D208 Multiple Regression for Predictive Modeling Performance Assessment, Task 1

Alexa R. Fisher

Western Governors University

Degree: M.S. Data Analytics

Table of Contents

Part I. Research Question for Data Analysis

A1. Research Question for Analysis

The research question for this thesis was, “ What variables contribute to the spread of Monthly Charges?”. This thesis was evaluated via the computation of multiple regression on the telecommunications dataset. When computing the regression model, the target or dependent variable was set to the attribute called “Monthly Charges”, which was the total amount charged to the customer every month for services. There were various explanatory variables utilized in forming the regression model. The variables included the following: “Population”, “Area”, “Children”, “Age”, “Income”, “Marital”, “Gender”, “Churn”, “Techie”, “Contract”, “Port_modem”, “Tablet”, “InternetService”, “Phone”, “Multiple”, “OnlineSecurity”, “OnlineBackup”, “DeviceProtection”, “TechSupport”, “StreamingTV”, “StreamingMovies”, “PaperlessBilling”, “PaymentMethod”, and “Tenure”. The predictive modeling technique called the Ordinary Least Squares or OLS regression model utilized the target and explanatory variables to estimate the coefficients of linear regression. The coefficients described the relationship between the dependent variable and the independent variables (Broeck, Introduction to Regression with statsmodels in Python, n.d.).

A2. Objective From Analysis

The object of this data analysis was to identify what variables contributed to the differences in monthly charges in the telecommunication dataset. This goal provided insightful information as to the sensitivity of fluctuation in charges based on various attributes. In the telecommunication services dataset, it was observed that customers vary in services, locations, tenure, and household sizing. The services included the addition of add-ons such as online security, online backup, device protection, and tech support. Services also included specific communications such as the internet, phone, or a combination of both. Another few variables were the type of area a client was located; how large their family size was or even how long they have been a client. Stakeholders may want to investigate if all of these have a positive or negative relationship as the monthly charge increases. It can provide insight on how to promote sales, retention, and other marketing techniques (Expert Panel, Forbes Agency Council, 2019).

Part II. Method Justification

B1. Summary for Assumptions of Multiple Regression Model

The assumptions of a multiple regression model can be summarized in the following points.

1. The explanatory variables in the dataset are independent (Bobbitt, 2021).
2. There is a linear relationship between the target variable and the explanatory variables (Bobbitt, 2021).
3. Normality of Residuals (Bobbitt, 2021)
4. There is no multicollinearity (Bobbitt, 2021).
5. The residuals have homoscedasticity (Bobbitt, 2021).

The above points were verified in several ways. The linear relationship, independence, and homoscedasticity were confirmed via visualizations such as scatterplots. There was a high likelihood of a linear relationship between the variables if the plotted points were found along a straight line. The same was viewed for homoscedasticity. If the residual plots had constant variance and the same error at every point, they reflected homoscedasticity (Bobbitt, 2021). Multicollinearity was shown when independent variables are too highly correlated to each other. It can cause unreliable results. Multicollinearity can be determined by calculating the Variance Inflation Factor or VIF score. If the score is higher than ten there is a higher chance of multicollinearity. The residuals confirmed normality by reviewing the Prob(Omnibus) within the regression summary. A value of less then 0.05 indicated the residuals were not normally distributed.

B2. Benefits of Using Python

There were numerous benefits of using the selected programming language. Python was an object-oriented, general-purpose programming language. It allowed for libraries and packages to be imported for data analysis computations and visualizations. This tool had a robust collection of packages such as Pandas, NumPy, Sklearn, Scipy, Statsmodels, Matplotlib, and Seaborn to name a few. All of these allowed for datasets to be explored and provided insightful information. Pandas provided tools to explore, clean, and analyze data. NumPy allowed the ability to work with arrays during data cleaning and data wrangling. The Seaborn and Matplotlib provided ways to visualize data during all stages. Sklearn, Scipy, and Statsmodels allowed ways to compute statistics within the Python environment. The statistics computed within the packages also had various toolsets. The toolsets were noted as variance inflation factor, ordinary least squares, as well as train test split to name a few. The VIF method was utilized for checking on multicollinearity. The OLS regression models showed the coefficients within the dataset compared to the dependent variable. The Linear Regression and Train Test Split packages provided data for residual plotting. The noted packages permitted each step of analysis such as data cleaning, data exploration, data wrangling, and predictive modeling.

B3. Justification for Multiple Regression Technique

The appropriate technique to analyze the presented research question, " What variables contribute to the spread of Monthly Charges?" was multiple regression. Multiple regression was described as having multiple explanatory variables in a model (Broeck, Intermediate Regression with statsmodels in Python, n.d.). The additional variables presented more insight into the relationship between the target and explanatory variables. Better predictions were provided as well. The telecommunication dataset target variable, "MonthlyCharge" varies from customer to customer. Comparing the independent variables against this target variable revealed intuition on what can affect the monthly charge. This intuition can allow stakeholders to make future marketing and sales decisions (Expert Panel, Forbes Agency Council, 2019).

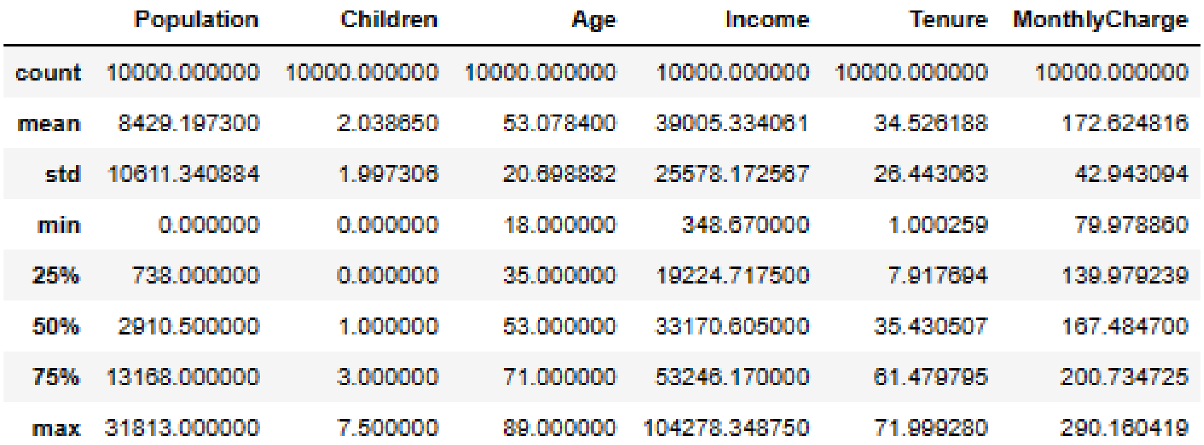
Part III. Data Preparation

C1. Data Preparation: Goals

There were several data preparation and manipulation goals for the telecommunication dataset to analyze the research thesis. The main goals of the data preparation included data cleaning, data exploration, and data wrangling. The data cleaning process ensured that the target and explanatory variables were free of any duplicates, null values, and outliers. It was resolved by manipulating the data to find these features. Then, the features were treated accordingly by either dropping the values or resolving them in another way. For example, in the case of outliers found, the values were not dropped, yet treated via the IQR method. This allowed the distribution of the data to remain the same across all variables. The data exploration goal was resolved via summary statistics and visualizations such as boxplots, scatterplots, and bar plots. The explanatory variables were explored by viewing their univariate and bivariate statistics along with their degree of cardinality. Data Wrangling was presented by re-expressing the categorical variables and various encoding methods. The ordinal and one-hot encoding were applied to the categorical variables. The conversion to numerical values put the attributes in the ideal format for further predictive modeling. The completion of these goals allowed for multiple regression techniques to be used.

C2. Summary of Statistics

The summary of statistics for this data set, telecommunications, was a key part of the exploratory data analysis. This was provided by using the .describe() function in Python. The .describe() function provided the mean, standard deviation, count, quantile information of each percentage grouping, the minimum, and the maximum value for each quantitative variable. The quantitative explanatory variables were "Population", "Children", "Age", "Income", and "Tenure". The mean described the measure of the location of the values in the variable by averaging the entries. The standard deviation noted the average amount of variability. It explained how far on average a value lies from the mean. The count is the total of entries in the variable. The minimum was the lowest value found in the variable. The maximum was the highest value found in the variable. The quantile percentages were split into three percentiles. The 25% percentile was the lower quantile. The 50% percentile was the same as the median, which is the middle value in the sorted list. The 75% percentile was the upper quantile. In the below output, the count was the same across all the quantitative explanatory variables as there were no missing values.



	Population	Children	Age	Income	Tenure	MonthlyCharge
count	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
mean	8429.197300	2.038650	53.078400	39005.334061	34.526188	172.624816
std	10611.340884	1.997306	20.698882	25578.172567	26.443063	42.943094
min	0.000000	0.000000	18.000000	348.670000	1.000259	79.978860
25%	738.000000	0.000000	35.000000	19224.717500	7.917694	139.979239
50%	2910.500000	1.000000	53.000000	33170.605000	35.430507	167.484700
75%	13168.000000	3.000000	71.000000	53246.170000	61.479795	200.734725
max	31813.000000	7.500000	89.000000	104278.348750	71.999280	290.160419

Summary statistics were also described using analysis of variance or ANOVA for the bivariate statistical analysis for the target variable, “MonthlyCharge” against each of the explanatory categorical variables to allow them to be pictured, “Area”, “Marital”, “Gender”, “Churn”, “Techie”, “Contract”, “Port_modem”, “Tablet”, “InternetService”, “Phone”, “Multiple”, “OnlineSecurity”, “OnlineBackup”, “DeviceProtection”, “TechSupport”, “StreamingTV”, “StreamingMovies”, “PaperlessBilling”, and “PaymentMethod”. The one-way ANOVA was used to gain information about the relationship between dependent and independent variables. It described if the means of the variables were equal. The summary provided two values, the F-Statistic and the p-value. The F-statistic was the ratio between the variations of variables’ means and the variation within variables. The larger the F-statistic, the greater the difference. The opposite was true as well. The smaller the F-statistic the more jointly significant the variables were to each other. The p-value offered insight into if there was a significant difference between the variables. In the below code, we concluded that “Area”, “Marital”, “Gender”, “Techie”, “Contract”, “Port_modem”, “Tablet”, “InternetService”, “Phone”, “OnlineSecurity”, “OnlineBackup”, “DeviceProtection”, “TechSupport”, “PaperlessBilling”, and “PaymentMethod” all have p-values that caused us to fail to reject the null hypothesis and there was not enough evidence of suggesting a significant difference between the means of the variables against the target.

```
anova('Area', 'MonthlyCharge')
F  onewayResult(statistic=0.103728065024303, pvalue=0.9014713756427815)

anova('Marital', 'MonthlyCharge')
F  onewayResult(statistic=0.5735193600068232, pvalue=0.6818537061016405)

anova('Gender', 'MonthlyCharge')
F_onewayResult(statistic=0.4427132769463601, pvalue=0.6423039321761266)

anova('Churn', 'MonthlyCharge')
F  onewayResult(statistic=1615.1940392182648, pvalue=0.0)

anova(‘Techie’, 'MonthlyCharge')
F  onewayResult(statistic=0.5161180276428803, pvalue=0.47251912274138896)

anova('Contract', 'MonthlyCharge')
F  onewayResult(statistic=0.9876286273257879, pvalue=0.37249522134022617)

anova(Port_modem, 'MonthlyCharge')
F  onewayResult(statistic=0.0002423751239766553, pvalue=0.987579025235609
8)

anova('Tablet', 'MonthlyCharge')
F  onewayResult(statistic=0.5299539211519984, pvalue=0.46664328080158246)

anova('InternetService', 'MonthlyCharge')
F  onewayResult(statistic=532.6418693576522, pvalue=1.550441148202294e-22
0)

anova('Phone', 'MonthlyCharge')
F  onewayResult(statistic=3.6018697419456513, pvalue=0.057743401981852384
)

anova('Multiple', 'MonthlyCharge')
F  onewayResult(statistic=1750.2519831572727, pvalue=0.0)

anova('OnlineSecurity', 'MonthlyCharge')
F  onewayResult(statistic=22.83294385591984, pvalue=1.7922673421231386e-0
6)
```

```
anova('OnlineBackup', 'MonthlyCharge')
F  onewayResult(statistic=721.5203386837194, pvalue=1.608783095447509e-15
3)

anova('DeviceProtection', 'MonthlyCharge')
F  onewayResult(statistic=271.9759582091938, pvalue=2.624785165819671e-60
)

anova('TechSupport', 'MonthlyCharge')
F  onewayResult(statistic=146.82000446459222, pvalue=1.4759008600208933e-
33)
anova('StreamingTV', 'MonthlyCharge')
F  onewayResult(statistic=3030.832978655445, pvalue=0.0)

anova('StreamingMovies', 'MonthlyCharge')
F  onewayResult(statistic=5866.89985084985, pvalue=0.0)

anova('PaperlessBilling', 'MonthlyCharge')
F_onewayResult(statistic=0.018162589890714235, pvalue=0.8927975036274793
)

anova('PaymentMethod', 'MonthlyCharge')
F  onewayResult(statistic=0.6704200251897235, pvalue=0.5700939802234821)
```

C3. Data Preparation Steps

The data preparation steps utilized on the telecommunication dataset were summarized in the following phases. The phases included data cleaning, data exploration, and data wrangling. Each of the noted phases needed to be completed on the dataset before doing any predictive modeling.

First, the data cleaning process of removing duplicates, null values, and outliers. The .info() function was used to visually see all of the variable names in the dataset. It allowed a visual inspection to identify any duplicated attributes. After inspecting the variable names, the .duplicated() function was applied to check the dataset for any duplicated values. This function further was filtered by specifying a variable or attribute. After the duplicates were resolved, the

.isnull() function was used to find any null or missing values in the telecommunication dataset. The null values were summed to provide a total count per variable by applying the .sum() function in conjunction with the .isnull() function. In the case of this dataset, there were no null values so the next step in the data-cleaning process was detecting and treating outliers. Outliers were located via a created function. The function provided the total amount of outliers per the quantitative variable. This representation provided the volume of outliers as well as the notation for the minimum and maximum outlier values for each of those variables. The function was labeled as “find_outliers”. It utilized the interquartile range or IQR method. The IQR was mathematically identified as the first quantile subtracted from the third quantile with separate equations to calculate the lower and upper bounds, i.e., Q1 – 1.5 * IQR = lower bound value and Q3 + 1.5 * IQR = upper bound value In the IQR method, any values that were noted as less than the lower bound and greater than the upper bound are considered outliers. The outliers were treated by creating another function called “find_boundary”. This utilized the same IQR method to return the upper limit and lower limit of each continuous explanatory variable. Once the limits were identified, the NumPy .where() function was used to manipulate the values found outside of those limits and set them to the specific limit accordingly. For example, if a value was found above the upper limit it was set to the upper limit for that variable and vice versa for the lower limit.

