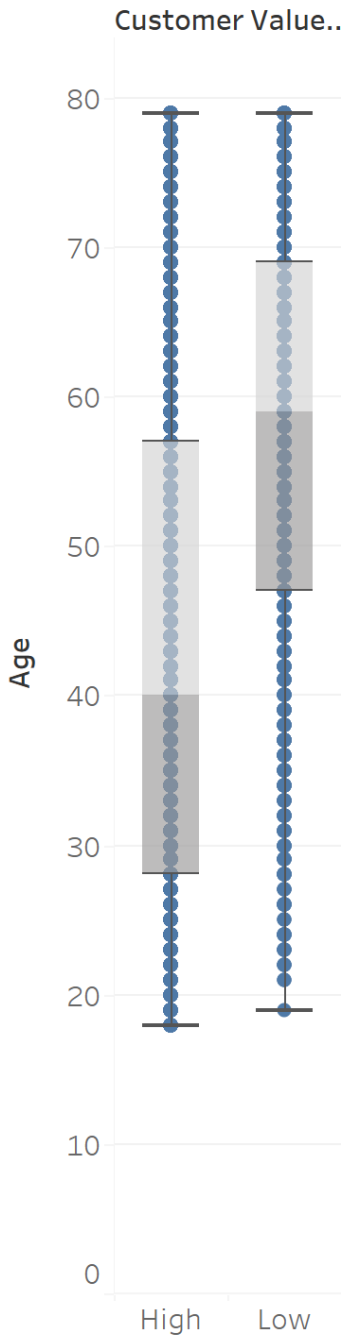
Looking at the segmented clusters from the unsupervised approach, Cluster 2 has the highest average total amount spent on telecommunication services in the last month.

Age Distribution in Rule-Based Segmentation



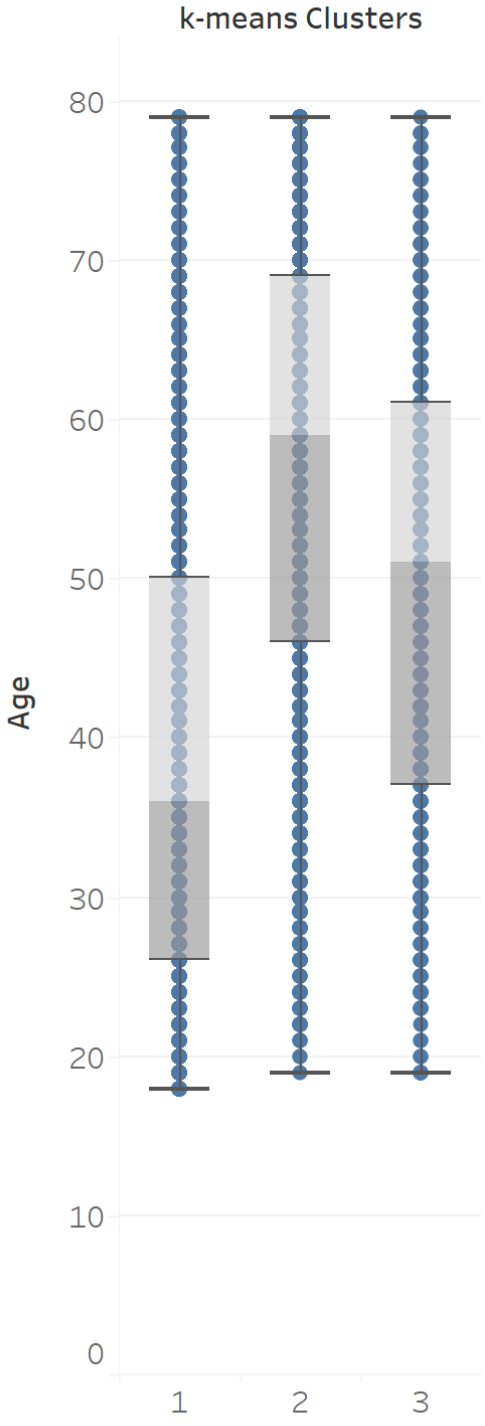
Age for each Customer Value.

Age Distribution Using Supervised Segmentation



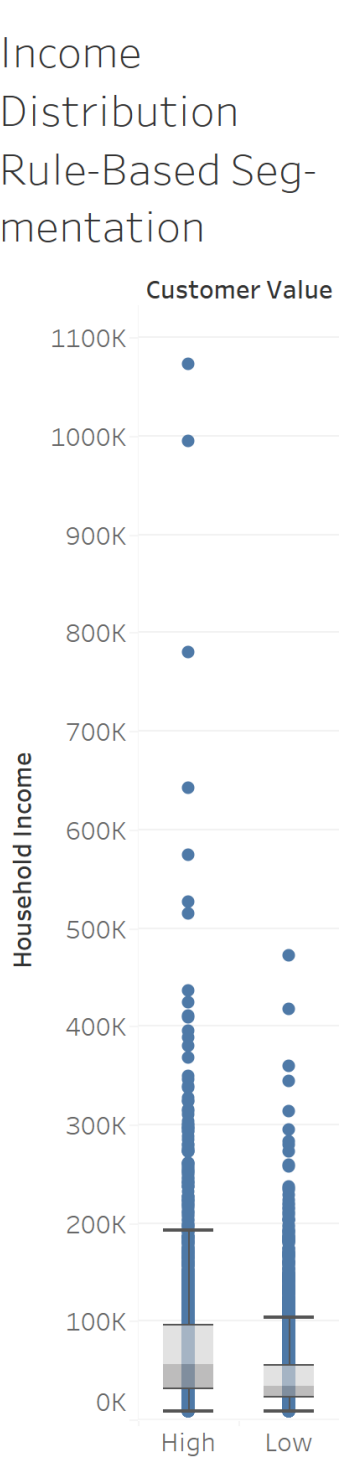
Age for each Customer Value - Supervised.

Age Distribution Using Unsupervised Segmentation

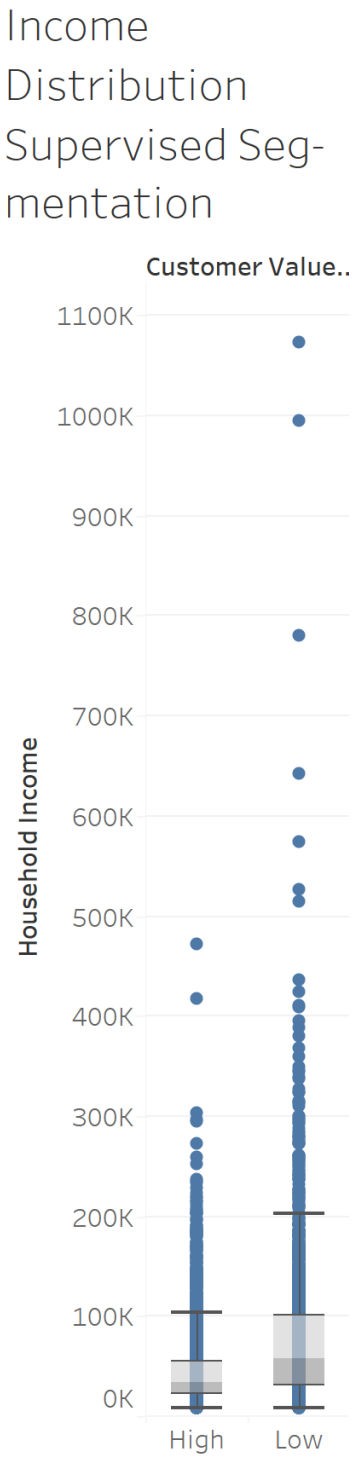


Age for each k-means Clusters.

