
**MPS Analytics**

**Course: ALY6015 - Intermediate Analytics**

**Module 6 - Final Project Report**

**Submitted On:**

Feb 17th , 2023

**Submitted to:**

Professor : Paromita Guha

**Submitted by:**

Bhagyashri R.Kadam

Shyamala Venkatakrishnan

Xiaoge Zhang

Yuchen Zhao

# Module 2 Final Project: Proposal/Dataset Selection

## Introduction

**What is Customer Churn and Why is it Important for companies?**

Customer churn is the percentage of customers that stopped using any company's product or service during a certain time frame. We can calculate churn rate by dividing the number of customers we lost during that time period by the number of customers we had at the beginning of that time period.
For example, if the company starts the quarter with 400 customers and ends with 380, the churn rate is 5% because the company lost 5% of the total customers. Obviously, the company should aim for a churn rate that is as close to 0% as possible.
In order to achieve that, the company must know at all times what the churn rate is, and treat it as its top priority.

## Telco Customer Churn Dataset

**The Telco Customer Churn** dataset contains information about a fictional Telco company that provides home phone and Internet services to 7043 Customers in California in Q3. It Indicates which customers have left , stayed , or signed up for their services.
As part of the final project, we are interested in finding the factors which can strongly predict the churn value for Telco. Various hypothesis tests are to be used to answer the questions and a model is to be built to predict whether the customer will be churned or not given the details related to the Customer demographics , account information and the services the Customer has signed up for.

Through our Analysis , Telco can predict the behavior to retain Customers and further develop focused Customer Retention Programs.

**This data set includes information about:**

- Customers who left within the last month – the column is called **Churn**
- Services that each customer has signed up for – phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies
- Customer account information – how long they've been a customer, contract, payment method, paperless billing, monthly charges, and total charges
- Demographic info about customers – gender, age range, and if they have partners and dependents

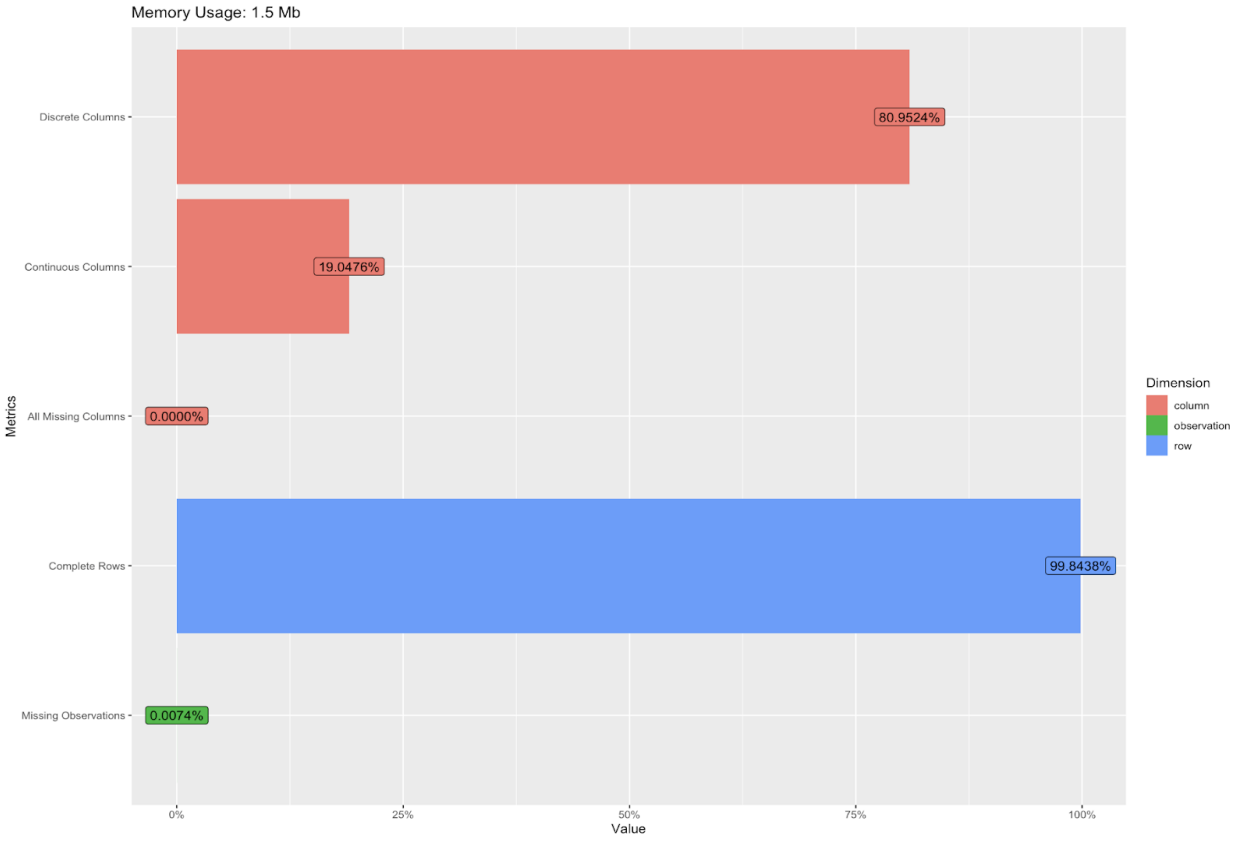
# Understanding the Dataset

This dataset was taken from the Kaggle platform and has information about the Customer data and different types of Home and Internet services they have signed up for from the Company.

It has a total of **7043 records and 21 columns**. This dataset contains both numerical and categorical types of data.

Raw Counts

| Name | Value |
|---|---|
| Rows | 7,043 |
| Columns | 21 |
| Discrete columns | 17 |
| Continuous columns | 4 |
| All missing columns | 0 |
| Missing observations | 11 |
| Complete Rows | 7,032 |
| Total observations | 147,903 |
| Memory allocation | 1.5 Mb |



**Description of the variables/features in the dataset:**

| Sr. No. | Feature | Dictionary |
|---|---|---|
| 1. | Gender | The customer's gender: Male, Female |
| 2. | Senior Citizen | Indicates if the customer is 65 or older: Yes, No |
| 3. | Partner | Indicates if the customer is married or not. |
| 4. | Dependents | Indicates if the customer lives with any dependents: Yes, No. |
| 5. | Tenure | Indicates the total amount of Tenure (In months) that the customer has been with the company by the end of the quarter |
| 6. | Phone Service | Indicates if the customer subscribes to home phone service with the company: Yes, No |
| 7. | Multiple Lines | Indicates if the customer subscribes to multiple telephone lines with the company: Yes, No |
| 8. | Internet Service | Indicates if the customer subscribes to Internet service with the company: No, DSL, Fiber Optic, Cable |
| 9. | Online Security | Indicates if the customer subscribes to an additional online security service provided by the company: Yes, No |
| 10. | Online Backup | Indicates if the customer subscribes to an additional online backup service provided by the company: Yes, No |
| 11. | Device Protection | Indicates if the customer subscribes to an additional device protection plan for their Internet equipment provided by the company: Yes, No |
| 12. | Tech Support | Indicates if the customer subscribes to an additional technical support plan from the company with reduced wait times: Yes, No |
| 13. | Streaming TV | Indicates if the customer uses their Internet service to stream television programing from a third party provider: Yes, No. The company does not charge an additional fee for this service. |

| | | |
|---|---|---|
| 14. | Streaming Movies | Indicates if the customer uses their Internet service to stream movies from a third party provider: Yes, No. The company does not charge an additional fee for this service. |
| 15. | Contract | Indicates the customer's current contract type: Month-to-Month, One Year, Two Year. |
| 16. | Paperless Billing | Indicates if the customer has chosen paperless billing: Yes, No |
| 17. | Payment Method | Indicates how the customer pays their bill: Bank Withdrawal, Credit Card, Mailed Check |
| 18. | Monthly Charges | Indicates the customer's current total monthly charge for all their services from the company. |
| 19. | Total Charges | Indicates the customer's total charges, calculated to the end of the quarter . |
| 20. | Churn | Yes = the customer left the company this quarter. No = the customer remained with the company. Directly related to Churn Value |
| 21. | Customer ID | A unique ID that identifies each customer. |

**Below are the questions we plan to answer through our Analysis:**

- What are the factors/services that would help predict the Customer Churn ?
- Do any of the Services that Customer has signed up for contribute to the Customer Churn ? (e.g -Online Security , Tech Support , Streaming movies etc.)
- Does demographic information of Customers (example - Gender , Senior Citizen , Dependents etc.) contribute to Customer churn rate ?
- Are Customers with less tenure more likely to be Churned ?
- Is the average total charge same across all the types of internet service?
- Are the churned customers independent of the contract type?

**Methods we plan to implement during our Analysis:**

- One Way Anova , Two –Way Anova
- Chi-Square
- Best Subset regression method
- Logistic Regression

# Module 4 : Final Project: Initial Analysis Report

## Data Cleaning

***Below is the structure of the Raw Dataset -***

```
> str(telco_df)
'data.frame':   7043 obs. of  21 variables:
 $ customerID      : chr  "7590-VHVEG" "5575-GNVDE" "3668-QPYBK" "7795-CFOCW" ...
 $ gender          : chr  "Female" "Male" "Male" "Male" ...
 $ SeniorCitizen   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Partner         : chr  "Yes" "No" "No" "No" ...
 $ Dependents      : chr  "No" "No" "No" "No" ...
 $ tenure          : int  1 34 2 45 2 8 22 10 28 62 ...
 $ PhoneService    : chr  "No" "Yes" "Yes" "No" ...
 $ MultipleLines   : chr  "No phone service" "No" "No" "No phone service" ...
 $ InternetService : chr  "DSL" "DSL" "DSL" "DSL" ...
 $ OnlineSecurity  : chr  "No" "Yes" "Yes" "Yes" ...
 $ OnlineBackup    : chr  "Yes" "No" "Yes" "No" ...
 $ DeviceProtection: chr  "No" "Yes" "No" "Yes" ...
 $ TechSupport     : chr  "No" "No" "No" "Yes" ...
 $ StreamingTV     : chr  "No" "No" "No" "No" ...
 $ StreamingMovies : chr  "No" "No" "No" "No" ...
 $ Contract        : chr  "Month-to-month" "One year" "Month-to-month" "One year" ...
 $ PaperlessBilling: chr  "Yes" "No" "Yes" "No" ...
 $ PaymentMethod   : chr  "Electronic check" "Mailed check" "Mailed check" "Bank transfer (automatic)" ...
 $ MonthlyCharges  : num  29.9 57 53.9 42.3 70.7 ...
 $ TotalCharges    : num  29.9 1889.5 108.2 1840.8 151.7 ...
 $ Churn           : chr  "No" "No" "Yes" "No" ...
```

***Let's check the missing values for each attribute -***

```
> #Checking data is clean?#
> colSums(is.na(telco_df)) # check & Returns the number of missing values in each column
      customerID           gender    SeniorCitizen          Partner       Dependents           tenure     PhoneService    MultipleLines 
               0                0                0                0                0                0                0                0 
 InternetService   OnlineSecurity     OnlineBackup DeviceProtection      TechSupport      StreamingTV  StreamingMovies         Contract 
               0                0                0                0                0                0                0                0 
PaperlessBilling    PaymentMethod   MonthlyCharges     TotalCharges            Churn 
               0                0                0               11                0 
>
```

### Observations:

From the above output , we can observe that there are a total **11** missing values for the TotalCharges feature. However , there is no missing value for any other attribute of the dataset.

***Now , let's handle the missing records for TotalCharges by replacing the missing values with their respective Mean and check if any missing values again in the dataframe -***

```
> sum(is.na(telco_df)) # Counts missing values in entire data frame
[1] 0
```

Also ,As we can see from the dataset , column – **Customer ID** is not required or significant for our analysis , hence we would drop it .

Further , some of the features in the dataset have values - **Yes & No** , so we would Recode the variables - **Churn , Partner , Dependents , PhoneService & PaperlessBilling** to **1 & 0** respectively and make them numeric features.

**Below is the structure of the final Dataset after above changes :**

```
> str(telco_df)
'data.frame':   7043 obs. of  20 variables:
 $ gender          : chr  "Female" "Male" "Male" "Male" ...
 $ SeniorCitizen   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Partner         : num  1 0 0 0 0 0 0 0 1 0 ...
 $ Dependents      : num  0 0 0 0 0 0 1 0 0 1 ...
 $ tenure          : num  1 34 2 45 2 8 22 10 28 62 ...
 $ PhoneService    : num  0 1 1 0 1 1 1 0 1 1 ...
 $ MultipleLines   : chr  "No phone service" "No" "No" "No phone service" ...
 $ InternetService : chr  "DSL" "DSL" "DSL" "DSL" ...
 $ OnlineSecurity  : chr  "No" "Yes" "Yes" "Yes" ...
 $ OnlineBackup    : chr  "Yes" "No" "Yes" "No" ...
 $ DeviceProtection: chr  "No" "Yes" "No" "Yes" ...
 $ TechSupport     : chr  "No" "No" "No" "Yes" ...
 $ StreamingTV     : chr  "No" "No" "No" "No" ...
 $ StreamingMovies : chr  "No" "No" "No" "No" ...
 $ Contract        : chr  "Month-to-month" "One year" "Month-to-month" "One year" ...
 $ PaperlessBilling: num  1 0 1 0 1 1 1 0 1 0 ...
 $ PaymentMethod   : chr  "Electronic check" "Mailed check" "Mailed check" "Bank transfer (automatic)"
 $ MonthlyCharges  : num  29.9 57 53.9 42.3 70.7 ...
 $ TotalCharges    : num  29.9 1889.5 108.2 1840.8 151.7 ...
 $ Churn           : num  0 0 1 0 1 1 0 0 1 0 ...
```

## Descriptive Characteristics of the Dataset

Descriptive Statistics (N, mean , median , Standard deviation, min , max , range etc.) for the entire dataset using the Describe function :

Descriptive statistics summary of the Telco Customer Churn Dataset

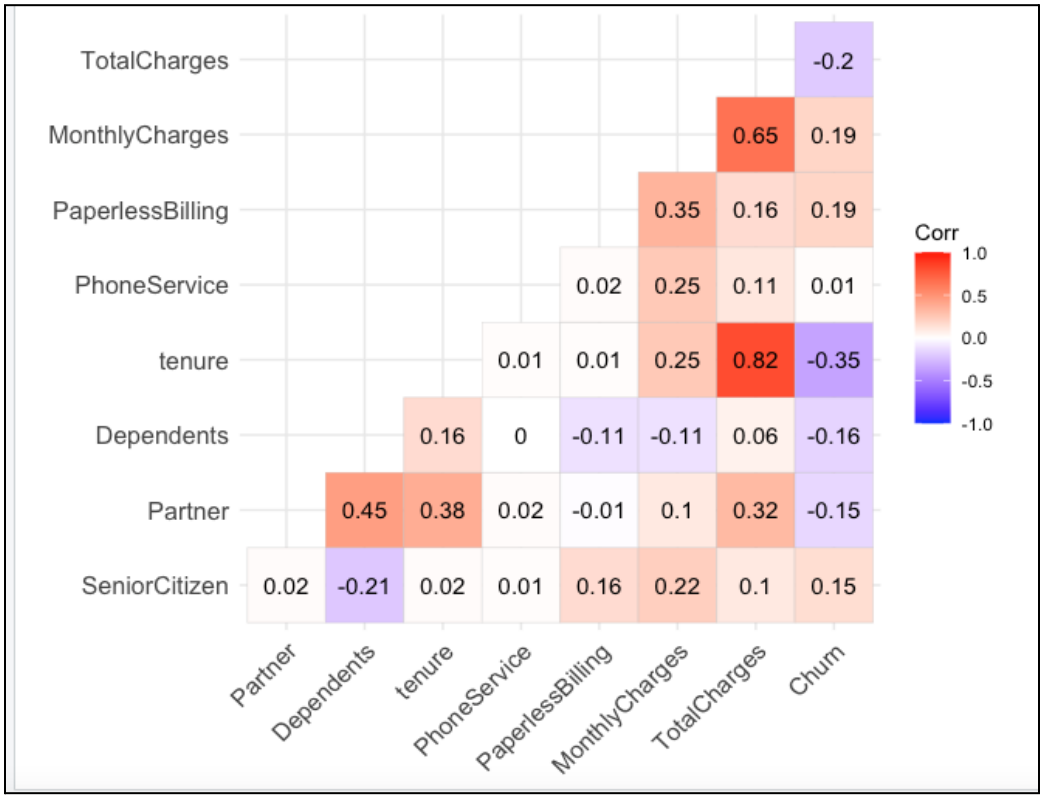
| | vars | n | mean | sd | median | trimmed | mad | min | max | range | skew | kurtosis | se |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| gender* | 1 | 7043 | 1.5047565 | 0.5000129 | 2.00 | 1.505945e+00 | 0.00000 | 1.00 | 2.00 | 1.0 | -0.01902279 | -1.9999220 | 0.005958025 |
| SeniorCitizen | 2 | 7043 | 0.1621468 | 0.3686116 | 0.00 | 7.772848e-02 | 0.00000 | 0.00 | 1.00 | 1.0 | 1.83285177 | 1.3595387 | 0.004392281 |
| Partner | 3 | 7043 | 0.4830328 | 0.4997475 | 0.00 | 4.787933e-01 | 0.00000 | 0.00 | 1.00 | 1.0 | 0.06789345 | -1.9956738 | 0.005954863 |
| Dependents | 4 | 7043 | 0.2995882 | 0.4581102 | 0.00 | 2.495120e-01 | 0.00000 | 0.00 | 1.00 | 1.0 | 0.87482582 | -1.2348551 | 0.005458723 |
| tenure | 5 | 7043 | 32.3711487 | 24.5594810 | 29.00 | 3.142786e+01 | 32.61720 | 0.00 | 72.00 | 72.0 | 0.23943773 | -1.3876966 | 0.292644483 |
| PhoneService | 6 | 7043 | 0.9031663 | 0.2957522 | 1.00 | 1.000000e+00 | 0.00000 | 0.00 | 1.00 | 1.0 | -2.72599140 | 5.4318004 | 0.003524108 |
| MultipleLines* | 7 | 7043 | 1.9405083 | 0.9485540 | 2.00 | 1.925643e+00 | 1.48260 | 1.00 | 3.00 | 2.0 | 0.11866889 | -1.8782148 | 0.011302727 |
| InternetService* | 8 | 7043 | 1.8729235 | 0.7377963 | 2.00 | 1.841171e+00 | 1.48260 | 1.00 | 3.00 | 2.0 | 0.20533596 | -1.1460705 | 0.008791392 |
| OnlineSecurity* | 9 | 7043 | 1.7900043 | 0.8598475 | 2.00 | 1.737533e+00 | 1.48260 | 1.00 | 3.00 | 2.0 | 0.41680747 | -1.5211578 | 0.010245723 |
| OnlineBackup* | 10 | 7043 | 1.9064319 | 0.8801625 | 2.00 | 1.883052e+00 | 1.48260 | 1.00 | 3.00 | 2.0 | 0.18285247 | -1.6849210 | 0.010487790 |
| DeviceProtection* | 11 | 7043 | 1.9044441 | 0.8799489 | 2.00 | 1.880568e+00 | 1.48260 | 1.00 | 3.00 | 2.0 | 0.18676763 | -1.6832382 | 0.010485245 |
| TechSupport* | 12 | 7043 | 1.7971035 | 0.8615506 | 2.00 | 1.746406e+00 | 1.48260 | 1.00 | 3.00 | 2.0 | 0.40219313 | -1.5351795 | 0.010266016 |
| StreamingTV* | 13 | 7043 | 1.9853756 | 0.8850019 | 2.00 | 1.981721e+00 | 1.48260 | 1.00 | 3.00 | 2.0 | 0.02847350 | -1.7228219 | 0.010545455 |
| StreamingMovies* | 14 | 7043 | 1.9924748 | 0.8850907 | 2.00 | 1.990594e+00 | 1.48260 | 1.00 | 3.00 | 2.0 | 0.01465041 | -1.7235138 | 0.010546514 |
| Contract* | 15 | 7043 | 1.6904728 | 0.8337552 | 1.00 | 1.613132e+00 | 0.00000 | 1.00 | 3.00 | 2.0 | 0.63069036 | -1.2726490 | 0.009934814 |
| PaperlessBilling | 16 | 7043 | 0.5922192 | 0.4914569 | 1.00 | 6.152618e-01 | 0.00000 | 0.00 | 1.00 | 1.0 | -0.37523586 | -1.8594620 | 0.005856075 |
| PaymentMethod* | 17 | 7043 | 2.5743291 | 1.0681040 | 3.00 | 2.592902e+00 | 1.48260 | 1.00 | 4.00 | 3.0 | -0.17005696 | -1.2122656 | 0.012727254 |
| MonthlyCharges | 18 | 7043 | 64.7616925 | 30.0900471 | 70.35 | 6.496565e+01 | 35.65653 | 18.25 | 118.75 | 100.5 | -0.22043051 | -1.2577140 | 0.358545291 |
| TotalCharges | 19 | 7043 | 2283.3004402 | 2265.0002578 | 1400.55 | 1.970375e+03 | 1812.40437 | 18.80 | 8684.80 | 8666.0 | 0.96198413 | -0.2289450 | 26.989162711 |
| Churn | 20 | 7043 | 0.2653699 | 0.4415613 | 0.00 | 2.067436e-01 | 0.00000 | 0.00 | 1.00 | 1.0 | 1.06257868 | -0.8710502 | 0.005261531 |