Please see the below annotated code for finding duplicates, null values, and outliers.

```
# checking for duplicates
data.duplicated()
duplicates = data.duplicated('Customer_id')
data_duplicated = data[duplicates].sort_values(by='Customer_id')
print(data_duplicated[['Customer_id']])
```

```
#checking for null values
data.isnull().sum()

#finding outliers
def find_outliers(df, var):
    q1 = df[var].quantile(0.25)
    q3 = df[var].quantile(0.75)
    IQR = q3 - q1
    lowerbound = q1-(1.5*IQR)
    upperbound = q3+(1.5*IQR)
    outliers = df[var][((df[var] < (lowerbound)) | (df[var] > (upperbound)))]
    return outliers

#running created function on quantitative variables
outliers = find_outliers(data, 'Population')
print("number of outliers in Population: "+ str(len(outliers)))
print("max outlier value: "+ str(outliers.max()))
print("min outlier value: "+ str(outliers.min()))

outliers = find_outliers(data, 'Children')
print("number of outliers in Children: "+ str(len(outliers)))
print("max outlier value: "+ str(outliers.max()))
print("min outlier value: "+ str(outliers.min()))

outliers = find_outliers(data, 'Age')
print("number of outliers in Age: "+ str(len(outliers)))
print("max outlier value: "+ str(outliers.max()))
print("min outlier value: "+ str(outliers.min()))
```

```
outliers = find_outliers(data, 'Income')
print("number of outliers in Income: "+ str(len(outliers)))
print("max outlier value: "+ str(outliers.max()))
print("min outlier value: "+ str(outliers.min()))

outliers = find_outliers(data, 'Outage_sec_perweek')
print("number of outliers in Outage_sec_perweek: "+ str(len(outliers)))
print("max outlier value: "+ str(outliers.max()))
print("min outlier value: "+ str(outliers.min()))

outliers = find_outliers(data, 'Email')
print("number of outliers in Email: "+ str(len(outliers)))
print("max outlier value: "+ str(outliers.max()))
print("min outlier value: "+ str(outliers.min()))

outliers = find_outliers(data, 'Contacts')
print("number of outliers in Contacts: "+ str(len(outliers)))
print("max outlier value: "+ str(outliers.max()))
print("min outlier value: "+ str(outliers.min()))

outliers = find_outliers(data, 'Yearly_equip_failure')
print("number of outliers in Yearly_equip_failure: "+ str(len(outliers)))
print("max outlier value: "+ str(outliers.max()))
print("min outlier value: "+ str(outliers.min()))

outliers = find_outliers(data, 'Tenure')
print("number of outliers in Tenure: "+ str(len(outliers)))
print("max outlier value: "+ str(outliers.max()))
print("min outlier value: "+ str(outliers.min()))
```

```
outliers = find_outliers(data, 'MonthlyCharge')
print("number of outliers in MonthlyCharge: "+ str(len(outliers)))
print("max outlier value: "+ str(outliers.max()))
print("min outlier value: "+ str(outliers.min()))

outliers = find_outliers(data, 'Bandwidth_GB_Year')
print("number of outliers in Bandwidth_GB_Year: "+ str(len(outliers)))
print("max outlier value: "+ str(outliers.max()))
print("min outlier value: "+ str(outliers.min()))

#treating outliers found.

def find_boundary(df, var):
   Q1 = df[var].quantile(0.25)
   Q3 = df[var].quantile(0.75)
   IQR = Q3-Q1
   lower = Q1-(1.5*IQR)
   upper = Q3+(1.5*IQR)
   return lower , upper



lower_pop, upper_pop = find_boundary(data, 'Population' )
print("Upper limit for population is" , upper_pop)
print("Lower limit for population is" , lower_pop)
data.Population = np.where(data.Population > upper_pop, upper_pop,
                       np.where(data.Population < lower_pop, lower_pop,
                       data.Population))

lower_kid, upper_kid = find_boundary(data, 'Children')
print("Upper limit for children is" , upper_kid)
print("Lower limit for children is" , lower_kid)

data.Children = np.where(data.Children > upper_kid, upper_kid,

                     np.where(data.Children < lower_kid, lower_kid,
                     data.Children))


lower_inc, upper_inc = find_boundary(data, 'Income')

print("Upper limit for Income is" , upper_inc)

print("Lower limit for Income is" , lower_inc)

data.Income = np.where(data.Income > upper_inc, upper_inc,

                 np.where(data.Income < lower_inc, lower_inc, data.Income))


lower_osp, upper_osp = find_boundary(data, 'Outage_sec_perweek')

print("Upper limit for Outage_sec_perweek is" , upper_osp)

print("Lower limit for Outage_sec_perweek is" , lower_osp)

data.Outage_sec_perweek = np.where(data.Outage_sec_perweek > upper_osp,
upper_osp,

                               np.where(data.Outage_sec_perweek < lower_osp,
                        lower_osp, data.Outage_sec_perweek))


lower_eml, upper_eml = find_boundary(data, 'Email')

print("Upper limit for email is" , upper_eml)

print("Lower limit for email is" , lower_eml)

data.Email = np.where(data.Email > upper_eml, upper_eml,

                     np.where(data.Email < lower_eml, lower_eml, data.Email))


lower_contct, upper_contct = find_boundary(data, 'Contacts')

print("Upper limit for contacts is" , upper_contct)

print("Lower limit for contacts is" , lower_contct)

data.Contacts = np.where(data.Contacts > upper_contct, upper_contct,

                np.where(data.Contacts < lower_contct, lower_contct,
                data.Contacts))
```

```
        lower_yef, upper_yef = find_boundary(data, 'Yearly_equip_failure')
        print("Upper limit for Yearly_equip_failure is" , upper_yef)
        print("Lower limit for Yearly_equip_failure is" , lower_yef)
        data.Yearly_equip_failure = np.where(data.Yearly_equip_failure > upper_yef,
    upper_yef,
                              np.where(data.Yearly_equip_failure < lower_yef, lower_yef,
                              data.Yearly_equip_failure))
```

Second, the data exploration process included identifying high cardinality as well as univariate and bivariate statistics on the explanatory variables. The .nunique() function was used to provide the count of unique values in each variable. Once each value count was found the categorical variables with high cardinality and quantitative values that were deemed unnecessary to the research these were removed. High cardinality was deemed as values over five. The variables were dropped using the .drop() function.

Univariate statistics were completed on both the quantitative variables as well as the qualitative explanatory variables that were remaining. The seaborn library was used to make histograms for the quantitative variables of “Population”, “Children”, “Age”, “Income”, and “Tenure”. The qualitative explanatory variables univariate statistics are shown via a bar plot. The bar plot plotted after grouping the variable by the size. This was completed by using the .groupby() and .size() function. The Matplotlib library was used in providing the graphical visualization of the specified data. The qualitative variables that were plotted were the following: “Area”, “Marital”, “Gender”, “Churn”, “Techie”, “Contract”, “Port_modem”, “Tablet”, “InternetService”, “Phone”, “Multiple”, “OnlineSecurity”, “OnlineBackup”, “DeviceProtection”, “TechSupport”, “StreamingTV”, “StreamingMovies”, “PaperlessBilling”, and “PaymentMethod”. The bivariate statistics were visualized based on the type of explanatory variable. As the target variable was a continuous variable, when populating the bivariate statistics against the quantitative variables the scatterplot graphical representation was used. The Seaborn .histplot() function was applied to the categorical variables to graphical visualize the distributions. In all cases the x-axis value was the target variable of “MonthlyCharge”. The y-axis values were changed to each of the explanatory variables.

Please see the below annotated code for the data exploration phase.

```
print(f'CaseOrder: {data.CaseOrder.nunique()}')
print(f'Customer_id: {data.Customer_id.nunique()}')
print(f'Interaction: {data.Interaction.nunique()}')
print(f'UID: {data.UID.nunique()}')
print(f'City: {data.City.nunique()}')
print(f'State: {data.State.nunique()}')
print(f'County: {data.County.nunique()}')
print(f'Zip: {data.Zip.nunique()}')
print(f'Lat: {data.Lat.nunique()}')
print(f'Lng: {data.Lng.nunique()}')
print(f'Population: {data.Population.nunique()}')
print(f'Area: {data.Area.nunique()}')
print(f'TimeZone: {data.TimeZone.nunique()}')
print(f'Job: {data.Job.nunique()}')
print(f'Children: {data.Children.nunique()}')
print(f'Age: {data.Age.nunique()}')
print(f'Income: {data.Income.nunique()}')
print(f'Marital: {data.Marital.nunique()}')
print(f'Gender: {data.Gender.nunique()}')
print(f'Churn: {data.Churn.nunique()}')
print(f'Outage_sec_perweek: {data.Outage_sec_perweek.nunique()}')
print(f'Email: {data.Email.nunique()}')
print(f'Contacts: {data.Contacts.nunique()}')
print(f'Yearly_equip_failure: {data.Yearly_equip_failure.nunique()}')
print(f'Techie: {data.Techie.nunique()}')
print(f'Contract: {data.Contract.nunique()}')
print(f'Port_modem: {data.Port_modem.nunique()}')
print(f'Tablet: {data.Tablet.nunique()}')
print(f'InternetService: {data.InternetService.nunique()}')
print(f'Phone: {data.Phone.nunique()}')
print(f'Multiple: {data.Multiple.nunique()}')
print(f'OnlineSecurity: {data.OnlineSecurity.nunique()}')
print(f'OnlineBackup: {data.OnlineBackup.nunique()}')
print(f'DeviceProtection: {data.DeviceProtection.nunique()}')
print(f'TechSupport: {data.TechSupport.nunique()}')
print(f'StreamingTV: {data.StreamingTV.nunique()}')
print(f'StreamingMovies: {data.StreamingMovies.nunique()}')
print(f'PaperlessBilling: {data.PaperlessBilling.nunique()}')
print(f'PaymentMethod: {data.PaymentMethod.nunique()}')
print(f'Tenure: {data.Tenure.nunique()}')
print(f'MonthlyCharge: {data.MonthlyCharge.nunique()}')
print(f'Bandwidth_GB_Year: {data.Bandwidth_GB_Year.nunique()}')
print(f'Timely_Respd: {data.Timely_Respd.nunique()}')
print(f'Timely_Fixes: {data.Timely_Fixes.nunique()}')
print(f'Timely_Replc: {data.Timely_Replc.nunique()}')
print(f'Reliability: {data.Reliability.nunique()}')
print(f'Options: {data.Options.nunique()}')
print(f'Respect_Resp: {data.Respect_Resp.nunique()}')
print(f'Courteous_Exch: {data.Courteous_Exch.nunique()}')
print(f'Evidence_ActListen: {data.Evidence_ActListen.nunique()}')
#dropping categorical variables with high cardinality and unneeded variables
```

```
data.drop(['CaseOrder', 'Customer_id', 'Interaction', 'UID', 'City', 'State',
    'County', 'Zip', 'Lat', 'Lng', 'TimeZone', 'Job',
    'Outage_sec_perweek', 'Email', 'Contacts', 'Yearly_equip_failure', 'Bandwidth_GB_Year',
    'Timely_Respd', 'Timely_Fixes', 'Timely_Replc',
    'Reliability', 'Options', 'Respect_Resp', 'Courteous_Exch',
    'Evidence_ActListen'], axis=1, inplace=True)

#identification of distribution via univariate statistics for quantitative explanatory variables via histogram
sns.histplot(data.Population, kde=True)
sns.histplot(data.Children, kde=True)
sns.histplot(data.Age, kde=True)
sns.histplot(data.Income, kde=True)
sns.histplot(data.Tenure, kde=True)

#identification of distribution via univariate statistics for categorical explanatory variables via barplot
groupedArea= data.groupby(by='Area').size()
groupedArea
%matplotlib inline
groupedArea.plot.bar()
groupedMarital = data.groupby(by='Marital').size()
groupedMarital
%matplotlib inline
groupedMarital.plot.bar()
groupedGender = data.groupby(by='Gender').size()
groupedGender
%matplotlib inline
groupedGender.plot.bar()
groupedChurn = data.groupby(by='Churn').size()
```

```
groupedChurn
%matplotlib inline
groupedChurn.plot.bar()
groupedTechie = data.groupby(by='Techie').size()
groupedTechie
%matplotlib inline
groupedTechie.plot.bar()
groupedContract = data.groupby(by='Contract').size()
groupedContract
%matplotlib inline
groupedContract.plot.bar()
groupedPort = data.groupby(by='Port_modem').size()
groupedPort
%matplotlib inline
groupedPort.plot.bar()
groupedTablet = data.groupby(by='Tablet').size()
groupedTablet
%matplotlib inline
groupedTablet.plot.bar()
groupedInternetService = data.groupby(by='InternetService').size()
groupedInternetService
%matplotlib inline
groupedInternetService.plot.bar()
groupedPhone = data.groupby(by='Phone').size()
groupedPhone
%matplotlib inline
groupedPhone.plot.bar()
groupedMultiple = data.groupby(by='Multiple').size()
groupedMultiple
%matplotlib inline
```

```
groupedMultiple.plot.bar()
groupedSecurity = data.groupby(by='OnlineSecurity').size()
groupedSecurity
%matplotlib inline
groupedSecurity.plot.bar()
groupedBackup = data.groupby(by='OnlineBackup').size()
groupedBackup
%matplotlib inline
groupedBackup.plot.bar()
groupedDevice = data.groupby(by='DeviceProtection').size()
groupedDevice
%matplotlib inline
groupedDevice.plot.bar()
groupedSupport = data.groupby(by='TechSupport').size()
groupedSupport
%matplotlib inline
groupedSupport.plot.bar()
groupedStreamingTV = data.groupby(by='StreamingTV').size()
groupedStreamingTV
%matplotlib inline
groupedStreamingTV.plot.bar()
groupedStreamingMovies = data.groupby(by='StreamingMovies').size()
groupedStreamingMovies
%matplotlib inline
groupedStreamingMovies.plot.bar()
groupedBilling = data.groupby(by='PaperlessBilling').size()
groupedBilling
%matplotlib inline
groupedBilling.plot.bar()
groupedPayment = data.groupby(by='PaymentMethod').size()
```

```
groupedPayment
%matplotlib inline
groupedPayment.plot.bar()

#identification of distributions via bivariate statistics
# 2 continuous variables via scatterplot
pm_scatter = data.plot.scatter(x='MonthlyCharge', y='Population')
pm_scatter.set_xlabel('Monthly Charge')
pm_scatter.set_ylabel('Population')
cm_scatter = data.plot.scatter(x='MonthlyCharge', y='Children')
cm_scatter.set_xlabel('Outages per Week In Seconds')
cm_scatter.set_ylabel('Children')
am_scatter = data.plot.scatter(x='MonthlyCharge', y='Age')
am_scatter.set_xlabel('Monthly Charge')
am_scatter.set_ylabel('Age')
im_scatter = data.plot.scatter(x='MonthlyCharge', y='Income')
im_scatter.set_xlabel('Monthly Charge')
im_scatter.set_ylabel('Income')
tm_scatter = data.plot.scatter(x='MonthlyCharge', y='Tenure')
tm_scatter.set_xlabel('Monthly Charge')
tm_scatter.set_ylabel('Tenure')

#identification of distributions via bivariate statistics via one way Anova /histogram visualizations
def anova(feature, label):
   groups = data[feature].unique()
   group_values = []
   for group in groups:
      group_values .append(data[data[feature]==group][label])
   return stats.f_oneway(*group_values)
```

```
anova('Area', 'MonthlyCharge')
sns.histplot(data=data, x='MonthlyCharge', hue='Area', kde=False)
plt.show()
anova('Marital', 'MonthlyCharge')
sns.histplot(data=data, x='MonthlyCharge', hue='Marital', kde=False)
plt.show()
anova('Gender', 'MonthlyCharge')
sns.histplot(data=data, x='MonthlyCharge', hue='Gender', kde=False)
plt.show()
anova('Churn', 'MonthlyCharge')
sns.histplot(data=data, x='MonthlyCharge', hue='Churn', kde=False)
plt.show()
anova('Techie', 'MonthlyCharge')
sns.histplot(data=data, x='MonthlyCharge', hue='Techie', kde=False)
plt.show()
anova('Contract', 'MonthlyCharge')
sns.histplot(data=data, x='MonthlyCharge', hue='Contract', kde=False)
plt.show()
anova('Port_modem', 'MonthlyCharge')
sns.histplot(data=data, x='MonthlyCharge', hue='Port_modem', kde=False)
plt.show()
anova('Tablet', 'MonthlyCharge')
sns.histplot(data=data, x='MonthlyCharge', hue='Tablet', kde=False)
plt.show()
anova('InternetService', 'MonthlyCharge')
sns.histplot(data=data, x='MonthlyCharge', hue='InternetService', kde=False)
plt.show()
anova('Phone', 'MonthlyCharge')
sns.histplot(data=data, x='MonthlyCharge', hue='Phone', kde=False)
```

```
plt.show()
anova('Multiple', 'MonthlyCharge')
sns.histplot(data=data, x='MonthlyCharge', hue='Multiple', kde=False)
plt.show()
anova('OnlineSecurity', 'MonthlyCharge')
sns.histplot(data=data, x='MonthlyCharge', hue='OnlineSecurity', kde=False)
plt.show()
anova('OnlineBackup', 'MonthlyCharge')
sns.histplot(data=data, x='MonthlyCharge', hue='OnlineBackup', kde=False)
plt.show()
anova('DeviceProtection', 'MonthlyCharge')
sns.histplot(data=data, x='MonthlyCharge', hue='DeviceProtection', kde=False)
plt.show()
anova('TechSupport', 'MonthlyCharge')
sns.histplot(data=data, x='MonthlyCharge', hue='TechSupport', kde=False)
plt.show()
anova('StreamingTV', 'MonthlyCharge')
sns.histplot(data=data, x='MonthlyCharge', hue='StreamingTV', kde=False)
plt.show()
anova('StreamingMovies', 'MonthlyCharge')
sns.histplot(data=data, x='MonthlyCharge', hue='StreamingMovies', kde=False)
plt.show()
anova('PaperlessBilling', 'MonthlyCharge')
sns.histplot(data=data, x='MonthlyCharge', hue='PaperlessBilling', kde=False)
plt.show()
anova('PaymentMethod', 'MonthlyCharge')
sns.histplot(data=data, x='MonthlyCharge', hue='PaymentMethod', kde=False)
plt.show()
```