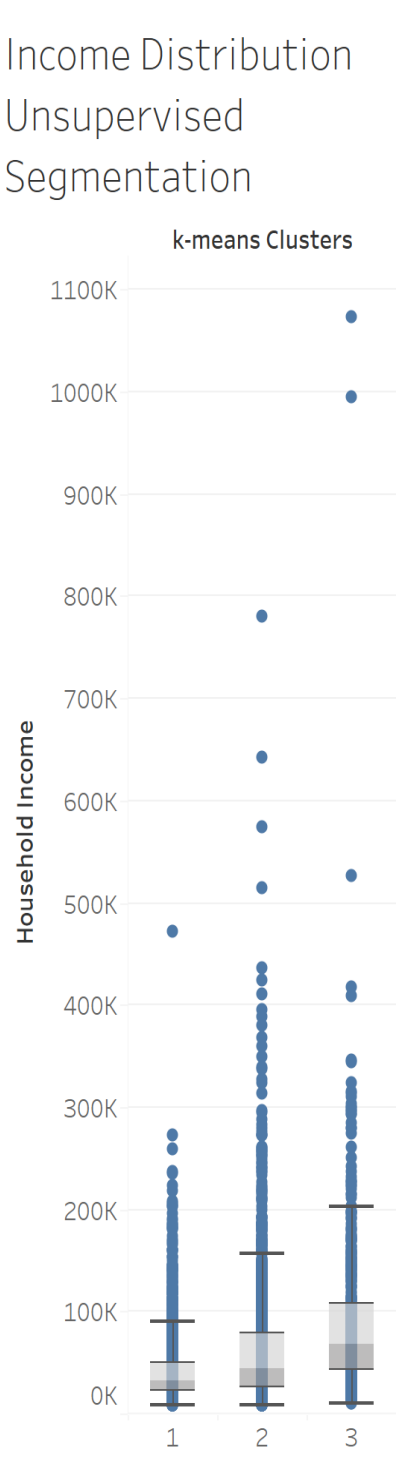
The distribution of ages for customers in the rules-based segments shows that high value customers tend to be older than those that are classified as low value. The opposite is true for segments determined by the supervised approach using logistic regression. For the unsupervised approach, the oldest cluster on average is Cluster 2, which is also the cluster that recently spent the most on services.



Household Income for each Customer Value.

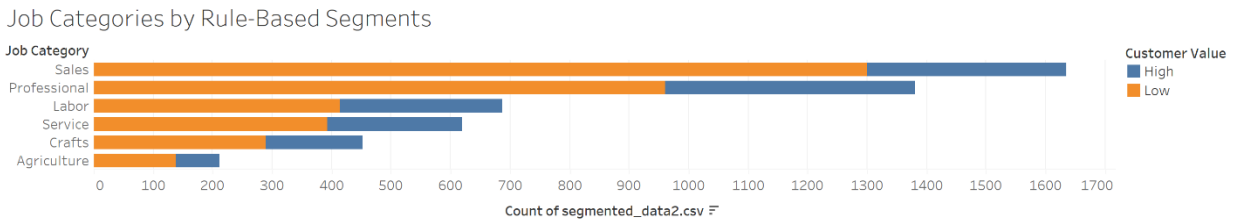


Household Income for each Customer Value - Supervised.

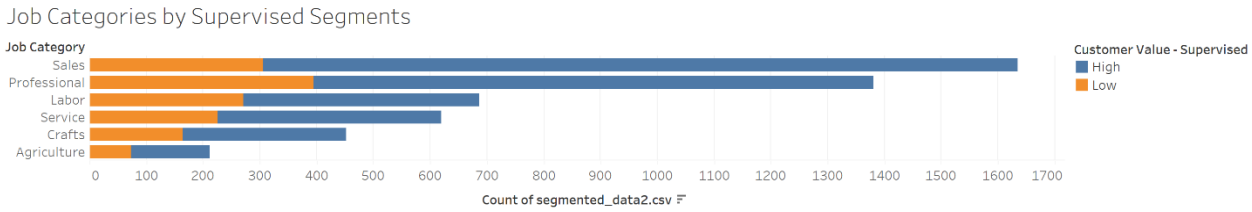


Household Income for each k-means Clusters.

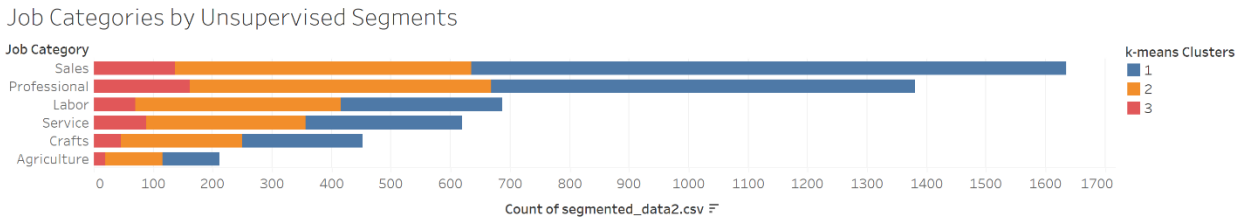
Median household income for high value customers according to rule-based segmentation is slightly higher than that of low value customers. More high value customers are also outliers and make $300K or more than low value customers. The opposite is true for the segments determined by supervised learning approach. In the unsupervised learning approach, Cluster 3 has the highest median household income amongst the three clusters, but Cluster 2 has more high-income households making above $200K.



Count of segmented_data2.csv for each Job Category. Color shows details about Customer Value. The view is filtered on Job Category, which excludes Null.