**We can understand the following observations from above –**

1. The mean of the Monthly Charges is $64.76, the minimum value is $18.25 and maximum is $118.75 with a standard deviation of 30.09
2. The mean of the Monthly Charges is $2283.30, the minimum value is $18.80 and maximum is $8684.80 with a standard deviation of 2265.
3. The mean Churn value is 0.26 with standard deviation of 0.44
4. The mean Tenure is 32.37 , the minimum value is 0 and maximum is 72 months with a standard deviation of 24.55

## Correlation Plot & Check for MultiCollinearity

Lets understand the correlation between all the numeric features of this dataset with the help of correlation plot & Matrix




## Correlation Matrix

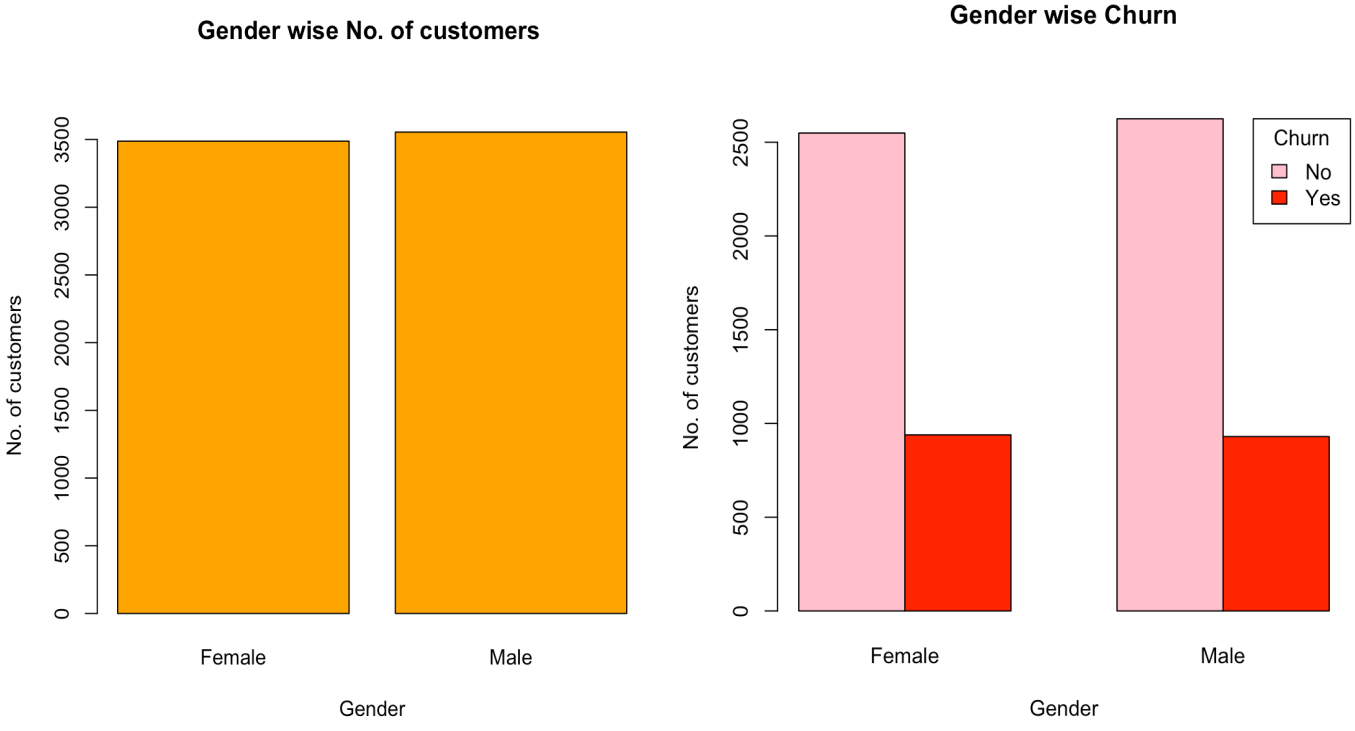
| | SeniorCitizen | Partner | Dependents | tenure | PhoneService | PaperlessBilling | MonthlyCharges | TotalCharges | Churn |
|---|---|---|---|---|---|---|---|---|---|
| SeniorCitizen | 1.000000000 | 0.01647866 | -0.211185088 | 0.016566878 | 0.008576401 | 0.156529559 | 0.22017334 | 0.10239510 | 0.15088933 |
| Partner | 0.016478658 | 1.00000000 | 0.452676283 | 0.379697461 | 0.017705663 | -0.014876622 | 0.09684794 | 0.31881156 | -0.15044754 |
| Dependents | -0.211185088 | 0.45267628 | 1.000000000 | 0.159712331 | -0.001761679 | -0.111377229 | -0.11389023 | 0.06453492 | -0.16422140 |
| tenure | 0.016566878 | 0.37969746 | 0.159712331 | 1.000000000 | 0.008448207 | 0.006152482 | 0.24789986 | 0.82475732 | -0.35222867 |
| PhoneService | 0.008576401 | 0.01770566 | -0.001761679 | 0.008448207 | 1.000000000 | 0.016504806 | 0.24739796 | 0.11285074 | 0.01194198 |
| PaperlessBilling | 0.156529559 | -0.01487662 | -0.111377229 | 0.006152482 | 0.016504806 | 1.000000000 | 0.35214997 | 0.15767630 | 0.19182533 |
| MonthlyCharges | 0.220173339 | 0.09684794 | -0.113890230 | 0.247899856 | 0.247397963 | 0.352149968 | 1.00000000 | 0.65046804 | 0.19335642 |
| TotalCharges | 0.102395103 | 0.31881156 | 0.064534917 | 0.824757316 | 0.112850743 | 0.157676301 | 0.65046804 | 1.00000000 | -0.19942771 |
| Churn | 0.150889328 | -0.15044754 | -0.164221402 | -0.352228670 | 0.011941980 | 0.191825332 | 0.19335642 | -0.19942771 | 1.00000000 |

## Observations:

1. From the above Correlation plot & Matrix , we can observe that the Target Variable – Churn is positively correlated with variables – Monthly Charges with 0.193 correlation value followed by Paperless Billing with value 0.191
2. The target variable - Churn is negatively correlated with feature - Tenure with value of -0.35
3. Further, we can also see that the **Monthly charges** feature is highly correlated with the T**otal Charges** with 0.65 correlation value.
4. Also , **Tenure** feature is highly correlated with the **Total Charges** with 0.82 correlation value.

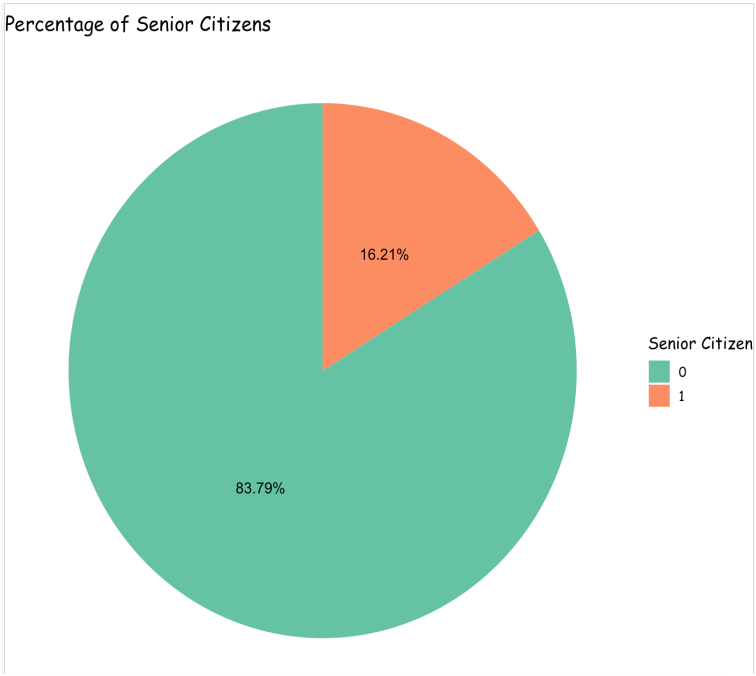
## Exploratory Data Analysis:

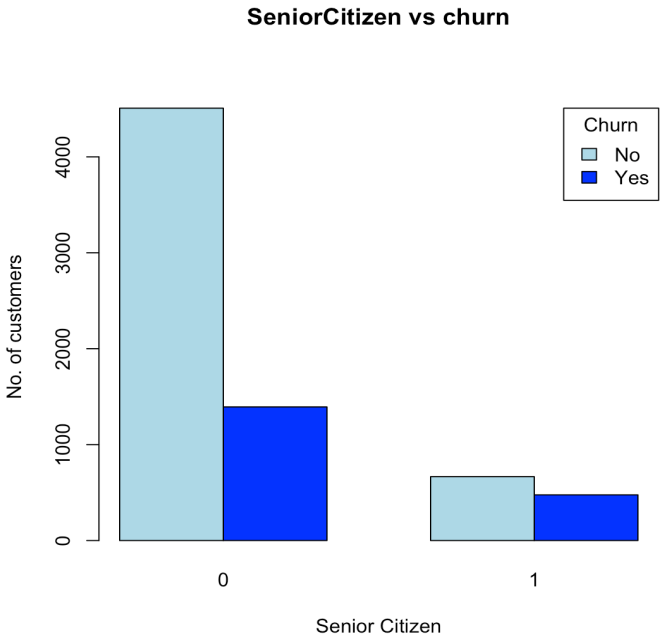
### 1. Does the gender attribute of a customer have an impact on the customer churn?



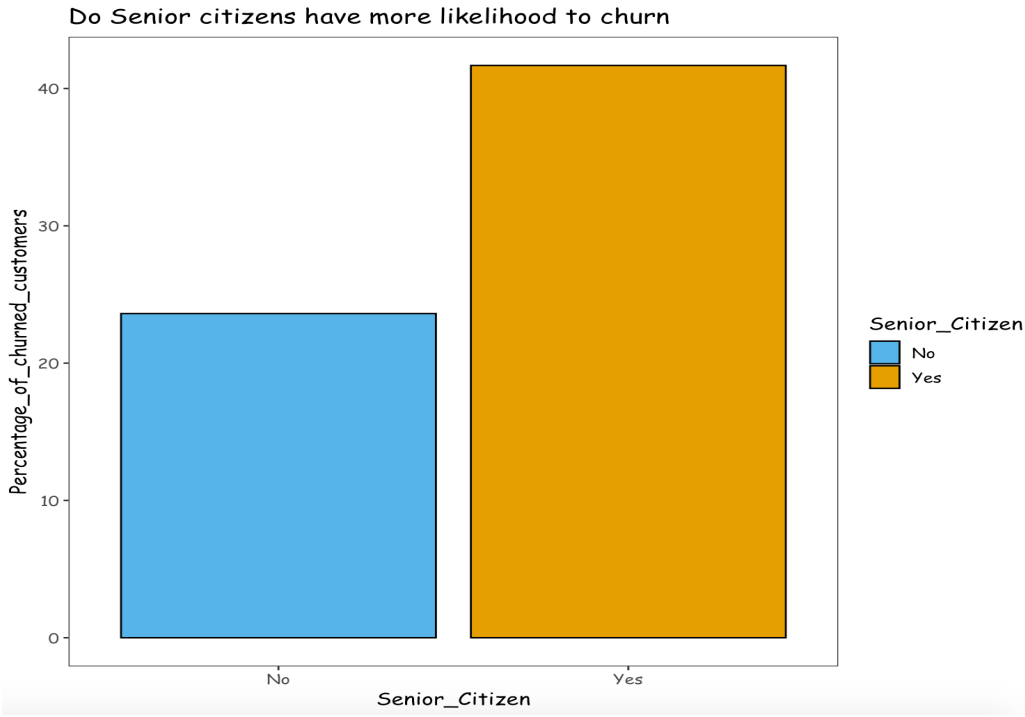

From the above graphs, we are able to infer that the churn rate is the same for both male and female customers and there is no impact of gender found in the churn rate.

## 2. Do senior citizens have more likelihood to churn?



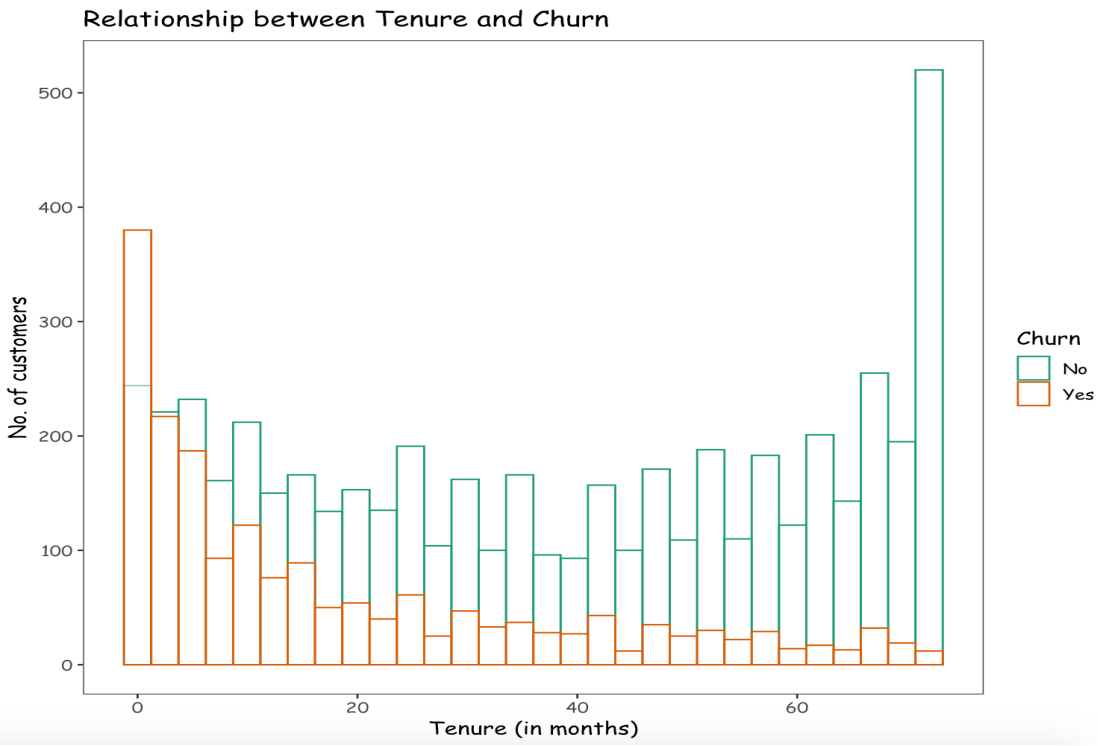




From the above graphs, it can be observed that there are a very less number of senior citizens signed up as a customer than the young/middle aged people. With respect to customer churn rate, the percentage of senior citizens churned is greater than the percentage of young/middle aged people.