The final stage of the data preparation phase was the data wrangling process. In this process the categorical variables are re-expressed into numerical variables. This is completed in two ways. The categorical variables that have only a yes or no value are converted using the ordinal encoding. The cat.codes method is used to change each “No” value to “0” and each “Yes” to “1”. The following variables were converted using this method, “Churn”, “Techie”, “Port_modem”, “Tablet”, “Phone”, “Multiple”, “OnlineSecurity”, “OnlineBackup”, “DeviceProtection”, “TechSupport”, “StreamingTV”, “StreamingMovies”, and “PaperlessBilling”.

The nominal categorical explanatory variables were converted using the Pandas get_dummies method. This method passed through the attributes and created dummy variables of the original attributes and their unique values. It was further expanded by putting a “1” in the column of the variable that previously was noted. Once completed, one of the columns in the grouping were dropped to reduce multicollinearity.

The last part of the data wrangling process was to check for multicollinearity. This was checked by using the variance inflation factor (VIF) method. The VIF value over ten has a high likelihood of multicollinearity. In this dataset there were no variables with a VIF score of greater than ten.

Please see the below annotated code for the data wrangling phase.

```
#re-expression of categorical variables
#convert ordinal categorical to numerical

data['Churn']=data['Churn'].astype('category')
data['Churn']=data['Churn'].cat.codes
```

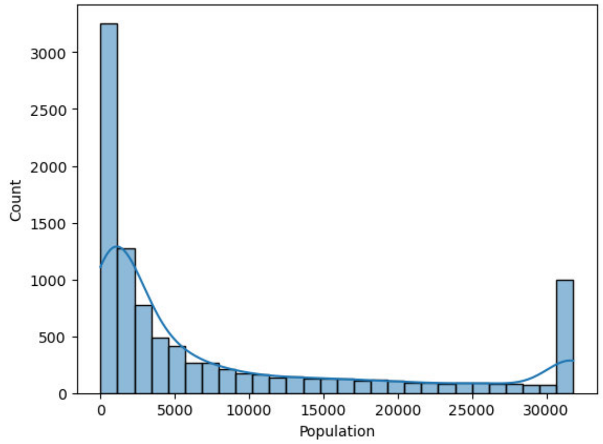
```
data['Techie']=data['Techie'].astype('category')
data['Techie']=data['Techie'].cat.codes
data['Port_modem']=data['Port_modem'].astype('category')
data['Port_modem']=data['Port_modem'].cat.codes
data['Tablet']=data['Tablet'].astype('category')
data['Tablet']=data['Tablet'].cat.codes
data['Phone']=data['Phone'].astype('category')
data['Phone']=data['Phone'].cat.codes
data['Multiple']=data['Multiple'].astype('category')
data['Multiple']=data['Multiple'].cat.codes
data['OnlineSecurity']=data['OnlineSecurity'].astype('category')
data['OnlineSecurity']=data['OnlineSecurity'].cat.codes
data['OnlineBackup']=data['OnlineBackup'].astype('category')
data['OnlineBackup']=data['OnlineBackup'].cat.codes
data['DeviceProtection']=data['DeviceProtection'].astype('category')
data['DeviceProtection']=data['DeviceProtection'].cat.codes
data['TechSupport']=data['TechSupport'].astype('category')
data['TechSupport']=data['TechSupport'].cat.codes
data['StreamingTV']=data['StreamingTV'].astype('category')
data['StreamingTV']=data['StreamingTV'].cat.codes
data['StreamingMovies']=data['StreamingMovies'].astype('category')
data['StreamingMovies']=data['StreamingMovies'].cat.codes
data['PaperlessBilling']=data['PaperlessBilling'].astype('category')
data['PaperlessBilling']=data['PaperlessBilling'].cat.codes
#utilizing get_dummies to convert nominal categorical to numerical


data= pd.get_dummies(data, columns=['Area', 'Marital', 'Gender', 'Contract',
                         'InternetService', 'PaymentMethod'], prefix_sep=" " ,
drop_first=True)
```

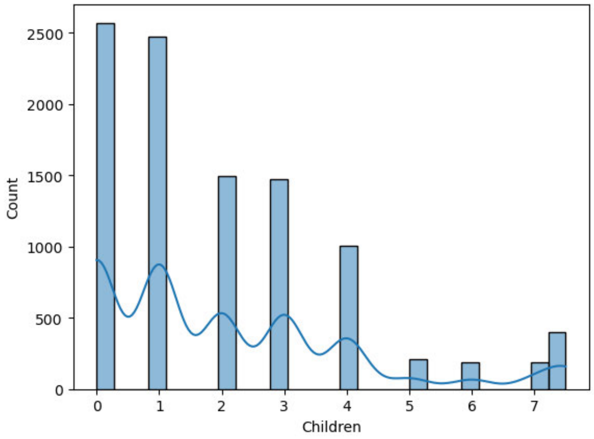
```
#VIF to check for multicollinearity, if greater than 10,  drop variables.

X= data[['Population', 'Children', 'Age', 'Income', 'Churn', 'Techie',
     'Port_modem', 'Tablet', 'Phone', 'Multiple', 'OnlineSecurity',
     'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',
     'StreamingMovies', 'PaperlessBilling', 'Tenure',
     'Area Suburban', 'Area Urban', 'Marital Married',
     'Marital Never Married', 'Marital Separated', 'Marital Widowed',
     'Gender Male', 'Gender Nonbinary', 'Contract One year',
     'Contract Two Year', 'InternetService Fiber Optic',
     'InternetService None', 'PaymentMethod Credit Card (automatic)',
     'PaymentMethod Electronic Check', 'PaymentMethod Mailed Check']]
vif_data = pd.DataFrame()
vif_data["Explanatory Variables"] = X.columns

vif_data["VIF"] = [variance_inflation_factor(X.values, i)
for i in range(len(X.columns))]

vif_data["VIF"]=round(vif_data["VIF"],2)
vif_data=vif_data.sort_values(by="VIF", ascending=False)
print(vif_data)
```

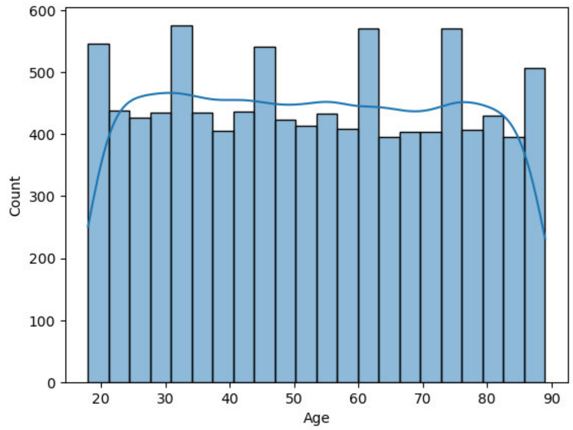
All the code can also be viewed in the attached AFD208CodeTk1.ipynb file.

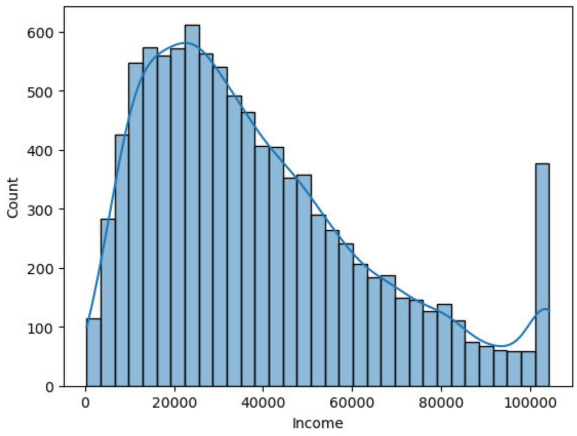
C4. Visualizations of Distributions via Univariate and Bivariate Statistics

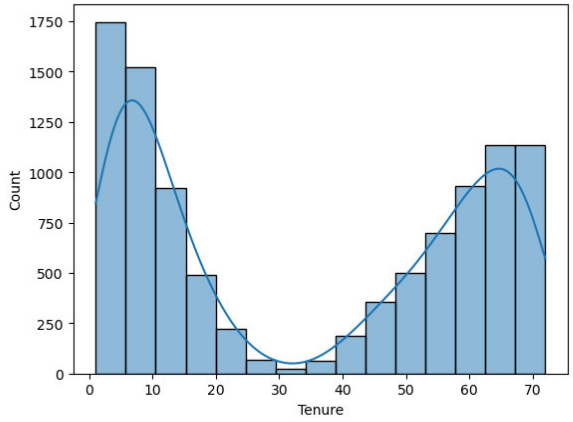
Please see below for my graphical visualizations of the univariate and bivariate analysis findings for each of the explanatory variables. The univariate statistics for the quantitative variables, “Population”, “Children”, “Age”, “Income”, and “Tenure”. These variables were plotted as histograms to view the distributions. The categorical variables were plotted via bar plots to visualize the distributions.