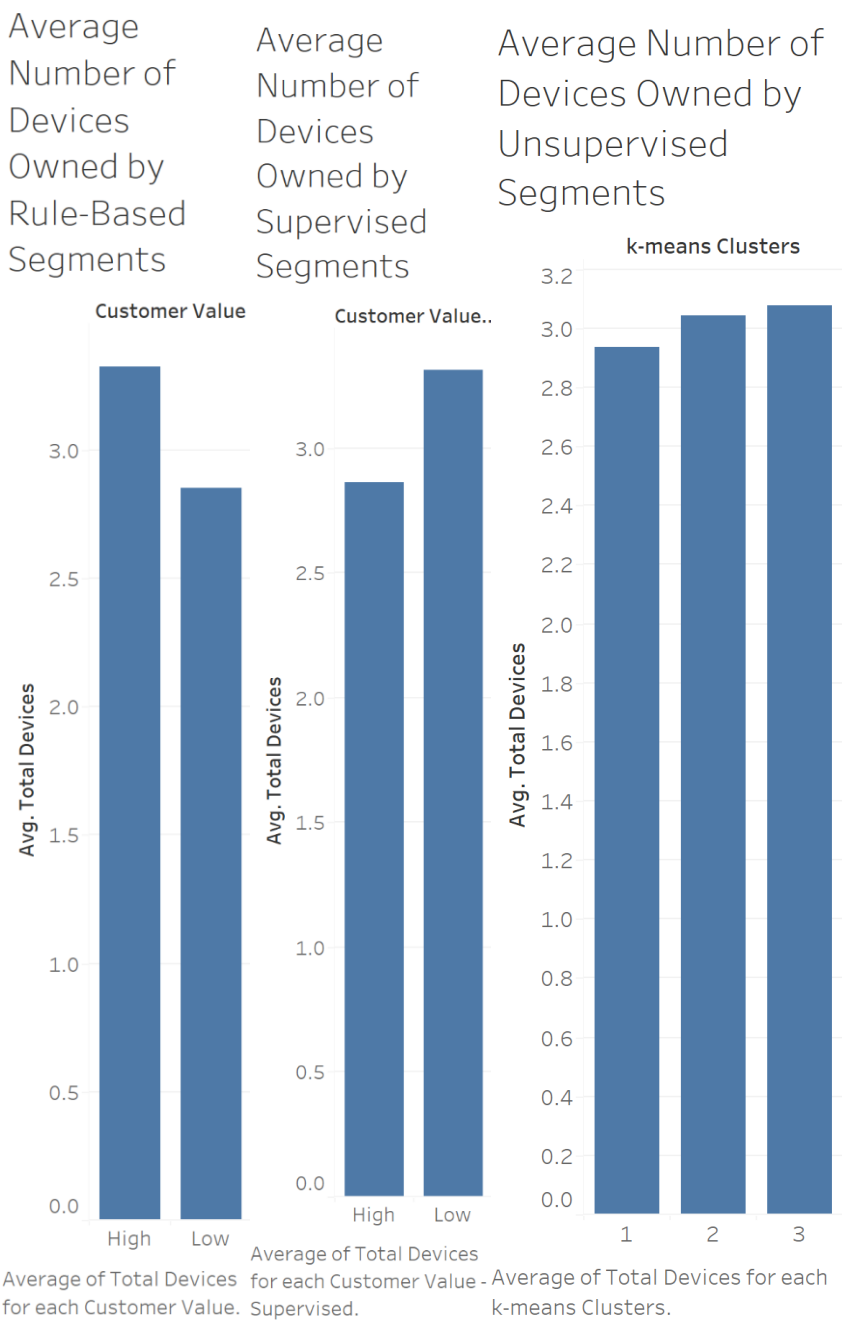


Count of segmented_data2.csv for each Job Category. Color shows details about Customer Value - Supervised. The view is filtered on Job Category, which excludes Null.

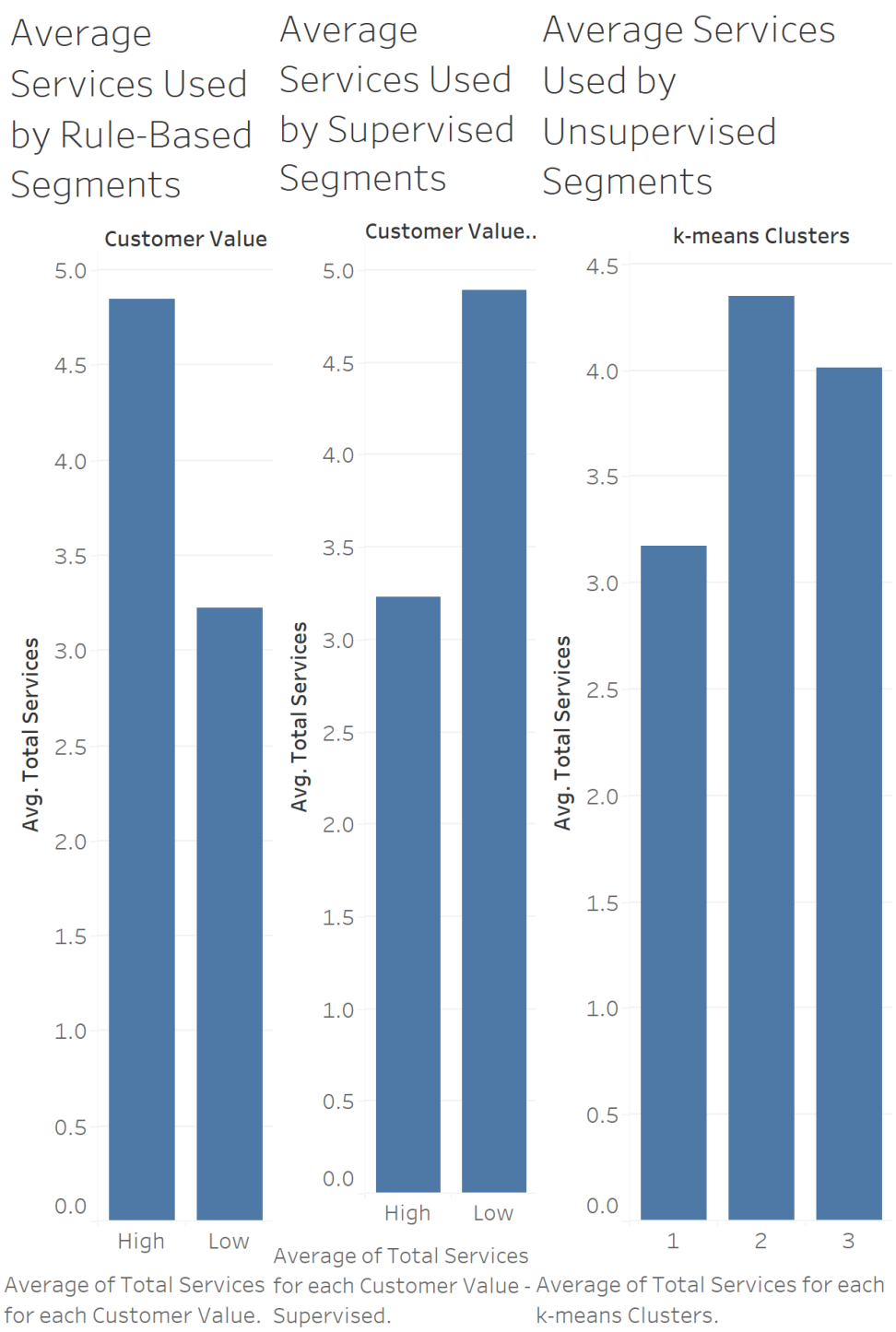


Count of segmented_data2.csv for each Job Category. Color shows details about k-means Clusters. The view is filtered on Job Category, which excludes Null.

Across the job categories, proportions of customers in each value group align with the proportions of each value group across the full data set. The job category with the most high value customers is “Professional.” Similar proportionality is seen in the supervised learning results, while the job category with most high value customers is “Sales.” In the unsupervised segments, the job categories are split fairly evenly between Clusters 1 and 2 while, unsurprisingly, a small portion of Cluster 3 is seen in each category.



High value customers determined by rules-based segmentation, on average, have more devices than low value customers, on average. The opposite is true in the supervised learning results. While Cluster 3 has the highest average amount of devices, there is not a significant difference between all the clusters in the results for the unsupervised learning approach.



