FMTV Insurance: Renewal Offer and Customer Lifetime Value Analysis

IST 707, Summer 2020, Sunday

Fu, Maloney, Tran, Ventosa

2020

Contents

[Introduction 3](#_Toc50623868)

[Analysis and Models 4](#_Toc50623869)

[Subsection 1: The Data 4](#_Toc50623870)

[Data Structure 4](#_Toc50623871)

[Data Cleaning 6](#_Toc50623872)

[Data Transformation 8](#_Toc50623873)

[Subsection 2: Exploratory Data Analysis 10](#_Toc50623874)

[Client demographic background by Policy Type 13](#_Toc50623875)

[Exploring Renewal Offers 20](#_Toc50623876)

[Key Factors to CLV 39](#_Toc50623877)

[Subsection 3: Models 61](#_Toc50623878)

[Association Rules Mining 62](#_Toc50623879)

[Decision Tree: Offer responses 68](#_Toc50623880)

[Naive Bayes 69](#_Toc50623881)

[Support Vector Machine 72](#_Toc50623882)

[Random Forest 73](#_Toc50623883)

[Cluster Analysis 83](#_Toc50623884)

[Decision Tree: CLV 95](#_Toc50623885)

[Results 99](#_Toc50623886)

[Conclusion 104](#_Toc50623887)

Table of Figures

[Figure 1: Histogram of the Income attribute 10](#_Toc50623888)

[Figure 2: Histogram of the Total.Claim.Amount attribute 10](#_Toc50623889)

[Figure 3: Histogram of the Monthly.Premium.Auto attribute 11](#_Toc50623890)

[Figure 4: Histogram of the Months.Since.Last.Claim attribute 11](#_Toc50623891)

[Figure 5: Histogram of the Months.Since.Policy.Inception attribute 12](#_Toc50623892)

[Figure 6: Histogram of the Customer.Lifetime.Value attribute 12](#_Toc50623893)

[Figure 7: Bar charts of Policy Type by Gender and by Marital Status 14](#_Toc50623894)

[Figure 8: Bar charts of Policy Type by State and Policy Type by Region 15](#_Toc50623895)

[Figure 9: Bar chart of Policy Type by Education Level 17](#_Toc50623896)

[Figure 10: Bar chart of Policy Type by Employment Status 18](#_Toc50623897)

[Figure 11: Bar chart of Policy Type by Income 19](#_Toc50623898)

[Figure 12: Bar chart of Renewal Offer Type by Policy Type 20](#_Toc50623899)

[Figure 13: Bar chart of distribution of Policy Types across each Offer subgroup 21](#_Toc50623900)

[Figure 14: Distributions of Offer Types by Gender and by Marital Status 23](#_Toc50623901)

[Figure 15: Distribution of Renewal Offer Types across States and Regions 25](#_Toc50623902)

[Figure 16: Distribution of Renewal Offer Types by Education Level 27](#_Toc50623903)

[Figure 17: Distribution of Renewal Offer Types by Employment Status 28](#_Toc50623904)

[Figure 18: Distribution of Renewal Offer Type by Income Bin 28](#_Toc50623905)

[Figure 19: Distribution of Responses by Renewal Offer Types 30](#_Toc50623906)

[Figure 20: Distribution of Responses across Genders and Marital Statuses 31](#_Toc50623907)

[Figure 21: Distribution of Responses across States and Regions 32](#_Toc50623908)

[Figure 22: Distribution of Responses by Education Level 34](#_Toc50623909)

[Figure 23: Distribution of Responses by Employment Status 34](#_Toc50623910)

[Figure 24: Distribution of Responses by Income Bin 35](#_Toc50623911)

[Figure 25: Distribution of Responses across CLV bins 36](#_Toc50623912)

[Figure 26: Distribution of Responses (to Offers 1 & 2) by Marital Status 37](#_Toc50623913)

[Figure 27: Distrbution of Responses (to Offers 1 & 2) by Employment Status 37](#_Toc50623914)

[Figure 28: Distribution of Responses (to Offers 1 & 2) by Region 38](#_Toc50623915)

[Figure 29: Scatterplot of Key Factors to CLV (Open Complaints, Months Since Last Claim, and Monthly Premium) 39](#_Toc50623916)

[Figure 30: Scatterplot of Key Factors to CLV (Total Claim Amount, Months Since Policy Inception, and Number of Policies) 40](#_Toc50623917)

[Figure 31: Correlation Plot of Key Factors to CLV 41](#_Toc50623918)

[Figure 32: Histogram of CLV 42](#_Toc50623919)

[Figure 33: Bar chart of total CLV by Policy Type and Policy (a subset of Policy Type) 43](#_Toc50623920)

[Figure 34: CLV by Vehicle Type and Vehicle Class 44](#_Toc50623921)

[Figure 35: CLV by State and Region 45](#_Toc50623922)

[Figure 36: CLV by Marital Status and Gender 46](#_Toc50623923)

[Figure 38: CLV by Renewal Offer and Response 47](#_Toc50623924)

[Figure 39: CLV by State 48](#_Toc50623925)

[Figure 40: CLV mean by State 49](#_Toc50623926)

[Figure 41: CLV by Education Level 50](#_Toc50623927)

[Figure 42: CLV by Education Level 51](#_Toc50623928)

[Figure 43: Mean CLV by Gender 52](#_Toc50623929)

[Figure 44: Mean CLV by Income bin 53](#_Toc50623930)

[Figure 45: Scatterplot of CLV and Monthly.Premium.Auto 54](#_Toc50623931)

[Figure 46: Correlation between CLV and Number of Open Complaints 55](#_Toc50623932)

[Figure 47: Scatterplot of CLV and Total.Claim.Amount 56](#_Toc50623933)

[Figure 48: Correlation between CLV and Coverage type 57](#_Toc50623934)

[Figure 49: Correlation between CLV and Employment Status 58](#_Toc50623935)

[Figure 50: Correlation between CLV and Sales Channel 59](#_Toc50623936)

[Figure 51: Correlation between CLV and Vehicle Class 60](#_Toc50623937)

[Figure 52: Correlation between CLV and Vehicle Size 61](#_Toc50623938)

[Figure 53: Offer 1 Apriori 64](#_Toc50623939)

[Figure 54: Offer 2 Apriori 66](#_Toc50623940)

[Figure 55: Offer 3 Apriori 67](#_Toc50623941)

[Figure 56: Random Forest - minimum nodes 81](#_Toc50623942)

[Figure 57: Random Forest - maximum nodes 81](#_Toc50623943)

[Figure 58: Bar chart of redistributed offers 82](#_Toc50623944)

[Figure 59: Bar chart of probabilities of "yes" response after redistribution 83](#_Toc50623945)

[Figure 60: Optimal clusters for k-means analysis 88](#_Toc50623946)

[Figure 61: K-means planar clusters 89](#_Toc50623947)

[Figure 62: Customer segments by Employment Status 90](#_Toc50623948)

[Figure 63: Customer segments by location 91](#_Toc50623949)

[Figure 64: Customer segments by marital status 92](#_Toc50623950)

[Figure 65: Customer segments by annual income 93](#_Toc50623951)

[Figure 66: Customer segments by Total Claim Amount 94](#_Toc50623952)

[Figure 67: Decision Tree CLV Model - Accuracy & Complexity 96](#_Toc50623953)

[Figure 68: CLV Decision Tree Visualization 99](#_Toc50623954)

# Introduction

Automotive insurance is mandated in most states to provide drivers and automobile owners financial protection in the event of an accident. Laws vary from state to state and insurance coverages are designed to meet state laws while also providing customized policies to meet individual needs. Insurance policies are generally issued for six-month to one-year timeframes. If they are fortunate, typical consumers interact with their automotive insurer only rarely. Perhaps they negotiate a better rate every few years, make a switch upon seeing a competitor’s advertising, or file the occasional claim after damages occur. When nearing a policy's expiration, insurance companies send renewal offers to retain customers and collect premiums. As a highly competitive industry customer churn rate averages at approximately 22% every year (https://bit.ly/3fgvVXv). Accordingly, customer retention is one of the most important factors for success among insurance companies. This fact, combined with the volume of customers that are insuring with the average provider, means that even modest increases to customer retention can result in massive improvements to the bottom line.

FMTV Auto Insurance maintains and updates its customer records to discover new and effective ways to retain its clientele, which includes both personal and enterprise customers. As a means of improving long-term value, the company offers insurance renewals to customers approaching the end of their current policies. Ostensibly, these renewal offers are customized to best meet their clients' varying needs. Most recently, as part of its marketing efforts to retain clients, FMTV has sent out one of four renewal offers to this subset of soon-to-expire customers. A purchase or non-purchase response to the offers have been recorded to determine the marketing return on investment. Recent purchase response rates to their renewal offers have been quite low. The marketing department questions whether a more effective method of distributing renewal offers may increase purchases and reduce customer churn.

Additionally, Customer Lifetime Value (CLV) is one of the key metrics FMTV uses to measure its success. CLV is a measure of a customer's total value over the entire customer-business relationship (CLV = Customer revenue per year x Duration of the relationship in years – Total costs of acquiring and serving the customer). By developing strategies to reduce client churn, policy term businesses can increase the average CLV of its clients. This is, in part, because higher retention of existing customers costs less than acquiring new ones. In addition to its utility as a means of assessment, FMTV’s client CLV information can be used to develop predictive models to identify high CLV customers to acquire or maintain. Together, improvements to offer renewal notices and CLV identification could allow FMTV to increase the size of its client base and the profitability of its company.

# Analysis and Models

## Subsection 1: The Data

Loading Data

auto <-read.csv("https://raw.githubusercontent.com/jcventosa/IST-707/master/WA\_Fn-UseC\_-Marketing-Customer-Value-Analysis.csv")

### Data Structure

str(auto)

## 'data.frame': 9134 obs. of 24 variables:  
## $ Customer : Factor w/ 9134 levels "AA10041","AA11235",..: 601 5947 97 8017 2489 4948 8434 756 1352 548 ...  
## $ State : Factor w/ 5 levels "Arizona","California",..: 5 1 3 2 5 4 4 1 4 4 ...  
## $ Customer.Lifetime.Value : num 2764 6980 12887 7646 2814 ...  
## $ Response : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 2 1 2 1 ...  
## $ Coverage : Factor w/ 3 levels "Basic","Extended",..: 1 2 3 1 1 1 1 3 1 2 ...  
## $ Education : Factor w/ 5 levels "Bachelor","College",..: 1 1 1 1 1 1 2 5 1 2 ...  
## $ Effective.To.Date : Factor w/ 59 levels "1/1/11","1/10/11",..: 48 25 42 13 53 18 48 10 19 40 ...  
## $ EmploymentStatus : Factor w/ 5 levels "Disabled","Employed",..: 2 5 2 5 2 2 2 5 3 2 ...  
## $ Gender : Factor w/ 2 levels "F","M": 1 1 1 2 2 1 1 2 2 1 ...  
## $ Income : int 56274 0 48767 0 43836 62902 55350 0 14072 28812 ...  
## $ Location.Code : Factor w/ 3 levels "Rural","Suburban",..: 2 2 2 2 1 1 2 3 2 3 ...  
## $ Marital.Status : Factor w/ 3 levels "Divorced","Married",..: 2 3 2 2 3 2 2 3 1 2 ...  
## $ Monthly.Premium.Auto : int 69 94 108 106 73 69 67 101 71 93 ...  
## $ Months.Since.Last.Claim : int 32 13 18 18 12 14 0 0 13 17 ...  
## $ Months.Since.Policy.Inception: int 5 42 38 65 44 94 13 68 3 7 ...  
## $ Number.of.Open.Complaints : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Number.of.Policies : int 1 8 2 7 1 2 9 4 2 8 ...  
## $ Policy.Type : Factor w/ 3 levels "Corporate Auto",..: 1 2 2 1 2 2 1 1 1 3 ...  
## $ Policy : Factor w/ 9 levels "Corporate L1",..: 3 6 6 2 4 6 3 3 3 8 ...  
## $ Renew.Offer.Type : Factor w/ 4 levels "Offer1","Offer2",..: 1 3 1 1 1 2 1 1 1 2 ...  
## $ Sales.Channel : Factor w/ 4 levels "Agent","Branch",..: 1 1 1 3 1 4 1 1 1 2 ...  
## $ Total.Claim.Amount : num 385 1131 566 530 138 ...  
## $ Vehicle.Class : Factor w/ 6 levels "Four-Door Car",..: 6 1 6 5 1 6 1 1 1 1 ...  
## $ Vehicle.Size : Factor w/ 3 levels "Large","Medsize",..: 2 2 2 2 2 2 2 2 2 2 ...

In order to improve in the areas of customer retention and CLV identification, FMTV has provided a dataset containing demographic and policy information for customers who have recently received renewal offers.

The data has 9134 records with the following 24 variables:

|  |  |
| --- | --- |
| Variables | Description |
| Customer: | Customer ID number |
| State: | State of residence or business |
| Customer Lifetime Value: | Customer's total worth to business over life of the relationship |
| Response: | Yes or No response to a renewal offer |
| Coverage: | Type of policy (Basic, Extended, Premium) |
| Education: | Level of education (High School or less, College, BA, MA , PHD) |
| Effective To Date: | Date the policy expires |
| Employment Status: | (Employed, Unemployed, Retired, Disabled, Medical Leave) |
| Gender: | Male or Female |
| Income: | Customers' annual income |
| Location Code: | (Rural, Suburban, Urban) |
| Marital Status: | (Single, Married, Divorced) |
| Monthly Premium Auto | Amount of customers' monthly insurance payments |
| Months Since Last Claim | Number of months between customers' last reported insurance claim |
| Months Since Policy Inception | Number of months since customer began an insurance policy |
| Number of Open Complaints | Number of unresolved customer complaints |
| Number of Policies | Number of policies customer currently owns |
| Policy Type | (Corporate Auto, Personal Auto, Special Auto) |
| Policy | 3 levels (L1, L2, L3) per Policy Type (Corporate, Personal, Special) |
| Renew Offer Type | 4 types of renewal offers (Offer 1, Offer 2, Offer 3, Offer 4) |
| Sales Channel | Channels to purchase a policy (Agent, Branch, Call Center, Web) |
| Total Claim Amount | Cumulative amount of claims since policy inception |
| Vehicle Class | Type of vehicle (4-Door, Luxury, Luxury SUV, Sports Car, SUV, 2-Door) |
| Vehicle Size | (Large, Medium, Small) |

### Data Cleaning

Check for NA, missing and duplicated data

#install.packages("rapportools")  
library(rapportools)  
sum(is.na(auto))

## [1] 0

nrow(auto[duplicated(auto),])

## [1] 0

No missing or duplicated data.

Check data summary for unusual distributions or incorrect data types

summary(auto)

## Customer State Customer.Lifetime.Value Response   
## AA10041: 1 Arizona :1703 Min. : 1898 No :7826   
## AA11235: 1 California:3150 1st Qu.: 3994 Yes:1308   
## AA16582: 1 Nevada : 882 Median : 5780   
## AA30683: 1 Oregon :2601 Mean : 8005   
## AA34092: 1 Washington: 798 3rd Qu.: 8962   
## AA35519: 1 Max. :83325   
## (Other):9128   
## Coverage Education Effective.To.Date  
## Basic :5568 Bachelor :2748 1/10/11: 195   
## Extended:2742 College :2681 1/27/11: 194   
## Premium : 824 Doctor : 342 2/14/11: 186   
## High School or Below:2622 1/26/11: 181   
## Master : 741 1/17/11: 180   
## 1/19/11: 179   
## (Other):8019   
## EmploymentStatus Gender Income Location.Code Marital.Status  
## Disabled : 405 F:4658 Min. : 0 Rural :1773 Divorced:1369   
## Employed :5698 M:4476 1st Qu.: 0 Suburban:5779 Married :5298   
## Medical Leave: 432 Median :33890 Urban :1582 Single :2467   
## Retired : 282 Mean :37657   
## Unemployed :2317 3rd Qu.:62320   
## Max. :99981   
##   
## Monthly.Premium.Auto Months.Since.Last.Claim Months.Since.Policy.Inception  
## Min. : 61.00 Min. : 0.0 Min. : 0.00   
## 1st Qu.: 68.00 1st Qu.: 6.0 1st Qu.:24.00   
## Median : 83.00 Median :14.0 Median :48.00   
## Mean : 93.22 Mean :15.1 Mean :48.06   
## 3rd Qu.:109.00 3rd Qu.:23.0 3rd Qu.:71.00   
## Max. :298.00 Max. :35.0 Max. :99.00   
##   
## Number.of.Open.Complaints Number.of.Policies Policy.Type   
## Min. :0.0000 Min. :1.000 Corporate Auto:1968   
## 1st Qu.:0.0000 1st Qu.:1.000 Personal Auto :6788   
## Median :0.0000 Median :2.000 Special Auto : 378   
## Mean :0.3844 Mean :2.966   
## 3rd Qu.:0.0000 3rd Qu.:4.000   
## Max. :5.0000 Max. :9.000   
##   
## Policy Renew.Offer.Type Sales.Channel Total.Claim.Amount  
## Personal L3 :3426 Offer1:3752 Agent :3477 Min. : 0.099   
## Personal L2 :2122 Offer2:2926 Branch :2567 1st Qu.: 272.258   
## Personal L1 :1240 Offer3:1432 Call Center:1765 Median : 383.945   
## Corporate L3:1014 Offer4:1024 Web :1325 Mean : 434.089   
## Corporate L2: 595 3rd Qu.: 547.515   
## Corporate L1: 359 Max. :2893.240   
## (Other) : 378   
## Vehicle.Class Vehicle.Size   
## Four-Door Car:4621 Large : 946   
## Luxury Car : 163 Medsize:6424   
## Luxury SUV : 184 Small :1764   
## Sports Car : 484   
## SUV :1796   
## Two-Door Car :1886   
##

Initial inspection of the dataset summary provides the proportional split for some key nominal variables as follows:

**Response:** 0.14 Yes 0.86 No

**Gender:** 0.51 Female 0.49 Male

**Marital Status:** 0.58 Married 0.15 Divorced 0.27 Single

**Location Code:** 0.19 Rural 0.63 Suburban 0.17 Urban

**Coverage:** 0.61 Basic 0.30 Extended 0.09 Premium

**Policy Type:** 0.74 Personal 0.04 Special 0.22 Corporate

**Vehicle Size:** 0.10 Large 0.70 Medsize 0.19 Small

### Data Transformation

Converting the Effective.To.Date variable’s data type from factor to date.

auto$Effective.To.Date <- as.Date(auto$Effective.To.Date, format = "%m/%d/%y")

In addition the following discretized variables will be added to the dataset.

CLV\_Bins Discretized Customer.Lifetime.Value

IncomeBin Discretized Income

MoPremiumBin Discretized Monthly.Premium

ClaimBin Discretized Total.Claim

# Adding columns for CLV, Income, Monthly Premium and Total Claim bins.  
  
# Cutomer Lifetime Value bins  
auto$CLV\_Bins <- cut(auto$Customer.Lifetime.Value,   
 breaks = c(0,1999,3999,5999,7999,9999,Inf),  
 labels = c("< $2000", "$2000-$3999","$4000-$5999",  
 "$6000-$7999", "$8000-$9999", "$10000+"))  
# Income bins  
auto$IncomeBin <- cut(auto$Income, breaks = c(0,14999,29999,44999,59999,74999,Inf),  
 labels = c("< $15000", "$15000-$29999","$30000-$44999",  
 "$45000-$59999", "$60000-$74999", "$75000+"))  
# Monthly Premium bins  
auto$MoPremiumBin <- cut(auto$Monthly.Premium.Auto,   
 breaks = c(0,74,99,124,149,Inf),  
 labels = c("< $75", "$75-$99","$100-$124","$125-$149","$150+"))  
  
# Total Claim bins  
auto$ClaimBin <- cut(auto$Total.Claim.Amount,   
 breaks = c(0,249,499,749,999,Inf),  
 labels = c("< $250", "$250-$499","$500-$749","$750-$999","$1000+"))

## Subsection 2: Exploratory Data Analysis

hist(auto$Income)

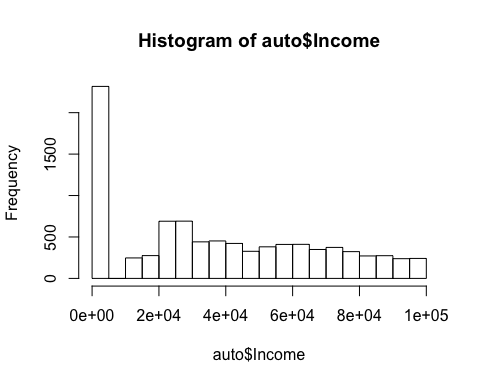


Figure 1: Histogram of the Income attribute

hist(auto$Total.Claim.Amount)

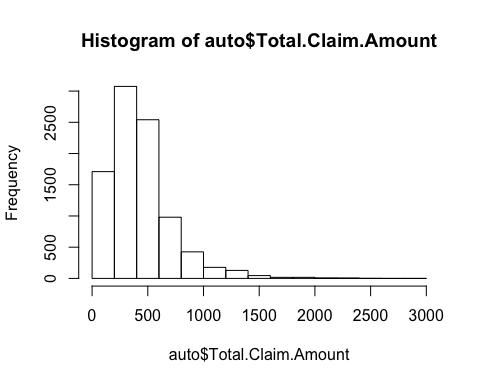


Figure 2: Histogram of the Total.Claim.Amount attribute

hist(auto$Monthly.Premium.Auto)

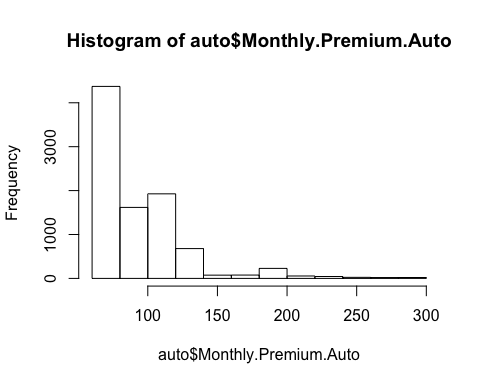


Figure 3: Histogram of the Monthly.Premium.Auto attribute

hist(auto$Months.Since.Last.Claim)

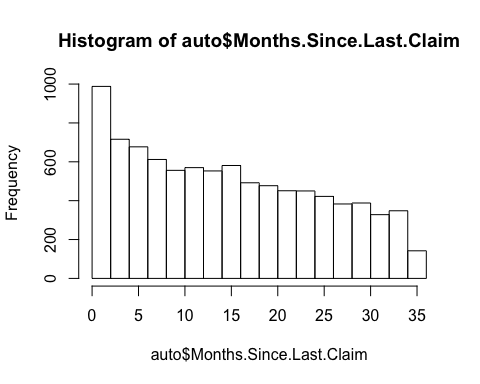


Figure 4: Histogram of the Months.Since.Last.Claim attribute

hist(auto$Months.Since.Policy.Inception)

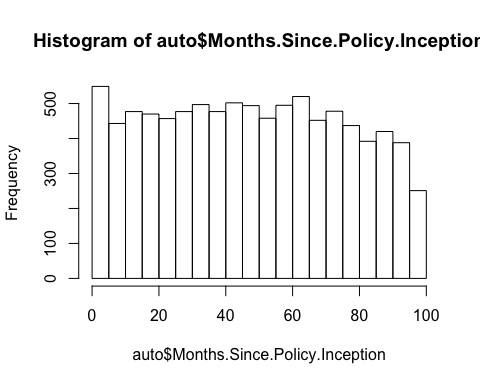


Figure 5: Histogram of the Months.Since.Policy.Inception attribute

hist(auto$Customer.Lifetime.Value)

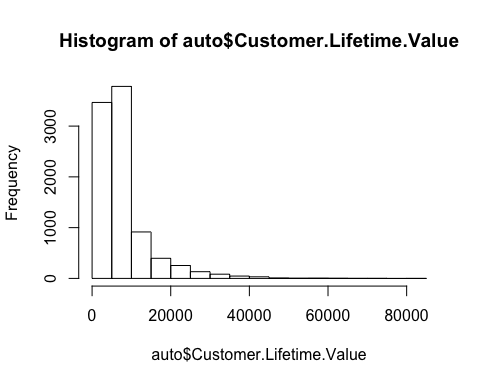


Figure 6: Histogram of the Customer.Lifetime.Value attribute

### Client demographic background by Policy Type

#install.packages("ggplot2", "RColorBrewer", "gridExtra"")  
library(ggplot2)  
library(RColorBrewer)  
library(gridExtra)  
  
PTGender <- ggplot(auto, aes(x= Policy.Type, fill=Gender)) + geom\_bar(stat = "count") +   
 xlab("Policy Type") + ylab("Policy Count") + ggtitle("Policy Type by Gender") +  
 scale\_fill\_brewer (palette = "Set1", labels = c("Female", "Male"))  
  
PTMarital <- ggplot(auto, aes(x=Policy.Type, fill = Marital.Status)) + geom\_bar(stat = "count") +  
 xlab("Policy Type") + ylab("Policy Count") + ggtitle("Policy Type by Marital Status") +  
 scale\_fill\_brewer (palette = "Dark2", name = "Marital Status")   
  
table(auto$Policy.Type, auto$Gender)

##   
## F M  
## Corporate Auto 1005 963  
## Personal Auto 3454 3334  
## Special Auto 199 179

table(auto$Policy.Type, auto$Marital.Status)

##   
## Divorced Married Single  
## Corporate Auto 311 1127 530  
## Personal Auto 1013 3947 1828  
## Special Auto 45 224 109

grid.arrange(PTGender, PTMarital)

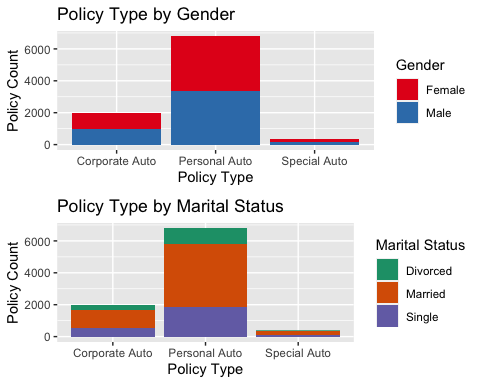


Figure 7: Bar charts of Policy Type by Gender and by Marital Status

There is an approximate 50/50 split by Gender for both Corporate and Personal policy types with women just slightly outnumbering men by a percentage point or less. Women make up 55% of the Special auto policy types. The distribution of policy types by marital status is within four percentage points of the overall distribution of 15% Divorced, 58% Married and 27% Single.