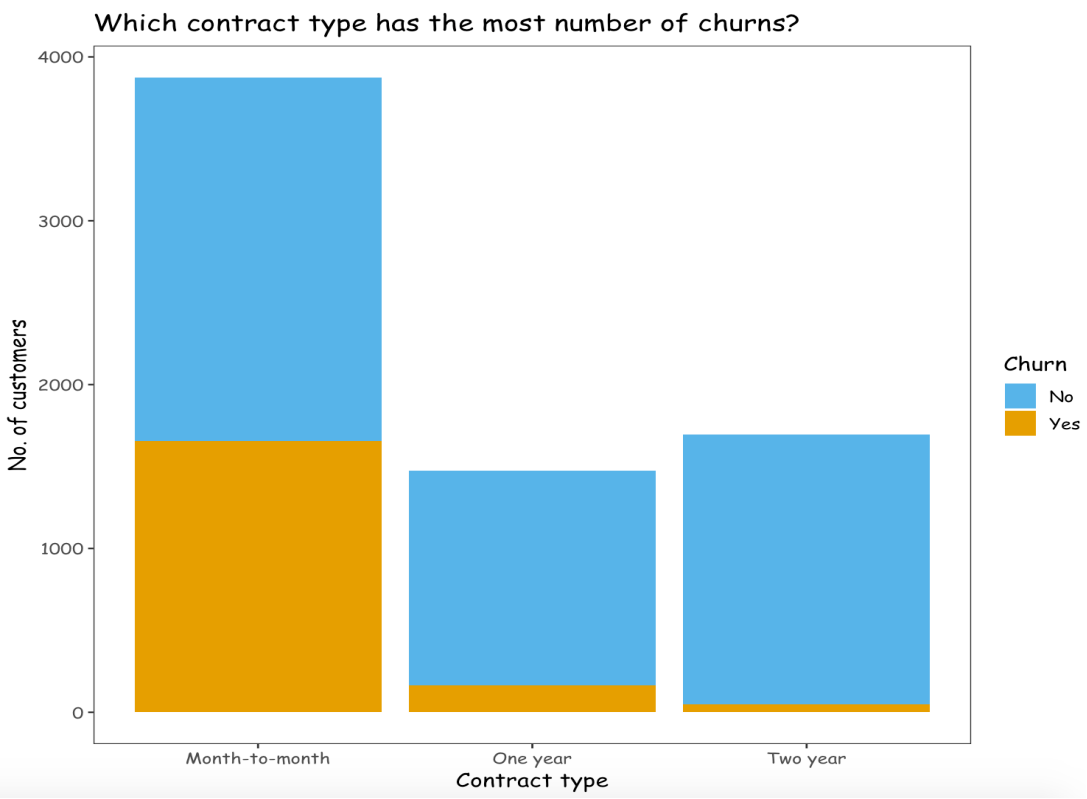
Thus, it can be inferred that the company is only able to retain around 60% of the Senior citizens and the remaining 40% of the Senior citizens have left the company. Compared to young/middle aged people, senior citizens have more likelihood to churn.

### 3. Are the customers having a longer tenure with the company less likely to churn?

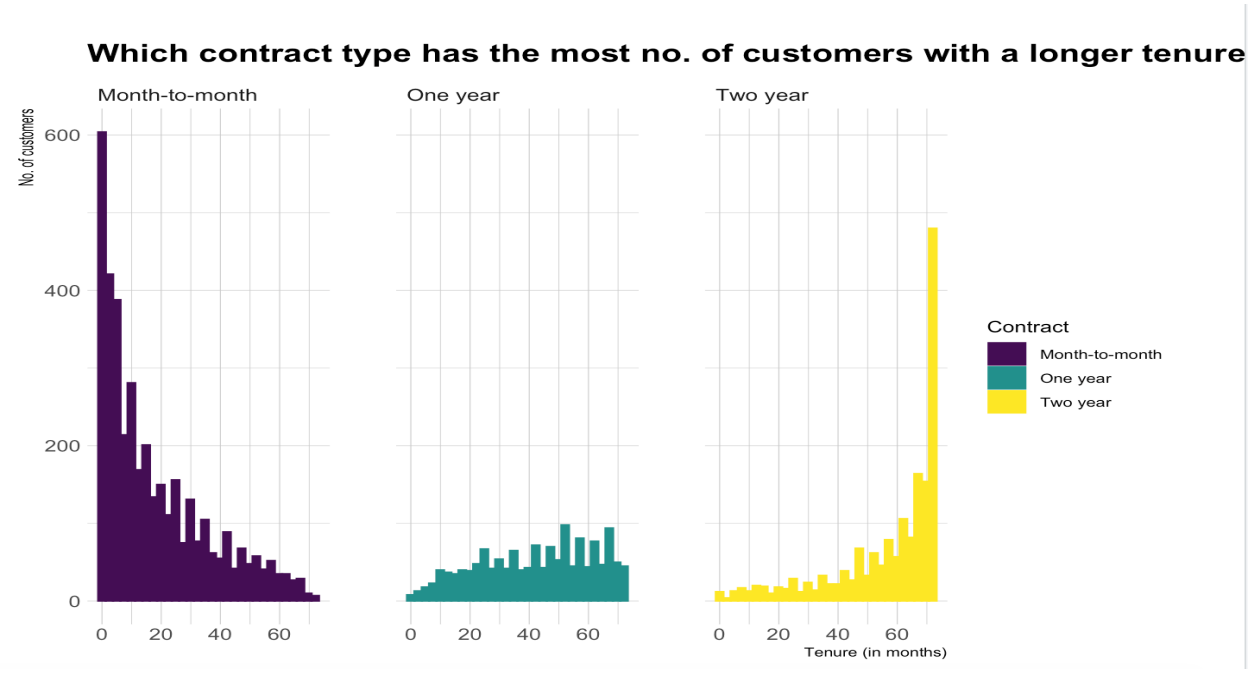



From the above graph, it can be inferred that as a customer has a longer tenure with the company, there is a reduction in the likelihood to churn.

**4. Which contract type has the most number of churns?**



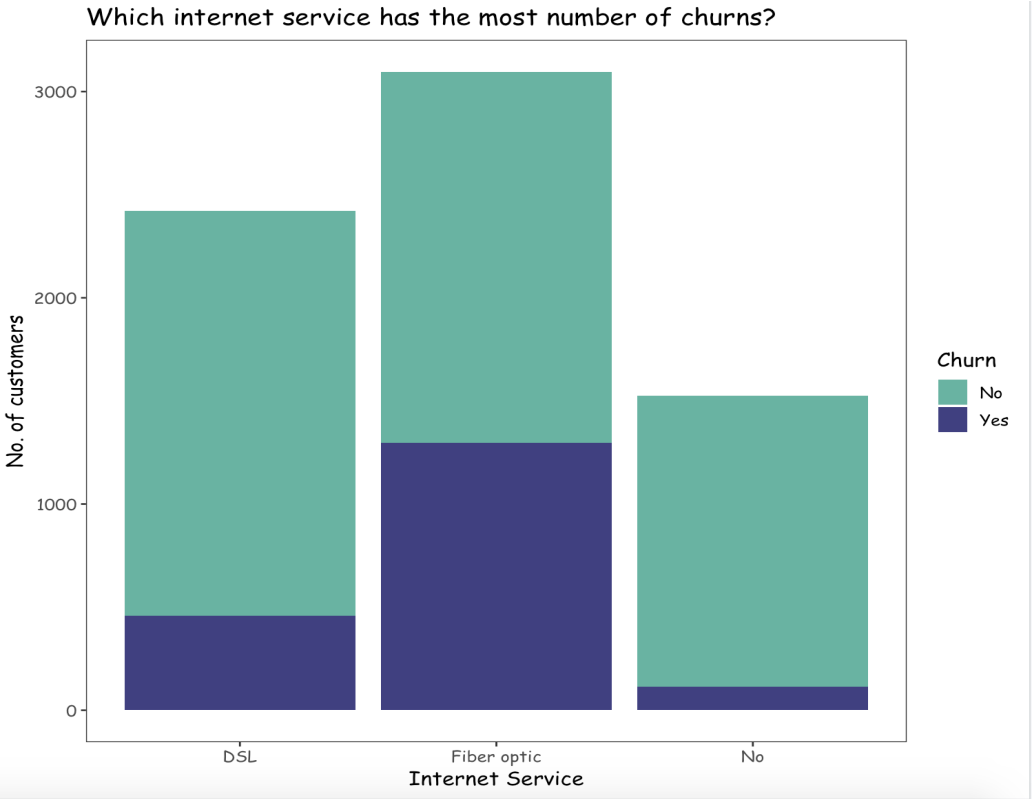

The customers with a contract type of Month-to-month are more likely to churn than the customers with other contract types.




The number of customers having a Two-year contract type with a longer tenure period are comparatively greater and hence two-year contract type can result in lesser number of customers getting churned.

The number of customers having a Month-to-month contract type with a longer tenure period are comparatively lesser and hence Month-to-month contract type can result in more number of customers getting churned.

**5. Which Internet service contributes to the maximum number of churned customers?**




From the above graph, it can be observed that the customers opting for an internet service type of Fiber optic, are more likely to churn than with other internet services.

## Methods - Chi Square (test of independence)

### 1. Are the churned customers independent of the contract type?

**Null Hypothesis:** No. of churned customers are independent of the contract type attribute.

**Alternate Hypothesis:** No. of churned customers are dependent on the contract type attribute.

alpha <- 0.05

Creating a table to get the number of customers for each contract type and churn attribute.

```
> table(telco_df$Contract,telco_df$Churn)

                  No  Yes
  Month-to-month 2220 1655
  One year       1307  166
  Two year       1647   48
>
```

Creating a matrix to hold the rows of the above values:

```
r1 <- c(2220,1655)
r2 <- c(1307,166)
r3 <- c(1647,48)

#State the number of rows for the matrix
rows <- 3

#Create a matrix from the rows
mtrx <- matrix(c(r1,r2,r3), nrow = rows,byrow = TRUE)

#Name the rows and columns
rownames(mtrx) <- c("Month-to-month","One year","Two year")
colnames(mtrx) <- c("No","Yes")
```

The chisq.test() function can be used to apply the test of independence:

```
> result <- chisq.test(mtrx)
> result

        Pearson's Chi-squared test

data:  mtrx
X-squared = 1184.6, df = 2, p-value < 2.2e-16
```

**Conclusion** : The p-value is lesser than the significance level 0.05, thus we can reject the null hypothesis and conclude that the number of churned customers are dependent on the contract type attribute.

**2. Churn & PhoneService**

H0: There is no relationship between PhoneService and churn
H1: There is a relationship between PhoneService and churn
(significant level of 0.05)

```
> chisq.test(phoneServiceChurn)

        Pearson's Chi-squared test with Yates' continuity correction

data:  phoneServiceChurn
X-squared = 0.91503, df = 1, p-value = 0.3388
```

Since the p-value (0.3388) is greater than the significance level of 0.05, we do not reject H0. There is not sufficient evidence that supports there is a relationship between PhoneService and churn.

**3. Churn & Gender**

H0: There is no relationship between gender and churn
H1: There is a relationship between gender and churn
(significant level of 0.05)

```
> chisq.test(genderChurn)

        Pearson's Chi-squared test with Yates' continuity correction

data:  genderChurn
X-squared = 0.48408, df = 1, p-value = 0.4866
```

Since the p-value (0.4866) is greater than the significance level of 0.05, we do not reject H0. There is not sufficient evidence that supports there is a relationship between gender and churn.

**4. Churn & Partner**

H0: There is no relationship between partner and churn

H1: There is a relationship between partner and churn

(significant level of 0.05)

```
> chisq.test(partnerChurn)

        Pearson's Chi-squared test with Yates' continuity correction

data:  partnerChurn
X-squared = 165.85, df = 1, p-value < 2.2e-16
```

Since the p-value (0.000) is less than the significance level of 0.05, we reject H0. There is sufficient evidence that supports there is a relationship between partner and churn.

**5. Churn & Tenure**

H0: There is no difference in the duration of tenure.
H1: There is a difference in the duration of tenure.
(significant level of 0.05)

```
> chisq.test(tenureChurn)

        Pearson's Chi-squared test with Yates' continuity correction

data:  tenureChurn
X-squared = 348.71, df = 1, p-value < 2.2e-16
```

Since the p-value (0.000) is less than the significance level of 0.05, we reject H0. There is a difference in the duration of tenure. The duration of tenure is less than the mean that the customer churn is 38%, and the duration of tenure is more than the mean that the customer churn is 18.1%. So Customers with less tenure are more likely to be Churned.

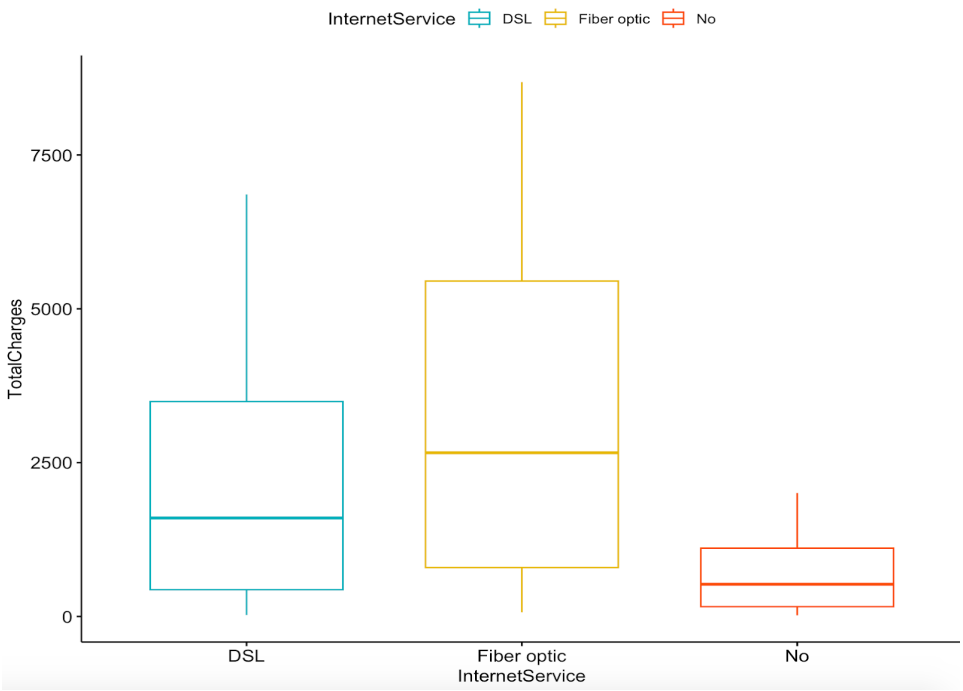
## One-way ANOVA

1. **Is the average total charge same across all the types of internet service?**

**Null Hypothesis:** The mean total charge is the same for all types of internet service.

**Alternate Hypothesis:** The mean total charge is different for all types of internet service.

**alpha <- 0.05**




Forming a data frame to hold the total chargers for each type of internet service.

```r
dsl <- sqldf("select * from telco_df where InternetService = 'DSL'")

fo <- sqldf("select * from telco_df where InternetService = 'Fiber optic'")

no <- sqldf("select * from telco_df where InternetService = 'No'")

dsl_df <- data.frame('total_charges' = dsl$TotalCharges,
                                'Internet_Service' = rep('DSL',nrow(dsl)),
                                stringsAsFactors = FALSE)

fo_df <- data.frame('total_charges' = fo$TotalCharges,
                    'Internet_Service' = rep('Fiber optic',nrow(fo)),
                    stringsAsFactors = FALSE)

no_df <- data.frame('total_charges' = no$TotalCharges,
                    'Internet_Service' = rep('No',nrow(no)),
                    stringsAsFactors = FALSE)


#Combine the dataframes into one
df <- rbind(dsl_df,fo_df,no_df)

df$Internet_Service <- as.factor(df$Internet_Service)
```

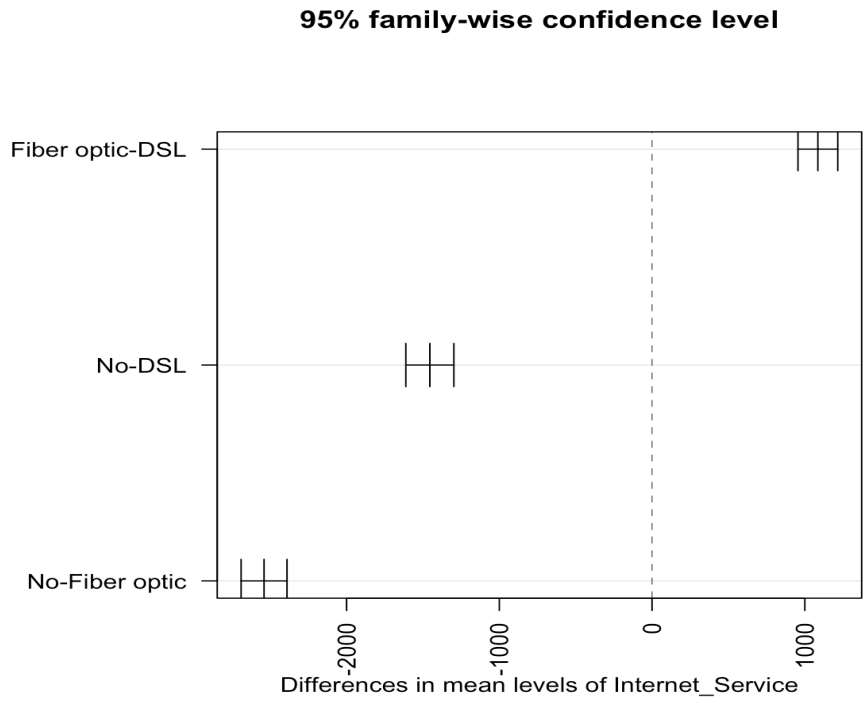
Running the one-way ANOVA test using aov() function:

```
> anova <- aov(total_charges ~ Internet_Service,data = df)
> #View the model summary
> summary(anova)
                   Df    Sum Sq   Mean Sq F value Pr(>F)
Internet_Service    2 6.676e+09 3.338e+09   796.7 <2e-16 ***
Residuals        7029 2.945e+10 4.190e+06
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
11 observations deleted due to missingness
>
```

The p-value is lesser than the significance level 0.05, thus we can reject the null hypothesis and conclude that the average total charge is different for all types of internet service.

Tukey test can be used to identify the pair of internet services which have a significantly different mean.

```
> TukeyHSD(anova)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = total_charges ~ Internet_Service, data = df)

$Internet_Service
                     diff       lwr       upr p adj
Fiber optic-DSL  1085.515   955.2578  1215.773     0
No-DSL          -1454.569 -1611.6610 -1297.477     0
No-Fiber optic  -2540.084 -2690.3665 -2389.802     0
```




From the above graph, we are able to observe that Fiber optic - DSL internet service pair has a significant mean difference than the remaining pairs.