High value customers according to rules-based segmentation approach utilize significantly more services than low value customers. The opposite observation is seen in the segments determined using supervised learning. Regarding the unsupervised learning segments, Cluster 2 uses the most services, on average, followed by Cluster 3 and then Cluster 1.

Appendix

```
original_data <- read.csv("data/Customer_Dataset_Data.csv", na.strings
= "?", stringsAsFactors = F)
original_data <- original_data %>% column_to_rownames(., var = "Custom
erID")

data <- original_data

############################## Data Pre-Processing ##################
##########

# Convert monetary columns from character to numeric

data$VoiceOverTenure <- parse_number(data$VoiceOverTenure)

## Warning: 3 parsing failures.
##  row col expected actual
##  842  -- a number #NULL!
## 2758  -- a number #NULL!
## 3480  -- a number #NULL!

data$VoiceLastMonth <- parse_number(data$VoiceLastMonth)

data$EquipmentOverTenure <- parse_number(data$EquipmentOverTenure)

## Warning: 3296 parsing failures.
## row col expected actual
##   3  -- a number    $ -
##   4  -- a number    $ -
##   5  -- a number    $ -
##   7  -- a number    $ -
##   8  -- a number    $ -
## ... ... ........ ......
## See problems(...) for more details.

data$EquipmentLastMonth <- parse_number(data$EquipmentLastMonth)

## Warning: 3296 parsing failures.
## row col expected actual
##   3  -- a number    $ -
##   4  -- a number    $ -
##   5  -- a number    $ -
##   7  -- a number    $ -
##   8  -- a number    $ -
## ... ... ........ ......
## See problems(...) for more details.
```

```
data$DataOverTenure <- parse_number(data$DataOverTenure)
```

```
## Warning: 3656 parsing failures.
## row col expected actual
##   1  -- a number    $ -
##   3  -- a number    $ -
##   4  -- a number    $ -
##   6  -- a number    $ -
##   7  -- a number    $ -
## ... ... ........ ......
## See problems(...) for more details.
```

```
data$DataLastMonth <- parse_number(data$DataLastMonth)
```

```
## Warning: 3656 parsing failures.
## row col expected actual
##   1  -- a number    $ -
##   3  -- a number    $ -
##   4  -- a number    $ -
##   6  -- a number    $ -
##   7  -- a number    $ -
## ... ... ........ ......
## See problems(...) for more details.
```

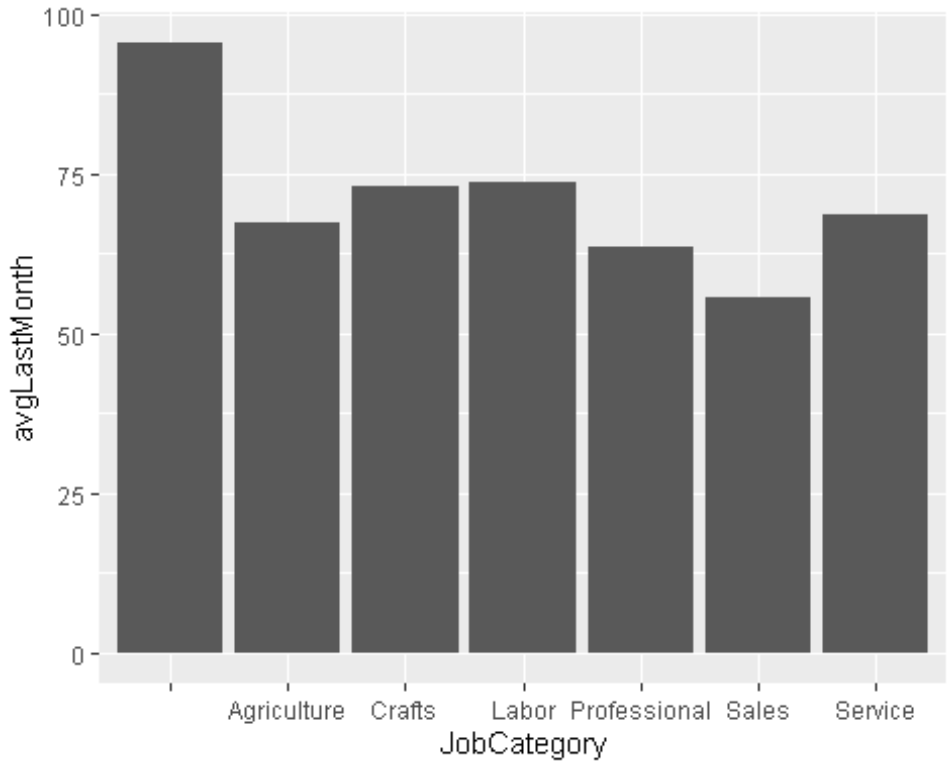
```
data$CardSpendMonth <- parse_number(data$CardSpendMonth)
```

```
## Warning: 7 parsing failures.
##  row col expected actual
## 1658  -- a number    $ -
## 1717  -- a number    $ -
## 2800  -- a number    $ -
## 2879  -- a number    $ -
## 4100  -- a number    $ -
## .... ... ........ ......
## See problems(...) for more details.
```

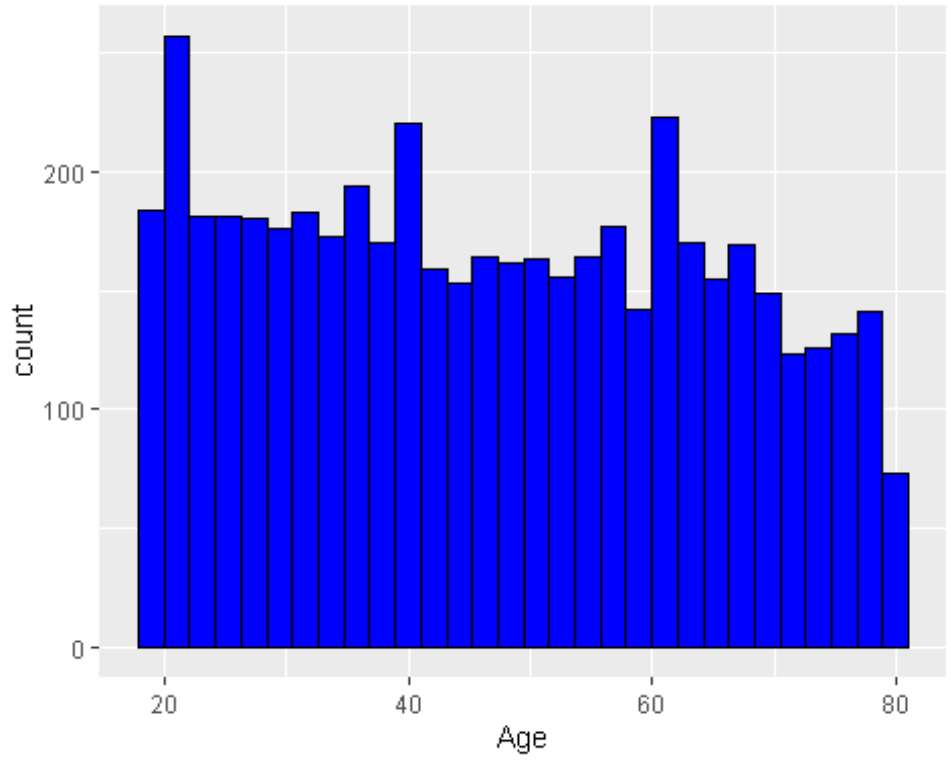
```
data$HHIncome <- parse_number(data$HHIncome)

# Handle null values in monetary columns

data$VoiceOverTenure <- replace_na(data$VoiceOverTenure, median(data$V
oiceOverTenure))
data$VoiceLastMonth <- replace_na(data$VoiceLastMonth, median(data$Voi
ceLastMonth))


data$VoiceOverTenure[is.na(data$VoiceOverTenure)] <- median(data$Voice
OverTenure)
```

```
data$VoiceLastMonth[is.na(data$VoiceLastMonth)] <- median(data$VoiceLa
stMonth)

handle_na <- function(data, col1, col2) {
  med_val <- median(data[[col1]], na.rm = TRUE)  # Calculate median of
Column 1
  for (i in 1:nrow(data)) {
    if (is.na(data[i, col1])) {
      if (data[i, col2] == "Yes") {
        data[i, col1] <- med_val
      } else {
        data[i, col1] <- 0
      }
    }
  }
  return(data)
}

data <- handle_na(data, "EquipmentOverTenure", "EquipmentRental")
data <- handle_na(data, "EquipmentLastMonth", "EquipmentRental")

data <- handle_na(data, "DataOverTenure", "WirelessData")
data <- handle_na(data, "DataLastMonth", "WirelessData")

data$CardSpendMonth[is.na(data$CardSpendMonth)] <- median(data$CardSpe
ndMonth, na.rm = T)
data$TVWatchingHours[is.na(data$TVWatchingHours)] <- mean(data$TVWatch
ingHours, na.rm = T)


# Derived Features

data$TotalOverTenure <- data$VoiceOverTenure + data$EquipmentOverTenur
e + data$DataOverTenure
data$TotalByTenure <- data$TotalOverTenure / data$PhoneCoTenure
data$TotalLastMonth <- data$VoiceLastMonth + data$EquipmentLastMonth +
data$DataLastMonth


data$TotalServices <- 0

data[2,"EquipmentRental"]

## [1] "Yes"

for (x in (1:nrow(data))) {
  for (y in c("EquipmentRental","WirelessData","Multiline","VM","Pager
```