PTState <- ggplot(auto, aes(x=Policy.Type, fill = State)) + geom\_bar(stat = "count") +  
 xlab("Policy Type") + ylab("Policy Count") + ggtitle("Policy Type by State") +  
 scale\_fill\_brewer (palette = "Accent", name = "State")   
  
PTLocation <- ggplot(auto, aes(x=Policy.Type, fill = Location.Code)) + geom\_bar(stat = "count") +  
 xlab("Policy Type") + ylab("Policy Count") + ggtitle("Policy Type by Region") +  
 scale\_fill\_brewer (palette = "Dark2", name = "Region")   
  
table(auto$Policy.Type, auto$State)

##   
## Arizona California Nevada Oregon Washington  
## Corporate Auto 333 739 195 540 161  
## Personal Auto 1293 2298 652 1938 607  
## Special Auto 77 113 35 123 30

table(auto$Policy.Type, auto$Location.Code)

##   
## Rural Suburban Urban  
## Corporate Auto 400 1239 329  
## Personal Auto 1301 4308 1179  
## Special Auto 72 232 74

grid.arrange(PTState, PTLocation)

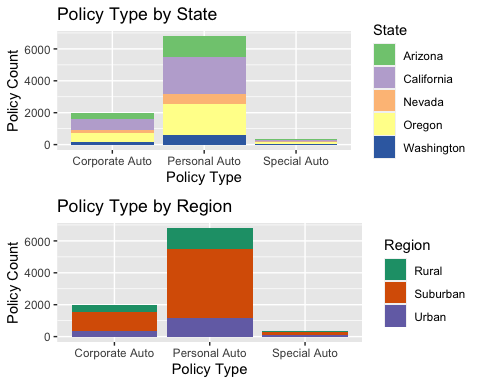


Figure 8: Bar charts of Policy Type by State and Policy Type by Region

The majority of all three policy types are in California followed by Oregon, Arizona, Nevada and Washington. Suburban regions have over three times the number of Corporate and Personal policy types compared to Rural and Urban regions. Sixty-0ne percent of Special policy types are also located in Suburban regions with Rural and Urban each making up approximately 14%-15%.

auto$Education <- factor(auto$Education, levels = c("High School or Below", "College",   
 "Bachelor", "Master", "Doctor"))  
auto$EmploymentStatus <- factor(auto$EmploymentStatus, levels = c("Employed", "Unemployed",  
 "Retired", "Disabled", "Medical Leave"))  
  
PTEmployment <- ggplot(auto, aes(x=Policy.Type, fill = EmploymentStatus)) + geom\_bar(stat = "count") +  
 xlab("Policy Type") + ylab("Policy Count") + ggtitle("Policy Type by Employment Status") +  
 scale\_fill\_brewer (palette = "Set1", name = "Employment Status")  
  
PTIncomeBin <- ggplot(auto, aes(x=Policy.Type, fill = IncomeBin)) + geom\_bar(stat = "count") +  
 xlab("Policy Type") + ylab("Policy Count") + ggtitle("Policy Type by Income") +  
 scale\_fill\_brewer (palette = "Accent", name = "Income")   
  
PTEducation <- ggplot(auto, aes(x=Policy.Type, fill = Education)) + geom\_bar(stat = "count") +  
 xlab("Policy Type") + ylab("Policy Count") + ggtitle("Policy Type by Education Level") +  
 scale\_fill\_brewer (palette = "Accent", name = "Education")   
  
table(auto$Policy.Type, auto$Education)

##   
## High School or Below College Bachelor Master Doctor  
## Corporate Auto 576 572 616 142 62  
## Personal Auto 1934 1992 2024 571 267  
## Special Auto 112 117 108 28 13

table(auto$Policy.Type, auto$EmploymentStatus)

##   
## Employed Unemployed Retired Disabled Medical Leave  
## Corporate Auto 1238 504 54 82 90  
## Personal Auto 4223 1720 217 305 323  
## Special Auto 237 93 11 18 19

table(auto$Policy.Type, auto$IncomeBin)

##   
## < $15000 $15000-$29999 $30000-$44999 $45000-$59999  
## Corporate Auto 553 341 281 266  
## Personal Auto 1909 1251 972 817  
## Special Auto 102 64 62 35  
##   
## $60000-$74999 $75000+  
## Corporate Auto 234 293  
## Personal Auto 849 990  
## Special Auto 51 64

PTEducation

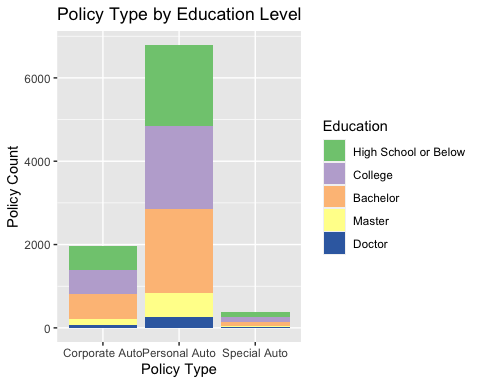


Figure 9: Bar chart of Policy Type by Education Level

PTEmployment

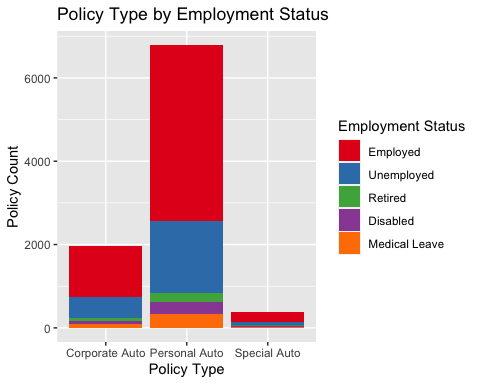


Figure 10: Bar chart of Policy Type by Employment Status

PTIncomeBin

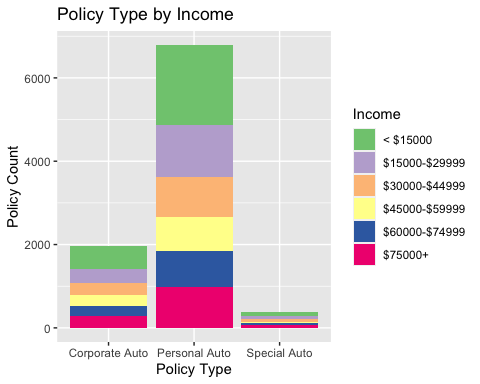


Figure 11: Bar chart of Policy Type by Income

There is nearly an equal distribution of clients with High School or Less, College and Bachelors where each education level makes up approximately 30% of each policy type. Employed and Unemployed are likewise equally distributed across policy types, making up approximately 62% and 25% of each policy type, respectively. Lastly 44%-45% of all three policy types have incomes less than $30,000.

### Exploring Renewal Offers

table(auto$Renew.Offer.Type, auto$Policy.Type)

##   
## Corporate Auto Personal Auto Special Auto  
## Offer1 827 2776 149  
## Offer2 597 2201 128  
## Offer3 308 1067 57  
## Offer4 236 744 44

# MyTheme to format font sizes for my graphs  
MyTheme <- theme(axis.text = element\_text(size = 14),   
 axis.title = element\_text(size = 16, face ="bold"),   
 plot.title = element\_text(size = 20, face = "bold"))  
MyTheme2 <- theme(axis.text = element\_text(size = 10),   
 axis.title = element\_text(size = 18, face ="bold"),   
 plot.title = element\_text(size = 20, face = "bold"),   
 axis.text.x = element\_text(angle = 90))  
  
ggplot(auto, aes(x=Renew.Offer.Type, fill = Policy.Type)) + geom\_bar(stat="count")+  
 xlab("Renewal Offer Type") + ylab("Policy Count") + ggtitle("Renewal Offer Type by Policy Type") +  
 scale\_fill\_brewer(palette = "YlOrBr") + MyTheme

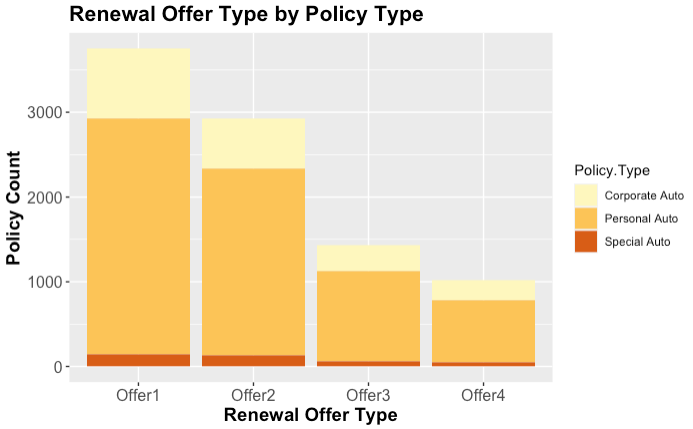


Figure 12: Bar chart of Renewal Offer Type by Policy Type

ggplot(auto, aes(x=Renew.Offer.Type, fill = Policy.Type)) + geom\_bar(stat="count", position = "fill")+  
 xlab("Renewal Offer Type") + ylab("") + ggtitle("Percent of Renewal Offer Types by Policy Type") +  
 scale\_fill\_brewer(palette = "YlOrBr") + MyTheme

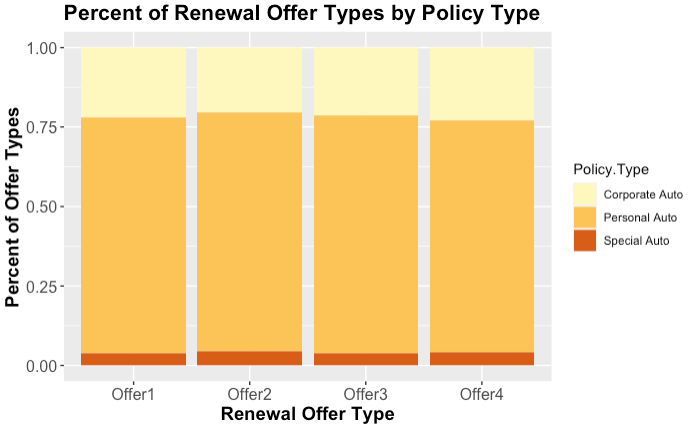


Figure 13: Bar chart of distribution of Policy Types across each Offer subgroup

Each client from the three policy types received a single renewal offer type of 1, 2, 3 or 4. Offer 1 had the highest distribution, having been received by 41% of the clients. Offer 2 was received by 32%, Offer 3 by 16%, and Offer 4 by 11% of the clients. Interestingly, the offer types were distributed at nearly equal percentages by each policy type.

ROGender <- ggplot(auto, aes(x= Renew.Offer.Type, fill=Gender)) +   
 geom\_bar(stat = "count", position = "fill") +   
 xlab("Renew Offer Type") + ylab("") +   
 ggtitle("Percent of Renew Offer Type by Gender") +  
 scale\_fill\_brewer (palette = "Set1", labels = c("Female", "Male"))  
  
ROMarital <- ggplot(auto, aes(x=Renew.Offer.Type, fill = Marital.Status)) +   
 geom\_bar(stat = "count", position = "fill") +  
 xlab("Renew Offer Type") + ylab("") +   
 ggtitle("Percent of Renew Offer Type by Marital Status") +  
 scale\_fill\_brewer (palette = "Dark2", name = "Marital Status")   
  
table(auto$Renew.Offer.Type, auto$Gender)

##   
## F M  
## Offer1 2003 1749  
## Offer2 1464 1462  
## Offer3 726 706  
## Offer4 465 559

table(auto$Renew.Offer.Type, auto$Marital.Status)

##   
## Divorced Married Single  
## Offer1 705 1794 1253  
## Offer2 362 1920 644  
## Offer3 194 872 366  
## Offer4 108 712 204

grid.arrange(ROGender, ROMarital)

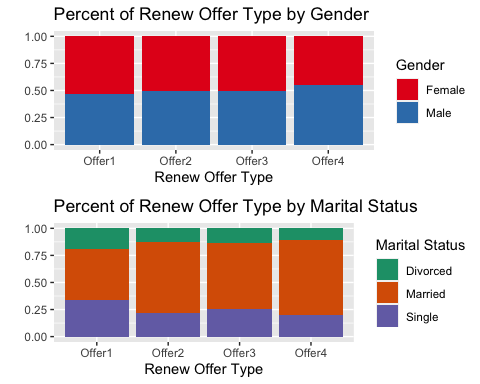


Figure 14: Distributions of Offer Types by Gender and by Marital Status

The offers were also distributed at nearly equal proportions by Gender and Marital Status.

ROState <- ggplot(auto, aes(x=Renew.Offer.Type, fill = State)) +   
 geom\_bar(stat = "count", position = "fill") +  
 xlab("Renew Offer Type") + ylab("") +   
 ggtitle("Percent of Renew Offer Types by State") +  
 scale\_fill\_brewer (palette = "Accent", name = "State")   
  
ROLocation <- ggplot(auto, aes(x=Renew.Offer.Type, fill = Location.Code)) +   
 geom\_bar(stat = "count", position = "fill") +  
 xlab("Renew Offer Type") + ylab("") +   
 ggtitle("Percent of Renew Offer Types by Region") +  
 scale\_fill\_brewer (palette = "Dark2", name = "Region")   
  
table(auto$Renew.Offer.Type, auto$State)

##   
## Arizona California Nevada Oregon Washington  
## Offer1 696 1276 366 1075 339  
## Offer2 551 1010 283 834 248  
## Offer3 265 512 130 402 123  
## Offer4 191 352 103 290 88

table(auto$Renew.Offer.Type, auto$Location.Code)

##   
## Rural Suburban Urban  
## Offer1 625 2557 570  
## Offer2 598 1765 563  
## Offer3 301 851 280  
## Offer4 249 606 169

grid.arrange(ROState, ROLocation)

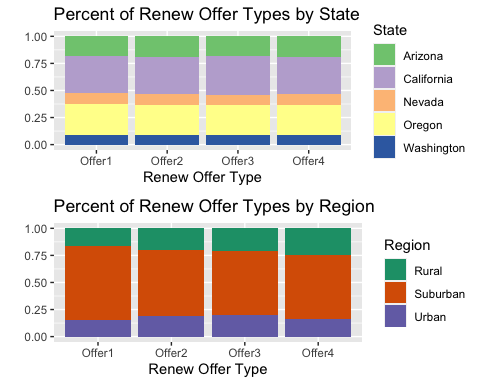


Figure 15: Distribution of Renewal Offer Types across States and Regions

The Renewal Offers were also distributed at nearly equal proportions by State and Region.

Mytheme <- theme(axis.text = element\_text(size = 12),   
 axis.title = element\_text(size = 14, face ="bold"),   
 plot.title = element\_text(size = 16, face = "bold"))  
  
ROEmployment <- ggplot(auto, aes(x=Renew.Offer.Type, fill = EmploymentStatus)) +   
 geom\_bar(stat = "count", position = "fill") +  
 xlab("Renew Offer Type") + ylab("") +   
 ggtitle("Percent of Renew Offer Type by Employment Status") +  
 scale\_fill\_brewer (palette = "YlOrBr",name = "Employment Status") + Mytheme  
  
ROIncomeBin <- ggplot(auto, aes(x=Renew.Offer.Type, fill = IncomeBin)) +   
 geom\_bar(stat = "count", position = "fill") +  
 xlab("Renew Offer Type") + ylab("") +   
 ggtitle("Percent of Renew Offer Type by Income") +  
 scale\_fill\_brewer (palette = "OrRd", name = "Income") + Mytheme  
  
ROEducation <- ggplot(auto, aes(x=Renew.Offer.Type, fill = Education)) +   
 geom\_bar(stat = "count", position = "fill") +  
 xlab("Renew Offer Type") + ylab("") +   
 ggtitle("Percent of Renew Offer Type by Education Level") +  
 scale\_fill\_brewer (palette = "Oranges", name = "Education") + Mytheme  
  
table(auto$Renew.Offer.Type, auto$Education)

##   
## High School or Below College Bachelor Master Doctor  
## Offer1 985 1136 1249 281 101  
## Offer2 893 817 840 238 138  
## Offer3 400 437 413 125 57  
## Offer4 344 291 246 97 46

table(auto$Renew.Offer.Type, auto$EmploymentStatus)

##   
## Employed Unemployed Retired Disabled Medical Leave  
## Offer1 1896 1229 170 225 232  
## Offer2 2107 541 77 89 112  
## Offer3 943 357 23 57 52  
## Offer4 752 190 12 34 36

table(auto$Renew.Offer.Type, auto$IncomeBin)

##   
## < $15000 $15000-$29999 $30000-$44999 $45000-$59999 $60000-$74999  
## Offer1 1376 657 591 492 279  
## Offer2 603 531 387 333 495  
## Offer3 379 264 206 167 199  
## Offer4 206 204 131 126 161  
##   
## $75000+  
## Offer1 357  
## Offer2 577  
## Offer3 217  
## Offer4 196

ROEducation

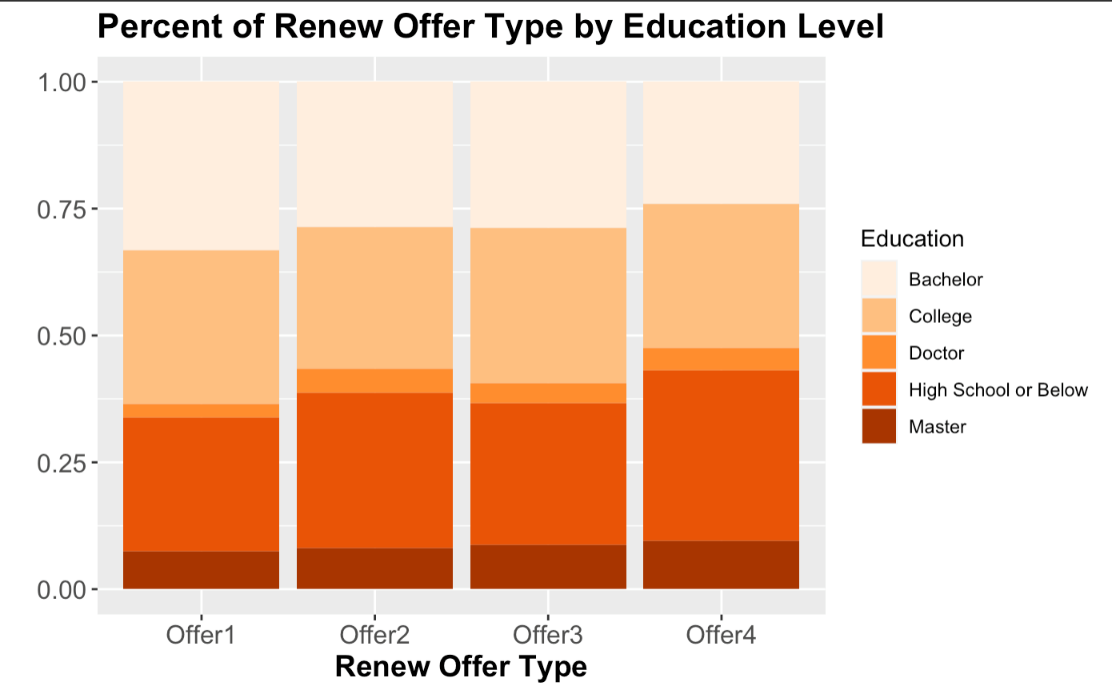


Figure 16: Distribution of Renewal Offer Types by Education Level

ROEmployment

A screenshot of a cell phone

Description automatically generated

Figure 17: Distribution of Renewal Offer Types by Employment Status

ROIncomeBin

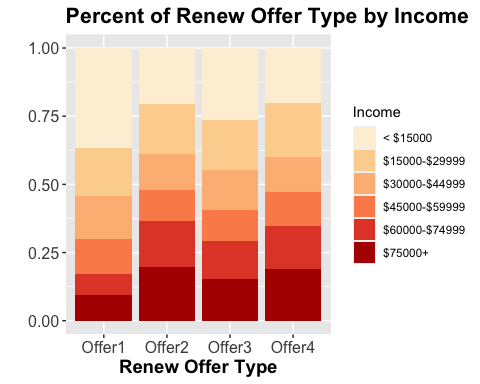


Figure 18: Distribution of Renewal Offer Type by Income Bin

There are minor differences in how the offers were distributed by education level, with Offer 1 more heavily distributed among College and Bachelor than the other offers. Offer 1 was also received by a lower proportion of employed clients compared to the other offers. This difference was also evident in the income levels, as nearly 70% of Offer 1 recipients had incomes below $45,000. Of the recipients of Offers 2, 3, and 4, the count of clients with incomes below $45,000 fell narrowly between 51% and 59% for all three offer groups.

library(ggplot2)  
library(RColorBrewer)  
ggplot(auto, aes(x=Renew.Offer.Type, fill = Response)) + geom\_bar(stat = "count", position = "fill") +  
 xlab("Renew Offer Type") + ylab("") + ggtitle("Percent of Renew Offer Type by Response") +  
 scale\_fill\_manual(values = c("darkorange1", "darkorange4")) + theme(axis.text = element\_text(size = 12),   
 axis.title = element\_text(size = 14, face ="bold"),   
 plot.title = element\_text(size = 18, face = "bold"))

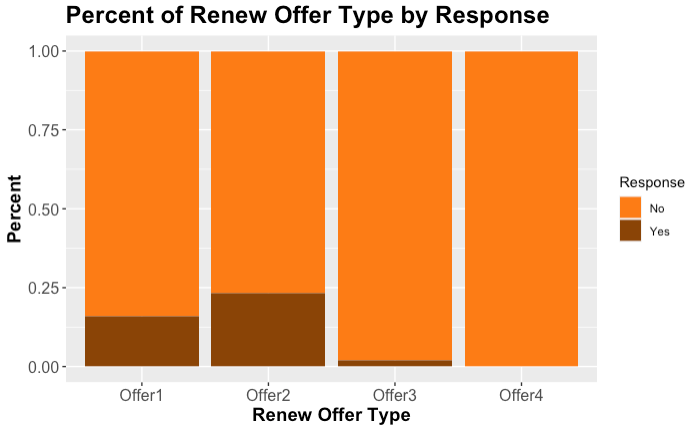


Figure 19: Distribution of Responses by Renewal Offer Types

The purchase response rates for the renew offers were very low. Offer 2 had the best purchase response rate at 23.4% followed by Offer 1 with a purchase response rate of 15.8%. Only 2% of Offer 3 recipients made a renewal while Offer 4 was declined by all client recipients.