## 2. Is the mean total charges for each contract type the same?

**Null Hypothesis:** The mean total charges for each contract type are the same.

**Alternate Hypothesis:** The mean total charges for at least one contract type is different.

alpha <- 0.05

Forming a data frame to hold the total chargers for each type of contract.

```
month_to_month <- telco_df[telco_df$Contract == 'Month-to-month', ]
one_year <- telco_df[telco_df$Contract == 'One year', ]
two_year <- telco_df[telco_df$Contract == 'Two year', ]

month_to_month_df <- data.frame('total_charges' = month_to_month$TotalCharges,
                                'Contract' = rep('Month-to-month', nrow(month_to_month)),
                                stringsAsFactors = FALSE)

one_year_df <- data.frame('total_charges' = one_year$TotalCharges,
                          'Contract' = rep('One year', nrow(one_year)),
                          stringsAsFactors = FALSE)

two_year_df <- data.frame('total_charges' = two_year$TotalCharges,
                          'Contract' = rep('Two year', nrow(two_year)),
                          stringsAsFactors = FALSE)

# Combine the dataframes into one
df <- rbind(month_to_month_df, one_year_df, two_year_df)
df$Contract <- as.factor(df$Contract)
```

Running the one-way ANOVA test using aov() function:

```
> # Run the ANOVA test
> anova <- aov(total_charges ~ Contract, data = df)
> # View the model summary
> summary(anova)
              Df    Sum Sq   Mean Sq F value Pr(>F)
Contract       2 7.569e+09 3.784e+09   932.9 <2e-16 ***
Residuals   7040 2.856e+10 4.057e+06
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> # Save summary to an object
> a.summary <- summary(anova)
> # Determine if we should reject the null hypothesis
> p.value <- a.summary[[1]]["Pr(>F)"]
> #Determine if we should reject the null hypothesis
> ifelse(p.value > alpha, "Fail to reject the null hypothesis",
+        "Reject the null hypothesis")
            Pr(>F)
Contract    "Reject the null hypothesis"
Residuals   NA
```

The p-value is lesser than the significance level 0.05, thus we can reject the null hypothesis and conclude that the average total charge is different for all types of contract.

### 3. Is the mean total charges for each PaymentMethod are the same?

**Null Hypothesis:** There is no difference in the mean Total Charges for different Payment Methods.

**Alternate Hypothesis:** There is a difference in the mean Total Charges for different Payment Methods.

alpha <- 0.05

Forming a data frame to hold the total chargers for each type of contract.

```
electronic_check <- telco_df[telco_df$PaymentMethod == 'Electronic check', ]
mailed_check <- telco_df[telco_df$PaymentMethod == 'Mailed check', ]
bank_transfer <- telco_df[telco_df$PaymentMethod == 'Bank transfer (automatic)', ]
credit_card <- telco_df[telco_df$PaymentMethod == 'Credit card (automatic)', ]

electronic_check_df <- data.frame('total_charges' = electronic_check$TotalCharges,
                                  'PaymentMethod' = rep('Electronic check', nrow(electronic_check)),
                                  stringsAsFactors = FALSE)

mailed_check_df <- data.frame('total_charges' = mailed_check$TotalCharges,
                              'PaymentMethod' = rep('Mailed check', nrow(mailed_check)),
                              stringsAsFactors = FALSE)

bank_transfer_df <- data.frame('total_charges' = bank_transfer$TotalCharges,
                               'PaymentMethod' = rep('Bank transfer (automatic)', nrow(bank_transfer)),
                               stringsAsFactors = FALSE)

credit_card_df <- data.frame('total_charges' = credit_card$TotalCharges,
                             'PaymentMethod' = rep('Credit card (automatic)', nrow(credit_card)),
                             stringsAsFactors = FALSE)

# Combine the dataframes into one
df <- rbind(electronic_check_df, mailed_check_df, bank_transfer_df, credit_card_df)
df$PaymentMethod <- as.factor(df$PaymentMethod)
```

Running the one-way ANOVA test using aov() function:

```
> # Run the ANOVA test
> anova <- aov(total_charges ~ PaymentMethod, data = df)
> # View the model summary
> summary(anova)
               Df    Sum Sq   Mean Sq F value Pr(>F)
PaymentMethod    3 4.417e+09 1.472e+09   326.9 <2e-16 ***
Residuals     7039 3.171e+10 4.505e+06
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> # Save summary to an object
> a.summary <- summary(anova)
> # Determine if we should reject the null hypothesis
> p.value <- a.summary[[1]]["Pr(>F)"]
> #Determine if we should reject the null hypothesis
> ifelse(p.value > alpha, "Fail to reject the null hypothesis",
+        "Reject the null hypothesis")
             Pr(>F)
PaymentMethod "Reject the null hypothesis"
Residuals     NA
```

The p-value is lesser than the significance level 0.05, thus we can reject the null hypothesis and conclude that the average total charge is different for all types of Payment Methods.

## Two-way ANOVA test

**Churn & SeniorCitizen, Dependents**
dependent variable: churn
independent variables: SeniorCitizen and Dependents
H0: SeniorCitizen and Dependents have no impact on churn
H1: SeniorCitizen and Dependents have an impact on churn
(significant level of 0.05)

```
> summary(anova2)
                Df Sum Sq Mean Sq F value Pr(>F)
Dependents       1   37.0   37.03   198.0 <2e-16 ***
SeniorCitizen    1   19.4   19.41   103.8 <2e-16 ***
Residuals     7040 1316.6    0.19
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Since the p-value dependents (<0.0000) are less than the significance level of 0.05, and the p-value senior citizen (<0.000) is less than the significance level of 0.05, we reject H0. There is sufficient evidence that supports SeniorCitizen and Dependents have an impact on churn.

In conclusion, demographic information of SeniorCitizen, Dependents, and Partners will affect customers' churn rate. Gender has no relationship with churn.

# Module 5:Final Project Draft Report

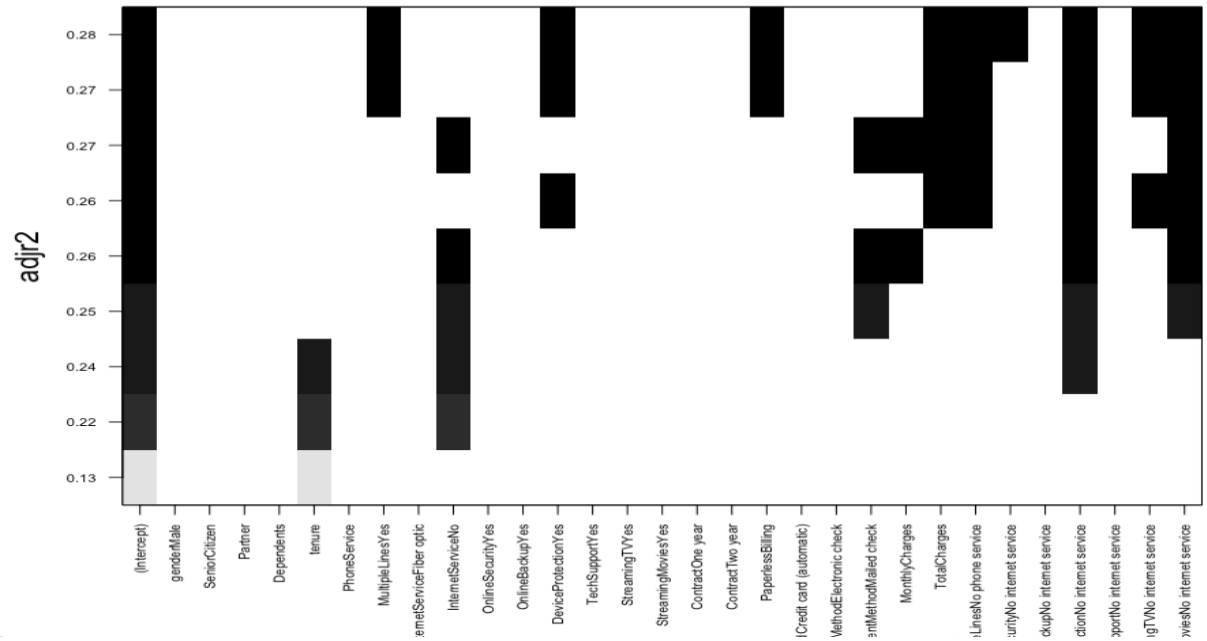
## Splitting the data into a train and test set

To separate the data into train and test set to a 70/30 split where 70% of the random observations are used for the training set and 30% is used for testing, I have used the **createDataPartition()** function from the 'caret' package

```
#Split the dataset into Test and Train data#
set.seed(123)
trainIndex <- createDataPartition(telco_df$Churn, p = 0.70, list = FALSE , times=1 )
telco_df_train <- telco_df[trainIndex,]
telco_df_test <- telco_df[-trainIndex,]
```

## Subset Regression Method

This method helps in the model selection process which tests all the possible combinations of the independent/explanatory variables and eventually helps in identifying the best possible Model using the **regsubsets** function from the leaps package.




**Observation:** The above plot function shows the attributes from the subset Regression method that can be selected for the best Model and are ranked as per their adjusted R-square value.

Below are the Final 9 predictors identified using the Subset Regression Method-

- MultipleLines-No
- OnlineSecurity-Yes
- TechSupport-Yes
- Contract- One year, Two Year
- PaperlessBilling
- PaymentMethod- Electronic
- MonthlyCharges
- Total Charges

Let us build the models using different algorithms using Glm Logistic regression, Stepwise selection, Ridge, Lasso for predicting the customer churn and select the best model by comparing various metrics like AIC, BIC, RMSE etc.,

## Logistic Regression Model Using GLM

```
> fit1<- glm(Churn~.,data = train,family = binomial(link = "logit"))
> summary(fit1)

Call:
glm(formula = Churn ~ ., family = binomial(link = "logit"), data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9375  -0.6760  -0.3370   0.6987   3.2480

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)      -6.475e-01  1.866e-01  -3.471 0.000519 ***
gender            5.718e-02  7.693e-02   0.743 0.457349
SeniorCitizen     2.873e-01  1.010e-01   2.846 0.004428 **
Partner           1.303e-01  9.301e-02   1.401 0.161286
Dependents       -2.950e-01  1.054e-01  -2.798 0.005149 **
tenure           -7.350e-02  6.863e-03 -10.710  < 2e-16 ***
PhoneService     -1.335e+00  1.603e-01  -8.330  < 2e-16 ***
MultipleLines     2.973e-02  9.764e-02   0.304 0.760772
OnlineSecurity   -6.958e-01  1.001e-01  -6.950 3.65e-12 ***
OnlineBackup     -2.513e-01  9.429e-02  -2.665 0.007707 **
DeviceProtection -3.324e-01  9.616e-02  -3.457 0.000547 ***
TechSupport      -7.928e-01  1.002e-01  -7.913 2.51e-15 ***
StreamingTV      -2.739e-01  1.041e-01  -2.633 0.008473 **
StreamingMovies  -8.025e-02  1.026e-01  -0.782 0.434208
PaperlessBilling  5.073e-01  8.811e-02   5.758 8.53e-09 ***
MonthlyCharges    3.512e-02  2.914e-03  12.052  < 2e-16 ***
TotalCharges      3.368e-04  7.802e-05   4.317 1.58e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5717.6  on 4924  degrees of freedom
Residual deviance: 4166.9  on 4908  degrees of freedom
  (因为不存在，5个观察量被删除了)
AIC: 4200.9

Number of Fisher Scoring iterations: 6
```

In the logistic regression model, I first used all the variables to create the model. According to the summary of the model, tenure, PhoneService, OnlineSecurity, DeviceProtection, TechSupport, PaperlessBilling, MonthlyCharges, TotalCharges, SeniorCitizen are significant variables. So in the next step, I will choose these variables to build a new model.

```
Call:
glm(formula = Churn ~ tenure + PhoneService + OnlineSecurity +
    DeviceProtection + TechSupport + PaperlessBilling + MonthlyCharges +
    TotalCharges + SeniorCitizen, family = binomial(link = "logit"),
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0060  -0.6826  -0.3419   0.7001   3.2664

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)      -6.961e-01  1.786e-01  -3.898 9.71e-05 ***
tenure           -7.336e-02  6.749e-03 -10.870  < 2e-16 ***
PhoneService     -1.167e+00  1.483e-01  -7.874 3.43e-15 ***
OnlineSecurity   -6.752e-01  9.863e-02  -6.846 7.58e-12 ***
DeviceProtection -3.367e-01  9.578e-02  -3.515 0.000439 ***
TechSupport      -8.188e-01  9.941e-02  -8.237  < 2e-16 ***
PaperlessBilling  5.173e-01  8.768e-02   5.901 3.62e-09 ***
MonthlyCharges    3.122e-02  2.352e-03  13.271  < 2e-16 ***
TotalCharges      3.070e-04  7.638e-05   4.019 5.85e-05 ***
SeniorCitizen     3.706e-01  9.859e-02   3.759 0.000171 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5717.6  on 4924  degrees of freedom
Residual deviance: 4190.3  on 4915  degrees of freedom
  (因为不存在，5个观察量被删除了)
AIC: 4210.3
```

From the summary of the new model, I can see all the variables are significant now, and this is maybe a better model. In the following steps, I will use a confusion matrix to calculate the accuracy of the model.Use the confusion matrix to see if the model is a good one.

```
> confusionMatrix(predicted.classes.min, train$Churn, positive = "TRUE")
Confusion Matrix and Statistics

          Reference
Prediction FALSE TRUE
     FALSE  3261  631
     TRUE    348  685

               Accuracy : 0.8012
                 95% CI : (0.7898, 0.8123)
    No Information Rate : 0.7328
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4552

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.5205
            Specificity : 0.9036
         Pos Pred Value : 0.6631
         Neg Pred Value : 0.8379
             Prevalence : 0.2672
         Detection Rate : 0.1391
   Detection Prevalence : 0.2097
      Balanced Accuracy : 0.7120

       'Positive' Class : TRUE
```

By looking at the confusion matrix, I observed an accuracy of 80.12% for this model.I don't think this number is that high, so I want to try to improve the accuracy of the model by removing outliers.

## Outliers Identification & Removal

```
#replace outliers with mean (tenure, MonthlyCharges,TotalCharges)
high <- mean(f1$tenure) + sd(f1$tenure) * 3
low <- mean(f1$tenure) - sd(f1$tenure) * 3
f1$Outlier <- (f1$tenure < low | f1$tenure > high)
f1$tenure[f1$outlier] = mean(f1$tenure[!is.na(f1$outlier)])

high <- mean(f1$MonthlyCharges) + sd(f1$MonthlyCharges) * 3
low <- mean(f1$MonthlyCharges) - sd(f1$MonthlyCharges) * 3
f1$Outlier <- (f1$MonthlyCharges < low | f1$MonthlyCharges > high)
f1$MonthlyCharges[f1$outlier] = mean(f1$MonthlyCharges[!is.na(f1$outlier)])

high <- mean(f1$TotalCharges) + sd(f1$TotalCharges) * 3
low <- mean(f1$TotalCharges) - sd(f1$TotalCharges) * 3
f1$Outlier <- (f1$TotalCharges < low | f1$TotalCharges > high)
f1$TotalCharges[f1$outlier] = mean(f1$TotalCharges[!is.na(f1$outlier)])a
```

Since the other variables except tenure, monthlycharges and totalcharges are Categorical variables, we will replace the outliers of the three non-categorical variables with the average of all the outliers. Then I will use the new cleaning data to build the model.

```
Call:
glm(formula = Churn ~ ., family = binomial(link = "logit"), data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9335  -0.6662  -0.3546   0.6922   3.1254

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)      -8.980e-01  1.837e-01  -4.890 1.01e-06 ***
gender            2.489e-02  7.670e-02   0.324 0.745575
SeniorCitizen     3.117e-01  9.923e-02   3.141 0.001683 **
Partner           8.958e-02  9.106e-02   0.984 0.325224
Dependents       -2.342e-01  1.042e-01  -2.247 0.024655 *
tenure           -6.173e-02  6.415e-03  -9.623  < 2e-16 ***
PhoneService     -1.317e+00  1.604e-01  -8.214  < 2e-16 ***
MultipleLines     6.330e-02  9.641e-02   0.657 0.511426
OnlineSecurity   -5.791e-01  9.965e-02  -5.812 6.17e-09 ***
OnlineBackup     -3.288e-01  9.328e-02  -3.525 0.000423 ***
DeviceProtection -2.986e-01  9.543e-02  -3.129 0.001752 **
TechSupport      -8.415e-01  1.000e-01  -8.412  < 2e-16 ***
StreamingTV      -1.547e-01  1.032e-01  -1.499 0.133860
StreamingMovies  -1.808e-01  1.025e-01  -1.763 0.077865 .
PaperlessBilling  4.158e-01  8.794e-02   4.728 2.27e-06 ***
MonthlyCharges    3.763e-02  2.936e-03  12.817  < 2e-16 ***
TotalCharges      2.294e-04  7.398e-05   3.101 0.001929 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5663.3  on 4929  degrees of freedom
Residual deviance: 4200.2  on 4913  degrees of freedom
AIC: 4234.2

Number of Fisher Scoring iterations: 6
```

Same step here, first used all the variables to create the model. According to the summary of the model, picked up all the significant variables to built a new model.

```
Call:
glm(formula = Churn ~ tenure + PhoneService + OnlineSecurity +
    TechSupport + PaperlessBilling + MonthlyCharges + SeniorCitizen +
    DeviceProtection, family = binomial(link = "logit"), data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0284  -0.6542  -0.3609   0.6574   3.1636

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      -1.198259   0.137535  -8.712  < 2e-16 ***
tenure           -0.047326   0.002171 -21.802  < 2e-16 ***
PhoneService     -1.242244   0.143433  -8.661  < 2e-16 ***
OnlineSecurity   -0.707964   0.100427  -7.050 1.79e-12 ***
TechSupport      -0.797516   0.099801  -7.991 1.34e-15 ***
PaperlessBilling  0.365469   0.086989   4.201 2.65e-05 ***
MonthlyCharges    0.039754   0.002009  19.784  < 2e-16 ***
SeniorCitizen     0.366862   0.097773   3.752 0.000175 ***
DeviceProtection -0.334683   0.095068  -3.520 0.000431 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5659.1  on 4929  degrees of freedom
Residual deviance: 4180.7  on 4921  degrees of freedom
AIC: 4198.7

Number of Fisher Scoring iterations: 5
```

From the summary of the new model, I can see all variables are significant now, and this is maybe a better model. In the AIC measure, if the score is lower than the other one, it means this model is the preferred model. But it didn’t improve too much if we look at the AIC measure, the model 2’s AIC is a little bigger than model 1’s. However, the parameter in Model 2 is less, which is better than Model 1.