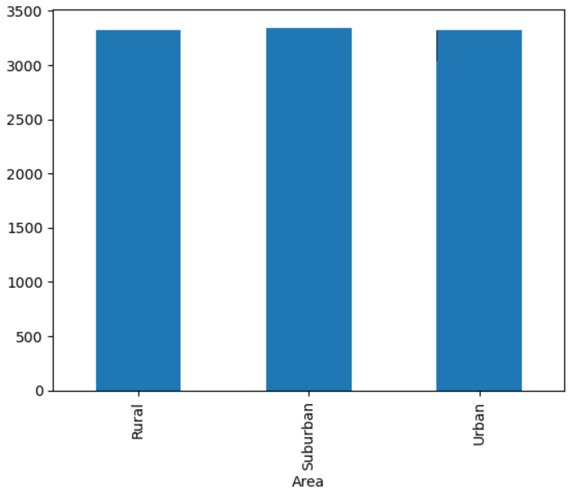


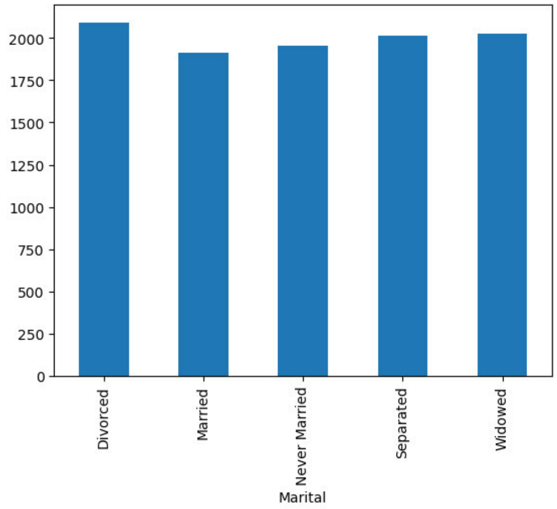


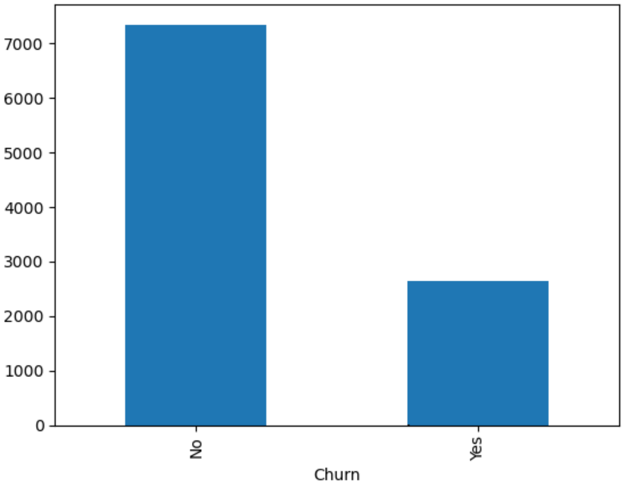


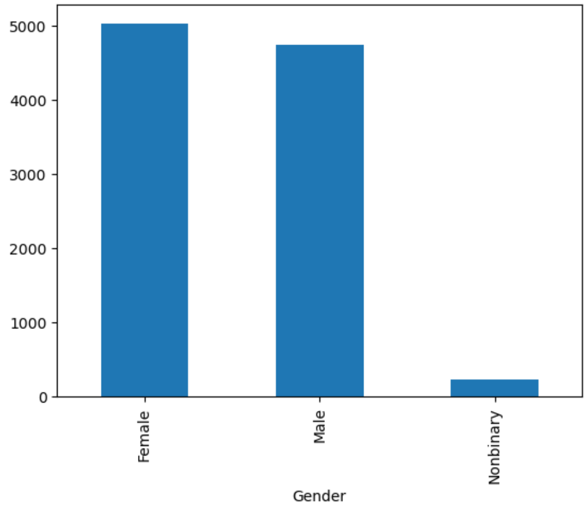


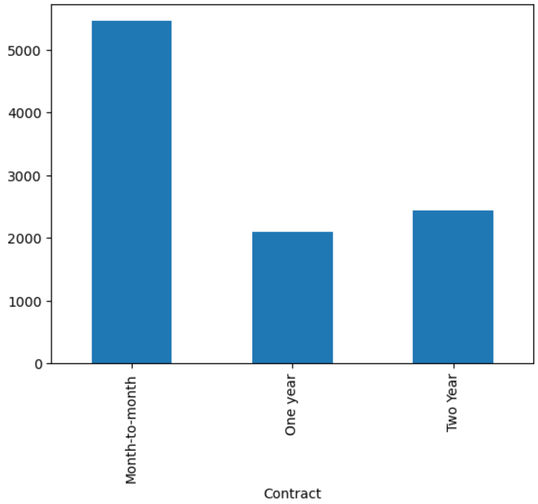

Count
0
250
500
750
1000
1250
1500
1750
0
10
20
30
40
50
60
70
Tenure

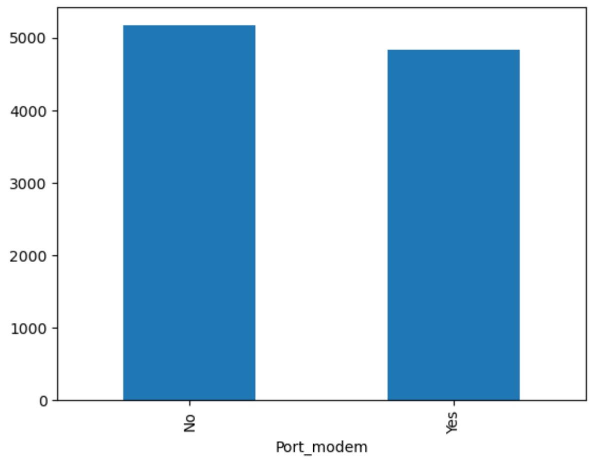

0
500
1000
1500
2000
2500
3000
3500
Rural
Suburban
Urban
Area

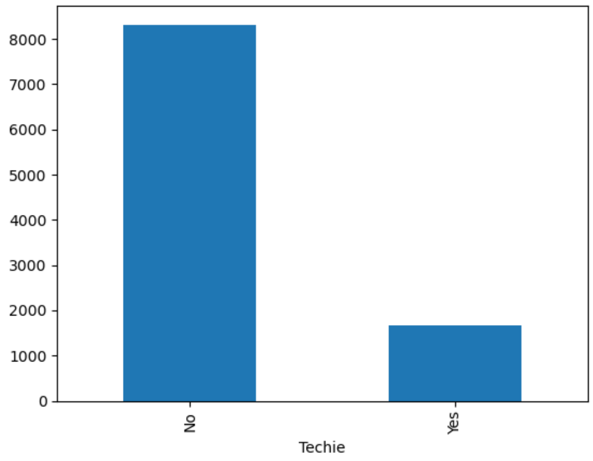

0
250
500
750
1000
1250
1500
1750
2000
Divorced
Married
Never Married
Separated
Widowed
Marital