RPSGender <- ggplot(auto, aes(x= Gender, fill=Response)) +   
 geom\_bar(stat = "count", position = "fill") +   
 xlab("Gender") + ylab("") +   
 ggtitle("Percent of Gender by Response") +  
 scale\_fill\_brewer (palette = "Set1")  
  
RPSMarital <- ggplot(auto, aes(x=Marital.Status, fill = Response)) +   
 geom\_bar(stat = "count", position = "fill") +  
 xlab("Marital.Status") + ylab("") +   
 ggtitle("Percent of Marital Status by Response") +  
 scale\_fill\_brewer (palette = "Dark2")   
  
table(auto$Response, auto$Gender)

##   
## F M  
## No 3998 3828  
## Yes 660 648

table(auto$Response, auto$Marital.Status)

##   
## Divorced Married Single  
## No 1045 4602 2179  
## Yes 324 696 288

grid.arrange(RPSGender, RPSMarital)

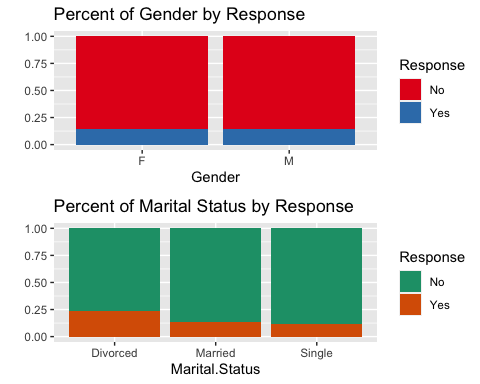


Figure 20: Distribution of Responses across Genders and Marital Statuses

Men and women were equally likely to respond with a “Yes.” Divorced clients were slightly more likely to say “Yes” to an offer.

RPSState <- ggplot(auto, aes(x=State, fill = Response)) +   
 geom\_bar(stat = "count", position = "fill") +  
 xlab("State") + ylab("") +   
 ggtitle("Percent of State Clients by Response") +  
 scale\_fill\_brewer (palette = "Accent")   
RPSLocation <- ggplot(auto, aes(x=Location.Code, fill = Response)) +   
 geom\_bar(stat = "count", position = "fill") +  
 xlab("Region") + ylab("") +   
 ggtitle("Percent of Region by Response") +  
 scale\_fill\_brewer (palette = "Dark2")   
table(auto$Response, auto$State)

##   
## Arizona California Nevada Oregon Washington  
## No 1460 2694 758 2225 689  
## Yes 243 456 124 376 109

table(auto$Response, auto$Location.Code)

##   
## Rural Suburban Urban  
## No 1611 4771 1444  
## Yes 162 1008 138

grid.arrange(RPSState, RPSLocation)

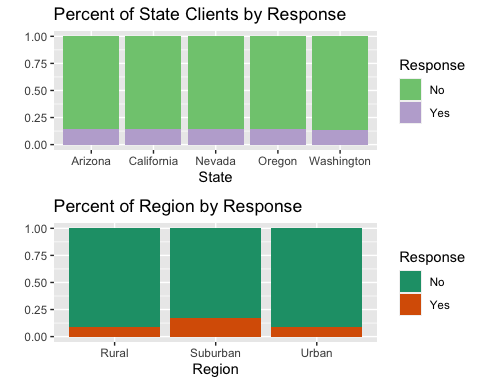


Figure 21: Distribution of Responses across States and Regions

The purchase response rates were more or less equal across the states. However, clients from Suburban areas did have a higher purchase response rate than Rural or Urban clients.

RPSEmployment <- ggplot(auto, aes(x=EmploymentStatus, fill = Response)) +   
 geom\_bar(stat = "count", position = "fill") +  
 xlab("Employment Status") + ylab("") +   
 ggtitle("Response Rate by Employment Status") +  
 scale\_fill\_manual (values = c("darkorange1", "darkorange4")) + Mytheme  
  
RPSIncomeBin <- ggplot(auto, aes(x=IncomeBin, fill = Response)) +   
 geom\_bar(stat = "count", position = "fill") +  
 xlab("Annual Income") + ylab("") +   
 ggtitle("Response Rate by Income") +  
 scale\_fill\_brewer (palette = "Accent") + MyTheme2  
  
RPSEducation <- ggplot(auto, aes(x=Education, fill = Response)) +   
 geom\_bar(stat = "count", position = "fill") +  
 xlab("Education Level") + ylab("") +   
 ggtitle("Response Rate by Education Level") +  
 scale\_fill\_brewer (palette = "YlOrBr", name = "Education") + MyTheme2  
  
table(auto$Response, auto$Education)

##   
## High School or Below College Bachelor Master Doctor  
## No 2280 2273 2370 621 282  
## Yes 342 408 378 120 60

table(auto$Response, auto$EmploymentStatus)

##   
## Employed Unemployed Retired Disabled Medical Leave  
## No 4942 2119 78 333 354  
## Yes 756 198 204 72 78

table(auto$Response, auto$IncomeBin)

##   
## < $15000 $15000-$29999 $30000-$44999 $45000-$59999 $60000-$74999 $75000+  
## No 2270 1308 1147 968 954 1179  
## Yes 294 348 168 150 180 168

RPSEducation

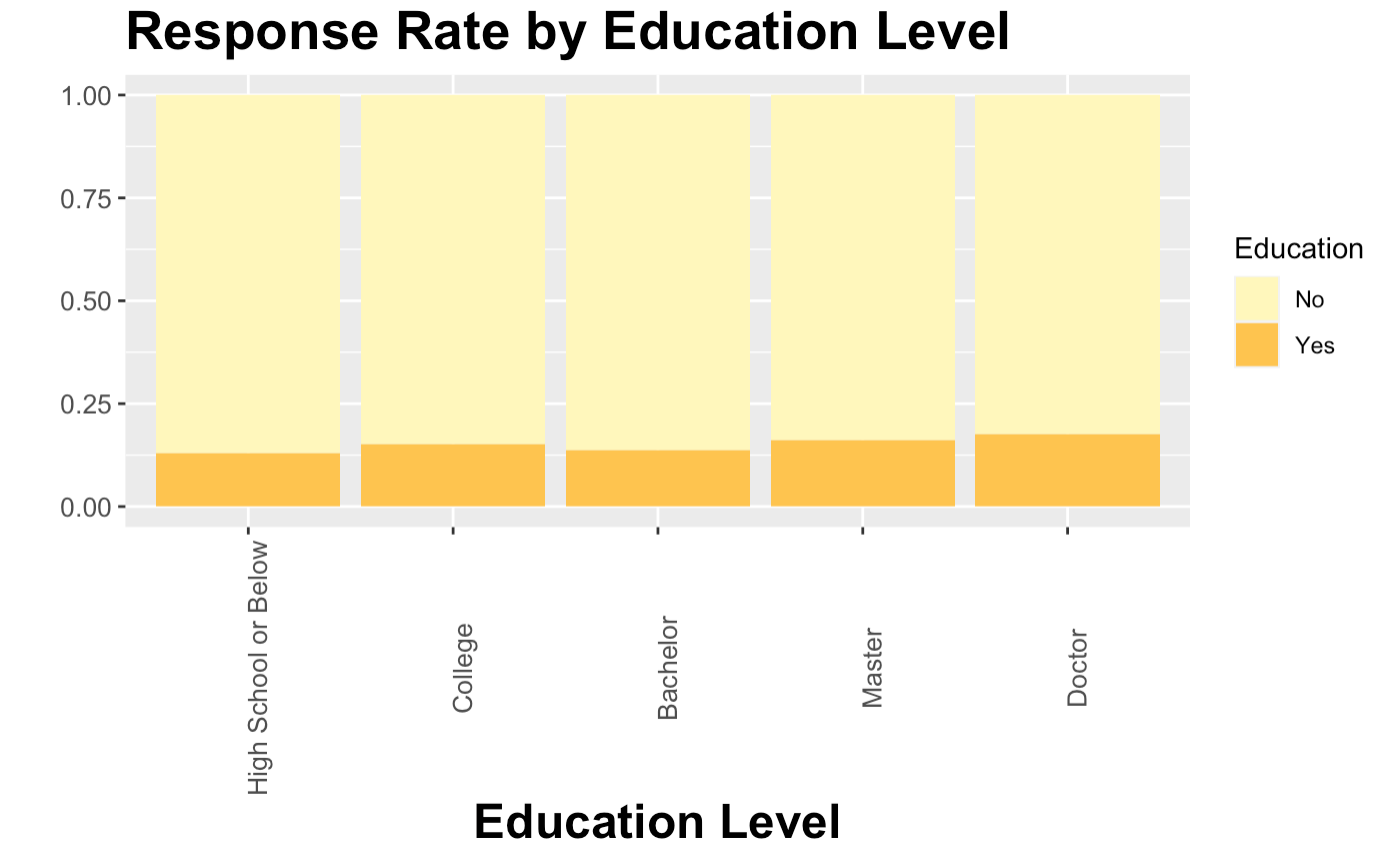


Figure 22: Distribution of Responses by Education Level

RPSEmployment

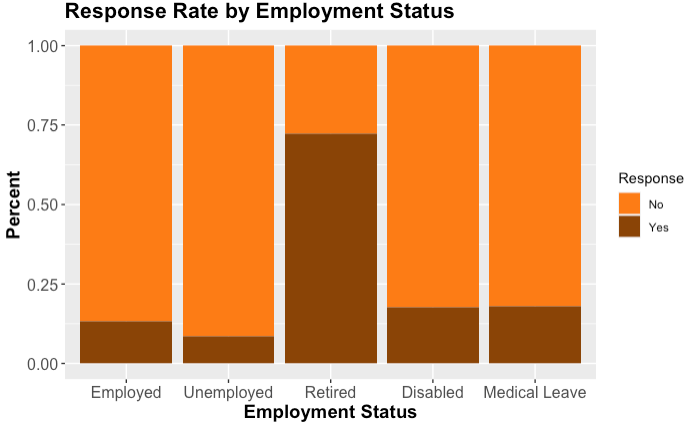


Figure 23: Distribution of Responses by Employment Status

RPSIncomeBin



Figure 24: Distribution of Responses by Income Bin

Clients with post graduate degrees were slightly more likely to accept the renewal offers they received. Curiously, nearly 75% or retired clients responded “Yes” to their renewal offers. Clients earning between 15,000-29,999 and 60,000-74,999 answered “Yes” to renewal offers more often than clients of other income ranges.

Mytheme2 <- theme(axis.text = element\_text(size = 10),   
 axis.title = element\_text(size = 14, face ="bold"),   
 plot.title = element\_text(size = 16, face = "bold"),   
 axis.text.x = element\_text(angle = 90))  
  
ggplot(auto, aes(x=CLV\_Bins, fill = Response)) +   
 geom\_bar(stat = "count", position = "fill") +  
 xlab("Customer Lifetime Value") + ylab("Percent") +   
 ggtitle("Response Rate by Customer Lifetime Value") +  
 scale\_fill\_brewer (palette = "Dark2", name = "Response") + Mytheme2

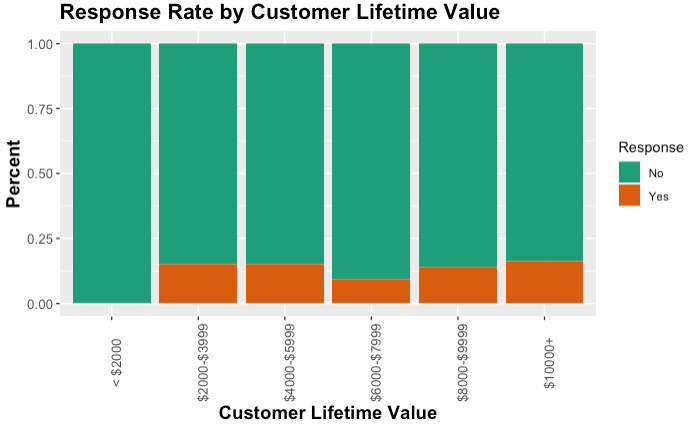


Figure 25: Distribution of Responses across CLV bins

Offer1 <- auto[auto$Renew.Offer.Type=="Offer1",]  
Offer2 <- auto[auto$Renew.Offer.Type=="Offer2",]  
Offer12 <-data.frame(rbind(Offer1,Offer2))  
  
ggplot(Offer12, aes(x=Marital.Status, fill = Response)) +   
 geom\_bar(stat = "count", position = "fill") +  
 xlab("Marital Status") + ylab("Percent") +   
 ggtitle("Response Rate to Offers 1 & 2 by Marital Status") +  
 scale\_fill\_manual (values = c("darkorange","darkorange4"), name = "Response") + Mytheme

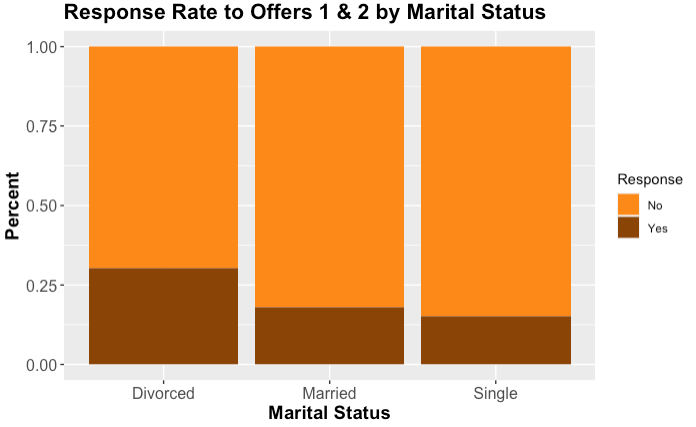


Figure 26: Distribution of Responses (to Offers 1 & 2) by Marital Status

ggplot(Offer12, aes(x=EmploymentStatus, fill = Response)) +   
 geom\_bar(stat = "count", position = "fill") +  
 xlab("Employment Status") + ylab("Percent") +   
 ggtitle("Response Rate to Offers 1 & 2 by Employment Status") +  
 scale\_fill\_brewer (palette = "Dark2", name = "Response") + Mytheme

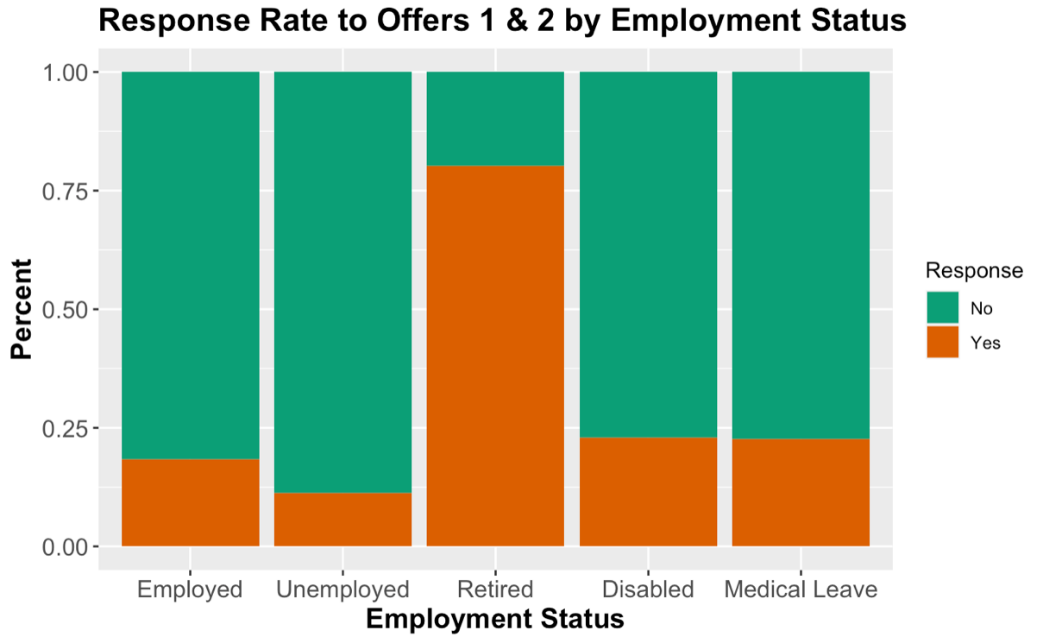


Figure 27: Distrbution of Responses (to Offers 1 & 2) by Employment Status

ggplot(Offer12, aes(x=Location.Code, fill = Response)) +   
 geom\_bar(stat = "count", position = "fill") +  
 xlab("Region") + ylab("Percent") +   
 ggtitle("Response Rate to Offers 1 & 2 by Region") +  
 scale\_fill\_manual (values = c("tan1","tan4"), name = "Response") + Mytheme



Figure 28: Distribution of Responses (to Offers 1 & 2) by Region

### Key Factors to CLV

ggplot(auto, aes(x=Monthly.Premium.Auto, y=Customer.Lifetime.Value)) +   
 geom\_point(aes(col = as.factor(Number.of.Open.Complaints), size = Months.Since.Last.Claim)) +  
 scale\_color\_brewer(palette="Spectral", name = "Number of Open Complaints") + xlab("Monthly Premium") +   
 ylab("Customer Lifetime Value") + ggtitle("Key Factors to Customer Lifetime Value") +  
 labs(size ="Months Since Last Claim") + theme(axis.text = element\_text(size = 10),   
 axis.title = element\_text(size = 12, face ="bold"),   
 plot.title = element\_text(size = 16, face = "bold"))

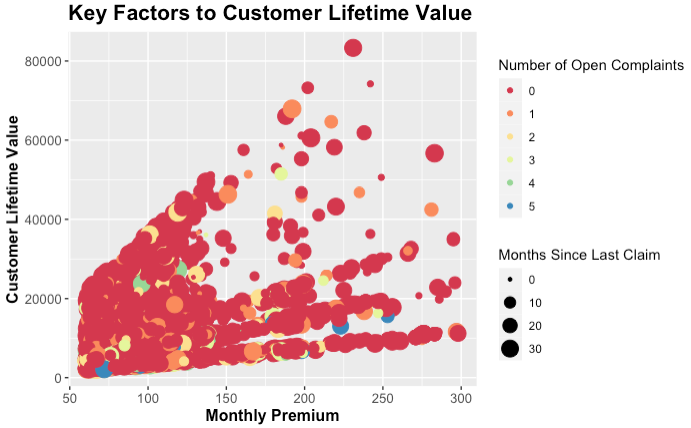


Figure 29: Scatterplot of Key Factors to CLV (Open Complaints, Months Since Last Claim, and Monthly Premium)

ggplot(auto, aes(x=Total.Claim.Amount, y=Customer.Lifetime.Value)) +   
 geom\_point(aes(col = as.factor(Number.of.Policies), size = Months.Since.Policy.Inception)) +  
 scale\_color\_brewer(palette="Spectral", name = "Number of Policies") + xlab("Total Claim Amount") +   
 ylab("Customer Lifetime Value") + ggtitle("Key Factors to Customer Lifetime Value") +  
 labs(size ="Months Since Policy Inception") + theme(axis.text = element\_text(size = 10),   
 axis.title = element\_text(size = 12, face ="bold"),   
 plot.title = element\_text(size = 16, face = "bold"))

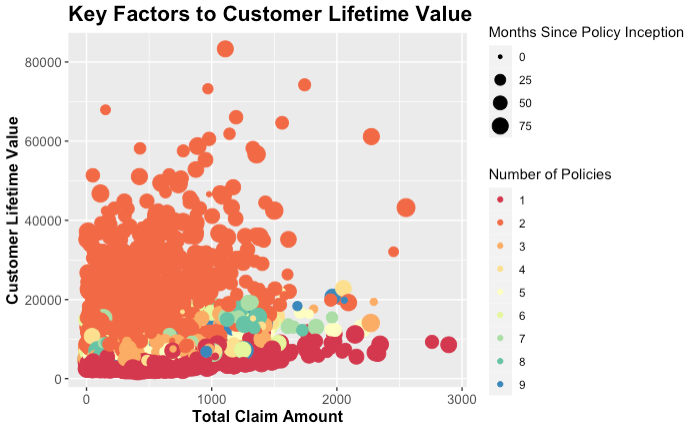


Figure 30: Scatterplot of Key Factors to CLV (Total Claim Amount, Months Since Policy Inception, and Number of Policies)

autoCorr <- auto[,c(3,10,13:17,22)]  
library(corrplot)

## corrplot 0.84 loaded

library(corrgram)

## Registered S3 method overwritten by 'seriation':  
## method from   
## reorder.hclust gclus

##   
## Attaching package: 'corrgram'

## The following object is masked \_by\_ '.GlobalEnv':  
##   
## auto

colnames(autoCorr) <- c("CLV", "Income", "Mon. Premium", "Mon Since Claim", "Mons Since Inception",  
 "Open Complaints", "Num of Poicies", "Total Claim Amt.")  
autoCorr <- cor(autoCorr)  
corrplot(autoCorr, method = "number")

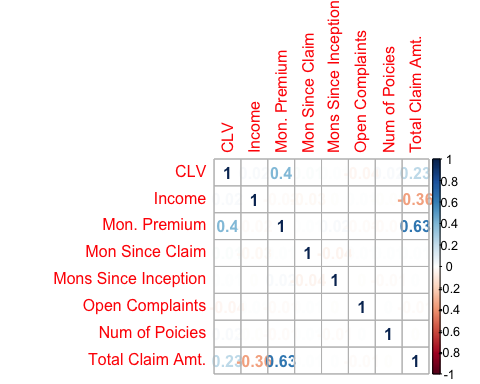


Figure 31: Correlation Plot of Key Factors to CLV

hist(auto$Customer.Lifetime.Value)

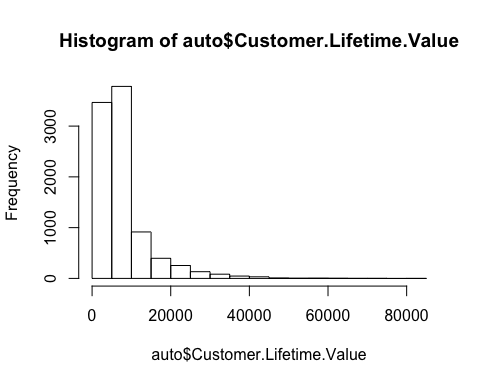


Figure 32: Histogram of CLV

summary(auto$Customer.Lifetime.Value)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1898 3994 5780 8005 8962 83325

sd(auto$Customer.Lifetime.Value)

## [1] 6870.968

Customer Lifetime Values (CLV) at FMTV Insurance ranges from $1,898 to $83,325 with a median of $5,780 and a standard deviation of $6,871. The mean CLV ($8,005) is greater than the median ($5,780), indicating a right skewed distribution with a decreasing number of customers as CLV increases.

library(RColorBrewer)  
library(ggplot2)  
CLV\_PolicyType <- ggplot(auto, aes(x=Policy.Type, y=Customer.Lifetime.Value, fill = Policy ))+  
 geom\_col() + xlab("Policy Type") + ylab("Customer Lifetime Value") +  
 ggtitle("Customer Lifetime Value by Policy Type and Policy") +  
 scale\_fill\_brewer(palette = "Set1")  
CLV\_PolicyType



Figure 33: Bar chart of total CLV by Policy Type and Policy (a subset of Policy Type)

Customers who purchase Personal Auto insurance provide the lion’s share of customer lifetime value to FMTV insurance.

#install.packages("RColorBrewer")  
library(RColorBrewer)  
CLV\_Vehicle <- ggplot(auto, aes(x=Vehicle.Size, y=Customer.Lifetime.Value,   
 fill = Vehicle.Class ))+geom\_col() +   
 xlab("Vehicle Type") + ylab("Customer Lifetime Value") +   
 ggtitle("Customer Lifetime Value by Vehicle Type and Vehicle Class") +  
 scale\_fill\_brewer(palette = "Dark2")  
   
CLV\_Vehicle

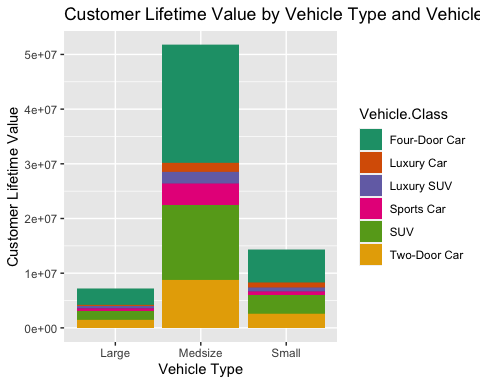


Figure 34: CLV by Vehicle Type and Vehicle Class

Customers with Midsize vehicles offer the highest customer lifetime values with Four-Door and SUV models.