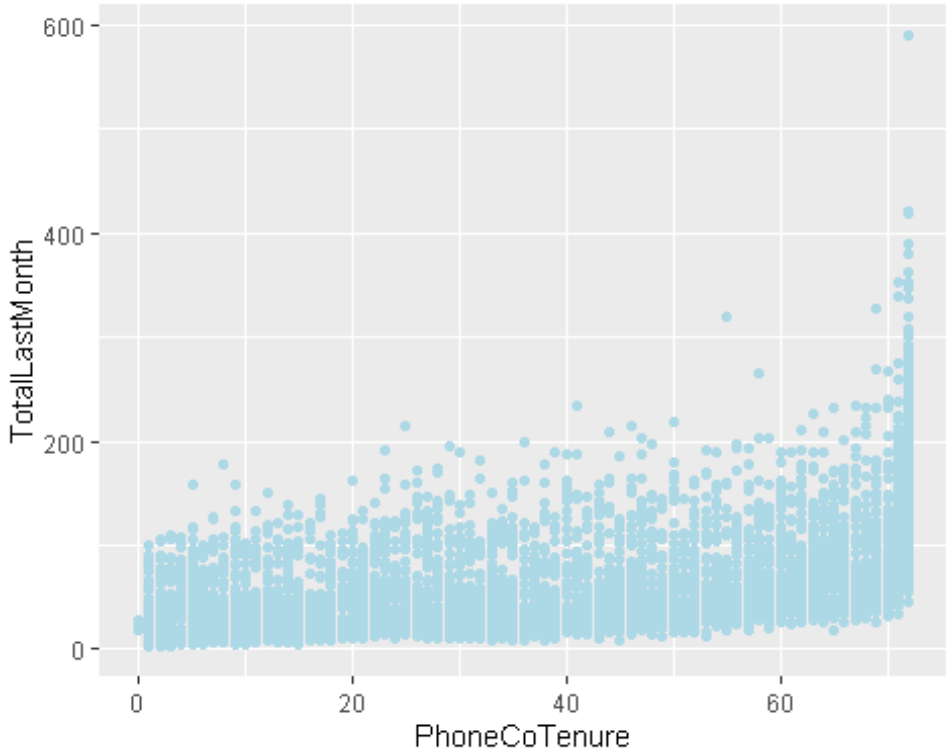
```
","Internet",
              "CallerID","CallWait","CallForward","ThreeWayCalling"))
{
    if (data[x, y]=="Yes") {
      data[x, "TotalServices"] <- data[x, "TotalServices"] + 1
    }
  }
}

data$TotalDevices <- 0

for (x in (1:nrow(data))) {
  for (y in c("Pager","OwnsPC","OwnsMobileDevice","OwnsGameSystem","Ow
nsFax")) {
    if (data[x, y]=="Yes") {
      data[x, "TotalDevices"] <- data[x, "TotalDevices"] + 1
    }
  }
  if (data[x,"TVWatchingHours"] != 0) {
    data[x, "TotalDevices"] <- data[x, "TotalDevices"] + 1
  }
}

# data$LogIncome <- log10(data$HHIncome)
# data$LogTotalLastMonth <- log10(data$TotalLastMonth)

############################ Exploratory Data Analysis ################
##########

data_occupation <- data %>%
  group_by(JobCategory) %>%
  summarise(avgLastMonth = mean(TotalLastMonth))

a <- ggplot(data = data_occupation) +
  geom_col(aes(x=JobCategory, y=avgLastMonth))
print(a)
```