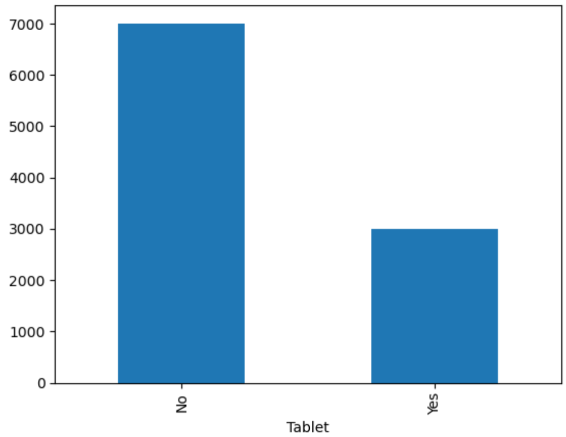

0
1000
2000
3000
4000
5000
6000
7000
No
Yes
Churn

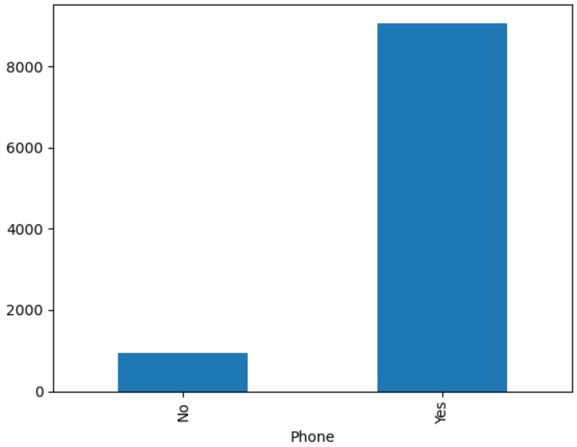

0
1000
2000
3000
4000
5000
Female
Male
Nonbinary
Gender

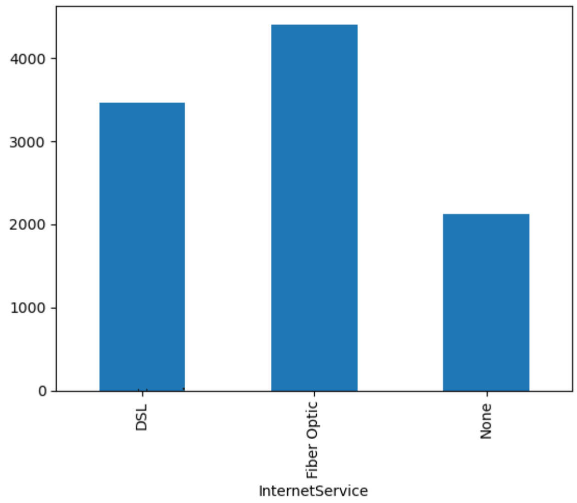

0
1000
2000
3000
4000
5000
Month-to-month
One year
Two Year
Contract

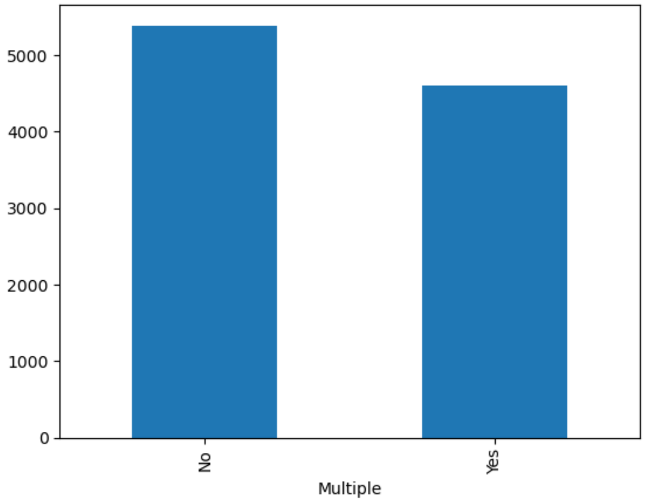

5000
4000
3000
2000
1000
0
No
Yes
Port_modem

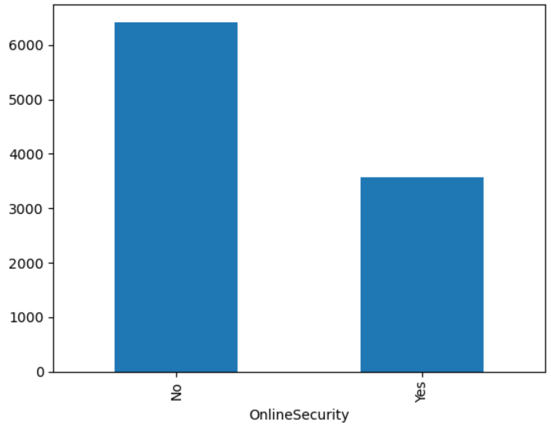

8000
7000
6000
5000
4000
3000
2000
1000
0
No
Yes
Techie

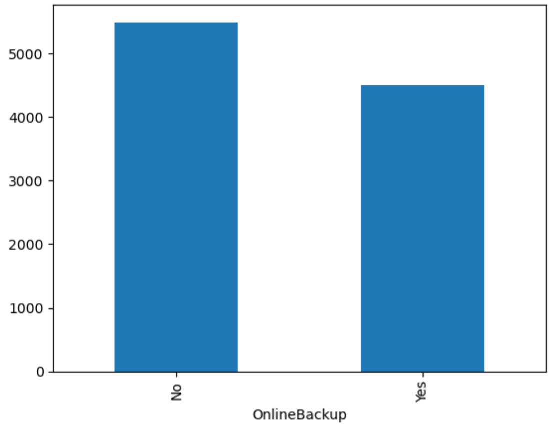

7000
6000
5000
4000
3000
2000
1000
0
No
Yes
Tablet

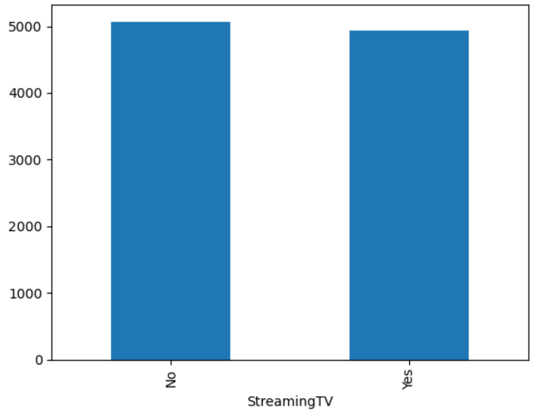

8000
6000
4000
2000
0
No
Yes
Phone

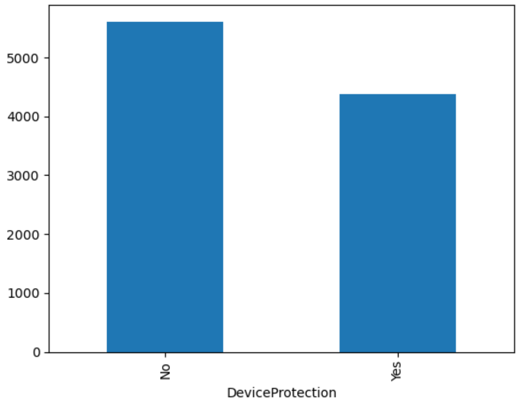

4000
3000
2000
1000
0
DSL
Fiber Optic
None
InternetService

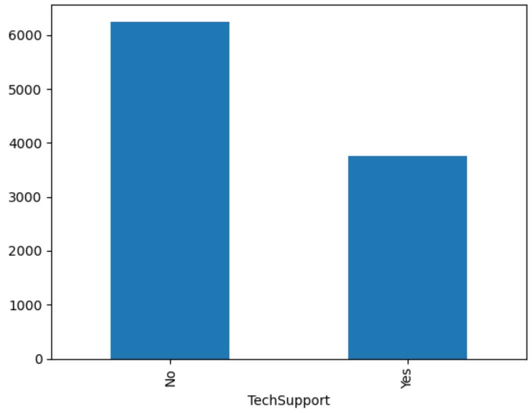

5000
4000
3000
2000
1000
0
No
Yes
Multiple

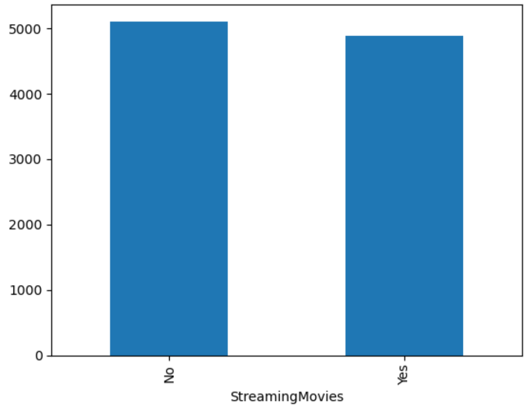

6000
5000
4000
3000
2000
1000
0
No
Yes
OnlineSecurity

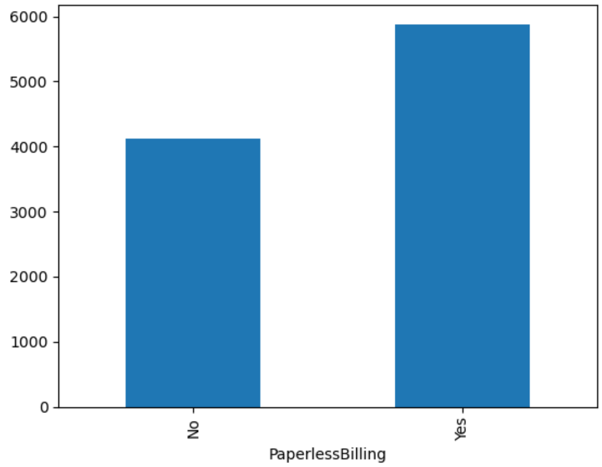

5000
4000
3000
2000
1000
0
No
Yes
OnlineBackup

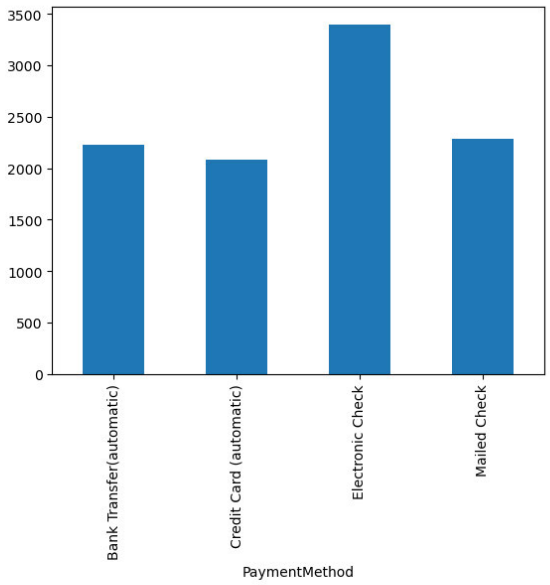

5000
4000
3000
2000
1000
0
No
Yes
StreamingTV


5000
4000
3000
2000
1000
0
No
Yes
DeviceProtection

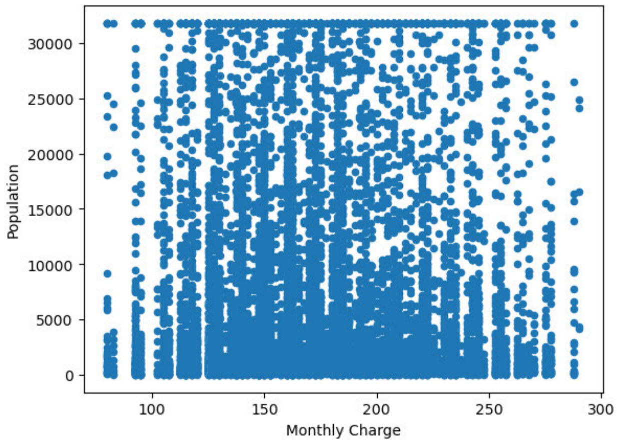

6000
5000
4000
3000
2000
1000
0
No
Yes
TechSupport

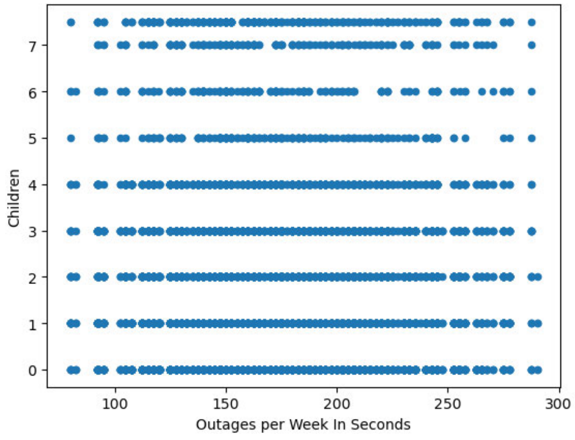

5000
4000
3000
2000
1000
0
No
Yes
StreamingMovies

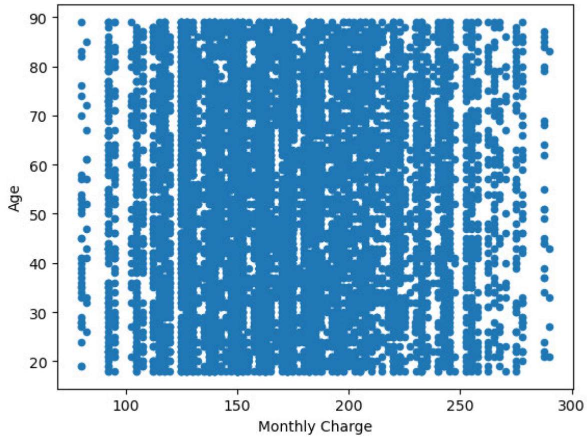


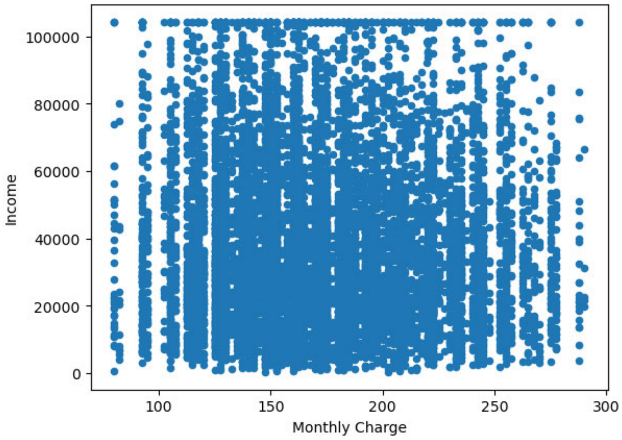


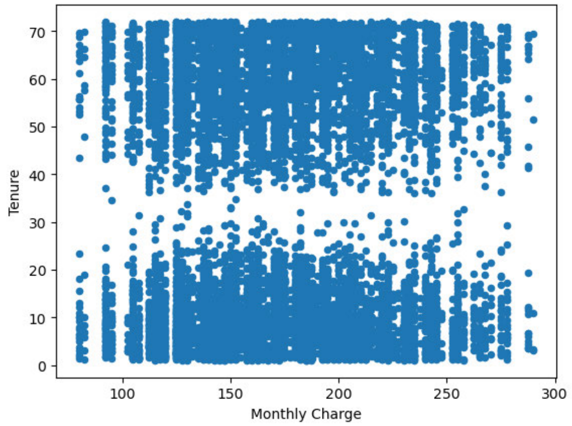
The bivariate statistics were visualized in two ways based on the type of variables being compared to the continuous target variable. The quantitative variables were plotted via a scatterplot to show the distributions. The target variable and the remaining categorical variables were plotted on a histogram.

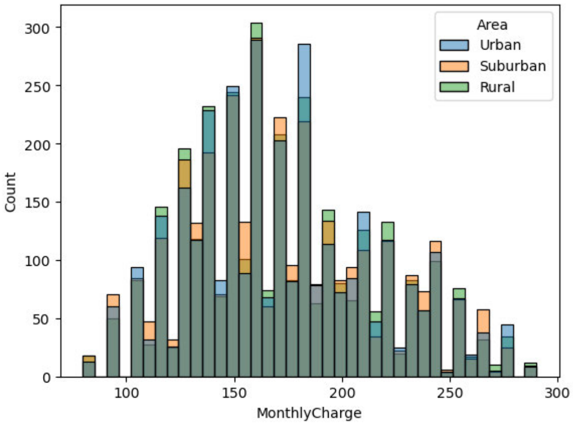


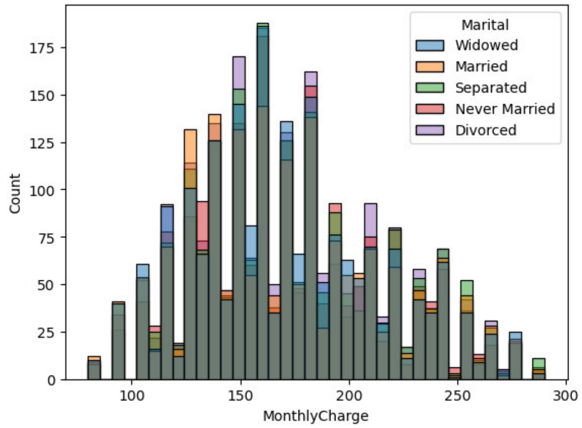


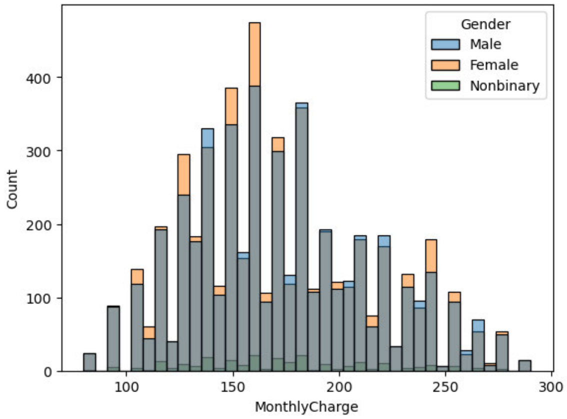

Age
90
80
70
60
50
40
30
20
100
150
200
250
300
Monthly Charge

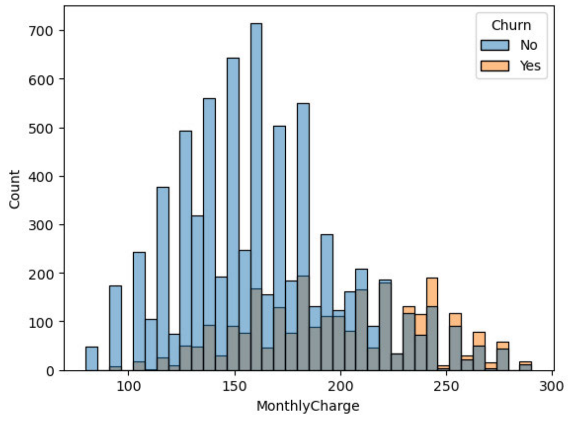

Income
100000
80000
60000
40000
20000
0
100
150
200
250
300
Monthly Charge

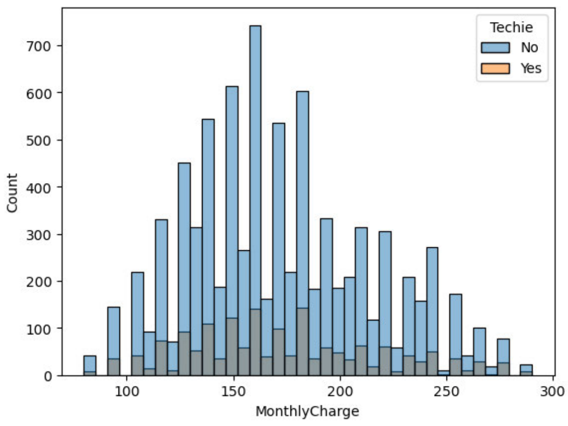

Tenure
70
60
50
40
30
20
10
0
100
150
200
250
300
Monthly Charge

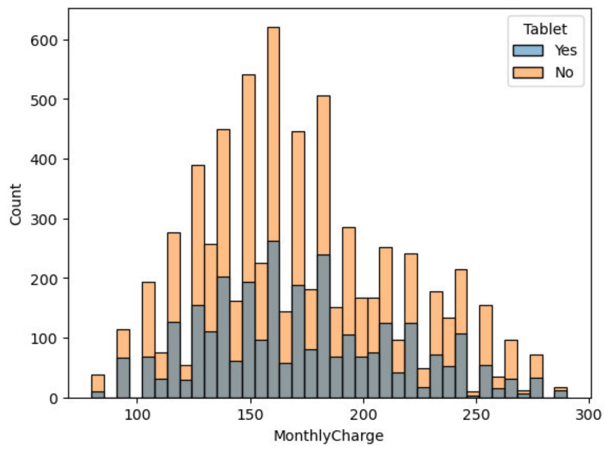

Area
Urban
Suburban
Rural
Count
300
250
200
150
100
50
0
100
150
200
250
300
MonthlyCharge

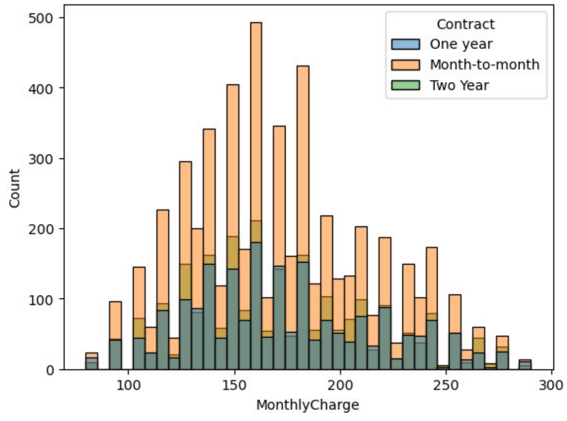

Marital
Widowed
Married
Separated
Never Married
Divorced
Count
175
150
125
100
75
50
25
0
100
150
200
250
300
MonthlyCharge

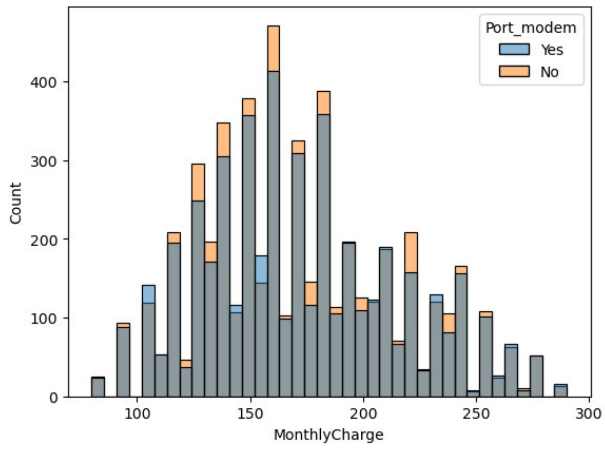

Gender
Male
Female
Nonbinary
Count
400
300
200
100
0
100
150
200
250
300
MonthlyCharge

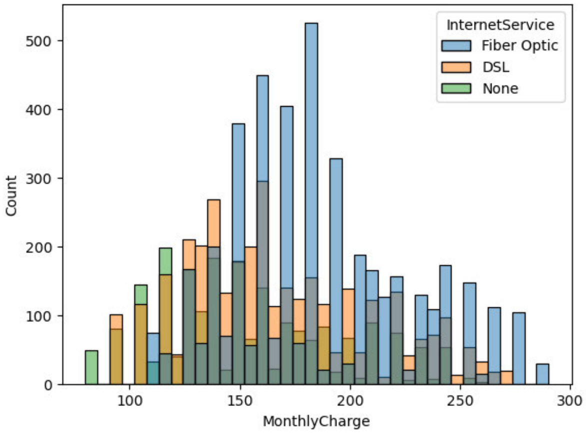

Churn
No
Yes
Count
0
100
200
300
400
500
600
700
100
150
200
250
300
MonthlyCharge

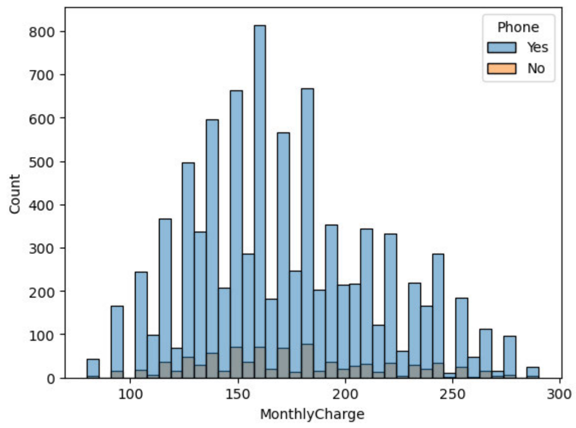

Techie
No
Yes
Count
0
100
200
300
400
500
600
700
100
150
200
250
300
MonthlyCharge