```
> confusionMatrix(predicted.classes.min, train$Churn, positive = "TRUE")
Confusion Matrix and Statistics

          Reference
Prediction FALSE TRUE
     FALSE  3316  632
     TRUE    328  654

               Accuracy : 0.8053
                 95% CI : (0.7939, 0.8162)
    No Information Rate : 0.7391
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4532

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.5086
            Specificity : 0.9100
         Pos Pred Value : 0.6660
         Neg Pred Value : 0.8399
             Prevalence : 0.2609
         Detection Rate : 0.1327
   Detection Prevalence : 0.1992
      Balanced Accuracy : 0.7093

       'Positive' Class : TRUE
```

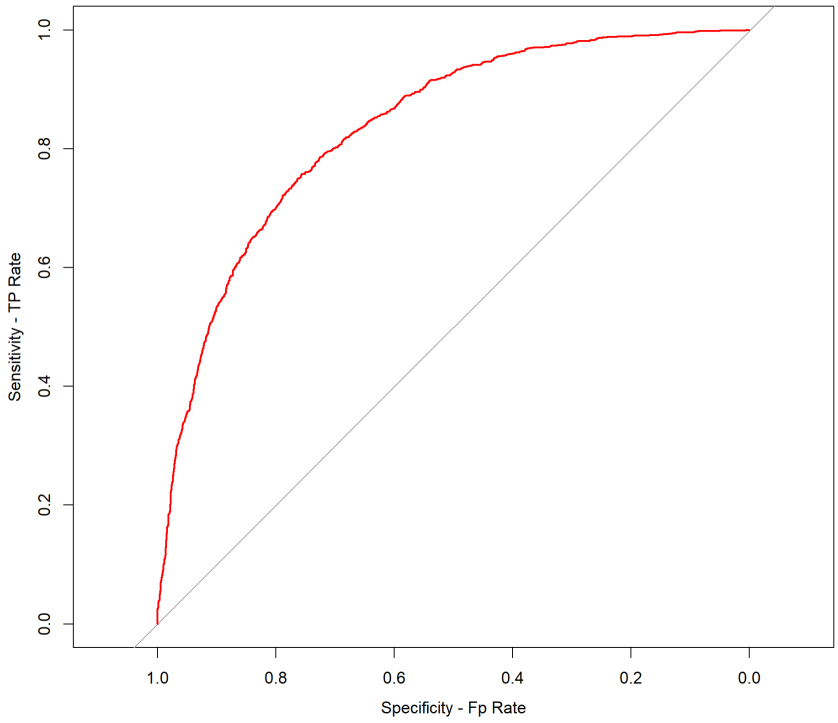
By looking at the confusion matrix, the prediction is the row and the actual value is the column.

**Accuracy**: By looking at the confusion matrix figure, the accuracy is 0.8053 (80.53%), this is better than the previous model.

**Precision**: Precision refers to the positive predictive value. As can be seen from the figure, the pos pred value is 0.6660 (66.6%). This indicates that 66.6% of the customers are predicted to be churned from the telco company.

**Recall**: Recall refers to sensitivity. As can be seen from the figure, the sensitivity value is 0.5086 (50.86%).

**Specificity**: means the proportion of customer churn correctly identified by the measurement was 91%.




```
> auc1
Area under the curve: 0.8351
```

The closer the curve is to the left and upper boundaries, the more accurate the test will be. In the example of this data set, the curve is in a medium to upper state, indicating that the model is good.

## Stepwise Selection

Forward selection

```
> fullModel = lm(Churn ~ ., data = telco_test) # model with all variables
> nullModel = lm(Churn ~ 1, data = telco_test) # model with the intercept only
```

The full model contains all variables in the dataset and the null model only has an intercept.

```
summary(stepAIC(nullModel, # start with a model containing no variables
                direction = 'forward', # run forward selection
                scope = list(upper = fullModel, # the maximum to consider is a model with all variables
                             lower = nullModel), # the minimum to consider is a model with no variables
                trace = 0)) # do not show the step-by-step process of model selection
```

```
Call:
lm(formula = Churn ~ Contract + InternetService + TotalCharges +
    PaymentMethod + StreamingTV + MultipleLines + SeniorCitizen +
    PaperlessBilling + StreamingMovies + OnlineSecurity + TechSupport +
    tenure, data = telco_test)

Residuals:
     Min       1Q   Median       3Q      Max
-0.71218 -0.24222 -0.05686  0.12945  1.12144

Coefficients:
                                        Estimate Std. Error t value Pr(>|t|)
(Intercept)                             3.140e-01  3.063e-02  10.248  < 2e-16 ***
ContractOne year                       -1.013e-01  2.482e-02  -4.083 4.62e-05 ***
ContractTwo year                       -4.936e-02  3.035e-02  -1.627  0.10399
InternetServiceFiber optic              1.852e-01  2.403e-02   7.706 1.99e-14 ***
InternetServiceNo                      -1.525e-01  2.933e-02  -5.200 2.19e-07 ***
TotalCharges                           -5.173e-05  1.044e-05  -4.956 7.75e-07 ***
PaymentMethodCredit card (automatic)   -3.017e-03  2.425e-02  -0.124  0.90101
PaymentMethodElectronic check           7.266e-02  2.387e-02   3.044  0.00236 **
PaymentMethodMailed check              -2.947e-04  2.526e-02  -0.012  0.99069
StreamingTV                             5.328e-02  2.179e-02   2.445  0.01457 *
MultipleLines                           5.880e-02  1.897e-02   3.100  0.00196 **
SeniorCitizen                           6.299e-02  2.320e-02   2.715  0.00668 **
PaperlessBilling                        3.830e-02  1.778e-02   2.154  0.03139 *
StreamingMovies                         3.631e-02  2.187e-02   1.660  0.09704 .
OnlineSecurity                         -3.989e-02  2.180e-02  -1.830  0.06746 .
TechSupport                            -4.116e-02  2.202e-02  -1.870  0.06168 .
tenure                                 -1.541e-03  8.280e-04  -1.860  0.06297 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3661 on 2095 degrees of freedom
Multiple R-squared:  0.2798,    Adjusted R-squared:  0.2743
F-statistic: 50.87 on 16 and 2095 DF,  p-value: < 2.2e-16
```

After running the stepwise forward selection, the final model selected by the AIC criterion includes the following predictors: Contract type (one year or two year), Internet service type (fiber optic or not), Total charges, payment method (electronic check), streaming TV, multiple lines, senior citizen, paperless billing, streaming movies, online security, tech support, and tenure.

- The residual standard error is 0.3661, which is a measure of the variability of the residuals.
- The multiple R-squared value of 0.2798 indicates that the model explains 27.98% of the variability in the response variable

- The adjusted R-squared value of 0.2743 indicates that the model accounts for 27.43% of the variability after adjusting for the number of predictors in the model.
- The F-statistic of 50.87 with a p-value of <2.2e-16 indicates that the model as a whole is a good fit to the data.

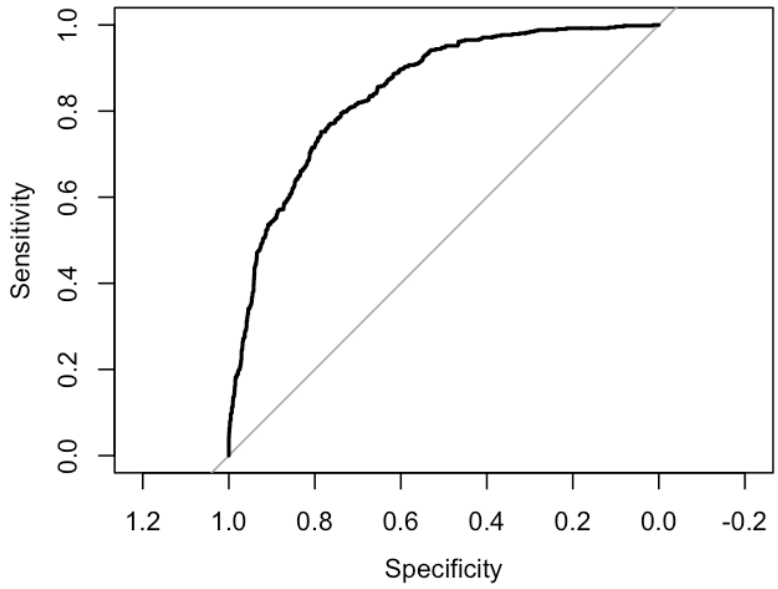
**Calculating the AIC and BIC metrics:**

```
> #Calculating the ROC and AUC scores:
> model <- lm(Churn ~ ., data = telco_test) # fit the model with all variables
> library(stats)
> stepAIC = stepAIC(nullModel, direction = 'forward',
+                   scope = list(upper = fullModel, lower = nullModel),
+                   trace = 0)
> # Calculate AIC
> AIC(stepAIC)
[1] 1768.383
> # Calculate BIC
> BIC(stepAIC)
[1] 1870.18
```

The AIC value of 1768.383 and BIC value of 1870.18.

**Calculating the ROC and AUC scores:**

```
> #Calculating the ROC and AUC scores:
> library(pROC)
> model <- lm(Churn ~ ., data = telco_test) # fit the model with all variables
> predictions <- predict(model, type = "response") # get the model's predicted probabilities of Churn
> predictions <- predict(fullModel, type = "response") # get the model's predicted probabilities of Churn
> roc_object <- roc(telco_test$Churn, predictions) # calculate the ROC curve
Setting levels: control = 0, case = 1
Setting direction: controls < cases
> auc(roc_object) # calculate the AUC score
Area under the curve: 0.8439
> plot(roc_object)
```




The AUC score of 0.8439 indicates that the model has a good level of accuracy in predicting the binary outcome variable. A score of 0.8439 suggests that the model has a relatively high level of accuracy in predicting the outcome variable.

**Calculating RMSE:**

```
> library(Metrics)
> rmse(telco_test$Churn, predict(stepAIC , telco_test))
[1] 0.3646527
> rmse(telco_train$Churn, predict(stepAIC , telco_train))
[1] 0.3783978
```

## LASSO REGRESSION

## Splitting the data into a train and test set

To separate the data into train and test set to a 70/30 split where 70% of the random observations are used for the training set and 30% is used for testing, I have used the **createDataPartition()** function from the 'caret' package

```
set.seed(123)
trainIndex <- createDataPartition(telco_df$Churn, p = 0.70, list = FALSE , times=1 )
telco_df_train <- telco_df[trainIndex,]
telco_df_test <- telco_df[-trainIndex,]
```

Let's Further define the Target variable for Train and Test dataset and define the matrix for predictor variables respectively.

```
#Define the Target variable for Train and Test Data set
y_train <- telco_df_train$Churn
y_test<- telco_df_test$Churn
```

```
#Define the matrix of predictor variables
x_train <- model.matrix(Churn ~ ., telco_df_train)[,-1]
x_test  <- model.matrix(Churn ~ ., telco_df_test)[,-1]
```

## Estimating the Lambda Value using Cross Validation for Lasso Regression

The best value of Lambda is calculated using the **cv.glmnet** function ,**alpha=1 and family = binomial** as the response type for computing the penalized Lasso Regression Model using the Training set of target and predictor variables as shown below :

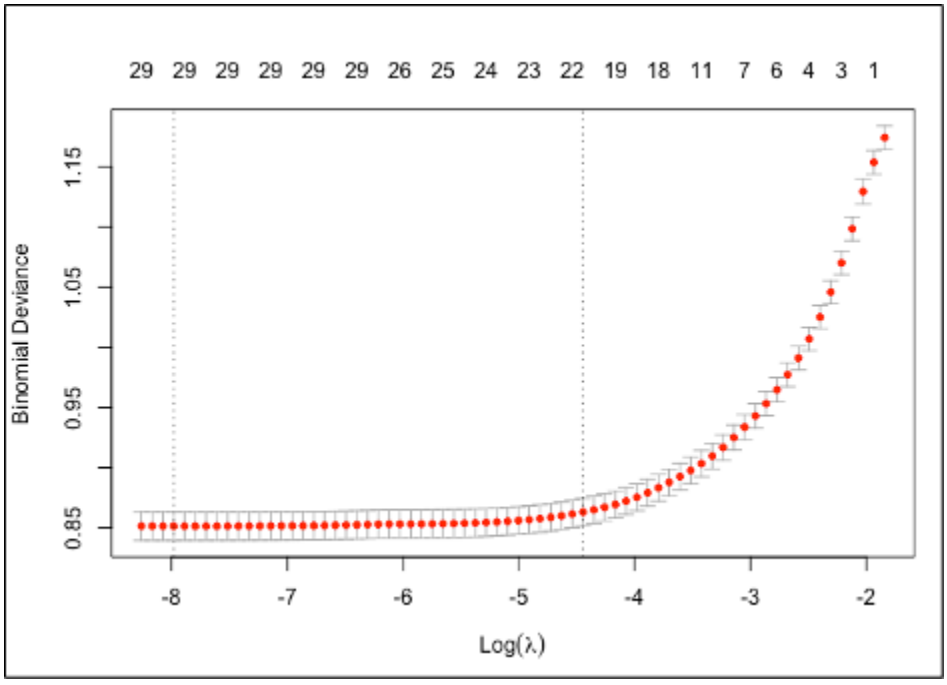
```
> #Perform k-fold cross-validation to find optimal lambda value for Lasso #
> set.seed(123)
> lasso_cv <- cv.glmnet(x_train, y_train, family="binomial", alpha=1,nfolds = 10)
> lasso_cv

Call:  cv.glmnet(x = x_train, y = y_train, nfolds = 10, family = "binomial",      alpha = 1)

Measure: Binomial Deviance

      Lambda Index Measure      SE Nonzero
min 0.000342    67  0.8513 0.01195      29
1se 0.011727    29  0.8631 0.01135      22
>
```

**Observation :** The minimum value of Lambda is **0.000342** and the value of Lambda at One Standard error is **0.011727**

**Below is the Logarithmic value of Lambda minimum and Lambda at One Standard Error –**

```
> log(lasso_cv$lambda.min)
[1] -7.981155
> log(lasso_cv$lambda.1se)
[1] -4.445872
```

**Now , lets plot Binomial Deviance and Log Lamba value as show below**




**Observation:** The above plot displays the cross validation error according to the log of Lambda

1. The left-dashed vertical line indicates that the log of optimal value of Lambda which is approx. -8 and is the one which minimizes the prediction error.
2. The log of lambda value at one standard error is approx -4.5

## Lasso Regression Model with Training Data:

**1.** Below is the Lasso regression model using the **glmnet** function **on the training data using the Minimum Lambda Value :**

```
> #Fitting the Lasso Regression Model on the Training Set Using Best Lamba value(Min value)#
> lasso_model_train_min  <- glmnet(x_train, y_train, alpha = 1, lambda = best_lambda1)
> lasso_model_train_min

Call:  glmnet(x = x_train, y = y_train, alpha = 1, lambda = best_lambda1)

  Df  %Dev    Lambda
1 27 28.66 0.0003418
```

**Lets , have a look at the coefficients for the above Lasso Regression Model using the Lamba Min**

```
> coef(lasso_model_train_min)
31 x 1 sparse Matrix of class "dgCMatrix"
                                              s0
(Intercept)                            3.926831e-01
genderMale                            -5.106572e-03
SeniorCitizen                          3.803871e-02
Partner                               -1.011146e-02
Dependents                            -2.368334e-02
tenure                                -1.995201e-03
PhoneService                          -2.728679e-02
MultipleLinesNo phone service          .
MultipleLinesYes                       4.621536e-02
InternetServiceFiber optic             1.718229e-01
InternetServiceNo                     -8.644111e-02
OnlineSecurityNo internet service      .
OnlineSecurityYes                     -5.130812e-02
OnlineBackupNo internet service       -1.847934e-02
OnlineBackupYes                       -1.972805e-02
DeviceProtectionNo internet service   -8.673059e-03
DeviceProtectionYes                   -1.052907e-02
TechSupportNo internet service        -4.569523e-03
TechSupportYes                        -5.379568e-02
StreamingTVNo internet service        -1.987172e-02
StreamingTVYes                         5.150453e-02
StreamingMoviesNo internet service    -8.918456e-03
StreamingMoviesYes                     6.098612e-02
ContractOne year                      -1.070512e-01
ContractTwo year                      -7.952482e-02
PaperlessBilling                       4.729550e-02
PaymentMethodCredit card (automatic) -6.409067e-03
PaymentMethodElectronic check          6.659604e-02
PaymentMethodMailed check             -9.109400e-03
MonthlyCharges                         .
TotalCharges                          -4.212142e-05
```

**Observation:** From the above plot coefficients of the Lasso Regression Model, we can understand the below:

1. Features with very less coefficient values have been penalized or shrunk to 0 by the Lasso Regression Model. The more they are penalized or shrunk, the less significant these features are to the model.
2. Below are the coefficients which we can understand have been shrunken to 0

- MultipleLines-No phone service

- OnlineSecurity-No internet service
- MonthlyCharges

**2.** Below is the Lasso regression model using the **glmnet** function on the **training data** using the **Lamba Value at 1se** :

```
> #Fitting the Lasso Regression Model on the Training Set Using 1se Lamba value#
> lasso_model_train_1se  <- glmnet(x_train, y_train, alpha = 1, lambda = lasso_cv$lambda.1se)
> lasso_model_train_1se

Call:  glmnet(x = x_train, y = y_train, alpha = 1, lambda = lasso_cv$lambda.1se)

  Df  %Dev  Lambda
1 19 27.72 0.01173
```

**Lets , have a look at the coefficients for the above Lasso Regression Model using the Lamba 1se**