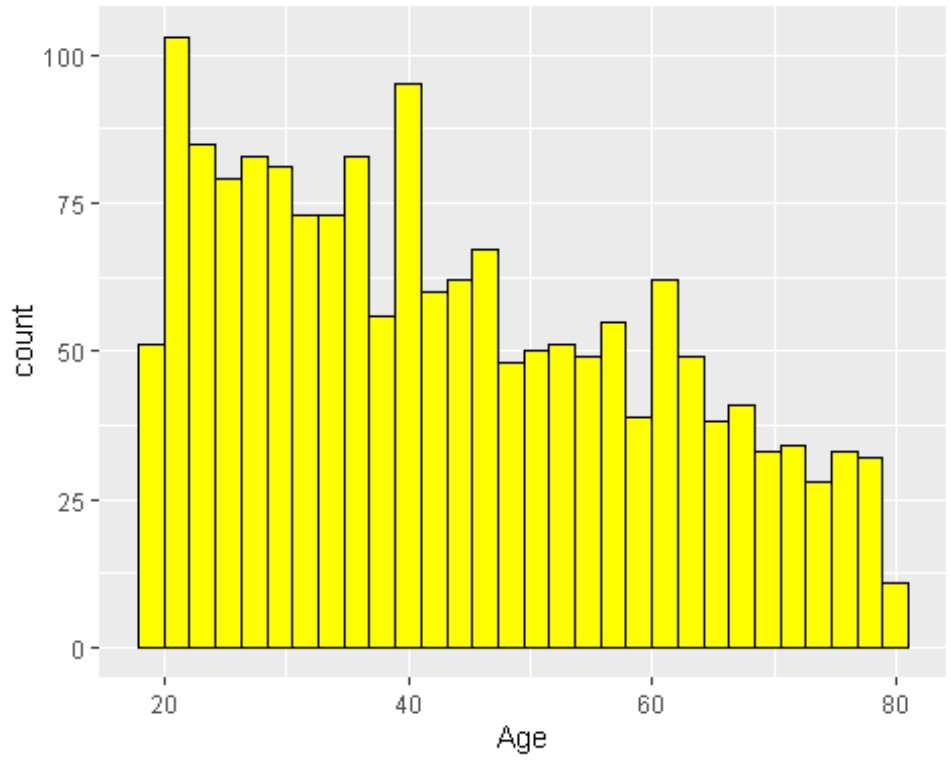

```
b <- ggplot(data = data) +
  geom_histogram(aes(x=Age), bins=30, color = "black", fill = "blue")
print(b)
```



```
c <- ggplot(data = data) +
  geom_point(aes(x=PhoneCoTenure, y=TotalLastMonth), color = "lightblu
e")
print(c)
```




```
data_equip <- data %>%
  add_count(EquipmentRental) %>%
  group_by(EquipmentRental,n) %>%
  summarise(avgLastMonth = mean(TotalLastMonth))

## `summarise()` has grouped output by 'EquipmentRental'. You can over
ride using
## the `.groups` argument.

d <- ggplot(data = data %>% filter(EquipmentRental == "Yes")) +
  geom_histogram(aes(x=Age), color="black", fill="yellow", bins=30)
print(d)
```

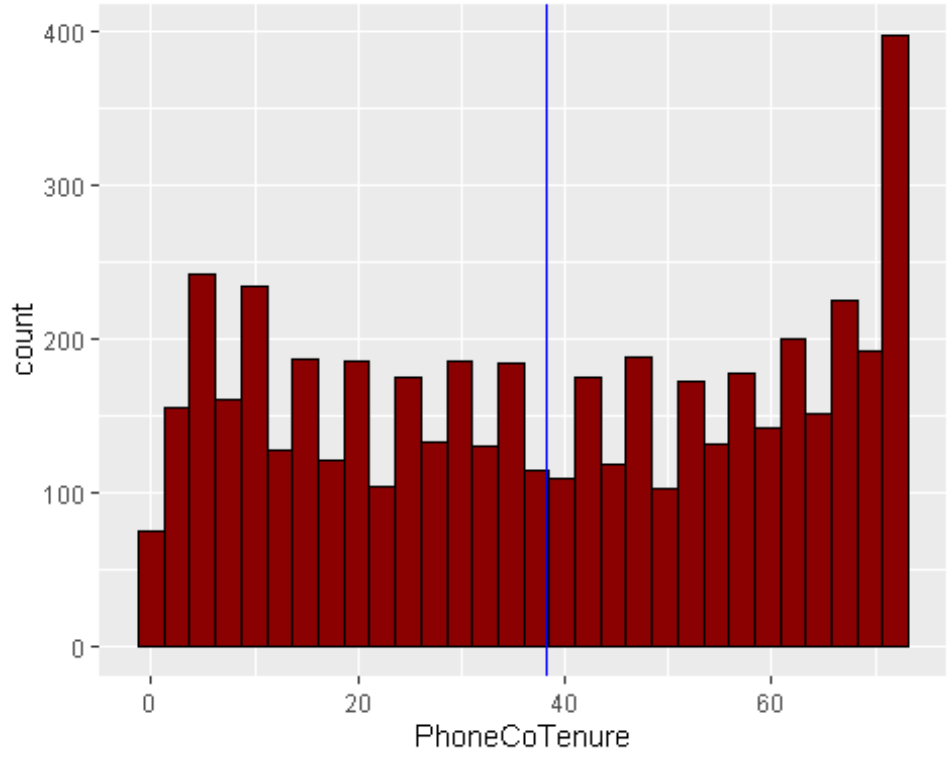



```
data_wireless <- data %>%
  add_count(WirelessData) %>%
  group_by(WirelessData,n) %>%
  summarise(avgLastMonth = mean(TotalLastMonth))
```

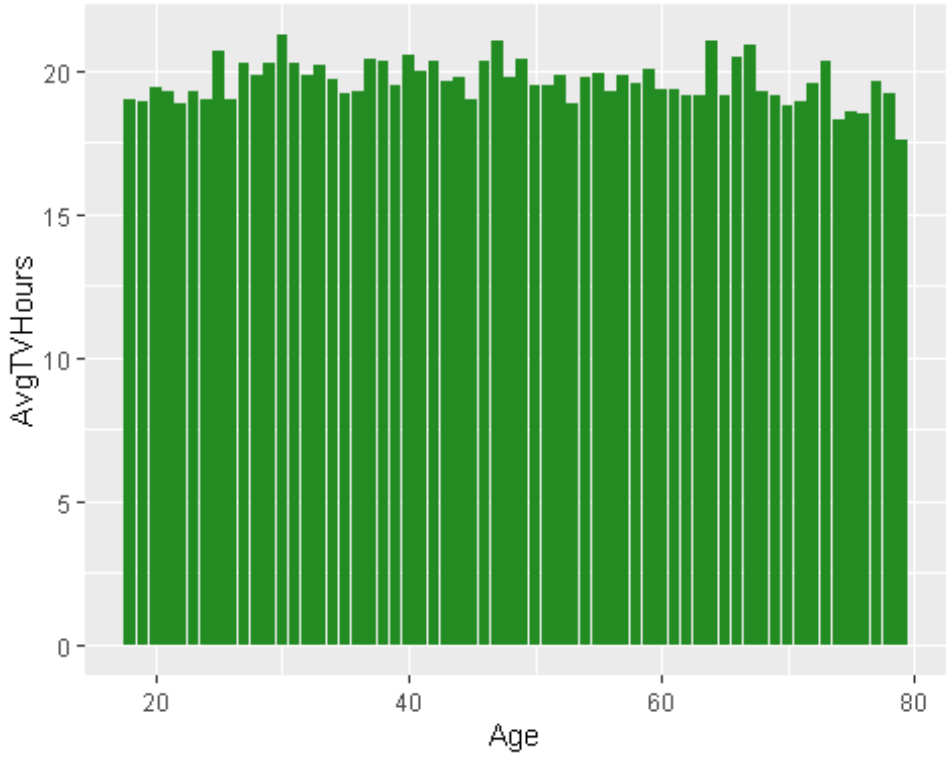
```
## `summarise()` has grouped output by 'WirelessData'. You can overrid
e using the
## `.groups` argument.
```

```
# data <- data %>% mutate(segment = case_when(
#   EquipmentRental == "Yes" & WirelessData == "Yes" ~ 1,
#   EquipmentRental == "Yes" & WirelessData == "No" ~ 2,
#   EquipmentRental == "No" & WirelessData == "Yes" ~ 3,
#   EquipmentRental == "No" & WirelessData == "No" ~ 4,
# ))
#
# data_segmented <- data %>% add_count(segment) %>%
#   group_by(segment,n) %>%
#   select_if(is.numeric) %>%
#   summarise_all("median")

e <- ggplot(data = data) +
  geom_histogram(aes(x=PhoneCoTenure), color="black", fill="darkred")
+
```

```
  geom_vline(xintercept = mean(data$PhoneCoTenure), color="blue")
print(e)

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

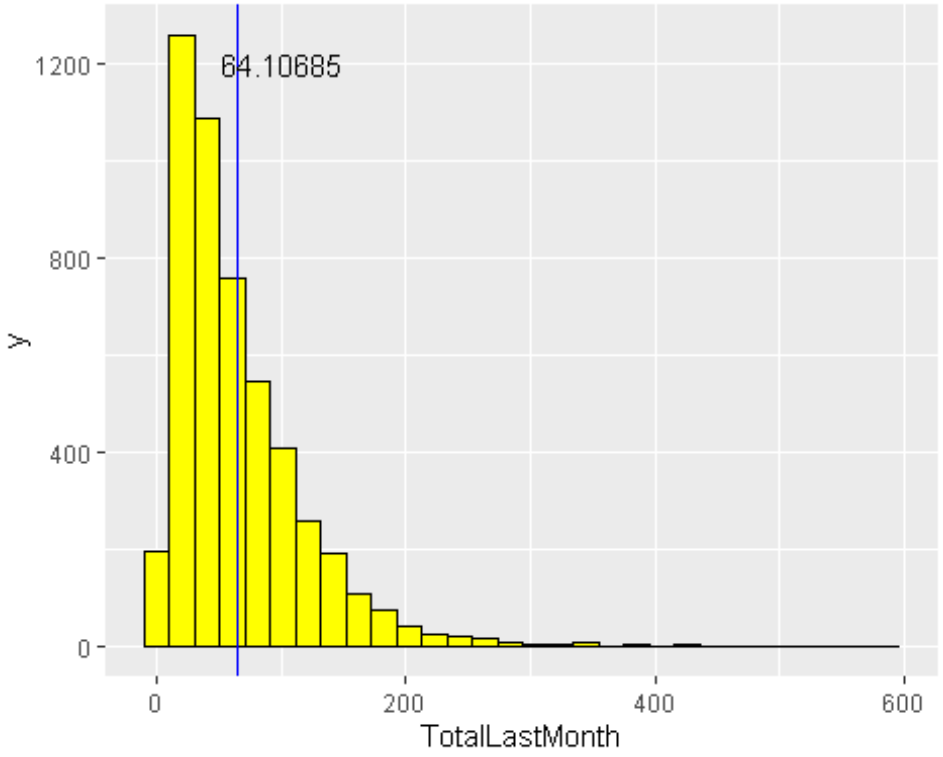