#install.packages("RColorBrewer")  
library(RColorBrewer)  
CLV\_Region <- ggplot(auto, aes(x=State, y=Customer.Lifetime.Value,   
 fill = Location.Code ))+geom\_col() +   
 xlab("State") + ylab("Customer Lifetime Value") +  
 ggtitle("Customer Lifetime Value by State and Region") +  
 scale\_fill\_brewer(palette = "Set1")  
   
CLV\_Region

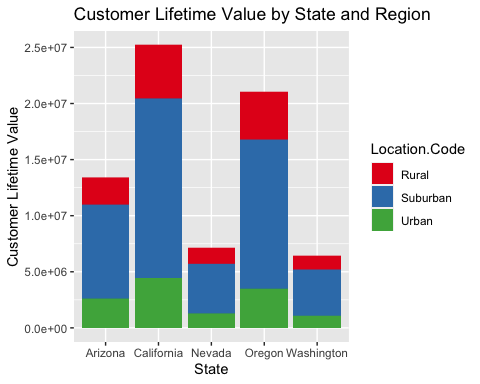


Figure 35: CLV by State and Region

California and Oregon are the states with the highest Customer Lifetime Value (CLV) customers. Suburban areas have higher CLV customers than other location codes.

#install.packages("RColorBrewer")  
library(RColorBrewer)  
CLV\_Marital <- ggplot(auto, aes(x=Marital.Status, y=Customer.Lifetime.Value,   
 fill = Gender ))+geom\_col() +   
 xlab("Marital") + ylab("Customer Lifetime Value") +  
 ggtitle("Customer Lifetime Value by Marital Status and Gender") +  
 scale\_fill\_brewer(palette = "Set1")  
   
CLV\_Marital

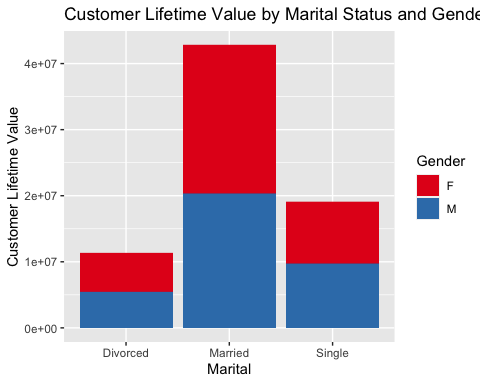


Figure 36: CLV by Marital Status and Gender

library(RColorBrewer)  
library(ggplot2)  
ggplot(auto, aes(x=Renew.Offer.Type, y=Customer.Lifetime.Value, fill = Response)) + geom\_col() +  
 xlab("Renewal Offer Type") + ylab("Customer Lifetime Value") +  
 ggtitle("Customer Lifetime Value by Renewal Offer and Response") +  
 scale\_fill\_brewer(palette = "Set1")

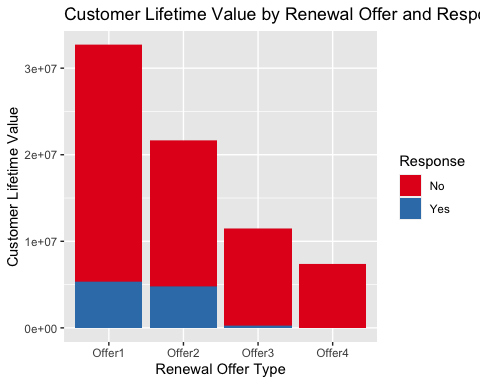


Figure 38: CLV by Renewal Offer and Response

**CLV and State Distribution**

CLV.statedf <- data.frame(auto$State, auto$Customer.Lifetime.Value)  
CLV.state <- tapply(CLV.statedf$auto.Customer.Lifetime.Value, CLV.statedf$auto.State, sum)  
CLV.state

## Arizona California Nevada Oregon Washington   
## 13387865 25211490 7106015 21010621 6401135

plot.CLV.state <- barplot(CLV.state, main = "CLV by State")

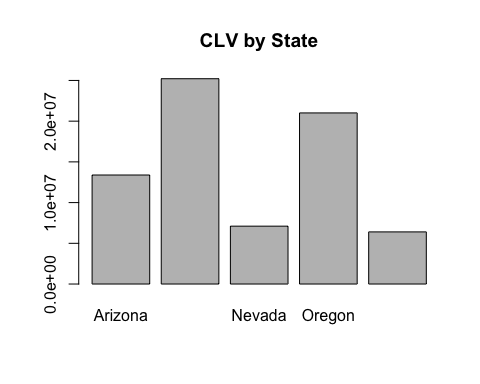


Figure 39: CLV by State

plot.CLV.state

## [,1]  
## [1,] 0.7  
## [2,] 1.9  
## [3,] 3.1  
## [4,] 4.3  
## [5,] 5.5

**CLV mean by state**

CLV.stateMean <- tapply(CLV.statedf$auto.Customer.Lifetime.Value, CLV.statedf$auto.State, mean)  
CLV.stateMean

## Arizona California Nevada Oregon Washington   
## 7861.341 8003.648 8056.707 8077.901 8021.472

plot.CLV.stateMean <- barplot(CLV.stateMean, main = "CLV Mean By State")

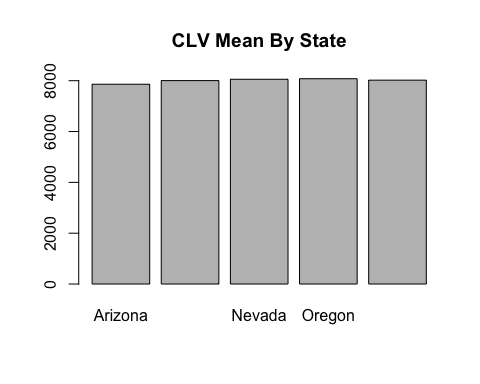


Figure 40: CLV mean by State

plot.CLV.stateMean

## [,1]  
## [1,] 0.7  
## [2,] 1.9  
## [3,] 3.1  
## [4,] 4.3  
## [5,] 5.5

**CLV and Education Distribution**

CLV.educationdf <- data.frame(auto$Customer.Lifetime.Value, auto$Education)  
CLV.education <- tapply(CLV.educationdf$auto.Customer.Lifetime.Value, CLV.educationdf$auto.Education, sum)  
CLV.education

## High School or Below College Bachelor   
## 21753971 21048704 21634071   
## Master Doctor   
## 6108422 2571958

plot.CLV.education <- barplot(CLV.education, main = "CLV by Education Level",   
 cex.names = 0.8) +  
 Mytheme2

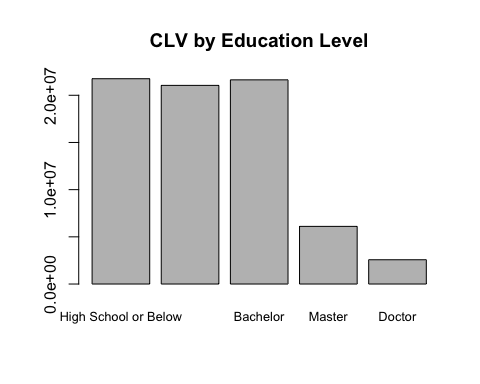


Figure 41: CLV by Education Level

plot.CLV.education

## NULL

**CLV mean by education**

CLV.educationMean <- tapply(CLV.educationdf$auto.Customer.Lifetime.Value, CLV.educationdf$auto.Education, mean)  
CLV.educationMean

## High School or Below College Bachelor   
## 8296.709 7851.065 7872.660   
## Master Doctor   
## 8243.485 7520.345

plot.CLV.educationMean <- barplot(CLV.educationMean, main = "CLV vs. Education Distribution")

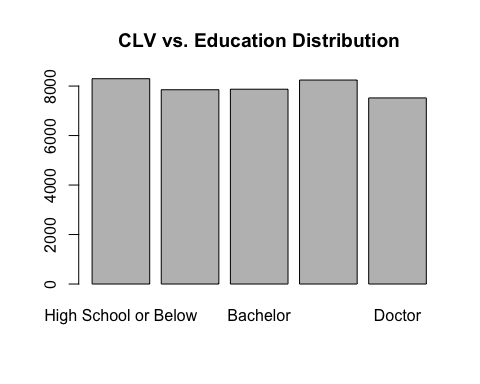


Figure 42: CLV by Education Level

plot.CLV.educationMean

## [,1]  
## [1,] 0.7  
## [2,] 1.9  
## [3,] 3.1  
## [4,] 4.3  
## [5,] 5.5

**CLV and Gender**

CLV.genderdf <- data.frame(auto$Customer.Lifetime.Value, auto$Gender)  
CLV.gender <- tapply(CLV.genderdf$auto.Customer.Lifetime.Value, CLV.genderdf$auto.Gender, mean)  
CLV.gender

## F M   
## 8096.602 7909.551

plot.CLV.gender <- barplot(CLV.gender, main = "Mean CLV by Gender")

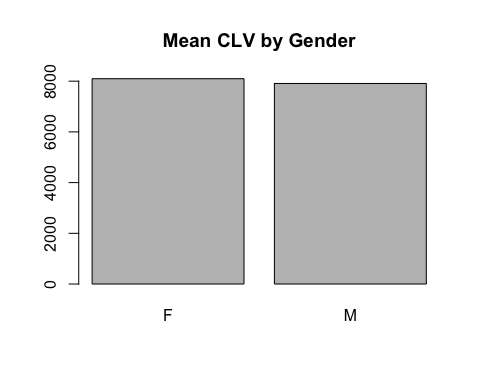


Figure 43: Mean CLV by Gender

plot.CLV.gender

## [,1]  
## [1,] 0.7  
## [2,] 1.9

**CLV and Incomebin**

CLV.IBdf <- data.frame(auto$Customer.Lifetime.Value, auto$IncomeBin)  
CLV.IB <- tapply(CLV.IBdf$auto.Customer.Lifetime.Value, CLV.IBdf$auto.IncomeBin, mean)  
CLV.IB

## < $15000 $15000-$29999 $30000-$44999 $45000-$59999 $60000-$74999   
## 7671.001 7878.946 8254.606 8369.973 8297.476   
## $75000+   
## 8002.502

plot.CLV.IB <- barplot(CLV.IB, main = "Mean CLV by Income", las = 2, cex.names = 0.65)

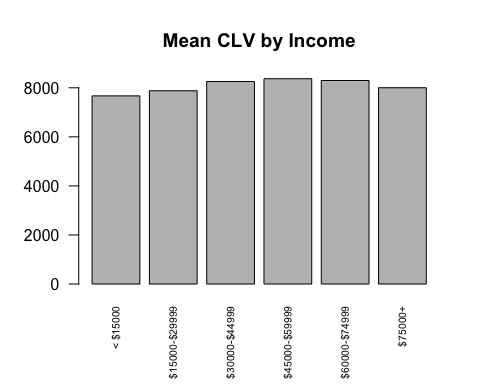


Figure 44: Mean CLV by Income bin

plot.CLV.IB

## [,1]  
## [1,] 0.7  
## [2,] 1.9  
## [3,] 3.1  
## [4,] 4.3  
## [5,] 5.5  
## [6,] 6.7

Discover the correlations between variables

premium.CLV <- ggplot(data=auto, aes(x=Customer.Lifetime.Value, y=Monthly.Premium.Auto)) +geom\_point()  
premium.CLV <- premium.CLV + ggtitle("Potential Correlation Between CLV and Monthly Premium Auto")  
premium.CLV

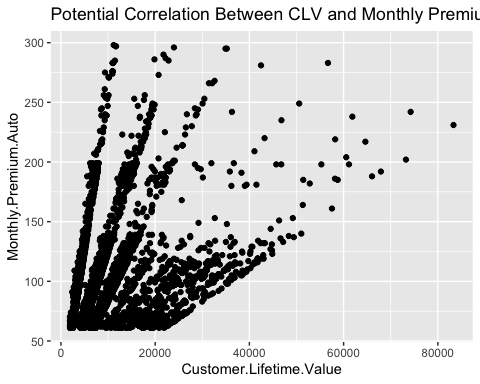


Figure 45: Scatterplot of CLV and Monthly.Premium.Auto

Here there is a positive correlation between the CLV and client’s monthly premium.

**Potential Correlation Between CLV and Number Of Open Complaints**

complaints.CLV <- ggplot(data=auto, aes(x=Customer.Lifetime.Value, y=Number.of.Open.Complaints)) +geom\_point()  
complaints.CLV <- complaints.CLV + ggtitle("Potential Correlation Between CLV and Number Of Open Complaints")  
complaints.CLV

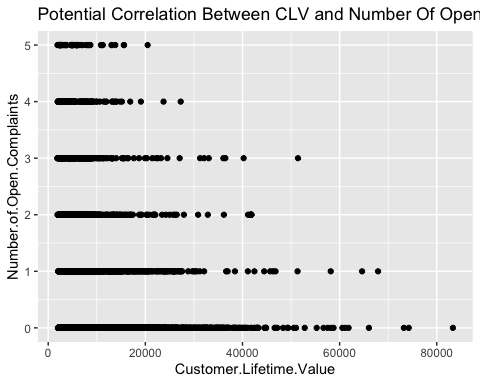


Figure 46: Correlation between CLV and Number of Open Complaints

The data indicate that a lower number of open complaints tends to correspond with a higher CLV.

**Total complaints**

totalClaim.CLV <- ggplot(data=auto, aes(x=Customer.Lifetime.Value, y=Total.Claim.Amount)) +geom\_point()  
totalClaim.CLV <- totalClaim.CLV + ggtitle("Potential Correlation Between CLV and Total Complaints")  
totalClaim.CLV

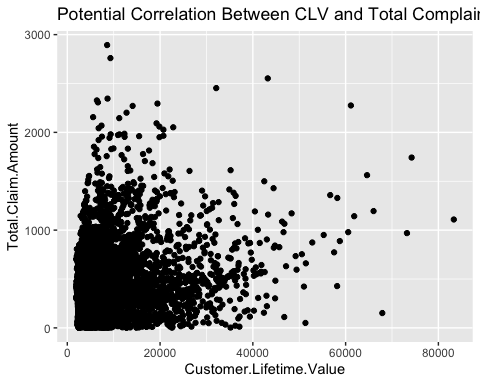


Figure 47: Scatterplot of CLV and Total.Claim.Amount

**Coverage VS CLV**

coverage.CLV <- ggplot(data = auto, aes(x = Customer.Lifetime.Value, y = Coverage)) + geom\_point()  
coverage.CLV <- coverage.CLV + ggtitle("Potential Correlation Between CLV and Coverage")  
coverage.CLV

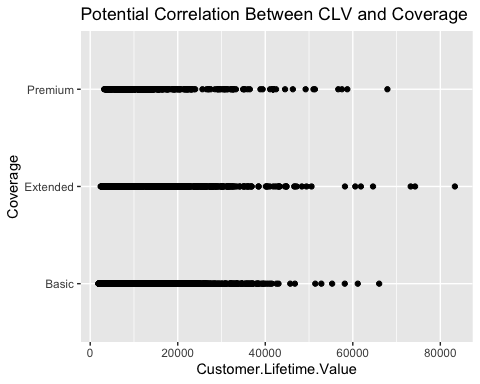


Figure 48: Correlation between CLV and Coverage type

Given the results above, it seems unlikely that coverage type has much impact on CLV.

**Employment VS CLV**

employeement.CLV <- ggplot(data = auto, aes(x = Customer.Lifetime.Value, y = EmploymentStatus)) + geom\_point()  
employeement.CLV <- employeement.CLV + ggtitle("Potential Correlation Between CLV and Employement Status")  
employeement.CLV

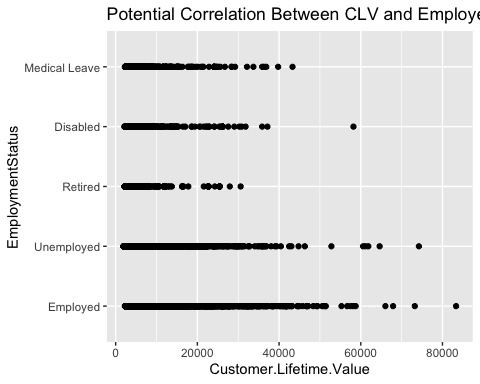


Figure 49: Correlation between CLV and Employment Status

**Sales Channel vs CLV**

SalesChannel.CLV <- ggplot(data = auto, aes(x = Customer.Lifetime.Value, y = Sales.Channel)) + geom\_point()  
SalesChannel.CLV <- SalesChannel.CLV + ggtitle("Potential Correlation Between CLV and Sales Channel")  
SalesChannel.CLV

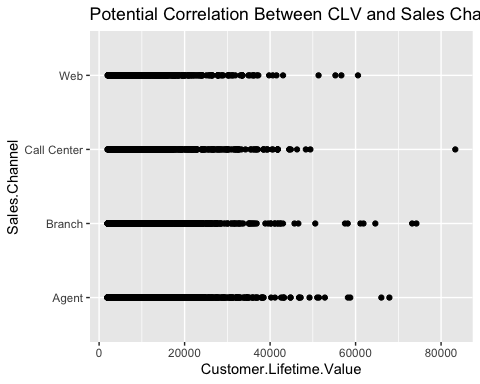


Figure 50: Correlation between CLV and Sales Channel

**Vechicle vs CLV**

vehicleClass.CLV <- ggplot(data = auto, aes(x = Customer.Lifetime.Value, y = Vehicle.Class)) + geom\_point()  
vehicleClass.CLV <- vehicleClass.CLV + ggtitle("Potential Correlation Between CLV and Vehicle Class")  
vehicleClass.CLV

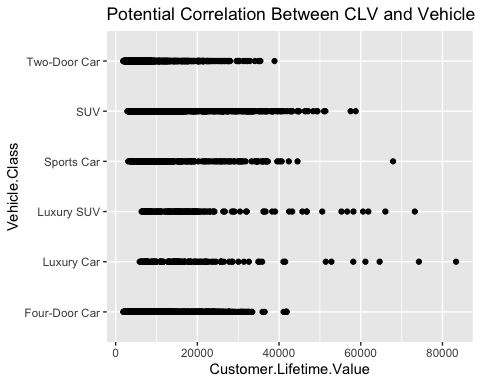


Figure 51: Correlation between CLV and Vehicle Class

**Vehicle Size vs CLV**

Vehicle.Size.CLV <- ggplot(data = auto, aes(x = Customer.Lifetime.Value, y = Vehicle.Size)) + geom\_point()  
Vehicle.Size.CLV <- Vehicle.Size.CLV + ggtitle("Potential Correlation Between CLV and Vehicle Size")  
Vehicle.Size.CLV

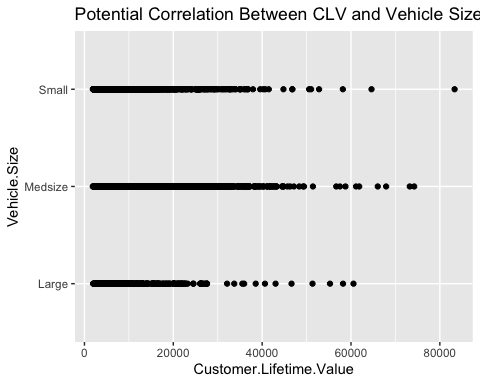


Figure 52: Correlation between CLV and Vehicle Size

## Subsection 3: Models

Load the data

auto <- read.csv("https://raw.githubusercontent.com/jcventosa/IST-707/master/auto.csv")  
# Remove row label  
auto <- auto[,-1]

Add discretized variables to be used in models

library(dplyr)  
# Adding predetermined Customer Lifetime Value categories.  
auto$CLV\_Levels <- as.factor(ifelse(auto$Customer.Lifetime.Value < 5000, "Low\_CLV",  
 ifelse(between(auto$Customer.Lifetime.Value, 5000, 8000), "Avg\_CLV",  
 ifelse(between(auto$Customer.Lifetime.Value, 8001, 15000), "High\_CLV",  
 "Super\_CLV"))))  
# Adding discretized variables for Months Since Policty Inception and   
# Months Since Last Claim  
auto$MoInception <- as.factor(ifelse(auto$Months.Since.Policy.Inception < 24, "<\_24\_months",  
 ifelse(between(auto$Months.Since.Policy.Inception, 24,48), "24-47\_months",  
 ifelse(between(auto$Months.Since.Policy.Inception,49,72), "48-71\_months",  
 "72+\_months"))))  
auto$MoLastClaim <- as.factor(ifelse(auto$Months.Since.Last.Claim < 6, "<\_6\_months",  
 ifelse(between(auto$Months.Since.Last.Claim, 6,11), "6-11\_months",  
 ifelse(between(auto$Months.Since.Last.Claim, 12,17), "12-17\_months",  
 ifelse(between(auto$Months.Since.Last.Claim, 18,23), "18-23\_months",  
 ifelse(between(auto$Months.Since.Last.Claim, 24,29), "24-29\_months",  
 "30+\_months"))))))

### Association Rules Mining

Association Rules Mining will be used to perform additional EDA on the customer characteristics associated with a “Yes” response to each of the offers.