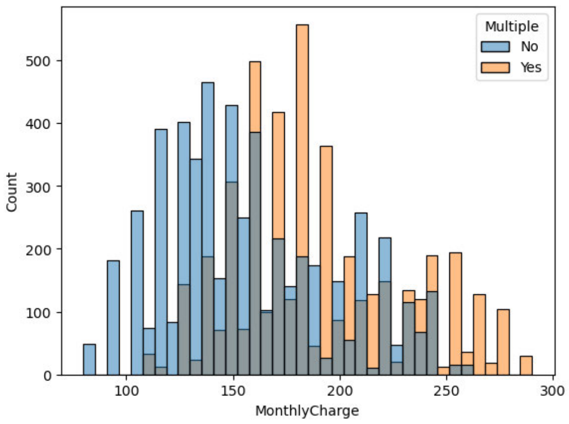

Tablet
Yes
No
Count
0
100
200
300
400
500
600
100
150
200
250
300
MonthlyCharge

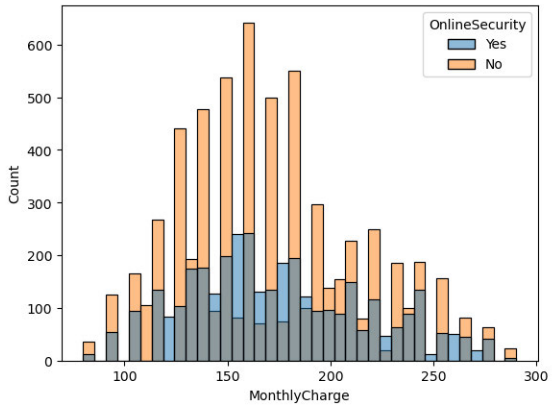

Contract
One year
Month-to-month
Two Year
Count
0
100
200
300
400
500
100
150
200
250
300
MonthlyCharge

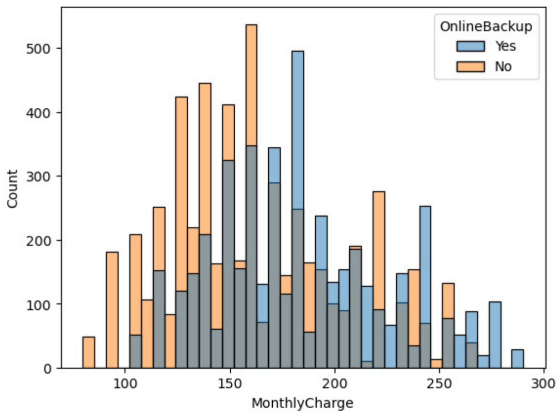

Port_modem
Yes
No
Count
0
100
200
300
400
100
150
200
250
300
MonthlyCharge

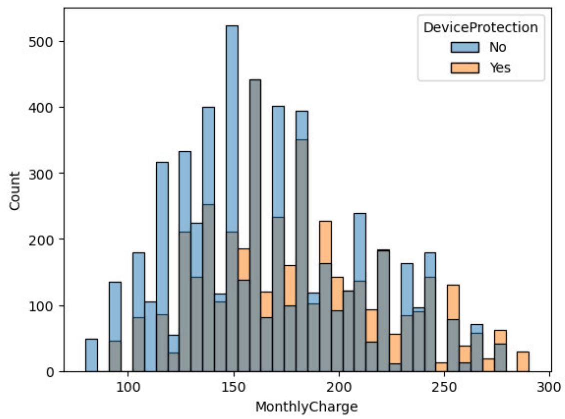

InternetService
Fiber Optic
DSL
None
Count
0
100
200
300
400
500
100
150
200
250
300
MonthlyCharge

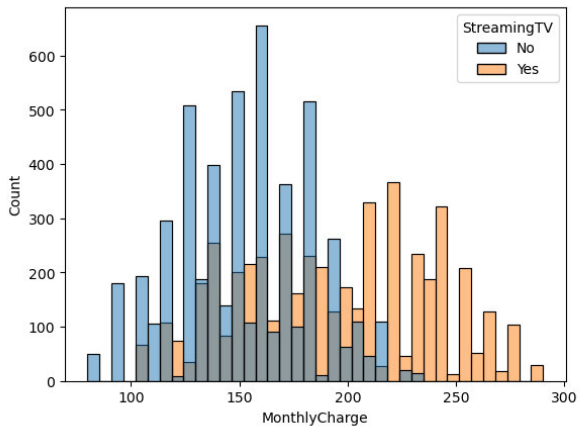

Phone
Yes
No
Count
0
100
200
300
400
500
600
700
800
100
150
200
250
300
MonthlyCharge

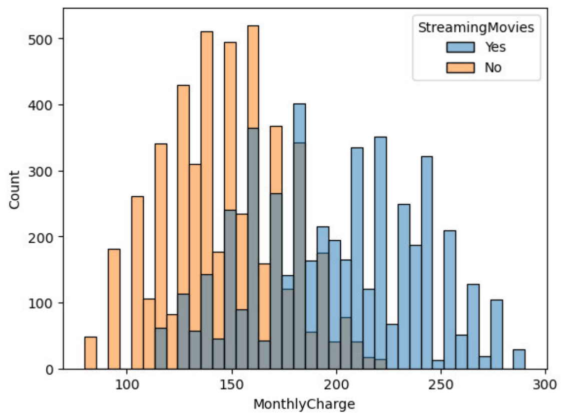

Multiple
No
Yes
Count
0
100
200
300
400
500
100
150
200
250
300
MonthlyCharge

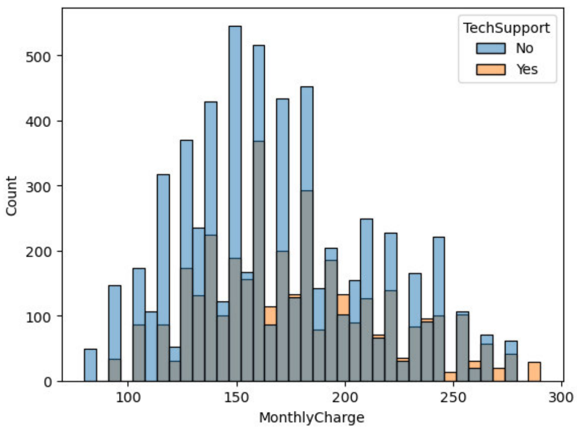

OnlineSecurity
Yes
No
Count
0
100
200
300
400
500
600
100
150
200
250
300
MonthlyCharge

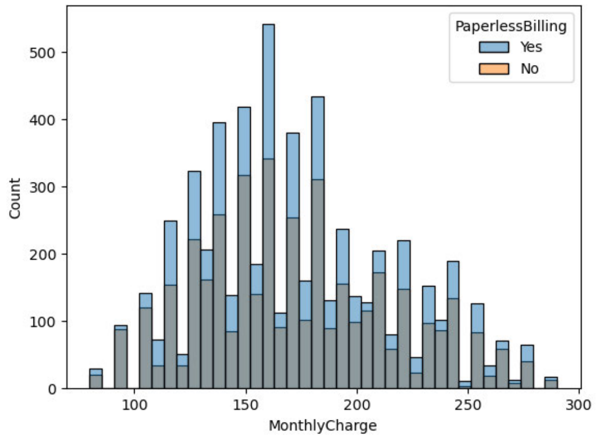

OnlineBackup
Yes
No
Count
0
100
200
300
400
500
100
150
200
250
300
MonthlyCharge

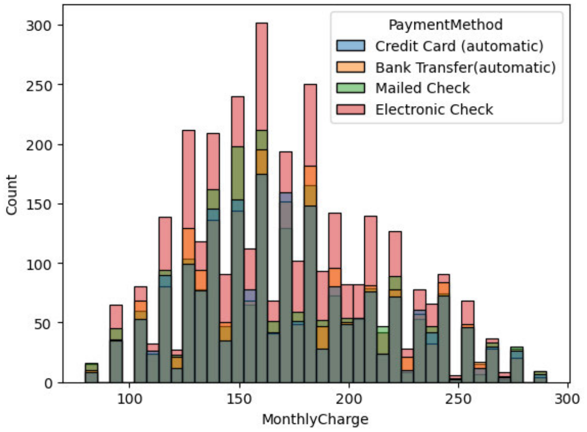

DeviceProtection
No
Yes
Count
0
100
200
300
400
500
100
150
200
250
300
MonthlyCharge


StreamingTV
No
Yes
Count
0
100
200
300
400
500
600
100
150
200
250
300
MonthlyCharge









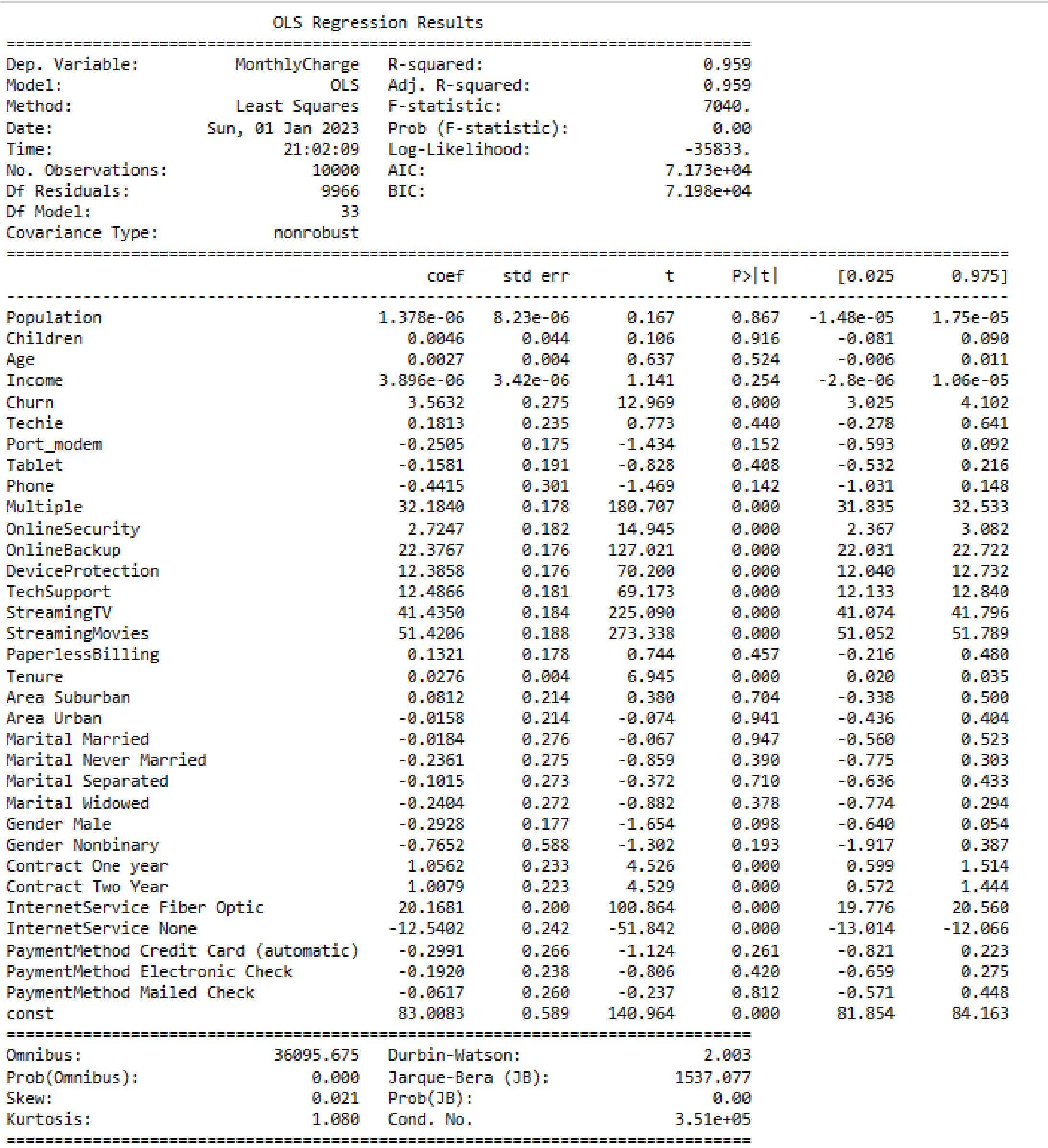
C5. Prepared Dataset

The attached CSV data file called AFCode208Task1_clean.csv is available to view the results of the prepared dataset.

Part IV. Model Comparison and Analysis

D1. Initial Multiple Regression Model

The target variable and re-expressed explanatory variables were analyzed in an OLS regression model. This model provided a regression summary based on the target variable, "MonthlyCharge". After the data preparation phases, the dataset was left with one continuous target variable, thirty-three explanatory variables. The thirty-three independent variables included continuous and categorical re-expressed attributes.



```
                            OLS Regression Results
==============================================================================
Dep. Variable:          MonthlyCharge   R-squared:                       0.959
Model:                            OLS   Adj. R-squared:                  0.959
Method:                 Least Squares   F-statistic:                     7040.
Date:                Sun, 01 Jan 2023   Prob (F-statistic):               0.00
Time:                        21:02:09   Log-Likelihood:                -35833.
No. Observations:               10000   AIC:                         7.173e+04
Df Residuals:                    9966   BIC:                         7.198e+04
Df Model:                          33
Covariance Type:            nonrobust
==========================================================================================================
                                           coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------------------------
Population                            1.378e-06   8.23e-06      0.167      0.867   -1.48e-05    1.75e-05
Children                                 0.0046      0.044      0.106      0.916      -0.081       0.090
Age                                      0.0027      0.004      0.637      0.524      -0.006       0.011
Income                                3.896e-06   3.42e-06      1.141      0.254    -2.8e-06    1.06e-05
Churn                                    3.5632      0.275     12.969      0.000       3.025       4.102
Techie                                   0.1813      0.235      0.773      0.440      -0.278       0.641
Port_modem                              -0.2505      0.175     -1.434      0.152      -0.593       0.092
Tablet                                  -0.1581      0.191     -0.828      0.408      -0.532       0.216
Phone                                   -0.4415      0.301     -1.469      0.142      -1.031       0.148
Multiple                                32.1840      0.178    180.707      0.000      31.835      32.533
OnlineSecurity                           2.7247      0.182     14.945      0.000       2.367       3.082
OnlineBackup                            22.3767      0.176    127.021      0.000      22.031      22.722
DeviceProtection                        12.3858      0.176     70.200      0.000      12.040      12.732
TechSupport                             12.4866      0.181     69.173      0.000      12.133      12.840
StreamingTV                             41.4350      0.184    225.090      0.000      41.074      41.796
StreamingMovies                         51.4206      0.188    273.338      0.000      51.052      51.789
PaperlessBilling                         0.1321      0.178      0.744      0.457      -0.216       0.480
Tenure                                   0.0276      0.004      6.945      0.000       0.020       0.035
Area Suburban                            0.0812      0.214      0.380      0.704      -0.338       0.500
Area Urban                              -0.0158      0.214     -0.074      0.941      -0.436       0.404
Marital Married                         -0.0184      0.276     -0.067      0.947      -0.560       0.523
Marital Never Married                   -0.2361      0.275     -0.859      0.390      -0.775       0.303
Marital Separated                       -0.1015      0.273     -0.372      0.710      -0.636       0.433
Marital Widowed                         -0.2404      0.272     -0.882      0.378      -0.774       0.294
Gender Male                             -0.2928      0.177     -1.654      0.098      -0.640       0.054
Gender Nonbinary                        -0.7652      0.588     -1.302      0.193      -1.917       0.387
Contract One year                        1.0562      0.233      4.526      0.000       0.599       1.514
Contract Two Year                        1.0079      0.223      4.529      0.000       0.572       1.444
InternetService Fiber Optic             20.1681      0.200    100.864      0.000      19.776      20.560
InternetService None                   -12.5402      0.242    -51.842      0.000     -13.014     -12.066
PaymentMethod Credit Card (automatic)   -0.2991      0.266     -1.124      0.261      -0.821       0.223
PaymentMethod Electronic Check          -0.1920      0.238     -0.806      0.420      -0.659       0.275
PaymentMethod Mailed Check              -0.0617      0.260     -0.237      0.812      -0.571       0.448
const                                   83.0083      0.589    140.964      0.000      81.854      84.163
==============================================================================
Omnibus:                    36095.675   Durbin-Watson:                   2.003
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1537.077
Skew:                           0.021   Prob(JB):                         0.00
Kurtosis:                       1.080   Cond. No.                     3.51e+05
==============================================================================
```