```
# f <- ggplot(data = data) +
#   geom_point(aes(x=LogIncome, y=LogTotalLastMonth), color="blue")
# print(f)

# cor(data$LogIncome, data$LogTotalLastMonth)

g <- ggplot(data = data %>% select(Age, TVWatchingHours) %>%
              group_by(Age) %>%
              summarise(AvgTVHours = mean(TVWatchingHours))) +
  geom_col(aes(x=Age, y=AvgTVHours), fill="forestgreen")
print(g)
```

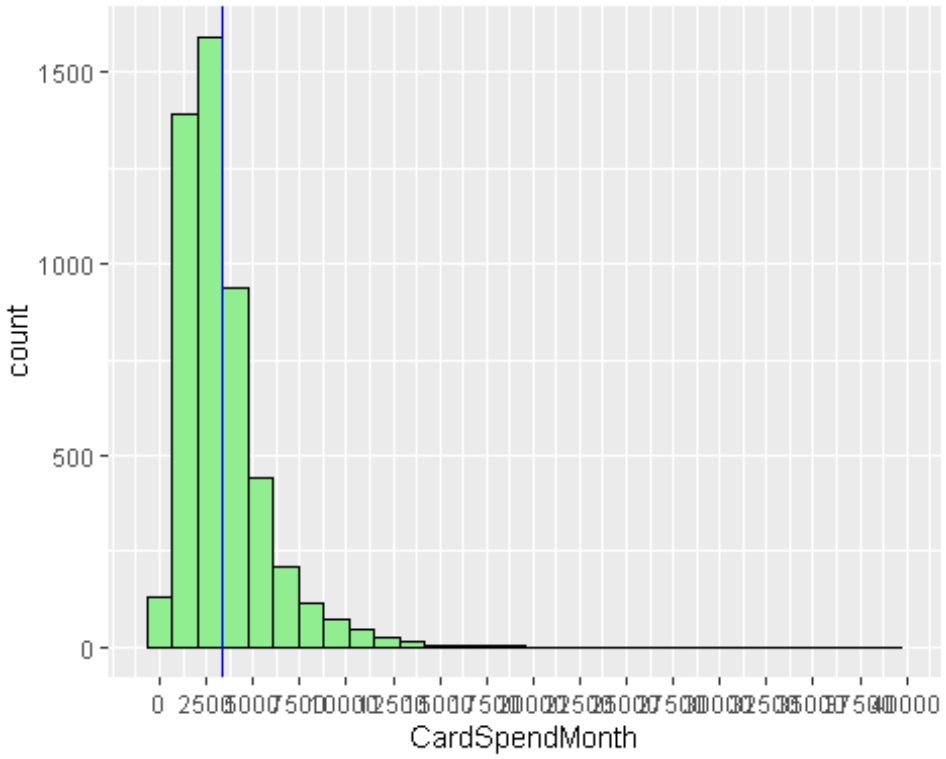



```
# missing_cols <- is.na(data$PhoneCoTenure) | is.na(data$CardSpendMont
h) | is.na(data$TotalLastMonth)
#
# data <- na.omit(data, subset = !missing_cols)


summary(data %>% select(PhoneCoTenure, TotalLastMonth, CardSpendMonth)
)

##  PhoneCoTenure  TotalLastMonth   CardSpendMonth
##  Min.   : 0.0   Min.   :  2.85   Min.   :   69.7
##  1st Qu.:18.0   1st Qu.: 26.70   1st Qu.: 1839.8
##  Median :38.0   Median : 49.73   Median : 2766.9
##  Mean   :38.2   Mean   : 64.11   Mean   : 3375.9
##  3rd Qu.:59.0   3rd Qu.: 87.30   3rd Qu.: 4185.4
##  Max.   :72.0   Max.   :590.40   Max.   :39264.1