Prep a transaction dataset for Apriori ARM

library(arules)  
library(arulesViz)  
  
# auto data set w/o continuous data. Includes previously discretized  
# CLV\_Levels, IncomeBin, MoPremiumBin, ClaimBin, MoInception, MoLastClaim  
autodiscrete <- auto[,c(2,4,5,6,8,9,11,12,18,19,20,21,23,24,26,27,28,29,30,31)]  
  
# Separte the Offers 1, 2, and 3 into their own datasets.  
Offer1 <- autodiscrete[autodiscrete$Renew.Offer.Type == "Offer1",]  
Offer2 <- autodiscrete[autodiscrete$Renew.Offer.Type == "Offer2",]  
Offer3 <- autodiscrete[autodiscrete$Renew.Offer.Type == "Offer3",]  
  
# Remove Renew.Offer.Type variable from the datasets.  
Offer1 <- Offer1[,-11]  
Offer2 <- Offer2[,-11]  
Offer3 <- Offer3[,-11]  
  
# Convert Offer 1, 2 and 3 datasets into transactions  
autotrans1 <- as(Offer1,"transactions")  
autotrans2 <- as(Offer2, "transactions")  
autotrans3 <- as(Offer3, "transactions")  
  
# check format of autotrans(x) data  
autotrans1

## transactions in sparse format with  
## 3752 transactions (rows) and  
## 83 items (columns)

autotrans2

## transactions in sparse format with  
## 2926 transactions (rows) and  
## 83 items (columns)

autotrans3

## transactions in sparse format with  
## 1432 transactions (rows) and  
## 83 items (columns)

YesRule1<-apriori(data=autotrans1, parameter=list(supp=0.019,conf = 0.70, maxlen=5),   
 appearance = list(default="lhs",rhs="Response=Yes"),  
 control = list(verbose=FALSE))  
YesRule1<-sort(YesRule1, decreasing=TRUE,by="confidence")  
inspect(YesRule1[1:5])

## lhs rhs support confidence coverage lift count  
## [1] {EmploymentStatus=Retired,   
## Sales.Channel=Agent,   
## Vehicle.Size=Medsize} => {Response=Yes} 0.01918977 0.9230769 0.02078891 5.830614 72  
## [2] {Coverage=Basic,   
## EmploymentStatus=Retired,   
## Vehicle.Size=Medsize} => {Response=Yes} 0.02078891 0.8965517 0.02318763 5.663067 78  
## [3] {EmploymentStatus=Retired,   
## Policy.Type=Personal Auto,   
## Sales.Channel=Agent} => {Response=Yes} 0.01945629 0.8902439 0.02185501 5.623224 73  
## [4] {Coverage=Basic,   
## EmploymentStatus=Retired,   
## Location.Code=Suburban,   
## Vehicle.Size=Medsize} => {Response=Yes} 0.01918977 0.8888889 0.02158849 5.614665 72  
## [5] {EmploymentStatus=Retired,   
## Location.Code=Suburban,   
## Policy.Type=Personal Auto,   
## Vehicle.Size=Medsize} => {Response=Yes} 0.02292111 0.8775510 0.02611940 5.543050 86

plot(YesRule1,method="graph",engine = "htmlwidget")

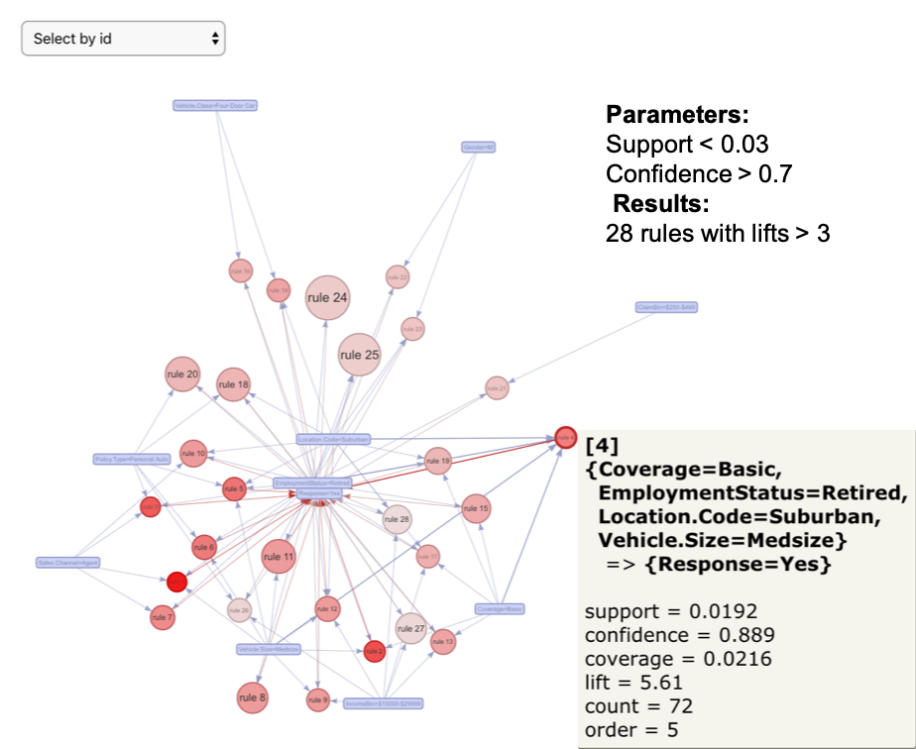


Figure 53: Offer 1 Apriori

Approximately 16% of Offer 1 recipients responded yes to the offer. The attributes commonly associated with a yes response to offer 1 were Suburban Retirees with Incomes in the 15k to 30k range with a Midsize Car and Basic Coverage. (Support < 0.03 and Confidence >0.7 generated 28 rules with lifts > 3)

YesRule2<-apriori(data=autotrans2, parameter=list(supp=0.02,conf = 0.50, minlen = 5),   
 appearance = list(default="lhs",rhs="Response=Yes"),  
 control = list(verbose=FALSE))  
YesRule2<-sort(YesRule2, decreasing=TRUE,by="confidence")  
inspect(YesRule2[1:5])

## lhs rhs support confidence coverage lift count  
## [1] {Coverage=Extended,   
## Gender=F,   
## Location.Code=Suburban,   
## Marital.Status=Married,   
## Sales.Channel=Agent} => {Response=Yes} 0.02050581 0.6896552 0.02973342 2.950192 60  
## [2] {Coverage=Extended,   
## Gender=F,   
## Location.Code=Suburban,   
## Sales.Channel=Agent} => {Response=Yes} 0.02870813 0.6131387 0.04682160 2.622871 84  
## [3] {Coverage=Extended,   
## Gender=F,   
## Location.Code=Suburban,   
## Policy.Type=Personal Auto,   
## Sales.Channel=Agent} => {Response=Yes} 0.02187286 0.6095238 0.03588517 2.607407 64  
## [4] {Coverage=Extended,   
## Location.Code=Suburban,   
## Marital.Status=Married,   
## Policy.Type=Personal Auto,   
## Sales.Channel=Agent} => {Response=Yes} 0.02050581 0.5825243 0.03520164 2.491909 60  
## [5] {Gender=F,   
## Location.Code=Suburban,   
## Sales.Channel=Agent,   
## MoPremiumBin=$75-$99} => {Response=Yes} 0.02050581 0.5769231 0.03554340 2.467949 60

plot(YesRule2,method="graph",engine = "htmlwidget")

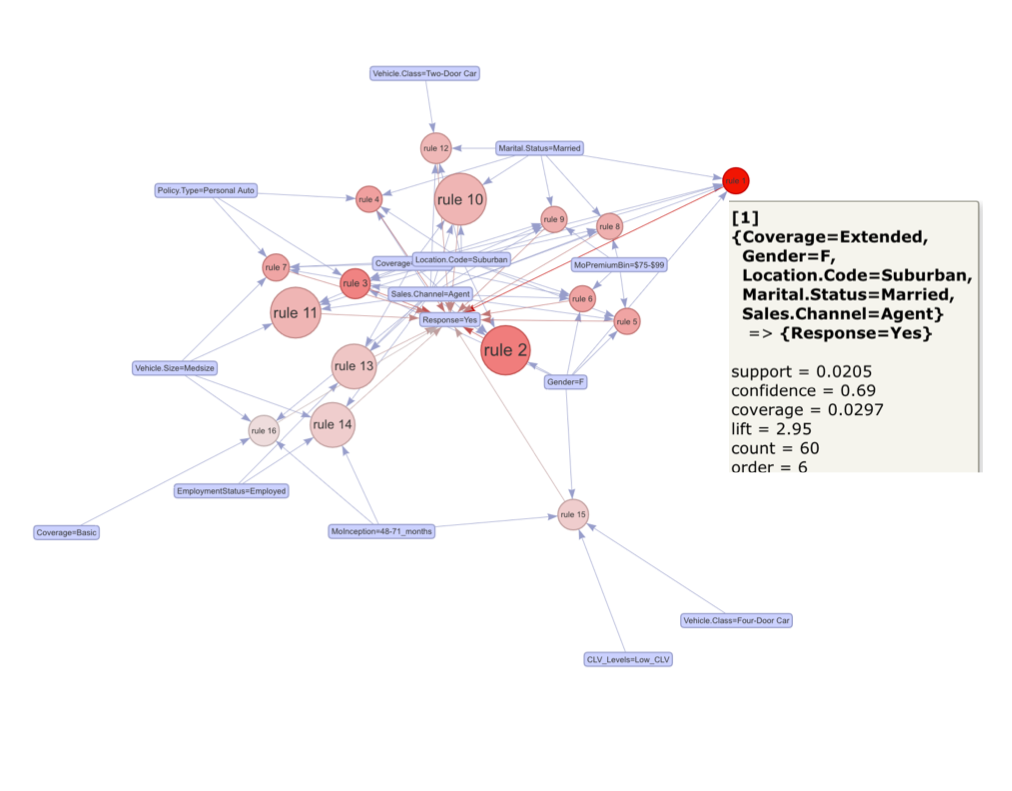


Figure 54: Offer 2 Apriori

Approximately 23% of Offer 2 recipients responded yes to the offer. The attributes commonly associated with a yes response to Offer 2 were Suburban Married Women with Extended Coverage usually acquired through a sales agent. (Support < 0.03 and Confidence > 0.5 generated 16 rules with lifts > 2)

YesRule3<-apriori(data=autotrans3, parameter=list(supp=0.01,conf = 0.10, maxlen=5),   
 appearance = list(default="lhs",rhs="Response=Yes"),  
 control = list(verbose=FALSE))  
YesRule3<-sort(YesRule3, decreasing=TRUE,by="support")  
inspect(YesRule3[1:5])

## lhs rhs support confidence coverage lift count  
## [1] {Gender=M,   
## ClaimBin=$250-$499,   
## CLV\_Levels=Low\_CLV} => {Response=Yes} 0.01256983 0.1363636 0.09217877 6.509091 18  
## [2] {Marital.Status=Married,   
## ClaimBin=$250-$499,   
## CLV\_Levels=Low\_CLV} => {Response=Yes} 0.01256983 0.1125000 0.11173184 5.370000 18  
## [3] {Gender=M,   
## Marital.Status=Married,   
## CLV\_Levels=Low\_CLV} => {Response=Yes} 0.01256983 0.1139241 0.11033520 5.437975 18  
## [4] {EmploymentStatus=Employed,   
## Marital.Status=Married,   
## Sales.Channel=Agent,   
## Vehicle.Class=Four-Door Car} => {Response=Yes} 0.01256983 0.1192053 0.10544693 5.690066 18  
## [5] {Gender=M,   
## Marital.Status=Married,   
## ClaimBin=$250-$499,   
## CLV\_Levels=Low\_CLV} => {Response=Yes} 0.01256983 0.2045455 0.06145251 9.763636 18

plot(YesRule2,method="graph",engine = "htmlwidget")

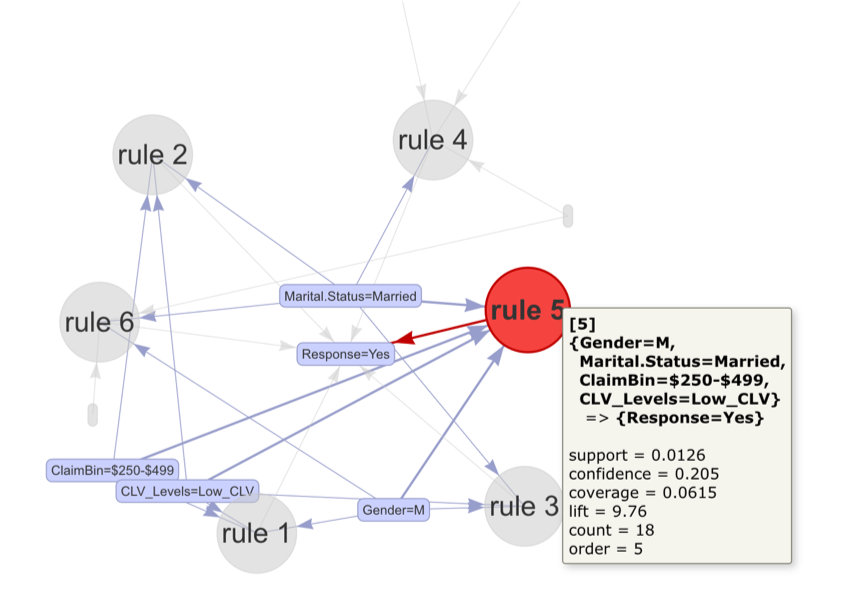


Figure 55: Offer 3 Apriori

Only 2% of Offer 3 recipients responded yes to the offer. The attributes that were most associated with a yes response to Offer 3 were Married Men with Low\_CLV and Total Claim Amounts ranging between 250 to 499 dollars. (Support < 0.013 and Confidence < 0.21 generated 6 rules with lifts > 5). The low support and low confidence values for these outcomes suggest there is a high likelihood of chance association.

The results from association rules mining determined that customer characteristics associated with a yes response differed by renew offer type. The findings suggest variables available in the data may be used to train classification models to provide better response predictions to the renew offers. Since each customer was given only one of four offers to accept, it would be reasonable to provide customers who declined their offer one of the other 3 offers as an alternative. To provide the customer with the alternative offer with the highest probability of renewal, 3 models will be needed. Each model will be trained on one of the 3 offers. The model that predicts the highest probability of a customer’s “Yes” response to an offer will determine the alternative offer the customer will receive.

Four algorithm candidates have been chosen for the models: Decision Tree, Naive Bayes, Support Vector Machine (SVM) and Random Forest. A cross validation for each of the algorithms will be executed to assess the models’ accuracy and precision at predicting a “Yes” or “No” response to Offers 1, 2 and 3. Note: Offer 4 was excluded because it did not receive a single “Yes” response.

### Decision Tree: Offer responses

SampleOffer1 <- sample.int(n=round(nrow(Offer1)\*.6))

SampleOffer2 <- sample.int(n=round(nrow(Offer2)\*.6))

TrainOffer1 <- Offer1[SampleOffer1,]

TrainOffer2 <- Offer2[SampleOffer2,]

TestOffer1 <- Offer1[-SampleOffer1,]

TestOffer2 <- Offer2[-SampleOffer2,]

## Remove Customer column

TrainOffer1 <- TrainOffer1[,-1]

TrainOffer2 <- TrainOffer2[,-1]

## Create decision tree for Offer1 subset

Offer1\_Tree <- rpart(Response ~ ., data = TrainOffer1, method = "class", control = rpart.control(cp=0))

Offer1\_Prediction = predict(Offer1\_Tree, TestOffer1, type = "class")

table(Response=Offer1\_Prediction, true = TestOffer1$Response)

> table(Response=Offer1\_Prediction, true = TestOffer1$Response)

true

Response No Yes

No 1152 79

Yes 115 155

Offer1\_PredictionTrain = predict(Offer1\_Tree, TrainOffer1, type = "class")

table(Response=Offer1\_PredictionTrain, true = TrainOffer1$Response)

> table(Response=Offer1\_PredictionTrain, true = TrainOffer1$Response)

true

Response No Yes

No 1819 60

Yes 72 300

## Create decision tree Offer2 subset

Offer2\_Tree <- rpart(Response ~ ., data = TrainOffer2, method = "class", control = rpart.control(cp=0))

Offer2\_Prediction <- predict(Offer2\_Tree, TestOffer2, type = "class")

table(Response=Offer2\_Prediction, true = TestOffer2$Response)

> table(Response=Offer2\_Prediction, true = TestOffer2$Response)

true

Response No Yes

No 797 58

Yes 112 203

## Create subset of dataset

## Include customers receiving Offers 2, 3, & 4

## Apply Offer1 model to subset

Offer234 <- auto$Renew.Offer.Type!="Offer1"

Offer234df <- auto[Offer234,]

Offer234df$Renew.Offer.Type <- "Offer1"

Offer1\_Model <- predict(Offer1\_Tree, Offer234df, type = "class")

table(Response=Offer1\_Model)

> table(Response=Offer1\_Model)

Response

No Yes

4933 449

## Create subset of dataset

## Include customers receiving Offers 1, 3, & 4

## Apply Offer2 model to subset

Offer134 <- auto$Renew.Offer.Type!="Offer2"

Offer134df <- auto[Offer134,]

Offer134df$Renew.Offer.Type <- "Offer2"

Offer2\_Model <- predict(Offer2\_Tree, Offer134df, type = "class")

table(Response=Offer2\_Model)

> table(Response=Offer2\_Model)

Response

No Yes

5350 858

### Naive Bayes

Libraries for Naive Bayes

library (naivebayes)

## naivebayes 0.9.7 loaded

library(klaR)

## Loading required package: MASS

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

library(e1071)  
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

EDA determined that none of the continuous variables in the data follow a Gaussian distribution based on visual inspection of histograms. Therefore, continuous variables require discretization to be included in Naive Bayes models. The discretized data sets (Offer1, Offer2, Offer3) created during the process of prepping the data for Association Rules Mining will be used to train the Naive Bayes models.

A 10 fold holdout cross validation for each of the Naive Bayes models will be performed to evaluate the model’s accuracy and precision.

Create a function to perform holdout cross validation evaluations of Naive Bayes models.

# Create result bins for holdoutcv function  
AllResultest <- list()  
AllLabelstest <- list()  
  
NBholdoutcv <- function (data, N, kfolds) {  
 holdout <- split(sample(1:N),1:kfolds)  
 for (k in 1:kfolds) {  
   
 test= data[holdout[[k]],]  
 train = data[-holdout[[k]],]  
  
# Make sure to remove Response from the testing data  
 test\_response <- test$Response  
 test\_no\_response <- test[,-2]  
  
# {naivebayes} package  
 model <- naive\_bayes(Response~., data = train, na.action = na.pass,   
 laplace = 1, adjust = 0)  
 predtest <- predict(model,test\_no\_response)  
   
 # Accumulate results from each fold  
 AllResultest <- c(AllResultest, predtest)  
 AllLabelstest <- c(AllLabelstest, test\_response)  
 ConfMatrix <- table(Actual = unlist(AllLabelstest), Prediction = unlist(AllResultest))}  
 return(ConfMatrix)  
}

Perform 10 fold cross validation Naive Bayes evaluation on each of the Offers 1, 2 and 3 recipients.

library(naivebayes)  
# Set up an experimental evaluation.  
# Number of observations  
N1 <- nrow(Offer1)  
N2 <- nrow(Offer2)  
N3 <- nrow(Offer3)  
# Number of desired splits  
kfolds <- 10  
# Perform NB holdout cross validation using NBholdoutcv function  
set.seed(123)  
NBholdoutcv(Offer1, N1, kfolds)

## Prediction  
## Actual 1 2  
## 1 3012 146  
## 2 388 206

set.seed(123)  
NBholdoutcv(Offer2, N2, kfolds)

## Prediction  
## Actual 1 2  
## 1 2122 120  
## 2 582 102

set.seed(123)  
NBholdoutcv(Offer3, N3, kfolds)

## Prediction  
## Actual 1 2  
## 1 1390 12  
## 2 8 22

Although the models holdout cross validation accuracies were all above 75%, precision on the “yes” response seemed inadequately low for all three models. Note: 1 = No and 2 = Yes in the confusion matrix results.

Offer 1 model accuracy = 0.858 with yes precision = 0.585

Offer 2 model accuracy = 0.760 with yes precision = 0.459

Offer 3 model accuracy = 0.986 with yes precision = 0.647

### Support Vector Machine

Prep the training and test data for SVM.

# Prep data set for SVM analysis  
# Unlike Naive Bayes continuous data in SVM algorithm need not be Gaussian.  
autosvm <- auto[,c(2:6,8:15,17:24)]  
  
# Segregate the data sets by Renew.Offer.Type to predict each offer  
Offer\_1 <- autosvm[autosvm$Renew.Offer.Type=="Offer1",]  
Offer\_2 <- autosvm[autosvm$Renew.Offer.Type == "Offer2",]  
Offer\_3 <- autosvm[autosvm$Renew.Offer.Type == "Offer3",]  
  
# Remove the Renew.Offer.Type variable from the data sets.  
Offer\_1 <- Offer\_1[,-17]  
Offer\_2 <- Offer\_2[,-17]  
Offer\_3 <- Offer\_3[,-17]

Create a function to perform holdout cross validation evaluations of SVM models.

library(e1071)  
# Create result bins for SVMholdoutcv function  
AllResultest <- list()  
AllLabelstest <- list()  
  
SVMholdoutcv <- function (data, N, kfolds) {  
 holdout <- split(sample(1:N),1:kfolds)  
 for (k in 1:kfolds) {  
   
 train= data[holdout[[k]],]  
 test = data[-holdout[[k]],]  
 test\_response <- test$Response  
  
# {e1071} package  
 model <- svm(Response~., data = train, kernel = "linear")  
 predtest <- predict(model,newdata = test, type=c("class"))  
   
 # Accumulate results from each fold  
 AllResultest <- c(AllResultest, predtest)  
 AllLabelstest <- c(AllLabelstest, test\_response)  
 ConfMatrix <- table(Actual = unlist(AllLabelstest), Prediction = unlist(AllResultest))}  
 return(ConfMatrix)  
}

Perform 10 fold holdout cross validation SVM evaluation on Offer 1, 2, and 3 recipients.

# Set up an experimental evaluation.  
# Number of observations  
N1 <- nrow(Offer\_1)  
N2 <- nrow(Offer\_2)  
N3 <- nrow(Offer\_3)  
# Number of desired splits  
kfolds <- 10  
# Perform NB holdout cross validation using NBholdoutcv function  
set.seed(123)  
SVMholdoutcv(Offer\_1, N1, kfolds)

## Prediction  
## Actual 1 2  
## 1 28128 294  
## 2 4101 1245

set.seed(123)  
SVMholdoutcv(Offer\_2, N2, kfolds)

## Prediction  
## Actual 1 2  
## 1 19066 1112  
## 2 5269 887

set.seed(123)  
SVMholdoutcv(Offer\_3, N3, kfolds)

## Prediction  
## Actual 1 2  
## 1 12459 159  
## 2 189 81

Although the SVM models’ holdout cross validation accuracies were all above 75%, precision on the “yes” response for Offers 2 & 3 were low (< 50%). Note: 1 = No and 2 = Yes in the confusion matrix results.

Offer 1 model accuracy = 0.870 with yes precision = 0.809

Offer 2 model accuracy = 0.758 with yes precision = 0.444

Offer 3 model accuracy = 0.973 with yes precision = 0.338

### Random Forest

Use {caret} package to run a cross validation on random forest train data to find best model.

library(caret)  
# Create train and test data sets using 80/20 split  
set.seed(49)  
sample\_1 <- createDataPartition(Offer\_1$Response, p=0.80, list = FALSE)  
train\_1 <- Offer\_1[sample\_1,]  
test\_1 <- Offer\_1[-sample\_1,]  
  
set.seed(49)  
sample\_2 <- createDataPartition(Offer\_2$Response, p=0.80, list = FALSE)  
train\_2 <- Offer\_2[sample\_2,]  
test\_2 <- Offer\_2[-sample\_2,]  
  
set.seed(49)  
sample\_3 <- createDataPartition(Offer\_3$Response, p=0.80, list = FALSE)  
train\_3 <- Offer\_3[sample\_3,]  
test\_3 <- Offer\_3[-sample\_3,]  
  
set.seed(123)  
RFmodel\_1 <- train(Response~., data=train\_1, method = "rf",  
 trControl = trainControl("cv", number = 10),  
 tuneLength = 10)  
  
set.seed(123)  
RFmodel\_2 <- train(Response~., data=train\_2, method = "rf",  
 trControl = trainControl("cv", number = 10),  
 tuneLength = 10)  
  
set.seed(123)  
RFmodel\_3 <- train(Response~., data=train\_3, method = "rf",  
 trControl = trainControl("cv", number = 10),  
 tuneLength = 10)  
  
RFmodel\_1$finalModel

##   
## Call:  
## randomForest(x = x, y = y, mtry = param$mtry)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 16  
##   
## OOB estimate of error rate: 0.53%  
## Confusion matrix:  
## No Yes class.error  
## No 2511 16 0.006331619  
## Yes 0 476 0.000000000

RFmodel\_2$finalModel

##   
## Call:  
## randomForest(x = x, y = y, mtry = param$mtry)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 16  
##   
## OOB estimate of error rate: 0.38%  
## Confusion matrix:  
## No Yes class.error  
## No 2786 8 0.004459309  
## Yes 1 547 0.001824818

RFmodel\_3$finalModel

##   
## Call:  
## randomForest(x = x, y = y, mtry = param$mtry)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 6  
##   
## OOB estimate of error rate: 0%  
## Confusion matrix:  
## No Yes class.error  
## No 1122 0 0  
## Yes 0 24 0

The Random Forest models’ cross validation accuracies were all above 99%. In addition the precision on the “yes” response were all above 95%.

Offer 1 model accuracy = 0.995 with yes precision = 0.967

Offer 2 model accuracy = 0.996 with yes precision = 0.986

Offer 3 model accuracy = 1.000 with yes precision = 1.000

The Random Forest models had the highest overall accuracy and the highest precision in correctly predicting a “yes” response. To test the models for over fitting the accuracies from the models results at predicting the train and test data will be compared.

# predict train and test using final models and compare accuracies.  
predtrain\_1 <- predict(RFmodel\_1, train\_1, type = "prob")  
predtest\_1 <- predict(RFmodel\_1, test\_1, type = "prob")  
resulttrain\_1 <- ifelse(predtrain\_1$Yes > 0.5,"Yes", "No")  
resulttest\_1 <- ifelse(predtest\_1$Yes > 0.5,"Yes", "No")  
table(Train1\_Actual = train\_1$Response, Prediction = resulttrain\_1)

## Prediction  
## Train1\_Actual No Yes  
## No 2527 0  
## Yes 0 476

table(Test1\_Actual = test\_1$Response, Prediction = resulttest\_1)

## Prediction  
## Test1\_Actual No Yes  
## No 628 3  
## Yes 0 118

predtrain\_2 <- predict(RFmodel\_2, train\_2, type = "prob")  
predtest\_2 <- predict(RFmodel\_2, test\_2, type = "prob")  
resulttrain\_2 <- ifelse(predtrain\_2$Yes > 0.5,"Yes", "No")  
resulttest\_2 <- ifelse(predtest\_2$Yes > 0.5,"Yes", "No")  
table(Train2\_Actual = train\_2$Response, Prediction = resulttrain\_2)

## Prediction  
## Train2\_Actual No Yes  
## No 1794 0  
## Yes 0 548

table(Test2\_Actual = test\_2$Response, Prediction = resulttest\_2)

## Prediction  
## Test2\_Actual No Yes  
## No 448 0  
## Yes 0 136

predtrain\_3 <- predict(RFmodel\_3, train\_3, type = "prob")  
predtest\_3 <- predict(RFmodel\_3, test\_3, type = "prob")  
resulttrain\_3 <- ifelse(predtrain\_3$Yes > 0.5,"Yes", "No")  
resulttest\_3 <- ifelse(predtest\_3$Yes > 0.5,"Yes", "No")  
table(Train3\_Actual = train\_3$Response, Prediction = resulttrain\_3)

## Prediction  
## Train3\_Actual No Yes  
## No 1122 0  
## Yes 0 24

table(Test3\_Actual = test\_3$Response, Prediction = resulttest\_3)

## Prediction  
## Test3\_Actual No Yes  
## No 280 0  
## Yes 0 6

All the models had 100% accuracy on the train data. Model for Offer 1 had a 97.5% accuracy on its test data while models for Offer 2 and 3 had 100% accuracy on their test data. The results indicate very low signs of overfitting with extremely high accuracy at predicting unknown data. With the highest accuracy results of all the algorithms and low signs of overfitting, the random forest models will be used to predict responses on customers who declined their original renew offer types to determine the best alternative offer upon redistribution.