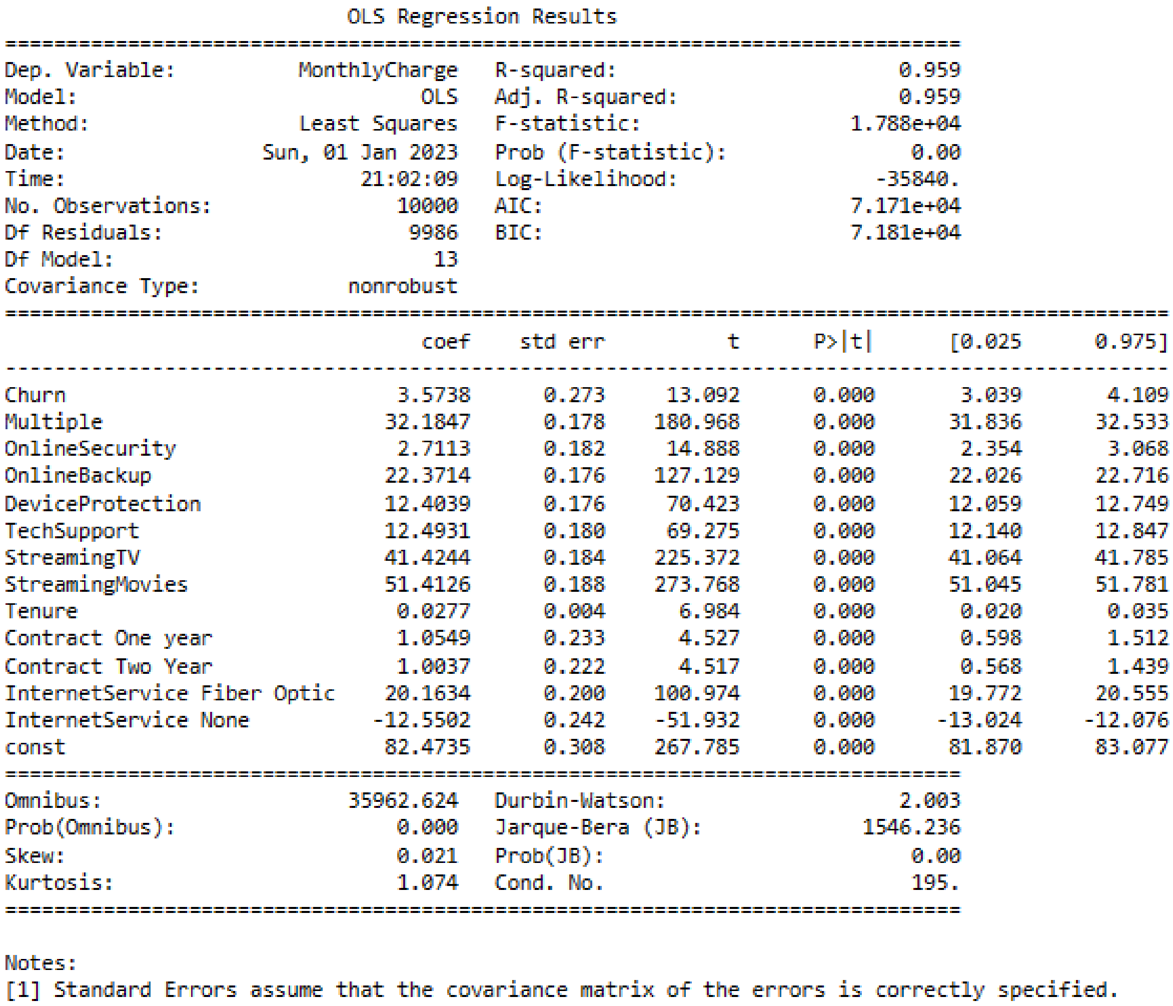
D2. Justification of Variable Selection and Evaluation Metric

The selection of the variables to remove for the reduced regression model was made by utilizing a wrapper method on the initial model. The wrapper method selected was the Backward Stepwise Elimination method. This method began with viewing the initial regression model. The initial model included a summary of the p-value for each feature or variable. The Backward Stepwise Elimination method was justified as it kept only the features that had a p-value that was deemed statistically significant. The p-value needed to be less than 0.05 to mean this criterion. The following continuous variable which met this criterion was “Tenure”. The remaining variables were all categorical. It included “Churn”, “Multiple”, “OnlineSecurity”, “OnlineBackup”, “TechSupport”, “DeviceProtection”, “StreamingTV”, “StreamingMovies”, “Contract One year”, “Contract Two Year”, “InternetService Fiber Optic”, “InternetService None” and the constant. All of these features had a p-value of 0.0.

The initial model and the reduced model both were shown with an evaluation metric R-squared value of 0.959 or about 96%. This value signified the percentage variation in the target variable could be explained by the independent variables (Broeck, Introduction to Regression with statsmodels in Python, n.d.). In the reduced model, the 96 % variation in “MonthlyCharge” was explained by the explanatory variables of “Tenure”, “Churn”, “Multiple”, “OnlineSecurity”, “OnlineBackup”, “TechSupport”, “DeviceProtection”, “StreamingTV”, “StreamingMovies”, “Contract One year”, “Contract Two Year”, “InternetService Fiber Optic”, and “InternetService None”. The closer the R-squared value was to 1 or 100% deemed the better goodness of fit measure for the regression model.

D3. Reduced Multiple Regression Model

The reduced multiple regression model was populated after the variable selection method, Backward Stepwise Elimination was applied. This left the explanatory variables of “Tenure”, “Churn”, “Multiple”, “OnlineSecurity”, “OnlineBackup”, “TechSupport”, “DeviceProtection”, “StreamingTV”, “StreamingMovies”, “Contract One year”, "Contract Two Year", "InternetService Fiber Optic", and "InternetService None" to be shown in the regression summary. The only continuous variable included in reduced model was "Tenure"; the twelve other variables were categorical.



```
                            OLS Regression Results
==============================================================================
Dep. Variable:          MonthlyCharge   R-squared:                       0.959
Model:                            OLS   Adj. R-squared:                  0.959
Method:                 Least Squares   F-statistic:                 1.788e+04
Date:                Sun, 01 Jan 2023   Prob (F-statistic):               0.00
Time:                        21:02:09   Log-Likelihood:                -35840.
No. Observations:               10000   AIC:                         7.171e+04
Df Residuals:                    9986   BIC:                         7.181e+04
Df Model:                          13
Covariance Type:            nonrobust
===============================================================================================
                                  coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------------
Churn                           3.5738      0.273     13.092      0.000       3.039       4.109
Multiple                       32.1847      0.178    180.968      0.000      31.836      32.533
OnlineSecurity                  2.7113      0.182     14.888      0.000       2.354       3.068
OnlineBackup                   22.3714      0.176    127.129      0.000      22.026      22.716
DeviceProtection               12.4039      0.176     70.423      0.000      12.059      12.749
TechSupport                    12.4931      0.180     69.275      0.000      12.140      12.847
StreamingTV                    41.4244      0.184    225.372      0.000      41.064      41.785
StreamingMovies                51.4126      0.188    273.768      0.000      51.045      51.781
Tenure                          0.0277      0.004      6.984      0.000       0.020       0.035
Contract One year               1.0549      0.233      4.527      0.000       0.598       1.512
Contract Two Year               1.0037      0.222      4.517      0.000       0.568       1.439
InternetService Fiber Optic    20.1634      0.200    100.974      0.000      19.772      20.555
InternetService None          -12.5502      0.242    -51.932      0.000     -13.024     -12.076
const                          82.4735      0.308    267.785      0.000      81.870      83.077
==============================================================================
Omnibus:                    35962.624   Durbin-Watson:                   2.003
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1546.236
Skew:                           0.021   Prob(JB):                         0.00
Kurtosis:                       1.074   Cond. No.                         195.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Part IV. Data Analysis: Outcome

E1. Discussion of Data Analysis Process

The data analysis process of comparing the initial and reduced multiple regression model included the following elements. The model evaluation metrics, variable selection, and the residual plotting. The model evaluation metrics shown in the initial and reduced regression models were R-Squared, Adj R-Squared, F-statistics, Prob(F-Statistics), and BIC. In the both models, the R-Squared and Adj R-squared were the same value of 0.959. The R-squared value showed what fraction of variation was linear (Vestuto, n.d.). The Adj R-Squared was the modified version of R-Squared but adjusted for the number of variables. The closer to a value of one, the better the goodness of fit measure was acknowledged. The F-Statistic varied between the initial and reduced models. The initial model had a F-statistic of 7040, whereas the reduced model's F-statistic was 1.788. This was used with the Prob(F-statistics) to tell the overall significance of the regression. In both cases, the variables implied significance in the regression model. Lastly, the BIC or Bayesian information criteria was a metric that compared the goodness of fit of different regression models (Vestuto, n.d.). The BIC value on the initial model was 7.198. The BIC value on the reduced model was 7.181. The reduced model had the lowest BIC value, which confirmed it was best fit for the data. The above metrics helped identify which model was better useful for predictive modeling.

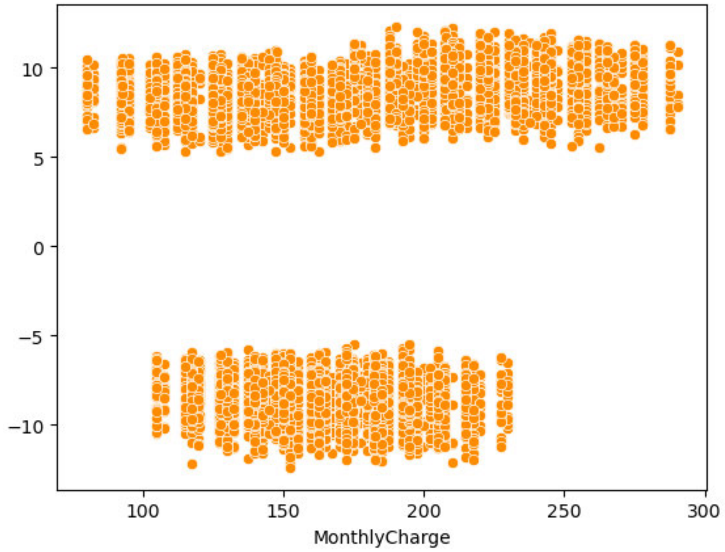
The variable selection process of Backward Stepwise Elimination was utilized in this data analysis. This was a wrapper method which started with the initial model and removed variables until a pre-specified stop rule, p-value less than 0.05. In the initial model, there were the following attributes: "Population", "Children", "Age", "Income", "Churn", "Techie", "Port_modem", "Tablet", "Phone", "Multiple", "OnlineSecurity", "OnlineBackup", "DeviceProtection", "TechSupport", "StreamingTV", "StreamingMovies", "PaperlessBilling", "Tenure", "Area Suburban", "Area Urban", "Marital Married", "Married Never Married", "Marital Separated", "Marital Widowed", “Gender Male”, “Gender Nonbinary”, “Contract One year”, “Contract Two Year”, “InternetService Fiber Optic, “InternetService None”, “PaymentMethod Credit Card (automatic), “PaymentMethod Electronic Check”, “PaymentMethod Mailed Check”, and “const”. The elimination method was completed and only the following variables met the stop rule specified; “Churn”, “Multiple”, “OnlineSecurity”, “OnlineBackup”, “DeviceProtection”, “TechSupport”, “StreamingTV”, “StreamingMovies”, “Tenure”, “Contract One year”, “Contract Two Year”, “InternetService Fiber Optic, “InternetService None”, and “const”. These were the variables that were included in the reduced model.

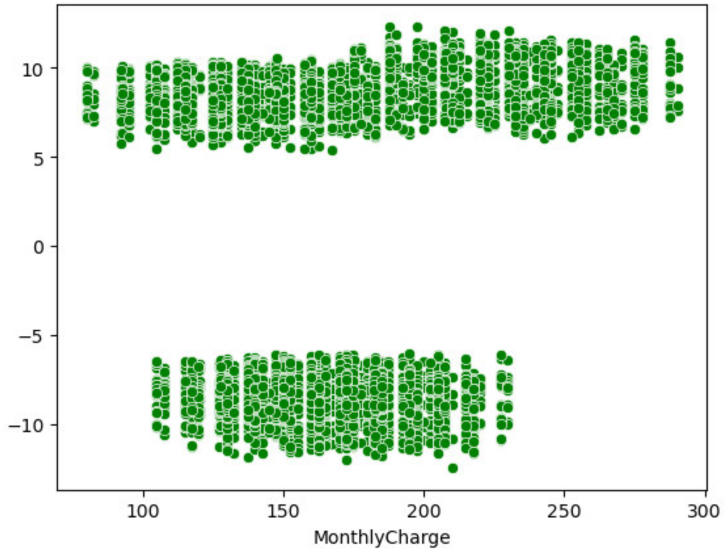
The last stage of the data analysis was the residual plotting. The residuals were plotted by first identifying the residuals. Residuals were the actual observed value minus the predicted value. The residual values measure how much regression line misses a data point. The residual error for both models was 8.72. This was further shown in the scatterplots of the residuals. The residuals were all found within the 5 through 13, and -5 through -13 ranges. It was noted as a bimodal distribution.

E2. Data Analysis Output Results

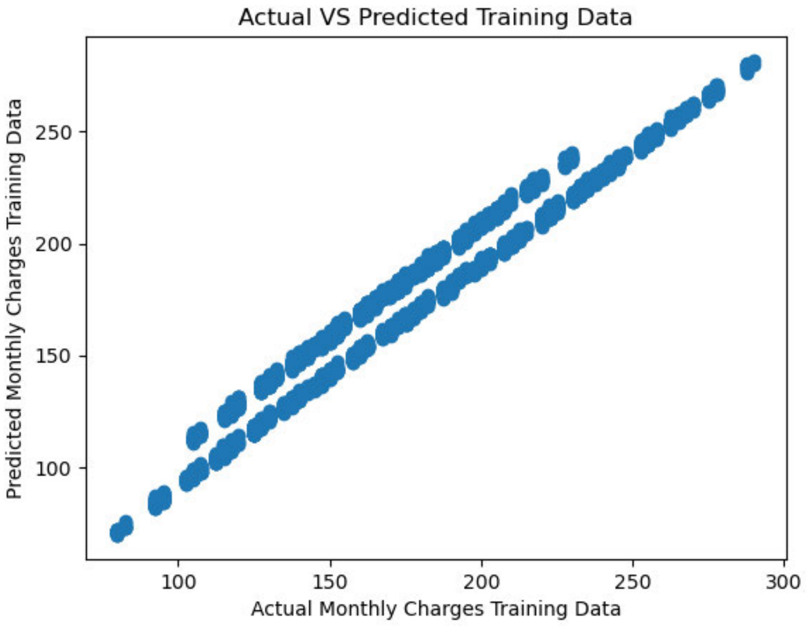
The below showed the results of the data analysis calculations as well as the residual plots via scatter plots and residual error.