```
> coef(lasso_model_train_1se)
31 x 1 sparse Matrix of class "dgCMatrix"
                                                s0
(Intercept)                             3.640615e-01
genderMale                              .
SeniorCitizen                           2.570816e-02
Partner                                 .
Dependents                             -1.726123e-02
tenure                                 -3.468170e-03
PhoneService                            .
MultipleLinesNo phone service           .
MultipleLinesYes                        6.591026e-03
InternetServiceFiber optic              1.538565e-01
InternetServiceNo                      -8.258471e-02
OnlineSecurityNo internet service       .
OnlineSecurityYes                      -4.534149e-02
OnlineBackupNo internet service        -1.633038e-02
OnlineBackupYes                        -9.362326e-04
DeviceProtectionNo internet service    -4.115967e-04
DeviceProtectionYes                     .
TechSupportNo internet service          .
TechSupportYes                         -4.236373e-02
StreamingTVNo internet service         -1.564884e-02
StreamingTVYes                          1.518554e-02
StreamingMoviesNo internet service      .
StreamingMoviesYes                      2.706048e-02
ContractOne year                       -8.561748e-02
ContractTwo year                       -6.393572e-02
PaperlessBilling                        4.136841e-02
PaymentMethodCredit card (automatic)    .
PaymentMethodElectronic check           8.037777e-02
PaymentMethodMailed check               .
MonthlyCharges                          .
TotalCharges                           -1.531121e-05
```

**Observation:** From the above plot coefficients of the Lasso Regression Model using Lambda 1se, we can understand the below:

1. Features with very less coefficient values have been penalized or shrunk to 0 by the Lasso Regression Model. The more they are penalized or shrunk, the less significant these features are to the model.
2. Below are the coefficients which we have been shrunken to 0 which are more as compared to the Model using the Lambda Min

- Gender - Male
- Partner
- Phone Service
- MultipleLines-No phone service
- OnlineSecurity-No internet service
- DeviceProtection -Yes
- TechSupport-No internet service
- StreamingMovies- No internet service
- PaymentMethod- Credit card (automatic)
- PaymentMethod-Mailed check
- MonthlyCharges

## Prediction on the Training Dataset and Determining Performance of the model :

```
> #Use the Lasso regression model for making predictions on Train data#
> predict_lasso_train <- predict(lasso_model_train_min, newx = x_train)
>
```

**Root Mean Squared Error :**

```
> #RMSE for Lasso regression model against the Training set#
> lasso_train_rmse <- rmse(y_train, predict_lasso_train)
> lasso_train_rmse
[1] 0.3768874
```

## Observation:

1. We first use the predict function on the Lasso Model we created using Lambda Min value to generate prediction on the training data.
2. Further , the performance of the Lasso Regression Model against the training dataset is determined by calculating the Root Mean Squared error (RMSE) using the rmse function on the predicted model.
3. Finally , we get the RMSE value on the training data which is **0.37**
4. This shows that the RMSE value is good enough for the prediction as the data points in the training data are not scattered much.

## Prediction on the Testing Dataset and Determining Performance of the model :

```
> #Use the Lasso regression model for making predictions on Test data#
> predict_lasso_test <- predict(lasso_model_train_min, newx = x_test)
>
```

**Root Mean Squared Error :**

```
> #RMSE for Lasso regression model against the Test set#
> lasso_test_rmse <- rmse(y_test, predict_lasso_test)
> lasso_test_rmse
[1] 0.3659978
```

## Observation:

1. We first use the predict function on the Lasso Model we created using Lambda Min value to generate prediction on the training data.
2. Further , the performance of the Lasso Regression Model against the training dataset is determined by calculating the Root Mean Squared error (RMSE) using the rmse function on the predicted model.

3. Finally , we get the RMSE value on the training data which is **0.36**
4. This shows that the RMSE value is good enough for the prediction as the data points in the training data are not scattered much.

## Is the Model Over-fitting ?

As we saw that the RMSE values for both the Training and Test data set are almost close, hence we can conclude that the Ridge Regression Model is not over-fitting.

## Lasso Model Metrics

**1. AIC $ BIC Score**

```
> #AIC SCORE#
> AIC.glmnet <- function(glm_fit) {
+   chisqLR <- glm_fit$nulldev - deviance(glm_fit)
+
+   chisqLR - 2*glm_fit$df
+ }
> AIC.glmnet(lasso_model_train_min)
[1] 227.3354
```

```
> #BIC SCORE#
> glmnet_cv_bic(lasso_cv)
$BIC
[1] -1397.502
```

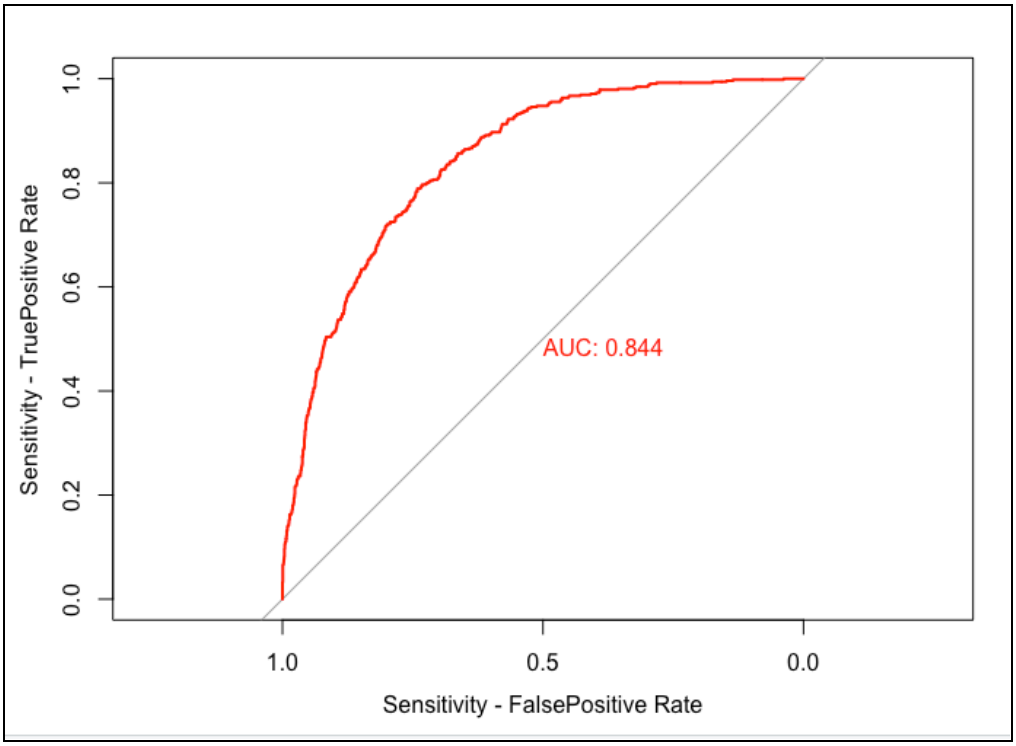
**2. The R-squared and Root Mean Square Error values (RMSE) for both Training & Test dataset is summarized as follows :**

```
> lasso_model_perf
  RMSE_test   R2_test RMSE_train  R2_train
1 0.3659978 0.2744586  0.3768874 0.2865638
>
```

**Observation :**

1. R-square value for Train : **0.28**
2. RMSE value for Train : **0.37**
3. R-square value for Test : **0.27**
4. RMSE value for Test : **0.36**

## ROC Curve (Receiver Operating Characteristics) and AUC




## Observation :

We can observe that the ROC curve is moderately closer or hugging the y-axis and is showing a curve and is not flat at the top which indicates that the Model has shown moderate performance results.

## Ridge Regression

In order to find the lambda value which produces the lowest test MSE, **k-fold cross validation** should be performed.

If a model is trained and tested using only one dataset, the test MSE can vary greatly depending on which observations were used in the training and testing datasets.

In order to avoid this problem, a model can be fit several times using different training and testing datasets each time and the overall test MSE can be calculated as the average of all of the test MSEs.

This method is known as **cross validation** and k-folds cross validation can be performed in R using **cv.glmnet() function**. It performs k-folds cross validation using **k = 10 folds.**

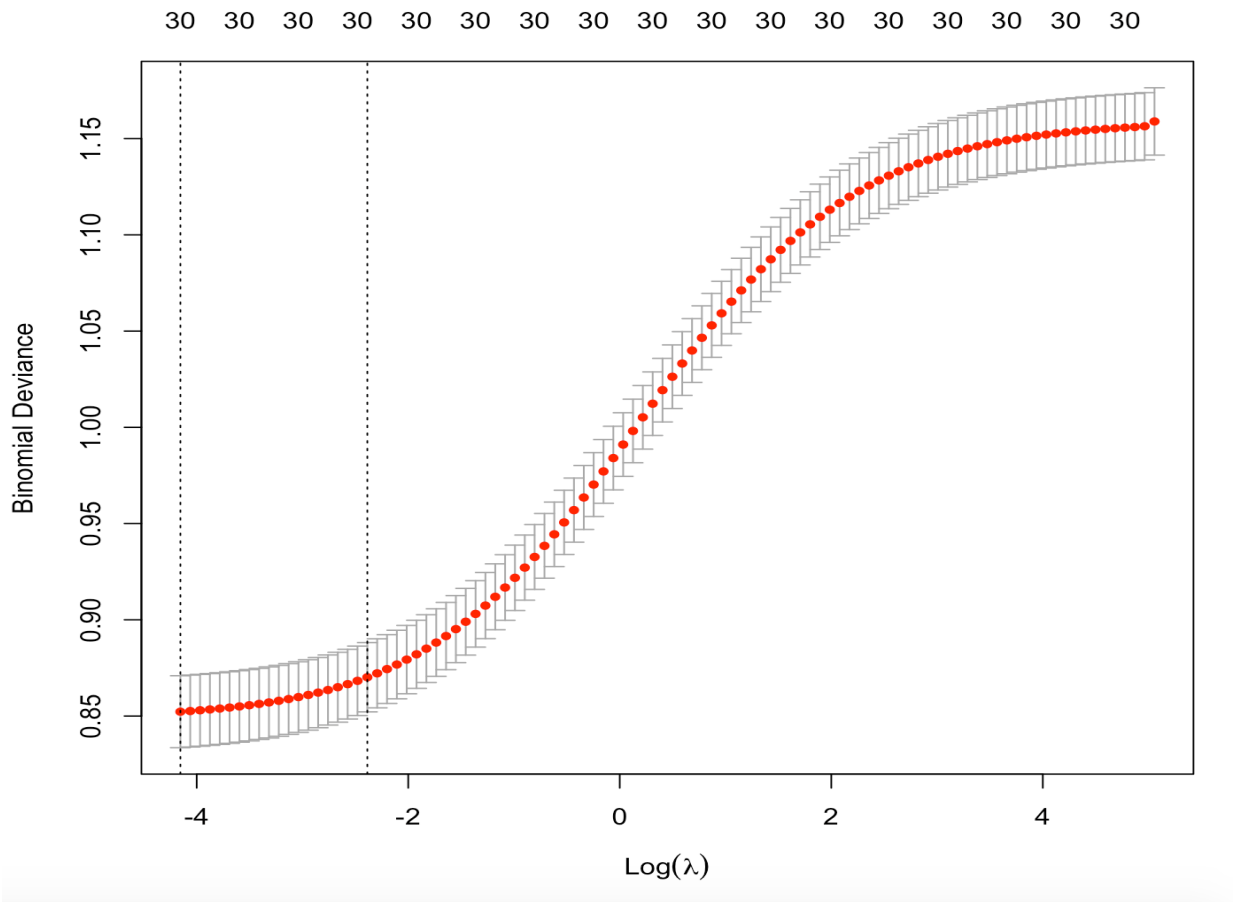
**Alpha value of 0** should be set for **Ridge regression** models.

```
> cv_model_ridge_train <- cv.glmnet(train_x, train_y, alpha = 0,family="binomial")
> cv_model_ridge_train$lambda.min
[1] 0.01569765
> cv_model_ridge_train$lambda.1se
[1] 0.09194148
> log(cv_model_ridge_train$lambda.min)
[1] -4.154244
> log(cv_model_ridge_train$lambda.1se)
[1] -2.386603
```

**Lambda.min** has a value of **0.015 and Lambda.1se** has a value of **0.09.**

**Lambda.min** gives the result with minimum mean cross-validation error. It is the lambda value at which the **smallest MSE and maximum accuracy** can be achieved. This lambda value will give the most accurate model.

**Lambda.1se** gives the result such that the cross-validation error is within 1 standard error of the minimum. This lambda value gives the **simplest model and also lies within 1 standard error of the optimal value of lambda**.




**The left dashed vertical line** gives the log of the optimal lambda value (**Lambda.min**) which produces the least prediction error , its value is approximately equal to **-4.15**.
**The right dashed vertical line** gives the log of the lambda value **(Lambda.1se)** which can produce the simplest model. Its value is approximately equal to **-2.38**.

**Fitting the Ridge regression model:**

```
> model.min.ridge.train <- glmnet(train_x, train_y, alpha = 0,
+                          lambda = cv_model_ridge_train$lambda.min)
> #Display regression coeff
> coef(model.min.ridge.train)
31 x 1 sparse Matrix of class "dgCMatrix"
                                                 s0
(Intercept)                              3.573022e-01
genderMale                               2.158546e-03
SeniorCitizen                            1.296309e-02
Partner                                  1.365491e-03
Dependents                              -2.964897e-02
tenure                                  -2.395068e-03
PhoneService                            -2.921509e-02
MultipleLinesNo phone service            2.740684e-02
MultipleLinesYes                         5.230255e-02
InternetServiceFiber optic               1.330155e-01
InternetServiceNo                       -4.321635e-03
OnlineSecurityNo internet service       -6.420333e-03
OnlineSecurityYes                       -5.965514e-02
OnlineBackupNo internet service         -1.565542e-02
OnlineBackupYes                         -1.703161e-02
DeviceProtectionNo internet service     -2.271977e-02
DeviceProtectionYes                     -7.951194e-03
TechSupportNo internet service          -2.320784e-02
TechSupportYes                          -4.930379e-02
StreamingTVNo internet service          -1.953655e-02
StreamingTVYes                           3.778230e-02
StreamingMoviesNo internet service      -1.505793e-02
StreamingMoviesYes                       4.177488e-02
ContractOne year                        -1.182481e-01
ContractTwo year                        -8.574157e-02
PaperlessBilling                         4.025599e-02
PaymentMethodCredit card (automatic)    -1.189565e-02
PaymentMethodElectronic check            6.526041e-02
PaymentMethodMailed check               -1.981635e-02
MonthlyCharges                           8.076421e-04
TotalCharges                            -3.446816e-05
```

Fitting the Ridge regression model using the best lambda value which can produce the least prediction error - Lambda.min - **glmnet() function** is used against the training dataset and the coefficients of the predictor variables are displayed.
All the significant variables are incorporated in the model. Thus the maximum possible accuracy of the model is achieved and the prediction error is reduced by imposing a penalty to the model for having too many variables.

**Making predictions for the training dataset:**

```
> preds.train.ridge <- predict(model.min.ridge.train, s = cv_model_ridge_train$lambda.min,
+                              newx = train_x)
> sse <- sum((train_y - preds.train.ridge)^2)
> sst <- sum((train_y - mean(train_y))^2)
> train_RMSE_ridge = rmse(train_y, preds.train.ridge)
> training_rsq <- 1 - (sse / sst)
> training_rsq #0.27
[1] 0.2758669
> train_RMSE_ridge #0.37
[1] 0.3759614
```

The Root mean square error value is calculated using the rmse() function of the Metrics package. The value of RMSE for this model is 0.37.

The R square value is calculated using the formula 1 - (sse/sst) and the value is 0.27.

Generally, **the smaller the RMSE , the better is the model performance**. The R square value for this model is quite less , with a value of 0.27 which means that 27 percent of the variance in the results is explained by the predictor variables. This model accuracy is very low and not reliable.

**Making predictions for the testing dataset:**

```
> preds.test.ridge <- predict(model.min.ridge.train, s = cv_model_ridge_train$lambda.min,
+                             newx = test_x)
> sse <- sum((test_y - preds.test.ridge)^2)
> sst <- sum((test_y - mean(test_y))^2)
> test_RMSE_ridge = rmse(test_y, preds.test.ridge)
> testing_rsq <- 1 - sse / sst
> testing_rsq #0.31
[1] 0.2974528
> test_RMSE_ridge #0.36
[1] 0.3701527
```

The **Root mean square error** value is calculated using the rmse() function of the Metrics package and the value with the testing dataset is calculated as **0.37**.

The **R square value** is calculated using the formula 1 - (sse/sst) and the value with the testing dataset is **0.29.**

We can determine if a model is overfit by calculating the difference between RMSE of the training and testing datasets.

= 0.37 - 0.37 = 0

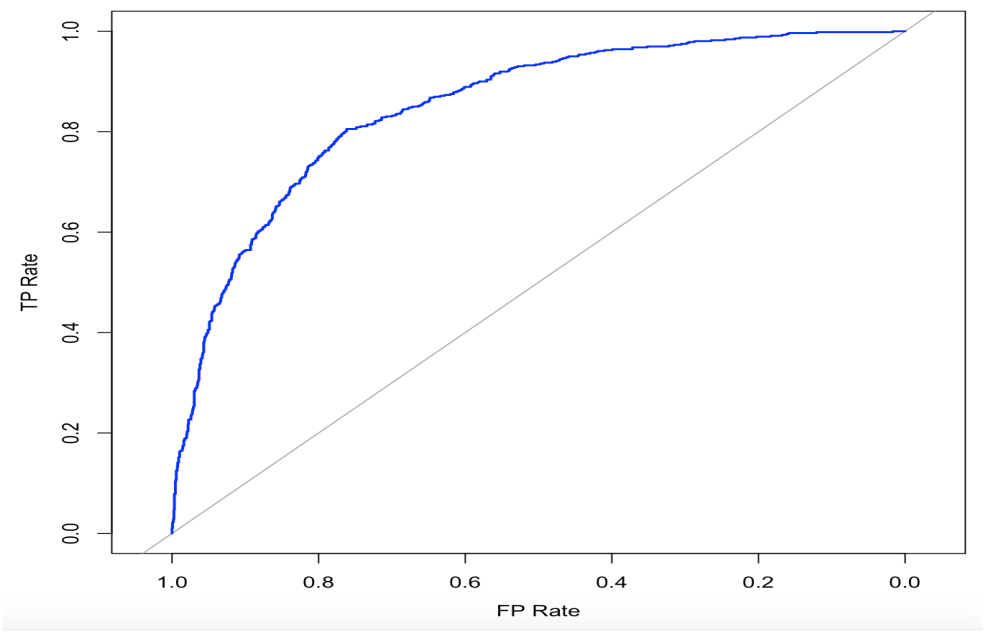
If this difference in RMSE between the training and testing datasets is close to 0, the model is not overfit. The difference of RMSE is 0 and also the R square value is almost equal in both training dataset and testing dataset, which means that **there is no problem of overfitting observed with this model**.

**Calculating the AIC and BIC metrics:**

```
> AIC.glmnet <- function(glm_fit) {
+   chisqLR <- glm_fit$nulldev - deviance(glm_fit)
+
+   chisqLR - 2*glm_fit$df
+ }
> AIC.glmnet(model.min.ridge.train)
[1] 205.0926
```