Extract all customers with “No” responses to their renew offer types and use each random forest Offer model to predict probability of “Yes” response to alternate offer.

library(dplyr)  
# Extract "No" responses into new data set  
OfferRespNo <- auto[auto$Response == "No",]  
# Extract the variables used in the train data of random forest models less Response  
autoRespNo <- OfferRespNo[,c(2,3,5,6,8:15,17:19,21:24)]  
  
# Predict probabilities of No and Yes responses based on random forest models  
RFpredict1 <- predict(RFmodel\_1, autoRespNo, type = "prob")  
RFpredict2 <- predict(RFmodel\_2, autoRespNo, type = "prob")  
RFpredict3 <- predict(RFmodel\_3, autoRespNo, type = "prob")  
  
# Add "Yes" probabilities to OfferRespNo and compare results for best "Yes" outcomes.  
OfferRespNo$Offer1\_Yes <- RFpredict1$Yes  
OfferRespNo$Offer2\_Yes <- RFpredict2$Yes  
OfferRespNo$Offer3\_Yes <- RFpredict3$Yes  
  
# Determine best alternative offer to distribute based on highest "Yes" probability  
OfferRespNo$BestOffer <- ifelse(OfferRespNo$Renew.Offer.Type == "Offer4",  
 ifelse(OfferRespNo$Offer1\_Yes > OfferRespNo$Offer2\_Yes,   
 ifelse(OfferRespNo$Offer1\_Yes > OfferRespNo$Offer3\_Yes,  
 "Offer1",  
 ifelse(OfferRespNo$Offer3\_Yes > OfferRespNo$Offer2\_Yes,  
 "Offer3","Offer2")),  
 ifelse(OfferRespNo$Offer2\_Yes > OfferRespNo$Offer3\_Yes,   
 "Offer2", "Offer3")),  
 ifelse(OfferRespNo$Renew.Offer.Type == "Offer1",  
 ifelse(OfferRespNo$Offer2\_Yes > OfferRespNo$Offer3\_Yes,   
 "Offer2", "Offer3"),  
 ifelse(OfferRespNo$Renew.Offer.Type == "Offer2",  
 ifelse(OfferRespNo$Offer1\_Yes > OfferRespNo$Offer3\_Yes,  
 "Offer1", "Offer3"),  
 ifelse(OfferRespNo$Offer1\_Yes > OfferRespNo$Offer2\_Yes,   
 "Offer1", "Offer2"))))  
  
# Record probability of "Yes" for Best Offer  
OfferRespNo$OfferProbability <- ifelse(OfferRespNo$Renew.Offer.Type == "Offer4",  
 ifelse(OfferRespNo$Offer1\_Yes > OfferRespNo$Offer2\_Yes,   
 ifelse(OfferRespNo$Offer1\_Yes > OfferRespNo$Offer3\_Yes,  
 OfferRespNo$Offer1\_Yes,  
 ifelse(OfferRespNo$Offer3\_Yes > OfferRespNo$Offer2\_Yes,  
 OfferRespNo$Offer3\_Yes, OfferRespNo$Offer2\_Yes)),  
 ifelse(OfferRespNo$Offer2\_Yes > OfferRespNo$Offer3\_Yes,   
 OfferRespNo$Offer2\_Yes, OfferRespNo$Offer3\_Yes)),  
 ifelse(OfferRespNo$Renew.Offer.Type == "Offer1",  
 ifelse(OfferRespNo$Offer2\_Yes > OfferRespNo$Offer3\_Yes,   
 OfferRespNo$Offer2\_Yes, OfferRespNo$Offer3\_Yes),  
 ifelse(OfferRespNo$Renew.Offer.Type == "Offer2",  
 ifelse(OfferRespNo$Offer1\_Yes > OfferRespNo$Offer3\_Yes,  
 OfferRespNo$Offer1\_Yes, OfferRespNo$Offer3\_Yes),  
 ifelse(OfferRespNo$Offer1\_Yes > OfferRespNo$Offer2\_Yes,   
 OfferRespNo$Offer1\_Yes, OfferRespNo$Offer2\_Yes))))  
  
# Provide discretized probability of "Yes" Response to Best Offer  
OfferRespNo$YesChance <- ifelse(OfferRespNo$OfferProbability < 0.25, "< 25% Chance",  
 ifelse(between(OfferRespNo$OfferProbability,0.25,0.5),   
 "25%-49% Chance",  
 ifelse(OfferRespNo$OfferProbability > 0.75, "> 75% Chance",  
 "50%-75% Chance")))

Print 10 sample rows illustrating the probability process of determining best offer.

head(OfferRespNo[order(-OfferRespNo$OfferProbability),c(20,32:35)],10)

## Renew.Offer.Type Offer1\_Yes Offer2\_Yes Offer3\_Yes BestOffer  
## 8301 Offer3 0.894 0.764 0.002 Offer1  
## 7042 Offer2 0.874 0.120 0.022 Offer1  
## 1244 Offer3 0.836 0.650 0.010 Offer1  
## 6376 Offer3 0.644 0.806 0.014 Offer2  
## 2321 Offer4 0.670 0.800 0.086 Offer2  
## 1691 Offer3 0.794 0.418 0.008 Offer1  
## 4661 Offer4 0.790 0.326 0.052 Offer1  
## 6988 Offer2 0.728 0.222 0.008 Offer1  
## 8751 Offer4 0.722 0.554 0.030 Offer1  
## 379 Offer2 0.704 0.174 0.000 Offer1

As illustrated in the printout above the Best Offer to be redistributed to the customer was determined by the model with the highest probability for a “Yes” response. The Renew.Offer.Type column represents the original offer the customer declined. The Offer1\_Yes, Offer2\_Yes and Offer3\_Yes columns represent each model’s prediction of a yes response by probability. BestOffer column represents the Offer with the highest probability for renewal.

Use Shirin Elsinghorst’s function to plot a sample random forest tree with least and most nodes.

library(dplyr)  
library(ggraph)  
library(igraph)

##   
## Attaching package: 'igraph'

## The following object is masked from 'package:arules':  
##   
## union

## The following objects are masked from 'package:dplyr':  
##   
## as\_data\_frame, groups, union

## The following objects are masked from 'package:stats':  
##   
## decompose, spectrum

## The following object is masked from 'package:base':  
##   
## union

# This is a function shared by Shirin Elsinghorst in her post   
# "Plotting trees from Random Forest models with ggraph"  
# (https://shiring.github.io/machine\_learning/2017/03/16/rf\_plot\_ggraph)  
  
tree\_func <- function(final\_model,   
 tree\_num) {  
   
 # get tree by index  
 tree <- randomForest::getTree(final\_model,   
 k = tree\_num,   
 labelVar = TRUE) %>%  
 tibble::rownames\_to\_column() %>%  
 # make leaf split points to NA, so the 0s won't get plotted  
 mutate(`split point` = ifelse(is.na(prediction), `split point`, NA))  
   
 # prepare data frame for graph  
 graph\_frame <- data.frame(from = rep(tree$rowname, 2),  
 to = c(tree$`left daughter`, tree$`right daughter`))  
   
 # convert to graph and delete the last node that we don't want to plot  
 graph <- graph\_from\_data\_frame(graph\_frame) %>%  
 delete\_vertices("0")  
   
 # set node labels  
 V(graph)$node\_label <- gsub("\_", " ", as.character(tree$`split var`))  
 V(graph)$leaf\_label <- as.character(tree$prediction)  
 V(graph)$split <- as.character(round(tree$`split point`, digits = 2))  
   
 # plot  
 plot <- ggraph(graph, 'dendrogram') +   
 theme\_bw() +  
 geom\_edge\_link() +  
 geom\_node\_point() +  
 geom\_node\_text(aes(label = node\_label), na.rm = TRUE, repel = TRUE) +  
 geom\_node\_label(aes(label = split), vjust = 2.5, na.rm = TRUE, fill = "white") +  
 geom\_node\_label(aes(label = leaf\_label, fill = leaf\_label), na.rm = TRUE,   
 repel = TRUE, colour = "white", fontface = "bold", show.legend = FALSE) +  
 theme(panel.grid.minor = element\_blank(),  
 panel.grid.major = element\_blank(),  
 panel.background = element\_blank(),  
 plot.background = element\_rect(fill = "white"),  
 panel.border = element\_blank(),  
 axis.line = element\_blank(),  
 axis.text.x = element\_blank(),  
 axis.text.y = element\_blank(),  
 axis.ticks = element\_blank(),  
 axis.title.x = element\_blank(),  
 axis.title.y = element\_blank(),  
 plot.title = element\_text(size = 18))  
   
 print(plot)  
}

Create the objects for the random forest tree samples using the Offer 3 model

tree\_min <- which(RFmodel\_3$finalModel$forest$ndbigtree == max(RFmodel\_3$finalModel$forest$ndbigtree))  
tree\_max <- which(RFmodel\_3$finalModel$forest$ndbigtree == max(RFmodel\_3$finalModel$forest$ndbigtree))

Plot the Offer 3 random forest model ‘tree’ with the least number of nodes.

t1 <- tree\_func(final\_model = RFmodel\_3$finalModel, tree\_min)

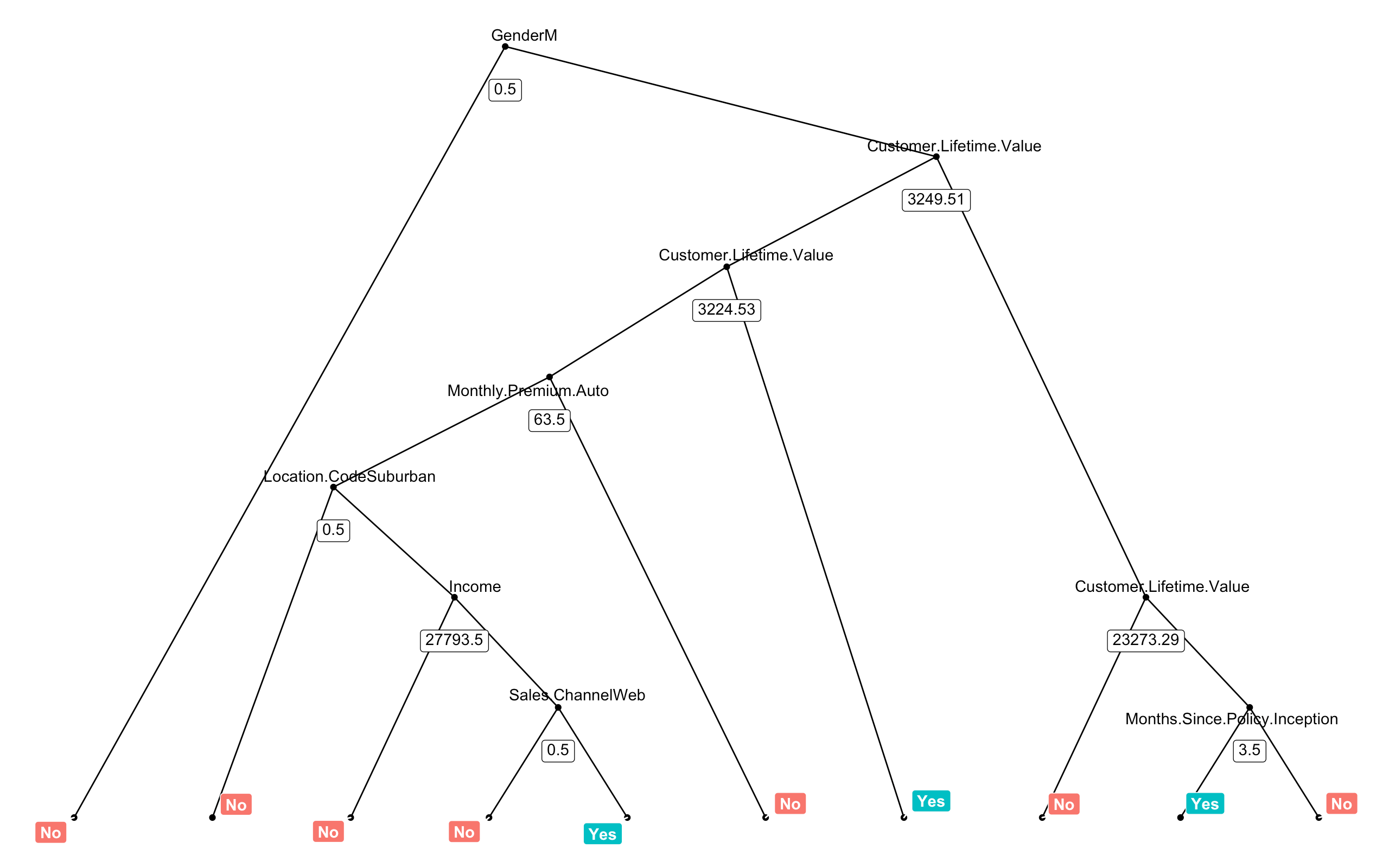


Figure 56: Random Forest - minimum nodes

Plot the Offer 3 random forest model ‘tree’ with the most number of nodes

t2 <- tree\_func(final\_model = RFmodel\_3$finalModel, tree\_max)

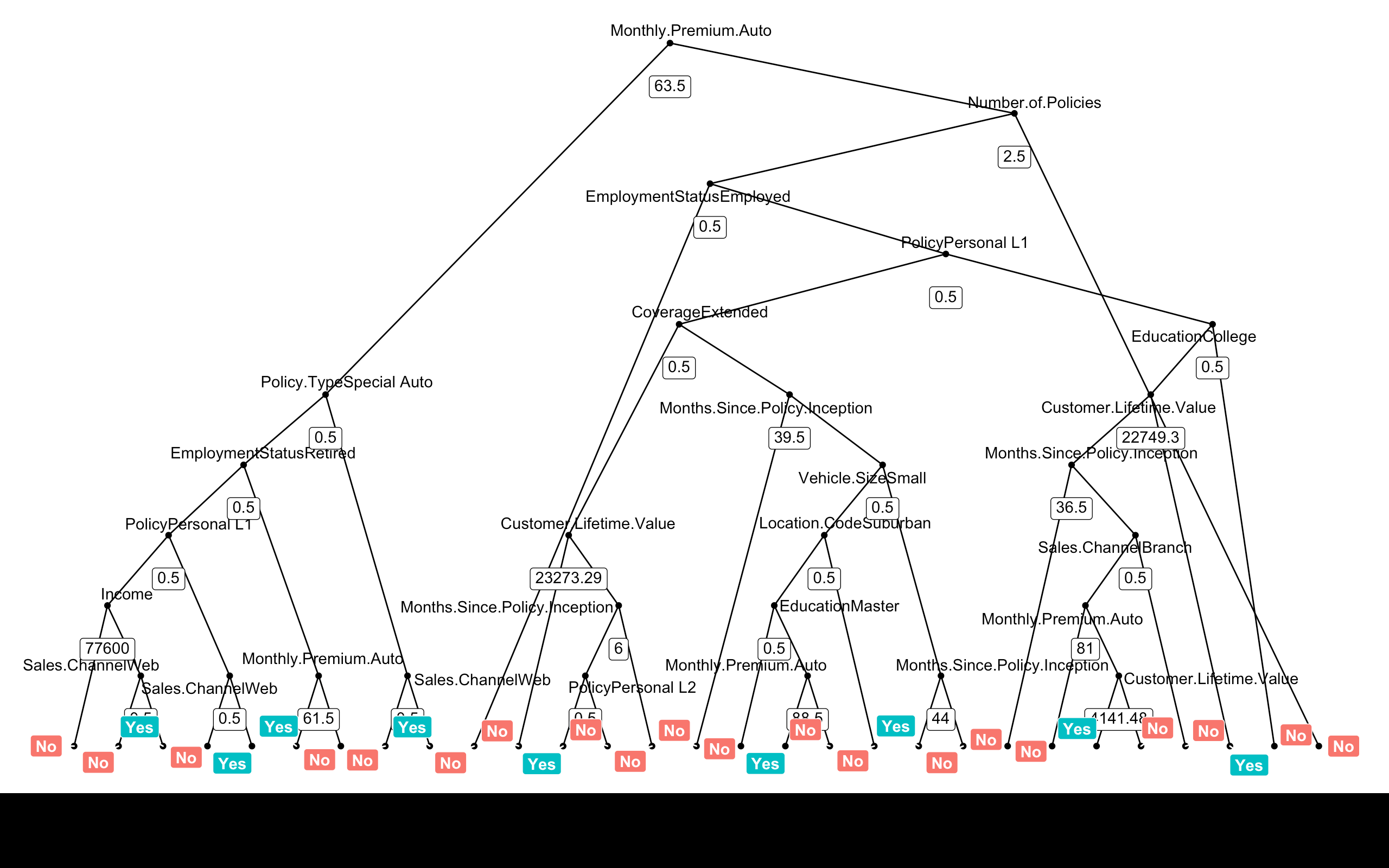


Figure 57: Random Forest - maximum nodes

Graph the model results on how the offers will be redistributed based on the best alternative offer. In addition, graph the model results on the probability the renew offers would be accepted.

library(ggplot2)  
  
OfferRespNo$YesChance <- factor(OfferRespNo$YesChance, levels = c("< 25% Chance","25%-49% Chance",  
 "50%-75% Chance","> 75% Chance"))  
  
ggplot(OfferRespNo, aes(x=Renew.Offer.Type, fill = BestOffer)) + geom\_bar()+  
 scale\_fill\_brewer(palette = "Dark2", name = "Best Offer") + labs(x="Original Offer Declined")+  
 ggtitle("Recommended Redistribution by Best Offer") +  
 scale\_x\_discrete(breaks=c("Offer1", "Offer2","Offer3","Offer4"),  
 labels = c("Declined Offer 1", "Declined Offer 2",  
 "Declined Offer 3", "Declined Offer 4"))



Figure 58: Bar chart of redistributed offers

ggplot(OfferRespNo, aes(x=YesChance, fill = BestOffer)) + geom\_bar() +  
 scale\_fill\_brewer(palette = "Dark2", name = "Best Offer") + labs(x= "Probability of a Yes Response")+  
 ggtitle("Recommended Redistribution of Best Offer by Probability of a Yes Conversion")

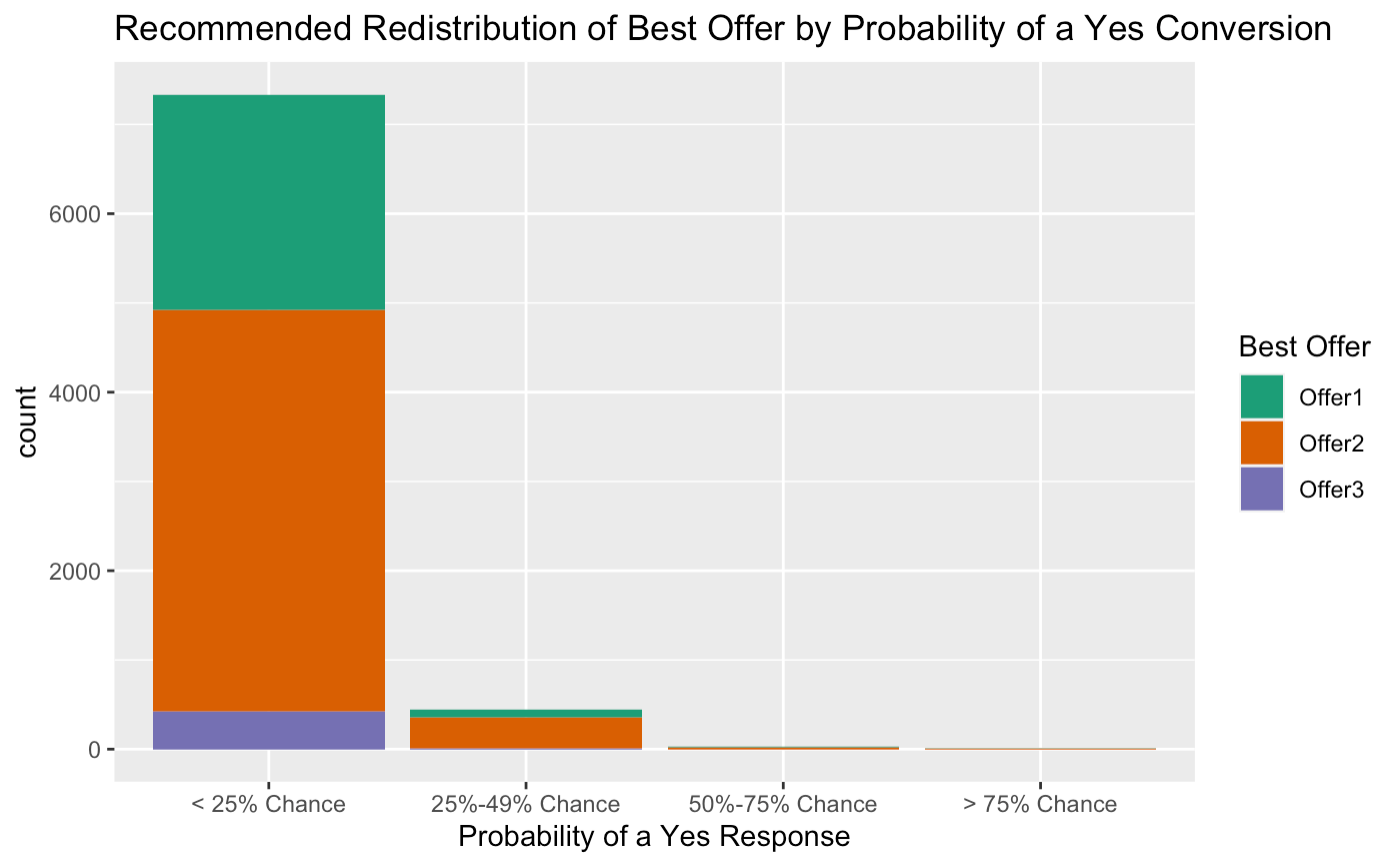


Figure 59: Bar chart of probabilities of "yes" response after redistribution

Table of Best Offer distribution by probability the offer will be accepted.

table(Best\_Offer = OfferRespNo$BestOffer,Probability\_Offer\_Accepted = OfferRespNo$YesChance)

## Probability\_Offer\_Accepted  
## Best\_Offer < 25% Chance 25%-49% Chance 50%-75% Chance > 75% Chance  
## Offer1 2412 91 15 5  
## Offer2 4503 353 18 2  
## Offer3 422 5 0 0

As illustrated in the table above, only 40 customers have a > 50% probability of renewing the alternative offer. Although the models had high accuracy and precision, the results predicted a low conversion rate upon redistribution of offers.

Given the poor performance of the renew offer types overall, a reexamination on the renew offer design may be needed to create offers that will achieve higher conversions. To aid in this process cluster analysis can be used to find distinct customer segments. The characteristics that make up the customer segments can provide insights and guidance to design renewal offers to make them more attractive to the target customers by segment.