######################## Rule Based Segmentation ########################
##########

h <- ggplot(data = data) +
  geom_histogram(aes(x=TotalLastMonth), color="black", fill="yellow")
+
  geom_vline(xintercept=mean(data$TotalLastMonth), color="blue") +
  annotate(geom="text", x=100, y=1200, label=as.character(mean(data$To
```

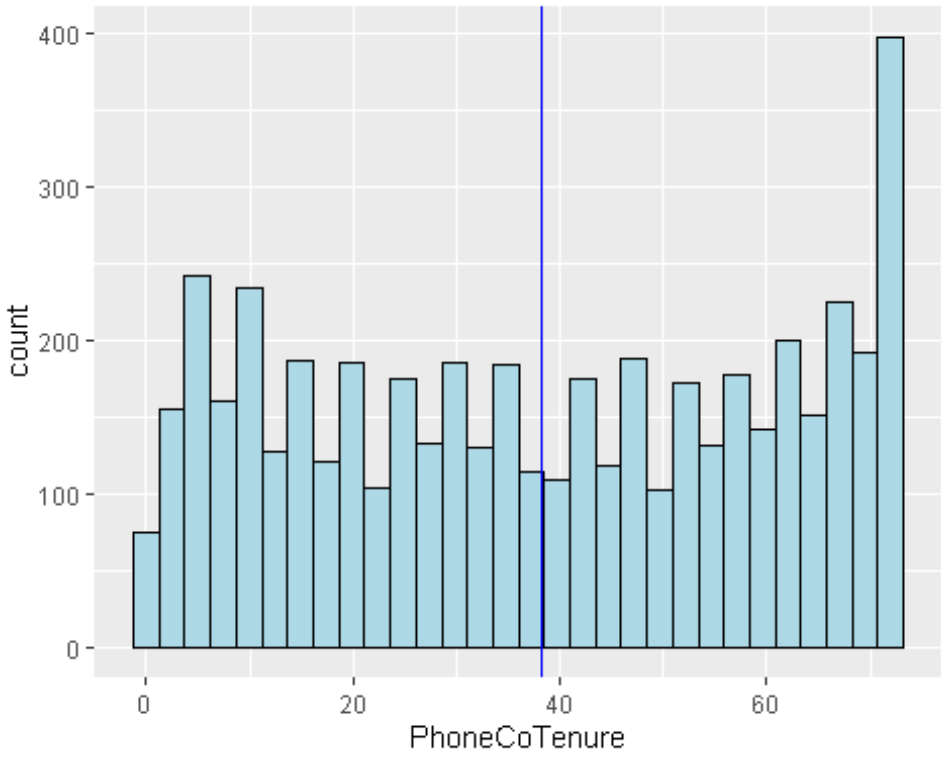
```
talLastMonth)))
print(h)

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




```
data$LastMonthGroup <- cut(data$TotalLastMonth, breaks=c(-1,70,600), l
abels=c("Low", "High"))

i <- ggplot(data = data) +
  geom_histogram(aes(x=CardSpendMonth), color="black", fill="lightgree
n") +
  geom_vline(xintercept=mean(data$CardSpendMonth, na.rm = T), color="b
lue") +
  scale_x_continuous(breaks = seq(0, 40000, by=2500))
print(i)

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

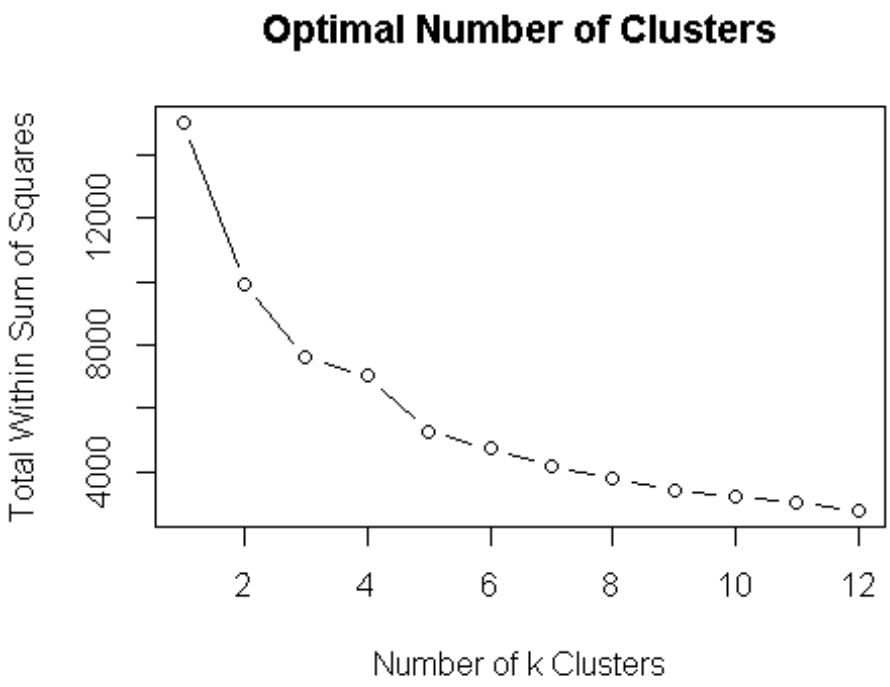



```
data$CardSpendGroup <- cut(data$CardSpendMonth, breaks=c(-1,5000,40000
), labels=c("Low", "High"))

j <- ggplot(data = data) +
  geom_histogram(aes(x=PhoneCoTenure), color="black", fill="lightblue"
) +
  geom_vline(xintercept=mean(data$PhoneCoTenure), na.rm = T, color="bl
ue")
print(j)

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




```
data$TenureGroup <- cut(data$PhoneCoTenure, breaks=c(-1,40,100), label
s=c("Short", "Long"))


data <- data %>% mutate(segment = case_when(
  LastMonthGroup == "Low" & CardSpendGroup == "Low" & TenureGroup == "
Short" ~ 1,
  LastMonthGroup == "Low" & CardSpendGroup == "Low" & TenureGroup == "
Long" ~ 2,
  LastMonthGroup == "Low" & CardSpendGroup == "High" & TenureGroup ==
"Short" ~ 3,
  LastMonthGroup == "Low" & CardSpendGroup == "High" & TenureGroup ==
"Long" ~ 4,
  LastMonthGroup == "High" & CardSpendGroup == "Low" & TenureGroup ==
"Short" ~ 5,
  LastMonthGroup == "High" & CardSpendGroup == "Low" & TenureGroup ==
"Long" ~ 6,
  LastMonthGroup == "High" & CardSpendGroup == "High" & TenureGroup ==
"Short" ~ 7,
  LastMonthGroup == "High" & CardSpendGroup == "High" & TenureGroup ==
"Long" ~ 8
))

data$CustomerValue <- ifelse(data$segment %in% c(4,6,7,8),"High","Low"
```

```
)
data$CustomerValue <- as.factor(data$CustomerValue)

###################### Supervised Learning Segmentation ##############
##########


log_reg <- glm(CustomerValue ~ TotalLastMonth + CardSpendMonth + Phone
CoTenure, data=data,
               family = "binomial")

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

data$probValue <- predict(log_reg, newdata = data, type="response")
data$ValueGroup <- cut(data$probValue, breaks=c(0.0,0.50,1.0),
                       labels=c("Low","High"))

###################### Unsupervised Learning Segmentation ############
##########

# Standardize numeric variables
k_points <- data %>% select(c("PhoneCoTenure", "TotalLastMonth", "Card
SpendMonth"))
k_points <- scale(k_points)



# Select best number of k-values
ks <- 1:12
tot_within_ss <- sapply(ks, function(k) {
  set.seed(1223)
  cl <- kmeans(k_points, k)
  cl$tot.withinss
})
plot(ks, tot_within_ss, type = "b", ylab = "Total Within Sum of Square
s",
     xlab = "Number of k Clusters", main = "Optimal Number of Clusters
")
```




```r
set.seed(1223)
NUM_CLUSTERS <- 3
kclust <- kmeans(k_points, centers = NUM_CLUSTERS, nstart=10)

#add segments to original dataset
data$kmeans_segment <- as.factor(kclust$cluster)
```