```
> glmnet_cv_aicc <- function(fit, lambda = 'lambda.1se'){
+   whlm <- which(fit$lambda == fit[[lambda]])
+   with(fit$glmnet.fit,
+        {
+          tLL <- nulldev - nulldev * (1 - dev.ratio)[whlm]
+          k <- df[whlm]
+          n <- nobs
+          return(list('BIC' = log(n) * k - tLL))
+        })
+ }
> #AIC and BIC scores for Ridge regression
> glmnet_cv_aicc(cv_model_ridge_train)
$BIC
[1] -1185.51
```

**Calculating the ROC and AUC scores:**




```
> plot(ROC1, col = "blue", ylab = "TP Rate",
+      xlab = "FP Rate")
> auc <- auc(ROC1)
> auc
Area under the curve: 0.8494
```

The ROC curve is approaching the central diagonal line which indicates that the model performance is moderate. The area under the curve value is 0.84. The higher the AUC, the model has good discriminatory ability and it will correctly predict the true positives and negatives for 84% of the time.

**Important metrics for the model:**

AIC : 205.09

BIC : -1185.51
RMSE (Training RMSE - Testing RMSE) : 0
R Square Test : 0.27
R Square Train : 0.27
AUC : 0.84

## COMPARISON OF THE MODELS

| METRICS | LOGISTIC MODEL - GLM | STEPWISE - Forward | LASSO | RIDGE |
|---|---|---|---|---|
| AIC | **4198.7** | **1768.383** | **227.33** | **205.09** |
| BIC | **4257** | **1870.18** | **-1397.50** | **-1185.51** |
| RMSE (Difference of Training & Test) | **0.022** | **0.01** | **0.01** | **0.01** |
| AUC | **0.8351** | **0.8439** | **0.844** | **0.84** |

## Observations:

The model built using the **Logistic regression algorithm with Glm() function** in R did a good job in predicting the customer churn with a RMSE difference value between training and testing datasets of **0.02.** This means that the difference between the actual and the predicted values is very less and it performed well in both the training and testing datasets. The AUC score of this model is **0.83**, indicating that this model can correctly classify the samples for 83% of the time.

AIC and BIC scores provide measures of the model performance that account for model complexity. These values are quite large and hence this model is quite complex though it can be accurate with the predictions.

In order to achieve a simpler model, we used the **Stepwise selection method, Ridge and Lasso algorithms.**

Used **Stepwise selection algorithm (forward)** to select the best set of predictor variables : Contract type (one year or two year), Internet service type (fiber optic or not), Total charges, payment method (electronic check), streaming TV, multiple lines, senior citizen, paperless billing, streaming movies. A comparatively lower AIC and BIC scores were achieved.

**Ridge and Lasso regularization** algorithms were used to build models to check if the complexity of the model can be reduced further. Both Lasso and Ridge models had a very low RMSE in both training and testing datasets and were able to predict the results accurately for **84%** of the time.

**All the above models achieved almost the same level of accuracy.**

In general, **BIC is more conservative than AIC and tends to select simpler models**, while AIC is more flexible and may select models with a higher complexity.

As we are trying to reduce the complexity of the model, we can select the **Lasso regression model as the best model as it has the lowest BIC value.**

# CONCLUSION

In this project , a dataset related to a Telecom company's customers was analyzed to study the various factors contributing towards Customer Churn. Data cleaning and Exploratory Data analysis were done on the dataset and some of the interesting insights were presented in the form of graphs and charts. The Telco Customer Churn dataset provides valuable information about the behavior of customers towards the company's products and services. By analyzing this dataset, we can understand the factors that influence customer churn, such as the services they subscribe to, demographic information, account information, and contract type. Additionally, we can use the insights gained from the analysis to develop targeted customer retention programs and minimize the customer churn rate. The dataset has 7043 records and 21 columns, including both numerical and categorical data. Through the analysis, the company aims to answer questions related to the predictors of customer churn and how different variables influence the decision of customers to leave the company.

**Chi-square and One-way Anova** methods were used to validate a few claims and the results and conclusions were documented. The best set of predictor variables used to predict the customer churn were identified using Best subsets regression method.

Following are some conclusions analyzed using the **Chi-square** and **ANOVA** methods -

- There is not sufficient evidence suggesting that there is a relationship between PhoneService and churn.
- There is not sufficient evidence that supports there is a relationship between gender and churn.
- There is sufficient evidence that supports there is a relationship between partner and churn.
- There is a difference in the duration of tenure. The duration of tenure is less than the mean that the customer churn is 38%, and the duration of tenure is more than the mean that the customer churn is 18.1%. So Customers with less tenure are more likely to be Churned.
- The average total charge is different for all types of internet service.
- The average total charge is different for all types of contract.
- The average total charge is different for all types of Payment Methods.
- Demographic information of SeniorCitizen, Dependents, and Partners will affect customers' churn rate. Gender has no relationship with churn.

In order to predict the customer Churn, we have built 4 models using Logistic regression with Glm function, Stepwise Selection method (Forward), Ridge and Lasso Regularization algorithms.
In order to select the best model, various model comparison metrics like AIC (Akaike information criterion) , BIC (Bayesian information criterion) , RMSE (Root Mean Square Error) , AUC (Area Under Curve) of ROC curve were compared.

All the four models achieved almost the same level of accuracy of 0.84, able to correctly make the predictions for 84% of the time and exhibited good discriminatory ability. In general, BIC is more conservative than AIC and tends to select simpler models and hence Lasso regression model was selected as the best model as it had the lowest BIC score.

**Key Insights:**

- The percentage of senior citizens churned is greater than the percentage of young/middle aged people.
- The customers having a longer tenure period tend to stay loyal with the company.
- The customers with a contract type of Month-to-month are more likely to churn.
- The customers having a Two-year contract type are less likely to churn.

- The customers opting for an internet service type of Fiber optic, are more likely to churn than with other internet services.
- Customers with partners and dependents have a lower churn rate than those who don't have partners and dependents.
- Customers opting for Electronic Check payment method are more likely to churn than the customers using other options.

## References


1. BlastChar. (2018, February 23). *Telco customer churn*. Kaggle. Retrieved February 5, 2023, from https://www.kaggle.com/datasets/blastchar/telco-customer-churn

2. Team, D. F. (2021, August 25). *Chi-square test in R: Explore the examples and essential concepts!* DataFlair. Retrieved January 22, 2023, from https://data-flair.training/blogs/chi-square-test-in-r/

3. *Chi-square goodness of fit test in R*. STHDA. (n.d.). Retrieved January 22, 2023, from http://www.sthda.com/english/wiki/chi-square-goodness-of-fit-test-in-r

4. Hayes, A. (2022, October 3). *Two-way anova: What it is, what it tells you, vs. one-way anova*. Investopedia. Retrieved January 22, 2023, from https://www.investopedia.com/terms/t/two-way-anova.asp

5. http://www.sthda.com/english/articles/37-model-selection-essentials-in-r/155-best-subsets-regression-essentials-in-r/

6. http://www.sthda.com/english/articles/36-classification-methods-essentials/149-penalized-logistic-regression-essentials-in-r-ridge-lasso-and-elastic-net/

## Appendix: R Code

```
#Importing the Dataset#

telco_df <- read.csv("WA_Fn-UseC_-Telco-Customer-Churn.csv",header=TRUE,
                     stringsAsFactors = FALSE,na.string = "")
str(telco_df)

#DATA CLEANING#
#Checking data is clean?#
colSums(is.na(telco_df)) # check & Returns the number of missing values in each column

sum(is.na(telco_df)) # Counts missing values in entire data frame

plot_missing(telco_df, title="Missing Data Profile",geom_label_args = list("size" = 2, "label.padding" =
unit(0.1, "lines")))


#Replacing missing value records of Total Charges column with their respective mean#

telco_df$TotalCharges[is.na(telco_df$TotalCharges)]<-round (mean(telco_df$TotalCharges,na.rm=TRUE),2)

#Dropping column CustomerID as they are not needed#
telco_df <-select(telco_df,-c(customerID))

#Converting Variables to Numeric
telco_df$tenure  <- as.numeric (telco_df$tenure)

#Recoding the Target variable Churn and converting it to Numeric
telco_df$Churn <- revalue(telco_df$Churn, c("Yes"="1", "No"="0"))
telco_df$Churn  <- as.numeric (telco_df$Churn)

#Recoding value for other attributes and converting it to Numeric
telco_df$Partner <- revalue(telco_df$Partner, c("Yes"="1", "No"="0"))
telco_df$Partner  <- as.numeric (telco_df$Partner)

telco_df$Dependents <- revalue(telco_df$Dependents, c("Yes"="1", "No"="0"))
telco_df$Dependents  <- as.numeric (telco_df$Dependents)

telco_df$PhoneService <- revalue(telco_df$PhoneService, c("Yes"="1", "No"="0"))
telco_df$PhoneService  <- as.numeric (telco_df$PhoneService)

telco_df$PaperlessBilling <- revalue(telco_df$PaperlessBilling, c("Yes"="1", "No"="0"))
telco_df$PaperlessBilling  <- as.numeric (telco_df$PaperlessBilling)

 #replace outliers with mean (tenure, MonthlyCharges,TotalCharges)

high <- mean(telco_df$tenure) + sd(telco_df$tenure) * 3
low <- mean(telco_df$tenure) - sd(telco_df$tenure) * 3
telco_df$Outlier <- (telco_df$tenure < low | telco_df$tenure > high)
telco_df$tenure[telco_df$outlier] = mean(telco_df$tenure[!is.na(telco_df$outlier)])
```

```
high <- mean(telco_df$MonthlyCharges) + sd(telco_df$MonthlyCharges) * 3
low <- mean(telco_df$MonthlyCharges) - sd(telco_df$MonthlyCharges) * 3
telco_df$Outlier <- (telco_df$MonthlyCharges < low | telco_df$MonthlyCharges > high)
telco_df$MonthlyCharges[telco_df$outlier] = mean(telco_df$MonthlyCharges[!is.na(telco_df$outlier)])

high <- mean(telco_df$TotalCharges) + sd(telco_df$TotalCharges) * 3
low <- mean(telco_df$TotalCharges) - sd(telco_df$TotalCharges) * 3
telco_df$Outlier <- (telco_df$TotalCharges < low | telco_df$TotalCharges > high)
telco_df$TotalCharges[telco_df$outlier] = mean(telco_df$TotalCharges[!is.na(telco_df$outlier)])
```

**#Descriptive Statistics for entire dataset#**

```
formattable(describe(telco_df),
          caption = "Descriptive statistics summary of the Telco Customer Churn Dataset")
```

**#Descriptive Statistics by Churn#**

```
describeBy(telco_df, group=telco_df$Churn)
```

**#Lets check the CORRELATION#**

```
telco_df_numeric = telco_df %>%   dplyr::select(where(is.numeric))

corr <- round(cor(telco_df_numeric), 2)

corrplot(corr, type = "upper", ,
        tl.col = "black", tl.cex=0.7, title = "Correlation of all Numeric Attributes ",
        mar=c(0,0,1,0))

ggcorrplot(corr, type = "lower",
          lab = TRUE)
```

**# Correlation Matrix (Table)**

```
View(cor(telco_df_numeric))
```

**#Split the dataset into Test and Train data#**

```
set.seed(123)
trainIndex <- createDataPartition(telco_df$Churn, p = 0.70, list = FALSE , times=1 )
telco_df_train <- telco_df[trainIndex,]
telco_df_test <- telco_df[-trainIndex,]
```

**#Subset Method#**

```
library(leaps)

telco_df_subset_model <- regsubsets(Churn ~ ., data = telco_df_train,nbest=1)
summary_telco_df_subset <- summary(telco_df_subset_model)
summary_telco_df_subset

par(cex.lab = 1.5 , cex.axis=0.7, las=3 )

plot(telco_df_subset_model , scale = "adjr2")
```

```
#Reset par
dev.off()


#Model selection criteria: Adjusted R2, Cp and BIC#
data.frame(
  Adj.R2 = which.max(summary_telco_df_subset$adjr2),
  CP = which.min(summary_telco_df_subset$cp),
  BIC = which.min(summary_telco_df_subset$bic)
)

#Final 9 Predictors -
#MultipleLinesNo , OnlineSecurityYes , TechSupportYes , ContractOne year , ContractTwo Year ,
#PaperlessBilling , PaymentMethodElectronic ,  MonthlyCharges & Total Charges
#EDA:

#1. Gender count:

gender <- barplot(table(telco_df$gender),
        xlab = "Gender",
        ylab = "No. of customers",
        main = "Gender wise No. of customers ",col = "orange")

gender_Churn <- table(telco_df$Churn, telco_df$gender)
gender_churn_bp <- barplot(gender_Churn,
        main = "Gender wise Churn",
        xlab = "Gender", ylab = "No. of customers",
        col = c("pink", "red"),
        legend.text = rownames(gender_Churn),
        beside = TRUE,
        args.legend = list(title = "Churn", x = "topright"))

#2. Senior citizen - pie chart? churn rate

sc <- table(telco_df$SeniorCitizen)
sc_perc <- round((sc / nrow(telco_df)) * 100, digits = 2)
sc_perc_df <- data.frame(sc_perc)
colnames(sc_perc_df)[1] <- "Senior_Citizen"
colnames(sc_perc_df)[2] <- "Percentage"

ggplot(sc_perc_df, aes(x = "", y = Percentage, fill = Senior_Citizen)) +
  geom_col() + geom_text(aes(label = paste(Percentage,"%", sep = ""))
                         ,position = position_stack(vjust = 0.5)) +
  guides(fill = guide_legend(title = "Senior Citizen")) +
  coord_polar(theta = "y") +
  theme_void() +
  scale_fill_brewer(palette = "Set2")+
  ggtitle("Percentage of Senior Citizens")+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        text=element_text(size=12,  family="Comic Sans MS", color= "black"))

sc_Churn <- table(telco_df$Churn, telco_df$SeniorCitizen)
```

```
sc_churn_bp <- barplot(sc_Churn,
                       main = "SeniorCitizen vs churn",
                       xlab = "Senior Citizen", ylab = "No. of customers",
                       col = c("lightblue", "blue"),
                       legend.text = rownames(sc_Churn),
                       beside = TRUE,
                       args.legend = list(title = "Churn", x = "topright"))

sc_1 <- round((1393*100/5901),2)
sc_0 <- round((476*100/1142),2)

sc_churn_df <- data.frame (
  Percentage_of_churned_customers = c(23.61,41.68),
                          Senior_Citizen = c('No','Yes'))

ggplot(sc_churn_df, aes(x=Senior_Citizen, y=Percentage_of_churned_customers,
                    fill=Senior_Citizen))+
  geom_bar(stat="identity", color="black")+ theme_bw()+
  scale_fill_manual(values=c("#56B4E9", "#E69F00"))+
  ggtitle("Do Senior citizens have more likelihood to churn")+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        text=element_text(size=12,  family="Comic Sans MS", color= "black"))

#3. Tenure and churn
ggplot(telco_df, aes(x=tenure, color=Churn)) +
  geom_histogram(fill="white", alpha=0.5, position="identity")+
  scale_color_brewer(palette="Dark2")+
  ggtitle("Relationship between Tenure and Churn")+
  xlab("Tenure (in months)")+
  ylab("No. of customers")+theme_bw()+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        text=element_text(size=12,  family="Comic Sans MS", color= "black"))

#4. Which contract type has the most number of churns?
install.packages("hrbrthemes")
library(hrbrthemes)

install.packages("viridis")
library(viridis)

g <- ggplot(telco_df, aes(Contract))
g + geom_bar(aes(fill = Churn))+
  ggtitle("Which contract type has the most number of churns?")+
  xlab("Contract type")+ scale_fill_manual(values=c("#56B4E9", "#E69F00"))+
  ylab("No. of customers")+theme_bw()+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        text=element_text(size=12,  family="Comic Sans MS", color= "black"))

ggplot(telco_df, aes(x=tenure, color=Contract, fill=Contract)) +
  geom_histogram()+
  scale_fill_viridis(discrete=TRUE) +
  scale_color_viridis(discrete=TRUE) +
  theme_ipsum() +
```

```
  facet_grid(. ~ Contract)+
  ggtitle("Which contract type has the most no. of customers with a longer tenure?")+
  xlab("Tenure (in months)")+
  ylab("No. of customers")

#Which internet service results in maximum no. of customer churn?

internet_service <- ggplot(telco_df, aes(InternetService))
internet_service + geom_bar(aes(fill = Churn))+
  ggtitle("Which internet service has the most number of churns?")+
  xlab("Internet Service")+scale_fill_manual(values=c("#69b3a2", "#404080"))+
  ylab("No. of customers")+theme_bw()+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        text=element_text(size=12,  family="Comic Sans MS", color= "black"))

#Chi Square (Test of independence):

#3. Contract type and churn

#H0 : No. of churned customers are independent of the contract type
#H1: No. of churned customers are dependent on the contract type.

table(telco_df$Contract,telco_df$Churn)

#Significance level:0.05
alpha <- 0.05

#Create one vector for each row

r1 <- c(2220,1655)
r2 <- c(1307,166)
r3 <- c(1647,48)

#State the number of rows for the matrix
rows <- 3

#Create a matrix from the rows
mtrx <- matrix(c(r1,r2,r3), nrow = rows,byrow = TRUE)

#Name the rows and columns
rownames(mtrx) <- c("Month-to-month","One year","Two year")
colnames(mtrx) <- c("No","Yes")

#Run the test
result <- chisq.test(mtrx)

#Compare the p-value to alpha and make a decision
ifelse(result$p.value > alpha, "Fail to reject the null hypothesis",
       "Reject the null hypothesis")


#4. Total charges and Internet service (One-way ANOVA):
```

```
#H0 : The mean total charges is the same for all types of internet service.
#H1: The mean total charges is different for all types of internet service.

library("ggpubr")
ggboxplot(telco_df, x = "InternetService", y = "TotalCharges",
        color = "InternetService", palette = c("#00AFBB", "#E7B800", "#FC4E07"),
        ylab = "TotalCharges", xlab = "InternetService")