### Cluster Analysis

Libraries for cluster analysis

# Libraries  
require(tidyverse)

## Loading required package: tidyverse

## ── Attaching packages ───── tidyverse 1.3.0 ──

## ✓ ggplot2 3.3.0 ✓ purrr 0.3.4  
## ✓ tibble 3.0.1 ✓ dplyr 0.8.5  
## ✓ tidyr 1.1.0 ✓ stringr 1.4.0  
## ✓ readr 1.3.1 ✓ forcats 0.5.0

## ── Conflicts ──────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

require(factoextra)

## Loading required package: factoextra

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

require(cluster)

## Loading required package: cluster

require(gridExtra)

## Loading required package: gridExtra

##   
## Attaching package: 'gridExtra'

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

require(dplyr)  
require(plyr)

## Loading required package: plyr

## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

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

require(VIM)

## Loading required package: VIM

## Loading required package: colorspace

## Loading required package: grid

## VIM is ready to use.

## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues

##   
## Attaching package: 'VIM'

## The following object is masked from 'package:datasets':  
##   
## sleep

require(lubridate)

## Loading required package: lubridate

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:dplyr':  
##   
## intersect, setdiff, union

## The following objects are masked from 'package:base':  
##   
## date, intersect, setdiff, union

require(DT)

## Loading required package: DT

library(arules)

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':  
##   
## expand, pack, unpack

##   
## Attaching package: 'arules'

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

## The following objects are masked from 'package:base':  
##   
## abbreviate, write

Prep the data to be used for cluster analysis. Variables to be included in the analysis: Gender, Income, Education, Marital Status, Employment Status, State, Region, Vehicle Size, Vehicle Class, Policy Type, Coverage, Sales Channel, Months Since Policy Inception, Monthly Premium, Months Since Last Claim, Total Claim Amount, Number of Open Complaints, Number of Policies and Customer Lifetime Value.

# move all continuous data into autoclust1  
autoclust1 <- auto[,c(3,10,13,14,15,16,17,22)]  
# transform nominal customer profile variables into ordinal  
autoclust2 <- data.frame(Education = ifelse(auto$Education == "High School or Below", 1,  
 ifelse(auto$Education == "College", 2,  
 ifelse(auto$Education == "Bachelor", 3,  
 ifelse(auto$Education == "Master",4, 5)))))  
autoclust2$Gender <- ifelse(auto$Gender == "F", 0, 1)  
autoclust2$MaritalStatus <- ifelse(auto$Marital.Status == "Divorced", 0,  
 ifelse(auto$Marital.Status == "Single",1, 2))  
autoclust2$EmployStatus <- ifelse(auto$EmploymentStatus == "Unemployed", 0,  
 ifelse(auto$EmploymentStatus == "Employed",1,  
 ifelse(auto$EmploymentStatus == "Retired", 2,  
 ifelse(auto$EmploymentStatus == "Disabled", 3, 4))))  
autoclust2$State <- ifelse(auto$State == "Arizona", 0,  
 ifelse(auto$State == "California", 1,  
 ifelse(auto$State == "Nevada", 2,  
 ifelse(auto$State == "Oregon", 3, 4))))  
autoclust2$Region <- ifelse(auto$Location.Code == "Rural", 0,  
 ifelse(auto$Location.Code == "Suburban", 1, 2))  
# transform nominal policy profile variables into ordinal  
autoclust3 <- data.frame(PolicyType = ifelse(auto$Policy.Type == "Corporate Auto",0,  
 ifelse(auto$Policy.Type == "Personal Auto",1, 2)))  
autoclust3$VehicleSize <- ifelse(auto$Vehicle.Size == "Large", 2,  
 ifelse(auto$Vehicle.Size == "Medsize", 1,0))  
autoclust3$VehicleClass <-ifelse(auto$Vehicle.Class == "Four-Door Car", 0,  
 ifelse(auto$Vehicle.Class == "Luxury Car", 1,  
 ifelse(auto$Vehicle.Class == "Luxury SUV", 2,  
 ifelse(auto$Vehicle.Class == "Sports Car", 3,  
 ifelse(auto$Vehicle.Class == "Two-Door Car", 4,5)))))  
autoclust3$Coverage <- ifelse(auto$Coverage == "Basic", 0,  
 ifelse(auto$Coverage == "Extended", 1, 2))  
autoclust3$SChannel <- ifelse(auto$Sales.Channel == "Agent", 0,  
 ifelse(auto$Sales.Channel == "Branch", 1,  
 ifelse(auto$Sales.Channel == "Call Center", 2, 3)))  
autoclust3$Response <- ifelse(auto$Response == "Yes", 1, 0)  
autoclust3$RenewOffer <- ifelse(auto$Renew.Offer.Type == "Offer1", 1,  
 ifelse(auto$Renew.Offer.Type == "Offer2", 2,  
 ifelse(auto$Renew.Offer.Type == "Offer3", 3, 4)))  
  
autoclust <- data.frame(autoclust1, autoclust2, autoclust3)

Use elbow method to determine optimal number of clusters by using the Total Within Sum Squares (wss) to the Number of Clusters.

# Determine optimal number of clusters using elbow method  
fviz\_nbclust(autoclust, kmeans, method = "wss") +   
 geom\_vline(xintercept = 3, linetype = 2)



Figure 60: Optimal clusters for k-means analysis

The wss to number clusters chart suggests the optimal number of clusters to be 3.

Run the kmeans algorithm with k= 3 clusters. Plot a 2-dimensional visualization of the resulting clusters.

# Run cluster analysis with k= 3 clusters  
set.seed(123)  
k3 <- kmeans(autoclust, centers = 3, nstart = 25)  
# Plot a 2 dimensional representation of the resulting clusters  
p3 <- fviz\_cluster(k3, geom = "point", data = autoclust) + ggtitle("Kmeans 3 Customer Clusters")  
p3

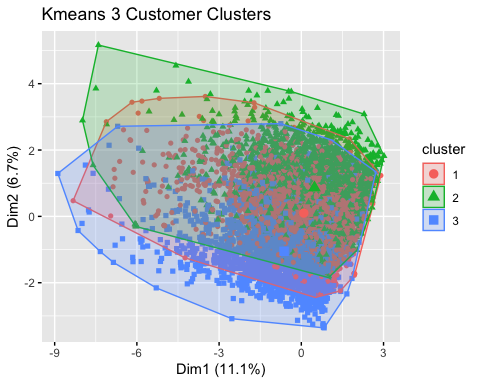


Figure 61: K-means planar clusters

Explore each cluster’s variables to determine the distinctive differences between the 3 clusters.

# Combine auto and cluster results into a data frame for EDA  
aCluster <- auto  
aCluster$Clustk3 <- k3$cluster  
  
MyTheme <- theme(axis.text = element\_text(size = 18),   
 axis.title = element\_text(size = 20, face ="bold"),   
 plot.title = element\_text(size = 24, face = "bold"),  
 legend.text = element\_text(size = 22),  
 legend.title = element\_text(size = 22))  
  
Xlabels <- scale\_x\_discrete(breaks=c("1", "2","3"),  
 labels = str\_wrap( c("Leisure Working-Class Provincials",  
 "Affluent Cautiously Vigilant Drivers",  
 "Single Out-of-Action Suburban Mavericks"), width = 20))  
  
aCluster$EmploymentStatus <- factor(aCluster$EmploymentStatus,   
 levels = c("Employed","Medical Leave",  
 "Disabled","Retired","Unemployed"))  
  
ggplot(aCluster,aes(x=as.factor(Clustk3), fill = EmploymentStatus )) + geom\_bar(position = "fill") +  
 scale\_y\_continuous(labels = scales::percent) +   
 labs(x="Customer Segments", y= "percent", fill = "Employment Status") +  
 ggtitle("Customer Segments by Employment Status") + scale\_fill\_brewer(palette = "YlOrBr") +  
 Xlabels + MyTheme

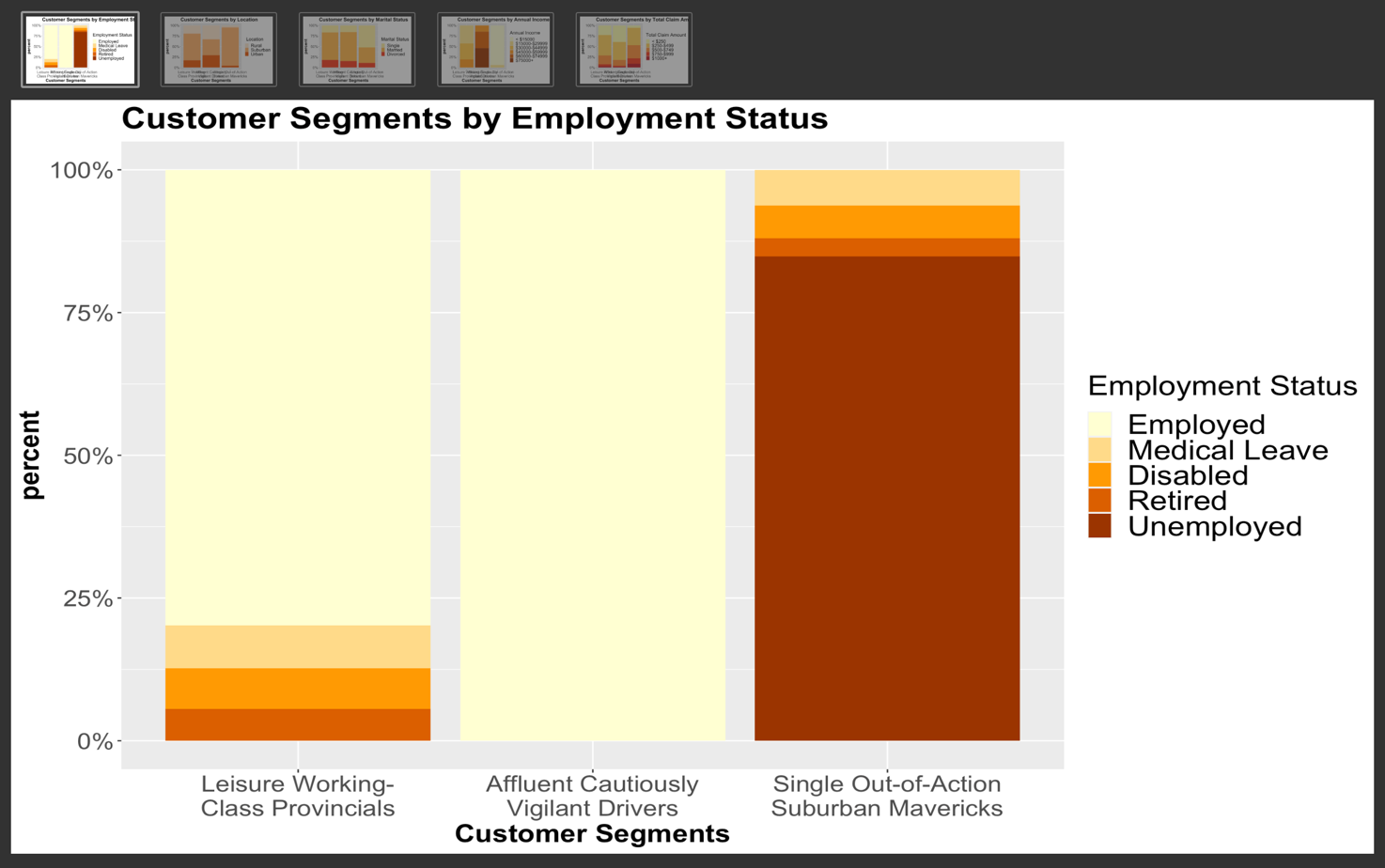


Figure 62: Customer segments by Employment Status

ggplot(aCluster,aes( x=as.factor(Clustk3), fill = Location.Code)) + geom\_bar(position = "fill") +  
 scale\_y\_continuous(labels = scales::percent) +   
 labs(x="Customer Segments", y= "percent", fill = " Location") +  
 ggtitle("Customer Segments by Location") + scale\_fill\_brewer(palette = "Oranges")+  
 Xlabels + MyTheme

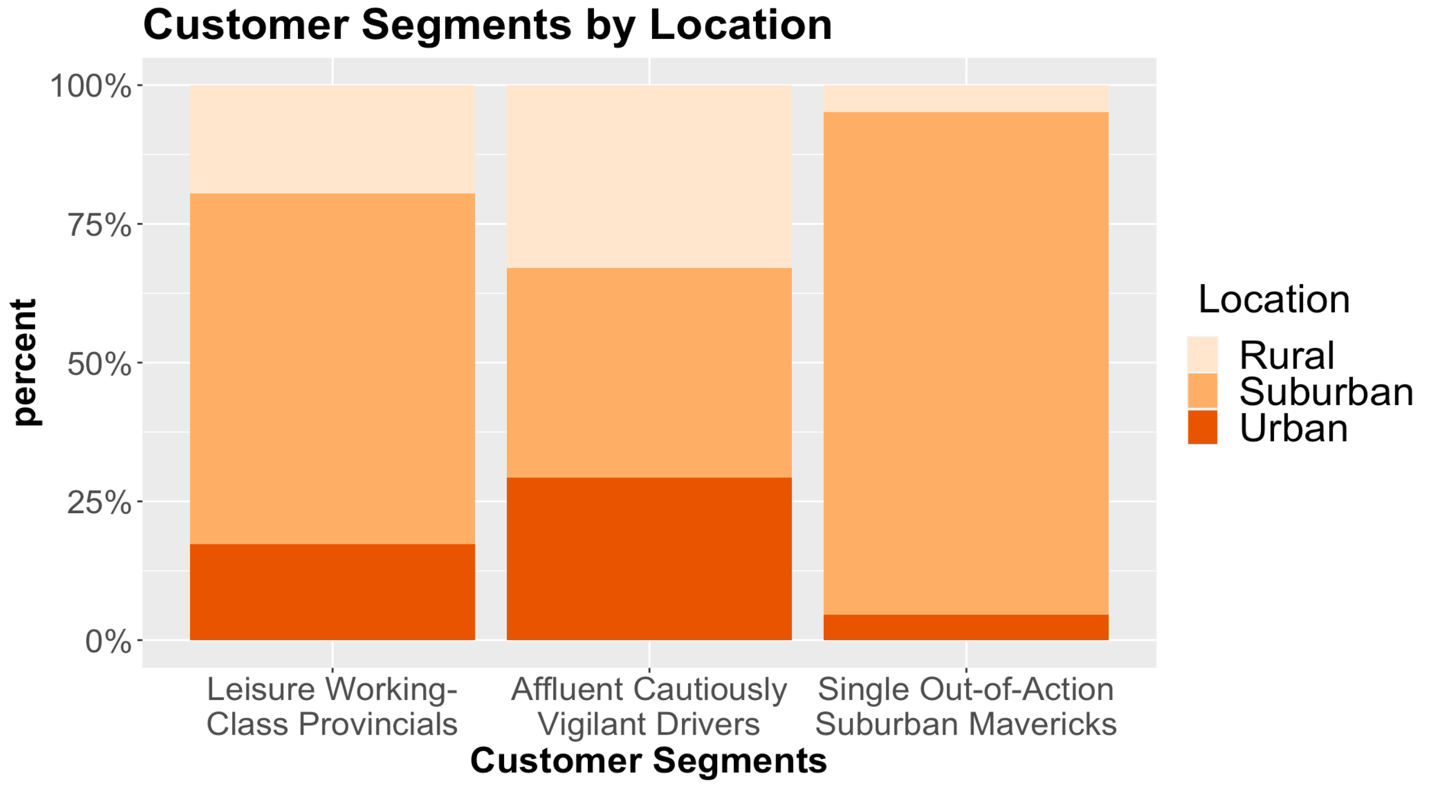


Figure 63: Customer segments by location

aCluster$Marital.Status <- factor(aCluster$Marital.Status,   
 levels = c("Single","Married", "Divorced"))  
  
ggplot(aCluster,aes( x=as.factor(Clustk3), fill = Marital.Status)) + geom\_bar(position = "fill") +  
 scale\_y\_continuous(labels = scales::percent) +   
 labs(x="Customer Segments", y= "percent", fill = "Marital Status") +  
 ggtitle("Customer Segments by Marital Status") + scale\_fill\_brewer(palette = "YlOrRd")+  
 Xlabels + MyTheme

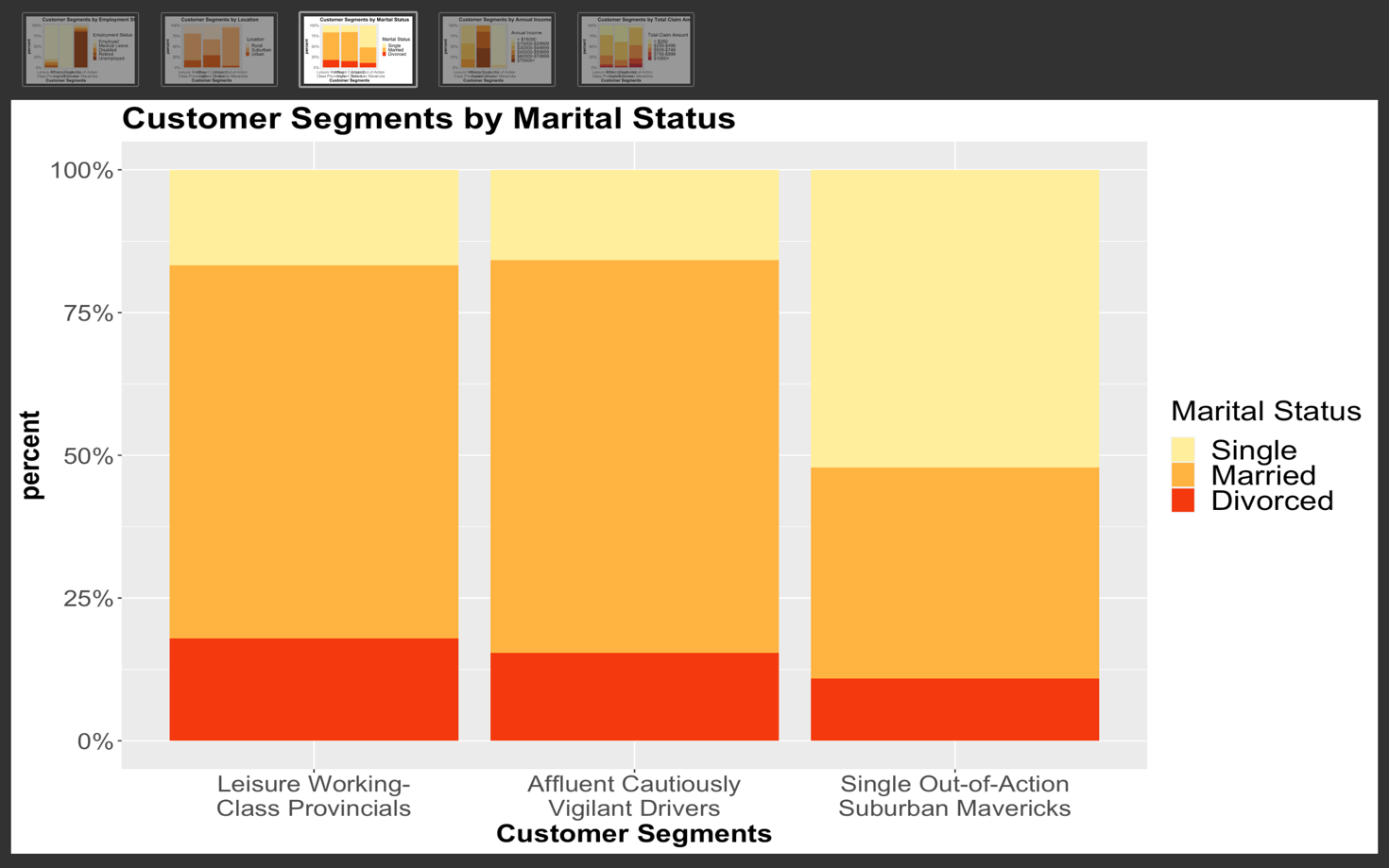


Figure 64: Customer segments by marital status

ggplot(aCluster,aes(x=as.factor(Clustk3), fill = IncomeBin)) + geom\_bar(position = "fill") +  
 scale\_y\_continuous(labels = scales::percent) +   
 labs(x="Customer Segments", y= "percent", fill = "Annual Income") +  
 ggtitle("Customer Segments by Annual Income") + scale\_fill\_brewer(palette = "YlOrBr")+  
 Xlabels + MyTheme

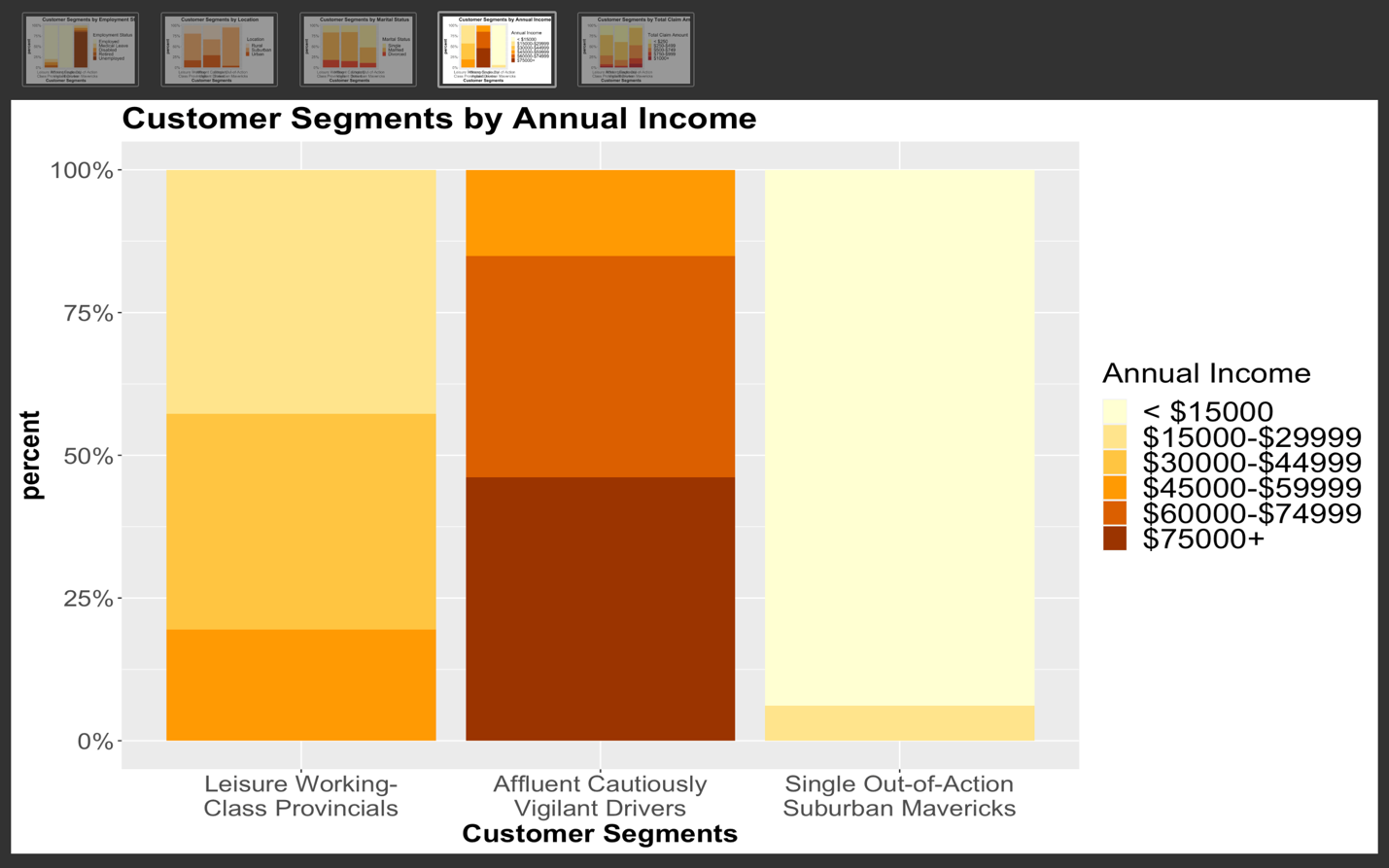


Figure 65: Customer segments by annual income

aCluster$ClaimBin <- factor(aCluster$ClaimBin, levels = c("< $250","$250-$499",  
 "$500-$749","$750-$999","$1000+"))  
  
ggplot(aCluster,aes(x=as.factor(Clustk3), fill = ClaimBin)) + geom\_bar(position = "fill") +  
 scale\_y\_continuous(labels = scales::percent) +   
 labs(x="Customer Segments", y= "percent", fill = "Total Claim Amount") +  
 ggtitle("Customer Segments by Total Claim Amount") + scale\_fill\_brewer(palette = "YlOrRd")+  
 Xlabels + MyTheme

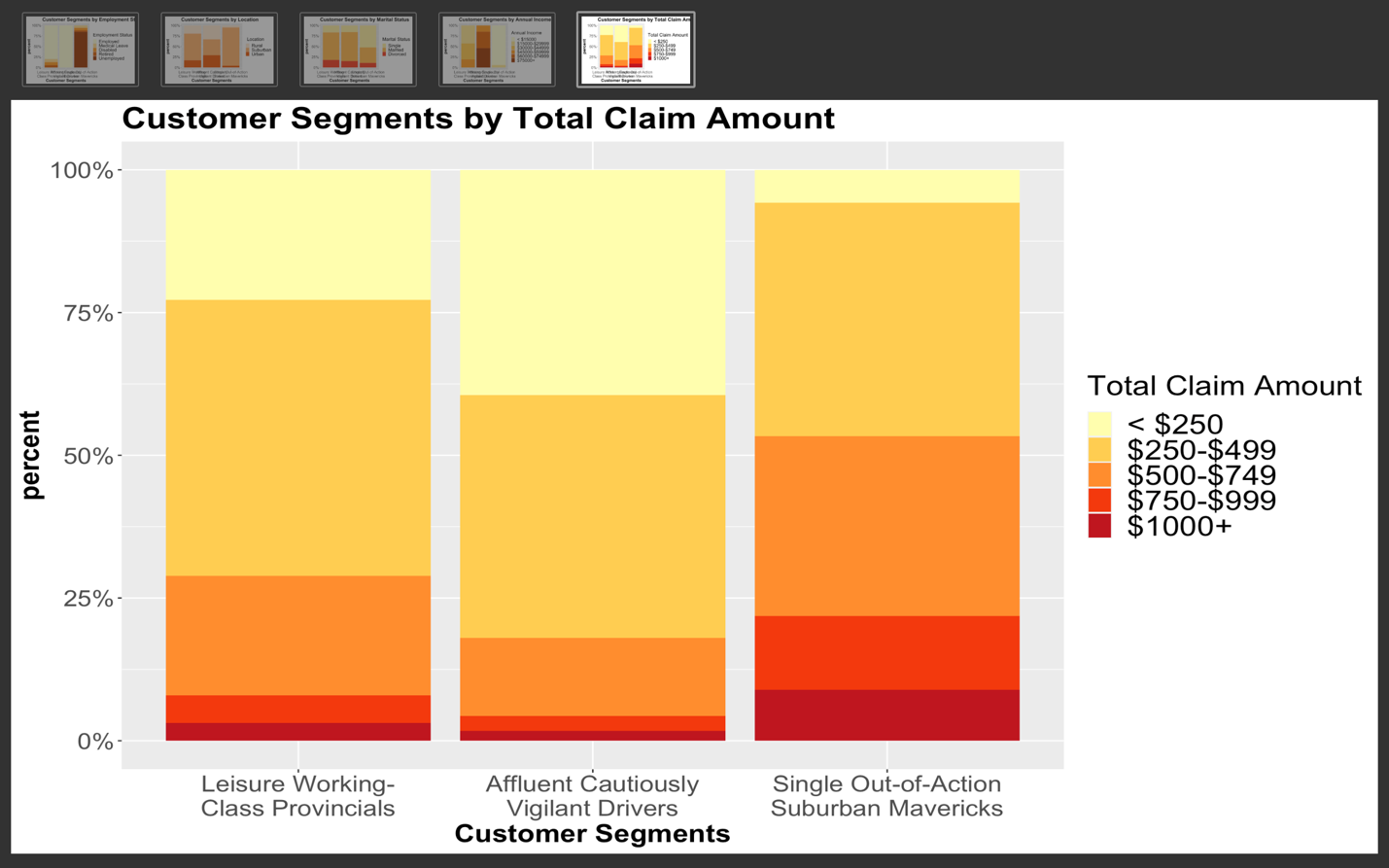


Figure 66: Customer segments by Total Claim Amount

Five key characteristics distinctly defined each customer segment: Employment Status, Location, Marital Status, Annual Income and Total Claim Amount. Below is a summary description of each segment based on the results seen in the graphs above.

CLUSTER 1: LEISURE WORKING-CLASS PROVINCIALS

* Disabled, Medical Leave, Retired, Employed
* Mostly Suburban
* Income 15k-60k

CLUSTER 2: AFFLUENT CAUTIOUSLY VIGILANT DRIVERS

* Employed
* Income: $45k or more
* Low Accident Claims

CLUSTER 3: SINGLE OUT-OF-ACTION SUBURBAN MAVERICKS

* Unemployed, Disabled, Medical Leave, Retired
* Single
* Suburban
* Income: < $30k
* High Accident Claims

In addition to providing customer segment insights to aid in designing improved renewal offers, a decision tree model can also be employed to assess the probable customer lifetime value (CLV) of prospective clients.

### Decision Tree: CLV

Load libraries

# Libraries  
library(tidyverse)  
library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

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

library(purrr)  
library(rpart)  
library(dplyr)  
library(rattle)

## Loading required package: bitops

## Rattle: A free graphical interface for data science with R.  
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.  
## Type 'rattle()' to shake, rattle, and roll your data.

##   
## Attaching package: 'rattle'

## The following object is masked from 'package:VIM':  
##   
## wine

As a client prospect tool, only variables likely to be known about customers at time of on-boarding will be included in the model. The response variable will be CLV\_Levels: Low\_CLV (< 5000), Avg\_CLV (5000 to 8000), High\_CLV (8001 to 15000) and Super\_CLV (> 15000). The predictor variables are: State, Coverage, Education, Employment Status, Gender, Income, Location Code, Marital Status, Monthly Premium Auto, Policy Type, Number of Policies, Policy, Sales Channel, Vehicle Class and Vehicle Size.

Prep the data for use with rpart decision tree algorithm. Create a train and test set using an 80/20 split.

# create data frame with the predictor and response variables  
autoDT <- auto[,c(2,5,6,8,9,10,11,12,13,17,18,19,21,23,24,29)]  
  
# autoDT 80/20 split train and test   
set.seed(123)  
sample8020 <- createDataPartition(autoDT$CLV\_Levels, p=0.80,list = FALSE)  
DTtrain8020 <- autoDT[sample8020,]  
DTtest8020 <- autoDT[-sample8020,]

Utilize caret package to run a 15-fold cross validation on the train data to assess model’s accuracy and to find best complexity parameter (cp) value for pruning. Plot the cross-validation results by accuracy and complexity parameter. Print cross validation results.

# Fit the model on the training sets  
set.seed(123)  
DTmodel8020 <- train(CLV\_Levels~., data = DTtrain8020 , method = "rpart",  
 trControl = trainControl("cv", number = 15),  
 tuneLength = 15)  
  
# Plot model accuracy vs different values of cp  
plot(DTmodel8020, main = "DECISION TREE CLV MODEL")

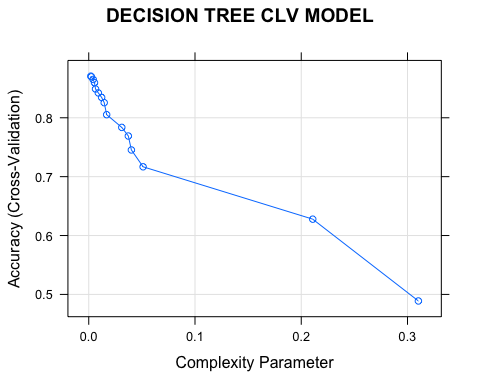


Figure 67: Decision Tree CLV Model - Accuracy & Complexity

DTmodel8020

## CART   
##   
## 7308 samples  
## 15 predictor  
## 4 classes: 'Avg\_CLV', 'High\_CLV', 'Low\_CLV', 'Super\_CLV'   
##   
## No pre-processing  
## Resampling: Cross-Validated (15 fold)   
## Summary of sample sizes: 6820, 6820, 6820, 6821, 6821, 6821, ...   
## Resampling results across tuning parameters:  
##   
## cp Accuracy Kappa   
## 0.001984564 0.8706867 0.8160401  
## 0.002425579 0.8695912 0.8139615  
## 0.004189636 0.8650788 0.8076230  
## 0.005402426 0.8596059 0.8000627  
## 0.006394708 0.8489271 0.7851143  
## 0.009040794 0.8422256 0.7756281  
## 0.012127894 0.8345565 0.7643380  
## 0.014553473 0.8256556 0.7517448  
## 0.016868798 0.8052827 0.7227941  
## 0.031091510 0.7836697 0.6925974  
## 0.037265711 0.7690154 0.6713689  
## 0.040132304 0.7453478 0.6372431  
## 0.051157663 0.7167462 0.5948860  
## 0.210804851 0.6277717 0.4568102  
## 0.310253583 0.4888537 0.2140420  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was cp = 0.001984564.

Best accuracy for the model was 87.1% with cp = 0.001984.

confusionMatrix(DTmodel8020)

## Cross-Validated (15 fold) Confusion Matrix   
##   
## (entries are percentual average cell counts across resamples)  
##   
## Reference  
## Prediction Avg\_CLV High\_CLV Low\_CLV Super\_CLV  
## Avg\_CLV 26.3 1.5 1.2 1.8  
## High\_CLV 1.1 20.0 0.0 4.8  
## Low\_CLV 1.3 0.0 36.8 0.0  
## Super\_CLV 0.1 1.2 0.0 4.0  
##   
## Accuracy (average) : 0.8707

Run the model against the test data to assess accuracy and possibility of over fitting.

# Model prediction on test  
set.seed(123)  
DTpredict8020 <- predict(DTmodel8020,DTtest8020)  
  
# Confusion Matrix on results  
table(Actual = DTtest8020$CLV\_Levels, Predict = DTpredict8020 )

## Predict  
## Actual Avg\_CLV High\_CLV Low\_CLV Super\_CLV  
## Avg\_CLV 484 21 19 0  
## High\_CLV 42 362 0 11  
## Low\_CLV 32 0 661 0  
## Super\_CLV 16 121 0 57

The best fit model had an 86.6% accuracy at predicting the test data. The model’s performance on the test data was only 0.5% less accurate the cross-validation results on the train data. The low difference in accuracy suggests low signs of overfitting. The precision values for each of the 4 levels predicted were all above 74%.

Model results on test data:

Accuracy = 0.866

Precisions: Low\_CLV = 0.959, Avg\_CLV = 0.870, High\_CLV = 0.741, Super\_CLV = 0.821

A model trained on the full data set will be used as the final model.

# Train the model on the entire dataset for a final model.  
set.seed(123)  
CLV\_model <- rpart(CLV\_Levels~.,data = autoDT, method = "class")  
  
# Plot the model  
fancyRpartPlot(CLV\_model, main = "Customer Lifetime Value Model")



Figure 68: CLV Decision Tree Visualization

# Results

The objective for all models employed were to ultimately improve FMTV Auto Insurance’s customer retention and customer lifetime value. The models utilized data from the customer responses to an FMTV marketing campaign distributing renewal offers. Customers received one of 4 renewal offers and all 4 offers had less than 25% conversion to renew. One of the 4 offers had a 0% conversion. Data mining and data analysis methods such as classification, cluster analysis and association rules mining were applied to the data for possible improvements in retention.

The Apriori Association Rules Mining (ARM) algorithm was employed to further explore the data for deeper insights into customer responses to the renewal offers. Of most interest were the customer characteristics associated with a “yes” response to the offer they received. To prepare the data for the algorithm, the following continuous variables were discretized: Income, Monthly Premium Auto, Total Claim Amounts, Months Since Policy Inception, Months Since Last Claim and Customer Lifetime Value. The discretized variables were combined with all the nominal variables (excluding Customer ID and Effective to Date) for a total of 20 variables. To explore each offer separately the data was segregated into 3 sets; one for each group of customers who received a renewal offer for either Offer 1, 2, or 3. The 3 data sets were transformed into sparse or transaction data sets resulting in 83 items for each set.

The ARM algorithm was applied to each of the data sets with the objective of finding associations to a “yes” response. The support, confidence and length of item sets were then leveraged to achieve the highest possible confidence to a maximum of 5 associations. To achieve this, Offer 1 parameters were set to a support = 0.019 for a threshold of associations occurring at least 1.9% of the time, a confidence = 0.70 for a threshold of associations appearing together at least 70% of the time and a maximum length of 5 for no more than 5 associations per rule. The right-hand side (rhs) associations were set to “Response = “Yes” to assure all resulting rules were associated with a “yes” response. This resulted in 28 rules with all lifts >3 in which the key customer characteristics associated with a “yes” response to Offer 1 were suburban retirees with incomes $15k-$30k, midsized cars and basic coverage. Offer 2 parameters were set to support =0.02, confidence = 0.5 and max length = 5, and Offer 3 parameters were set to a support =0.01, confidence = 0.10 and max length = 5. Offer 2 resulted in 16 rules with all lifts > 2 and characteristics associated with a “yes” response were suburban married women with extended coverage who acquired their policy through a sales agent. Offer 3 resulted in 6 rules with all lifts > 5 with characteristics associated with a ”yes” response as married men with customer lifetime values < $5000 and total claim amounts between $250 and $499. It should be noted that Offer 3 results may be of chance associations due to the low support (< 0.01) and low confidence (<0.21) values for all rules.

The ARM results provided clear distinctions between customers who had a “yes” response to each of the Offers. This strongly suggested that each offer appealed to different groups of customers, and implied that classification algorithms may be employed to predict which Offers would be most likely accepted by a customer based on the variables available in the data.

To determine the best classification algorithms for the job, four candidates were chosen for cross validation: Decision Tree, Naïve Bayes, Support Vector Machine (SVM) and Random Forest. Each of the algorithms were placed through a 10-fold cross validation to evaluate their accuracy and precision at predicting a response to each of the 3 offers.

To build the Decision Tree, training sets were built utilizing 60% of the customers who received Offer1 and 60% of the customers receiving Offer2 (called TrainOffer1 and TrainOffer2 in the code). Testing sets were created for both offer types with the other 40% of customers. Then, using the *rpart* package, a decision tree was created (Offer1\_Tree) using the method classification and the complexity parameter set to 0. After building the model, its prediction accuracy was verified in a table. Then, this same process was conducted on the Offer2 constituents.

With the decision trees created, the data was again subsetted to create objects Offer234 (which included all but customers receiving Offer1) and Offer 134 (which included all but customers receiving Offer2). The models were then tested against the subsets that did not include their respective training offer type. We see that 8.3% of the subset might have responded “yes” to Offer1, and 13.8% of the subset might have responded “yes” to Offer2.

For Naïve Bayes, the naivebayes package algorithm in R was applied. Through the exploratory data analysis process, it was discovered that none of the continuous variables had a Gaussian distribution. Therefore the 20 variables containing the nominal and discretized data from the ARM analysis was used for Naïve Bayes for a total of 19 nominal predictor variables and “Response” as the response variable. The Naïve Bayes algorithm parameters were set to laplace = 1, adjust = 0 and usekernel = FALSE. A 10-fold holdout cross validation was executed with 3 models trained on the responses for each of the 3 Offers: one model for each offer. The accuracy and precision results were as follows:

Offer 1 model accuracy = 0.858 with yes precision = 0.585

Offer 2 model accuracy = 0.760 with yes precision = 0.459

Offer 3 model accuracy = 0.986 with yes precision = 0.647

Only the model for Offer 3 provided an accuracy > 95%. Of key importance to the classification prediction was the precision of correctly predicting a “yes” response. In this regard the Naïve Bayes models failed to produce precisions above 65%.

The Support Vector Machine algorithm in the e1071 package for R was utilized for the SVM models. The data sets used to train the models contained 12 nominal predictor variables, 7 continuous predictor variables and “Response” as the response variable. A linear SVM model with cost = 1 was chosen for the task. A 10-fold holdout cross validation was performed using 3 models trained on the responses of each of the 3 Offers. The accuracy and precision results were as follows:

Offer 1 model accuracy = 0.870 with yes precision = 0.809

Offer 2 model accuracy = 0.758 with yes precision = 0.444

Offer 3 model accuracy = 0.973 with yes precision = 0.338

Like Naïve Bayes, only the SVM model for Offer 3 achieved an accuracy greater than 95%. In terms of correctly predicting a “yes” response, the SVM model for Offer 1 improved the precision to 80.9%, but the precisions for models on Offer 2 and 3 did not surpass 45%.

The Random Forest algorithm in the caret package for R was used for the models. The same datasets used to train the SVM models were also used to train the Random Forest models. The Random Forest data sets were split into an 80/20 ratio of train and test data sets respectively. Utilizing the cross-validation function in the caret package, the train data sets were executed through a 10-fold cross validation with a tune length of 10 to also test 10 randomly chosen mtry parameters. The mtry parameter is the number of variables randomly sampled at each node split. One of the features of the caret cross validation function is to find optimal model parameters that produce the best fit. For its Random Forest algorithm, the number of trees (ntree) and mtry are the key parameters used in the caret package cross validation function to define optimal models. The results of the cross validation on the 3 models trained on each of the offers were as follows:

Offer 1 model accuracy = 0.995 with yes precision = 0.967

Offer 2 model accuracy = 0.996 with yes precision = 0.986

Offer 3 model accuracy = 1.000 with yes precision = 1.000

The Random Forest model accuracies all exceeded 99% and the precisions for correctly predicting a “yes” response were all above 95%. The Offer 3 model had a 100% accuracy and a 100% precision. The best model parameters for each of the Offers were defined by the caret cross validation function as follows:

Offer 1 best fit model: ntree = 500 and mtry = 11

Offer 2 best fit model: ntree = 500 and mtry = 26

Offer 3 best fit model: ntree = 500 and mtry = 11

To inspect for over fitting, the best fit models were executed on the train and test data sets and the results were compared. The model executions on the train data sets all resulted in 100% accuracy. The model executions on the test data for Offer 2 and Offer 3 also resulted in 100% accuracy, while the model execution on the test data for Offer 1 resulted in 97.5% accuracy. The models similarly strong accuracies between the train and test data sets indicated low signs of overfitting.

Having achieved the best accuracies and best precisions, the Random Forest models were chosen for the task of classifying customer responses to the renew offers. To predict an appropriate redistribution of alternative offers, the models were applied to a data set of customers who originally declined their renew offer. The models predicted that only 40 out of 7826 customers would have a higher than 50% probability of a “yes” response to the redistribution of offers.

Given the high accuracy and high precision of the models’ predictions, other factors may contribute to the renew offers’ low conversion rates. To aid in improving the renew offers cluster analysis was employed to define distinct customer segments.

The unsupervised kmeans cluster analysis algorithm was employed for the task. Nine customer profile variables and ten policy profile variables were identified for the analysis. All eleven nominal variables were transformed into ordinal data types and the remaining 8 continuous variables kept their numerical data type for the algorithm. To determine an optimal number of clusters, the elbow method using the total within sum of squares in relation to the number of clusters was applied. A visual inspection for a point of an apparent bend or “elbow” crook in the resulting line graph suggested 3 clusters to be optimal. Executing the kmeans cluster analysis on the prepared data resulted in 3 distinct customer segments that were mostly defined by their employment status, marital status, location code, income and their total claim amounts. A summary of each of the customer segment key characteristics are:

CLUSTER 1: LEISURE WORKING-CLASS PROVINCIALS

· Disabled, Medical Leave, Retired, Employed

· Mostly Suburban

· Income 15k-60k

CLUSTER 2: AFFLUENT CAUTIOUSLY VIGILANT DRIVERS

· Employed

· Income: $45k or more

· Low Accident Claims

CLUSTER 3: SINGLE OUT-OF-ACTION SUBURBAN MAVERICKS

· Unemployed, Disabled, Medical Leave, Retired

· Single

· Suburban

· Income: < $30k

· High Accident Claims

In addition, a classification model can be used to define prospective clients into discrete customer lifetime values (CLV). Using the CLV distribution in the data as guidance, customers were assigned to one of four CLV levels.

* Low CLV = less than $5000
* Average CLV = $5000 to $8000
* High CLV = $8001 to $15000
* Super CLV = greater than $15000

For this task the rpart Decision Tree algorithm was chosen. The model was trained on 15 predictor variables that would normally be acquired during the policy application process: State, Coverage, Education, Employment Status, Gender, Income, Location, Marital Status, Monthly Premium Auto, Policy Type, Number of Policies, Policy, Sales Channel, Vehicle Class and Vehicle Size. The CLV levels represent the 4 level response variable.

The data was given an 80/20 split of train and test data respectively. Utilizing the caret cross validation function, a 15-fold cross validation of the train data was executed using the Decision Tree algorithm with a tune length of 15 to randomly test 15 complexity parameter (cp) values. The cp value ranges from 0 to 1 and essentially determines the pruning point of a decision tree with larger cp values equating to smaller trees. For Decision Tree models, the caret cross validation function uses the cp parameter to define an optimal model. Based on the caret cross validation results a Decision Tree with cp = 0.00198 produced the best fit model with an accuracy of 87.1%. The best fit model was executed against the test data and correctly predicted the response variables with an accuracy of 86.6%, which aligned with the cross-validation results indicating low signs of overfitting. The model’s precisions for predicting each CLV level are as follows: Low CLV = 95.9%, Avg. CLV = 87.0%, High CLV = 74.1% and Super CLV = 82.1%.

# Conclusion

After exploring the data provided by FMTV, these findings have a lot to say about the company’s customer base, its marketing efforts (as represented by the renewal offers), and its ability to retain customers and capitalize on their customer lifetime value. With FMTV’s stated business goals as “improving renewal offer responses” and “retaining customers,” steps to assist began at exploration of the data. Once completed, multiple modeling techniques were employed to identify the most significant aspects of the customer profile and to build tools that the company can use moving forward.

As stated above, the first steps were to investigate each of the attributes that made up the data set. Given that one aim was to improve renewal offer responses, it was clear that the two variables associated (Renew.Offer.Type and Response) would be important. But, by segmenting and comparing variables to one another, it became evident that Customer Lifetime Value would also be an important focus to improve the overall business goals of FMTV. After Customer Lifetime Value and Renewal Offer data had been identified as areas of interest, modeling efforts could begin.

Sparing the technical details, the first step in the modeling process was to identify how strongly each variable was associated with a “yes” response. When trying to predict future responses to renewal offers, it would be important to consider all attributes that were found to be connected to that response type. The attributes commonly associated with a yes response to Offer1 were Suburban Retirees with Incomes in the 15k to 30k range with a Midsize Car and Basic Coverage. For Offer2, those characteristics were Suburban Married Women with Extended Coverage, and these customers were usually acquired through a sales agent. And, finally, the attributes that were most associated with a “yes” response to Offer3 were Married Men with Low CLV and Total Claim Amounts ranging between 250 to 499 dollars.

Once important attributes like location, employment status, income, vehicle class, coverage type, marital status, gender, CLV, and total claim amounts, among others, had been identified as significant, modeling could begin. Without delving into the particulars of each model type, it is important for FMTV to know that multiple methodologies were employed. Ultimately, a tool was built that would allow for FMTV to analyze the characteristics of a particular customer in order to identify the optimal offer type for that individual. Unfortunately, given the low uptake rate of the existing offer types, it is perhaps unsurprising that even after redistributing the offers, the best model was not able to produce real improvements to the response rate. The model would be improved if there were more customers examples that accepted offers in order to increase the confidence in the model. Though maybe there is another reason why offer rates are low.

Given that redistributing the offers was unsuccessful, there is a strong likelihood that the underlying concern is the efficacy of the marketing materials themselves. FMTV might take this opportunity to improve its renewal offers themselves or their delivery methods. To aid in that process, customer segmentation analysis was employed (clustering) to group customers with similar profiles. What resulted were three defined segments: Leisure Working-Class Provincials, Affluent Cautiously Vigilant Drivers, and Single Out-of-Action Suburban Mavericks. More information about these groups can be found in the results section. Armed with this information, FMTV might be able to improve its offers in a more targeted manner moving forward.

As a final takeaway, FMTV has been provided with a model for predicting Customer Lifetime Value. By selecting attributes that would be available to FMTV at the time of a prospective customer’s application, this model should be useful in estimating an individual’s potential future value to the company before they even begin their policy. Additionally, this type of model (Decision Tree) allows for a conceptual visualization (Figure 68) that might be useful for those uninitiated in data science concepts.

It is the intent of this report to provide FMTV with clear actionable items to begin resolving its business questions. While work must be done on the marketing side before renewal offer responses can improve in any meaningful way, new ways of interpreting the customer base should aid in that process. Additionally, the chosen predictive model for offer responses can be viewed as iterative. This means that as the offers themselves change and improve; the model can be re-trained with new data. Lastly, predictive model for Customer Lifetime Value will empower FMTV to confidently classify prospective customers at the point of application moving forward. While the automotive insurance industry can be a challenging industry in which to operate, these new tools should give FMTV an advantage as they proceed.