#Significance level:0.05
alpha <- 0.05

dsl <- sqldf("select * from telco_df where InternetService = 'DSL'")

fo <- sqldf("select * from telco_df where InternetService = 'Fiber optic'")

no <- sqldf("select * from telco_df where InternetService = 'No'")

dsl_df <- data.frame('total_charges' = dsl$TotalCharges,
                     'Internet_Service' = rep('DSL',nrow(dsl)),
                     stringsAsFactors = FALSE)

fo_df <- data.frame('total_charges' = fo$TotalCharges,
                    'Internet_Service' = rep('Fiber optic',nrow(fo)),
                    stringsAsFactors = FALSE)

no_df <- data.frame('total_charges' = no$TotalCharges,
                    'Internet_Service' = rep('No',nrow(no)),
                    stringsAsFactors = FALSE)


#Combine the dataframes into one
df <- rbind(dsl_df,fo_df,no_df)

df$Internet_Service <- as.factor(df$Internet_Service)

#Run the ANOVA test
anova <- aov(total_charges ~ Internet_Service,data = df)

#View the model summary
summary(anova)

#Save summary to an object
a.summary <- summary(anova)

#Determine if we should reject the null hypothesis
ifelse(p.value > alpha, "Fail to reject the null hypothesis",
     "Reject the null hypothesis")

#See differences
TukeyHSD(anova)

par(mar = c(8, 8, 8, 8))
plot(TukeyHSD(anova, conf.level=.95), las = 2)
```

```
preparedData <- f1[sapply(f1, class) == 'numeric']
corMatrix <- cor(preparedData)
corrplot(corMatrix,tl.cex=0.6)

#sub mean
f1$TotalCharges[is.na(f1$TotalCharges)] <- 2283.3
#h0: there is no relationship between gender and churn
#h1: there is a realtionship between gender and churn
femal <-  f1[f1$gender == "1",]
femal$Churn <- as.factor(femal$Churn)
summary(femal) #churn yes 939, no 2549
male <-  f1[f1$gender == "0",]
male$Churn <- as.factor(male$Churn)
summary(male) #churn yes 930, no 2625
genderChurn<-matrix(c(939,930,2549,2625),nrow=2, ncol=2)
chisq.test(genderChurn)

#h0: there is no relationship between partner and churn
#h1: there is a realtionship between partner and churn
partnerYes <-  f1[f1$Partner == "1",]
partnerYes$Churn <- as.factor(partnerYes$Churn)
summary(partnerYes) #churn yes 669, no 2733
partnerNo <-  f1[f1$Partner == "0",]
partnerNo$Churn <- as.factor(partnerNo$Churn)
summary(partnerNo) #churn yes 1200, no 2441
partnerChurn<-matrix(c(669,1200,2773,2441),nrow=2, ncol=2)
chisq.test(partnerChurn)

#h0: there is no relationship between PhoneService and churn
#h1: there is a realtionship between PhoneService and churn
phoneYes <-  f1[f1$PhoneService == "1",]
phoneYes$Churn <- as.factor(phoneYes$Churn)
summary(phoneYes) #churn 1699, no 4662
phoneNo <-  f1[f1$PhoneService == "0",]
phoneNo$Churn <- as.factor(phoneNo$Churn)
summary(phoneNo) #churn 170, no 512
phoneServiceChurn<-matrix(c(1699,170,4662,512),nrow=2, ncol=2)
chisq.test(phoneServiceChurn)

#Two-way ANOVA test churn dependent variable
#SeniorCitizen and Dependents as the independent variables.
#H0 no impact on churn
#H1 impact on churn
f1$SeniorCitizen <- as.factor(f1$SeniorCitizen)
f1$Dependents <- as.factor(f1$Dependents)
anova2 <- aov(Churn~Dependents+SeniorCitizen,data=f1)
summary(anova2)

#H0: There is no difference in the duration of tenure.
#H1: There is a difference in the duration of tenure.
tenureLess <-  f1[f1$tenure <= "32.7",]
tenureLess$Churn <- as.factor(tenureLess$Churn)
summary(tenureLess) #churn 1134, no 1849
```

```
tenureMore <-  f1[f1$tenure > "32.7",]
tenureMore$Churn <- as.factor(tenureMore$Churn)
summary(tenureMore) #churn 735, no 3325
tenureChurn<-matrix(c(1134,735,1849,3325),nrow=2, ncol=2)
chisq.test(tenureChurn)
```

**#Total charges and contract(One-way ANOVA):**

```
#H0: The mean total charges for each contract type are the same.
#H1: The mean total charges for at least one contract type is different.
# Significance level: 0.05
alpha <- 0.05

month_to_month <- telco_df[telco_df$Contract == 'Month-to-month', ]
one_year <- telco_df[telco_df$Contract == 'One year', ]
two_year <- telco_df[telco_df$Contract == 'Two year', ]

month_to_month_df <- data.frame('total_charges' = month_to_month$TotalCharges,
                        'Contract' = rep('Month-to-month', nrow(month_to_month)),
                        stringsAsFactors = FALSE)

one_year_df <- data.frame('total_charges' = one_year$TotalCharges,
                    'Contract' = rep('One year', nrow(one_year)),
                    stringsAsFactors = FALSE)

two_year_df <- data.frame('total_charges' = two_year$TotalCharges,
                    'Contract' = rep('Two year', nrow(two_year)),
                    stringsAsFactors = FALSE)

# Combine the dataframes into one
df <- rbind(month_to_month_df, one_year_df, two_year_df)

df$Contract <- as.factor(df$Contract)

# Run the ANOVA test
anova <- aov(total_charges ~ Contract, data = df)

# View the model summary
summary(anova)

# Save summary to an object
a.summary <- summary(anova)

# Determine if we should reject the null hypothesis
p.value <- a.summary[[1]]["Pr(>F)"]

#Determine if we should reject the null hypothesis
ifelse(p.value > alpha, "Fail to reject the null hypothesis",
      "Reject the null hypothesis")
```

**#Total charges and PaymentMethod(One-way ANOVA):**

```
# H0: There is no difference in the mean Total Charges for different Payment Methods.
# Ha: There is a difference in the mean Total Charges for different Payment Methods.
# Significance level: 0.05
```

```
alpha <- 0.05

electronic_check <- telco_df[telco_df$PaymentMethod == 'Electronic check', ]
mailed_check <- telco_df[telco_df$PaymentMethod == 'Mailed check', ]
bank_transfer <- telco_df[telco_df$PaymentMethod == 'Bank transfer (automatic)', ]
credit_card <- telco_df[telco_df$PaymentMethod == 'Credit card (automatic)', ]

electronic_check_df <- data.frame('total_charges' = electronic_check$TotalCharges,
                          'PaymentMethod' = rep('Electronic check', nrow(electronic_check)),
                          stringsAsFactors = FALSE)

mailed_check_df <- data.frame('total_charges' = mailed_check$TotalCharges,
                        'PaymentMethod' = rep('Mailed check', nrow(mailed_check)),
                        stringsAsFactors = FALSE)

bank_transfer_df <- data.frame('total_charges' = bank_transfer$TotalCharges,
                         'PaymentMethod' = rep('Bank transfer (automatic)', nrow(bank_transfer)),
                         stringsAsFactors = FALSE)

credit_card_df <- data.frame('total_charges' = credit_card$TotalCharges,
                       'PaymentMethod' = rep('Credit card (automatic)', nrow(credit_card)),
                       stringsAsFactors = FALSE)

# Combine the dataframes into one
df <- rbind(electronic_check_df, mailed_check_df, bank_transfer_df, credit_card_df)

df$PaymentMethod <- as.factor(df$PaymentMethod)

# Run the ANOVA test
anova <- aov(total_charges ~ PaymentMethod, data = df)

# View the model summary
summary(anova)

# Save summary to an object
a.summary <- summary(anova)

# Determine if we should reject the null hypothesis
p.value <- a.summary[[1]]["Pr(>F)"]

#Determine if we should reject the null hypothesis
ifelse(p.value > alpha, "Fail to reject the null hypothesis",
      "Reject the null hypothesis")

#build logistic regression model
preparedData <- f1[sapply(f1, class) == 'numeric']
str(preparedData)

trainIndex <- sort(sample(x = nrow(preparedData), size = nrow(preparedData) * 0.7))
train <- preparedData[trainIndex,]
test <- preparedData[-trainIndex,]
dim(train)
dim(test)
```

```
fit1<- glm(Churn~.,data = train,family = binomial(link = "logit"))
summary(fit1)

fit2<- glm(Churn~tenure+PhoneService+OnlineSecurity+DeviceProtection+
             TechSupport+PaperlessBilling+MonthlyCharges+TotalCharges,
           data = train,family = binomial(link = "logit"))
summary(fit2)

fit3<- glm(Churn~tenure+PhoneService+OnlineSecurity+
             TechSupport+PaperlessBilling+MonthlyCharges+
             SeniorCitizen+DeviceProtection,
           data = train,family = binomial(link = "logit"))
summary(fit3)

train$Churn= ifelse(train$Churn== "1", TRUE, FALSE)
train$Churn <- as.factor(train$Churn)

probabilities.train <- predict(fit3, newdata=train, type="response")
predicted.classes.min <- as.factor(ifelse(probabilities.train>=0.5, "TRUE", "FALSE"))
confusionMatrix(predicted.classes.min, train$Churn, positive = "TRUE")

ROC1 <- roc(train$Churn, probabilities.train)
plot(ROC1, col="red", ylab="Sensitivity - TP Rate", xlab= "Specificity - Fp Rate")

auc1 <- auc(ROC1)
Auc1

# forward selection
# Create Train and Test set - random sample (70/30 split)
trainIndex <- sort(sample(x = nrow(telco_df), size = nrow(telco_df) * 0.7))
sample_train <- telco_df[trainIndex,]
sample_test <- telco_df[-trainIndex,]

# Create Train and Test set - maintain % of event rate (70/30 split)
library(caret)
set.seed(123)
trainIndex <- createDataPartition(telco_df$Churn, p = 0.7, list = FALSE, times = 1)
telco_train <- telco_df[ trainIndex,]
telco_test <- telco_df[-trainIndex,]

library(MASS)
fullModel = lm(Churn ~ ., data = telco_test) # model with all variables
nullModel = lm(Churn ~ 1, data = telco_test) # model with the intercept only

summary(stepAIC(nullModel, # start with a model containing no variables
            direction = 'forward', # run forward selection
            scope = list(upper = fullModel, # the maximum to consider is a model with all variables
                         lower = nullModel), # the minimum to consider is a model with no variables
            trace = 0)) # do not show the step-by-step process of model selection

#Calculating the AIC and BIC:
library(stats)
```

```
stepAIC = stepAIC(nullModel, direction = 'forward',
                  scope = list(upper = fullModel, lower = nullModel),
                  trace = 0)
install.packages("Metrics")
library(Metrics)
rmse(telco_test$Churn, predict(stepAIC , telco_test))
rmse(telco_train$Churn, predict(stepAIC , telco_train))

#Calculating the ROC and AUC scores:
library(pROC)
model <- lm(Churn ~ ., data = telco_test)
predictions <- predict(model, type = "response")
predictions <- predict(fullModel, type = "response")
roc_object <- roc(telco_test$Churn, predictions)
auc(roc_object) # calculate the AUC score
plot(roc_object)

#Ridge regression:
set.seed(123)
trainIndex <- createDataPartition(telco_df$Churn, p = 0.7, list = FALSE,
                                  times = 1)

train <- telco_df[ trainIndex,]
test <- telco_df[-trainIndex,]

install.packages("glmnet")
library(glmnet)

train_x <- model.matrix(Churn ~., train)[,-1]
test_x <- model.matrix(Churn ~., test)[,-1]

train_y <- train$Churn
test_y <- test$Churn

#Ridge
#Find the best values of lambda

#Training:
set.seed(123)

cv_model_ridge_train <- cv.glmnet(train_x, train_y, alpha = 0,family="binomial")

cv_model_ridge_train$lambda.min
cv_model_ridge_train$lambda.1se
log(cv_model_ridge_train$lambda.min)
log(cv_model_ridge_train$lambda.1se)
plot(cv_model_ridge_train)

coef(cv_model_ridge_train, cv_model_ridge_train$lambda.min)

#Fit models based on lambda

model.min.ridge.train <- glmnet(train_x, train_y, alpha = 0,
```

```
                    lambda = cv_model_ridge_train$lambda.min)

#Display regression coeff
coef(model.min.ridge.train)

summary(model.min.ridge.train)

#Train set predictions and RMSE
install.packages("Metrics")
library(Metrics)

preds.train.ridge <- predict(model.min.ridge.train, s = cv_model_ridge_train$lambda.min,
                    newx = train_x)

sse <- sum((train_y - preds.train.ridge)^2)
sst <- sum((train_y - mean(train_y))^2)
train_RMSE_ridge = rmse(train_y, preds.train.ridge)
training_rsq <- 1 - (sse / sst)
training_rsq #0.27
train_RMSE_ridge #0.37

#Testing:

preds.test.ridge <- predict(model.min.ridge.train, s = cv_model_ridge_train$lambda.min,
                    newx = test_x)

sse <- sum((test_y - preds.test.ridge)^2)

sst <- sum((test_y - mean(test_y))^2)

test_RMSE_ridge = rmse(test_y, preds.test.ridge)

testing_rsq <- 1 - sse / sst

testing_rsq #0.31
test_RMSE_ridge #0.36

glmnet_cv_aicc <- function(fit, lambda = 'lambda.min'){
  whlm <- which(fit$lambda == fit[[lambda]])
  with(fit$glmnet.fit,
       {
         tLL <- nulldev - nulldev * (1 - dev.ratio)[whlm]
         k <- df[whlm]
         n <- nobs
         return(list('BIC' = log(n) * k - tLL))
       })
}

#AIC and BIC scores for Ridge regression
glmnet_cv_aicc(cv_model_ridge_train)


AIC.glmnet <- function(glm_fit) {
```

```
  chisqLR <- glm_fit$nulldev - deviance(glm_fit)

  chisqLR - 2*glm_fit$df
}

AIC.glmnet(model.min.ridge.train)

probabilities.test <- predict(model.min.ridge.test, newx = test_x, type = "response")
predicted.classes.min <- as.factor(ifelse(probabilities.test >= 0.5, "Yes", "No"))
library(pROC)
probabilities.test <- predict(model.min.ridge.train, newx = test_x, type = "response")

ROC1 <- roc(test_y,probabilities.test)

plot(ROC1, col = "blue", ylab = "TP Rate",
    xlab = "FP Rate")

auc <- auc(ROC1)




#####################################################################
#LASSO REGULARIZATION#
#####################################################################
#Split the dataset into Test and Train data#
set.seed(123)
trainIndex <- createDataPartition(telco_df$Churn, p = 0.70, list = FALSE , times=1 )
telco_df_train <- telco_df[trainIndex,]
telco_df_test <- telco_df[-trainIndex,]

#Define the Target variable for Train and Test Data set
y_train <- telco_df_train$Churn
y_test<- telco_df_test$Churn

#Define the matrix of predictor variables
x_train <- model.matrix(Churn ~ ., telco_df_train)[,-1]
x_test  <- model.matrix(Churn ~ ., telco_df_test)[,-1]

#Perform k-fold cross-validation to find optimal lambda value for Lasso #
set.seed(123)
lasso_cv <- cv.glmnet(x_train, y_train, family="binomial", alpha=1,nfolds = 10)
lasso_cv

#Plot of MSE by Lambda value
plot(lasso_cv)

#####################################################################
# Optimal Value of Lambda; Minimizes the Prediction Error
# Lambda Min - Minimizes out of sample loss
```

```
# Lambda 1SE - Largest value of Lambda within 1 Standard Error of Lambda Min.
#####################################################################
log(lasso_cv$lambda.min)
log(lasso_cv$lambda.1se)

#Find optimal value that minimize MSE
best_lambda1 <- lasso_cv$lambda.min
best_lambda1

#Fitting the Lasso Regression Model on the Training Set Using Best Lamba value(Min value)#
lasso_model_train_min  <- glmnet(x_train, y_train, alpha = 1, lambda = best_lambda1)
lasso_model_train_min
coef(lasso_model_train_min)
plot(coef(lasso_model_train_min))


#Fitting the Lasso Regression Model on the Training Set Using 1se Lamba value#
lasso_model_train_1se  <- glmnet(x_train, y_train, alpha = 1, lambda = lasso_cv$lambda.1se)
lasso_model_train_1se
coef(lasso_model_train_1se)
plot(coef(lasso_model_train_1se))

#Lasso regression model for making predictions on Train data#
predict_lasso_train <- predict(lasso_model_train_min, newx = x_train)

#RMSE for Lasso regression model against the Training set#
lasso_train_rmse <- rmse(y_train, predict_lasso_train)
lasso_train_rmse

#Lasso regression model for making predictions on Test data#
predict_lasso_test <- predict(lasso_model_train_min, newx = x_test)

#RMSE for Lasso regression model against the Test set#
lasso_test_rmse <- rmse(y_test, predict_lasso_test)
lasso_test_rmse


#AIC and BIC scores for Lasso regression

glmnet_cv_bic <- function(fit, lambda = 'lambda.min'){
  whlm <- which(fit$lambda == fit[[lambda]])
  with(fit$glmnet.fit,
       {
         tLL <- nulldev - nulldev * (1 - dev.ratio)[whlm]
         k <- df[whlm]
         n <- nobs
         return(list('BIC' = log(n) * k - tLL))
       })
}

#BIC SCORE#
glmnet_cv_bic(lasso_cv)
```

```
#AIC SCORE#
AIC.glmnet <- function(glm_fit) {
  chisqLR <- glm_fit$nulldev - deviance(glm_fit)

  chisqLR - 2*glm_fit$df
}
AIC.glmnet(lasso_model_train_min)


# Sum of Squares Total and Error for Training and Test Data
sse_train <- sum((y_train - predict_lasso_train)^2)
sst_train <- sum((y_train - mean(y_train))^2)

#Calculating Training R-square
training_rsq <- 1 - sse_train / sst_train
training_rsq

sse_test <- sum((y_test - predict_lasso_test)^2)
sst_test <- sum((y_test - mean(y_test))^2)

#Calculating Testing R-square
testing_rsq <- 1 - sse_test / sst_test
testing_rsq

# Lasso Model performance Metrics
lasso_model_perf <- data.frame(
  RMSE_test = lasso_test_rmse,
  R2_test =testing_rsq ,
  RMSE_train =lasso_train_rmse,
  R2_train=training_rsq)

lasso_model_perf


#ROC & AUC Curve#

ROC_curve <- roc(telco_df_test$Churn ,predict_lasso_test)
plot(ROC_curve, col = "Red", ylab = "Sensitivity - TruePositive Rate",
     xlab = "Sensitivity - FalsePositive Rate" , plot=TRUE , print.auc=TRUE)
```