```
Initial model mse:  76.1028785896427

Initial model Residual Standard Error:  8.723696383394065

Reduced model mse:  76.06542091738008

Reduced Model Residual Standard Error:  8.723696383394065
```

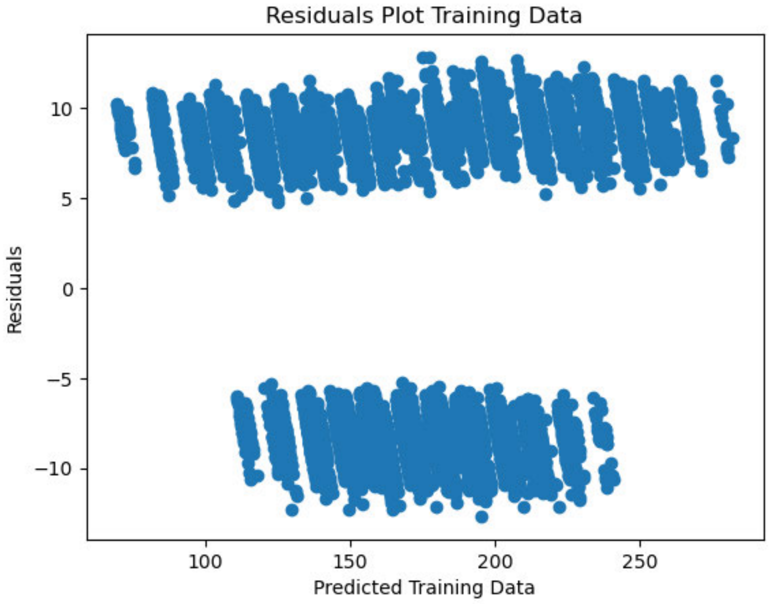




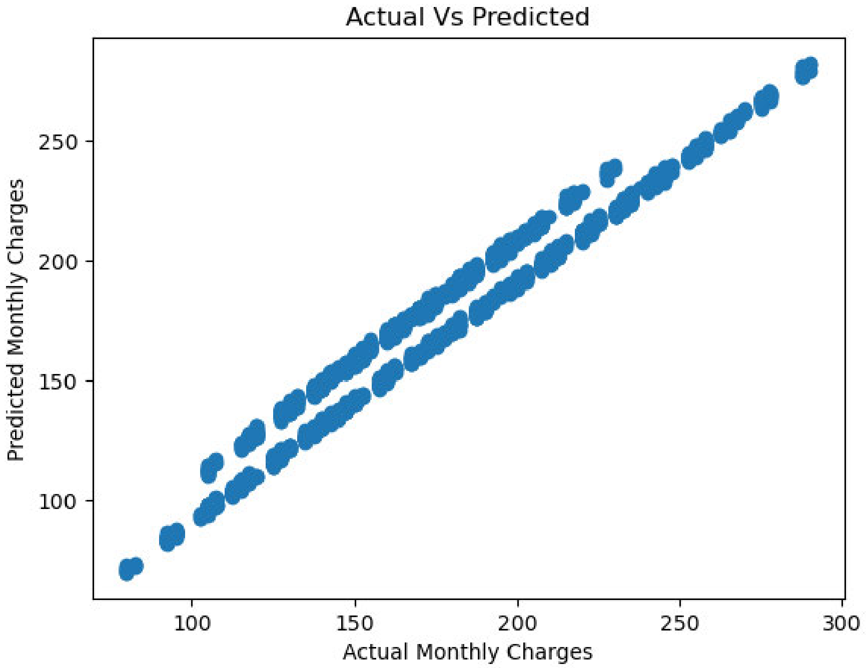
```
82.45076910966257

array([ 1.87195374e-06,  1.79058747e-02,  7.54573972e-03,  4.34189303e-06,
        3.66446385e+00, -1.00222184e-01, -3.74778300e-01, -3.52422197e-01,
       -3.84066346e-01,  3.19755076e+01,  2.75768503e+00,  2.25755375e+01,
        1.22664730e+01,  1.26680768e+01,  4.14116009e+01,  5.14090718e+01,
        2.75907427e-01,  3.05075296e-02,  2.89339690e-01,  8.76497427e-02,
        3.17833601e-01, -1.08059021e-01,  2.36898454e-04,  7.85640819e-02,

       -2.29681542e-01, -9.58122489e-01,  8.50723627e-01,  7.07665358e-01,

        2.00639459e+01, -1.28304523e+01, -1.97030510e-01, -1.30567528e-01,
        3.22677350e-02,  0.00000000e+00])
array([193.13841659, 155.08433269, 189.84100393, ..., 259.85926489,
       148.00924286, 183.14608288])
```



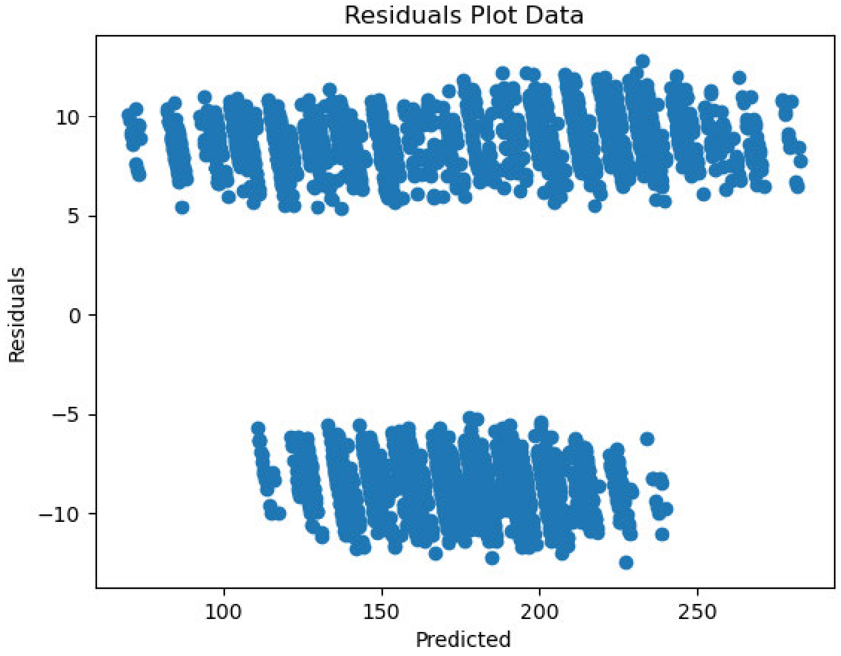
```
0.9589803022070595
```

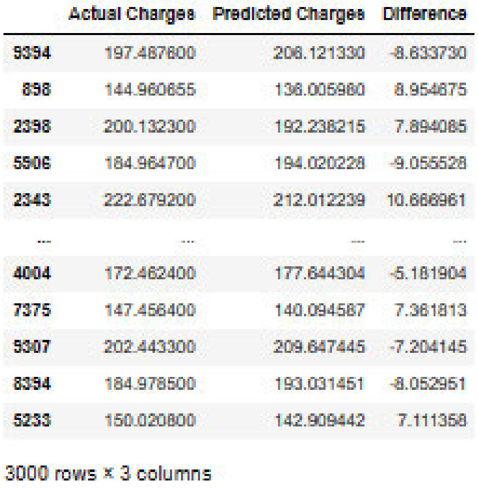


```
array([206.12133003, 136.00598038, 192.23821543, ..., 209.64744487,
       193.03145083, 142.90944207])
```



```
0.9583606984066744
```





	Actual Charges	Predicted Charges	Difference
9394	197.487600	206.121330	-8.633730
898	144.960655	136.005980	8.954675
2398	200.132300	192.238215	7.894085
5906	184.964700	194.020228	-9.055528
2343	222.679200	212.012239	10.666961
...	...	...	...
4004	172.462400	177.644304	-5.181904
7375	147.456400	140.094587	7.361813
9307	202.443300	209.647445	-7.204145
8394	184.978500	193.031451	-8.052951
5233	150.020800	142.909442	7.111358

3000 rows × 3 columns

E3. Data Analysis Code

```
#regression model
data['const']=1
y= data['MonthlyCharge']
X= data[['Population', 'Children', 'Age', 'Income', 'Churn', 'Techie',
      'Port_modem', 'Tablet', 'Phone', 'Multiple', 'OnlineSecurity',
      'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',
      'StreamingMovies', 'PaperlessBilling', 'Tenure', 'Area Suburban', 'Area
Urban',
       'Marital Married', 'Marital Never Married', 'Marital Separated',
       'Marital Widowed', 'Gender Male', 'Gender Nonbinary', 'Contract One year',
      'Contract Two Year', 'InternetService Fiber Optic',
      'InternetService None', 'PaymentMethod Credit Card (automatic)',
      'PaymentMethod Electronic Check', 'PaymentMethod Mailed Check', 'const']]

model = sm.OLS(y,X)
results = model.fit()
print(results.summary())

#reduced regression model utilizing the wrapper backward stepwise elimination,
#only keeping values that have a p-value of less than 0.05

data['const']=1
y= data['MonthlyCharge']
rX= data[['Churn','Multiple', 'OnlineSecurity', 'OnlineBackup',
      'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies',
      'Tenure', 'Contract One year', 'Contract Two Year',
       'InternetService Fiber Optic', 'InternetService None', 'const']]
```

```
reduced_model = sm.OLS(y, rX)
redu_results = reduced_model.fit()
print(redu_results.summary())

#Residual Standard Error of model
mse = results.mse_resid
print('Initial model mse: ', mse)
RSE = np.sqrt(mse)
print("Initial model Residual Standard Error: ", RSE)
r_mse = redu_results.mse_resid
print('Reduced model mse: ', r_mse)
r_RSE = np.sqrt(mse)
print("Reduced Model Residual Standard Error: ", r_RSE)

residuals = data["MonthlyCharge"] - results.predict(data[['Population', 'Children',
'Age', 'Income', 'Churn', 'Techie',
    'Port_modem', 'Tablet', 'Phone', 'Multiple', 'OnlineSecurity',
    'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',
    'StreamingMovies', 'PaperlessBilling', 'Tenure', 'Area Suburban', 'Area
Urban',
     'Marital Married', 'Marital Never Married', 'Marital Separated',
     'Marital Widowed', 'Gender Male', 'Gender Nonbinary', 'Contract One year',
    'Contract Two Year', 'InternetService Fiber Optic',
    'InternetService None', 'PaymentMethod Credit Card (automatic)',
    'PaymentMethod Electronic Check', 'PaymentMethod Mailed Check',
'const']])
sns.scatterplot(x=data["MonthlyCharge"], y=residuals, color='darkorange')
plt.show()
residuals = data["MonthlyCharge"] - redu_results.predict(data[['Churn','Multiple',
'OnlineSecurity', 'OnlineBackup',
    'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies',
    'Tenure', 'Contract One year', 'Contract Two Year',
     'InternetService Fiber Optic', 'InternetService None', 'const']])
sns.scatterplot(x=data["MonthlyCharge"], y=residuals, color='green')
plt.show()

#multiple linear regression followed by visualizations Using train test split

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.30,
random_state=0)
lr = LinearRegression()
lr.fit(X_train, y_train)
c = lr.intercept_
c
m= lr.coef_
m
```

```
y_pred_train = lr.predict(X_train)
y_pred_train

#residual plotting via scatterplot
#using training data
plt.scatter(y_train, y_pred_train)
plt.xlabel("Actual Monthly Charges Training Data")
plt.ylabel("Predicted Monthly Charges Training Data")
plt.title("Actual VS Predicted Training Data")
plt.show()
r2_score(y_train, y_pred_train)

residuals= y_train - y_pred_train
plt.scatter(y_pred_train, residuals)
plt.xlabel("Predicted Training Data")
plt.ylabel("Residuals")
plt.title("Residuals Plot Training Data")
plt.show()

y_pred_test = lr.predict(X_test)
y_pred_test

#residual plotting via scatterplot
#using actual test data
plt.scatter(y_test, y_pred_test)
plt.xlabel("Actual Monthly Charges")
plt.ylabel("Predicted Monthly Charges")
plt.title("Actual Vs Predicted")
plt.show()

r2_score(y_test, y_pred_test)

residuals= y_test - y_pred_test
plt.scatter(y_pred_test, residuals)
plt.xlabel("Predicted")
plt.ylabel("Residuals")
plt.title("Residuals Plot Data")
plt.show()

pred_y_data = pd.DataFrame({'Actual Charges': y_test, 'Predicted Charges':
y_pred_test, 'Difference': y_test-y_pred_test})
pred_y_data[0:10000]
```

Part V. Data Summary and Implications

F1. Summarization of Findings

The data analysis of the regression model resulted in the following elements.

1. Regression equation of the reduced model.
2. Interpretation of coefficients
3. The statistical and practical significance of the reduced model
4. The limitations of the data analysis.

The regression equation for the reduced model was set up based on the linear regression equation (Vestuto, n.d.).

$$Y_i = B_0 + B_1X_i + B_2X_i + B_3X_i + B_4X_i + B_5X_i \ldots. + B_{13}X_i$$

The reduced regression model formed the following regression equation.

MonthlyCharge = 82.47(const) + 3.57(Churn) + 32.18(Multiple) + 2.71(OnlineSecurity) + 22.37(OnlineBackup) + 12.40(DeviceProtection) + 12.49(TechSupport) + 41.42(StreamingTV) + 51.41(StreamingMovies) + 0.02(Tenure) + 1.05(Contract One year) + 1.00(Contract Two Year) + 20.16(InternetService Fiber Optic) + -12.55(InternetService None)

The regression equation used the target variable as Y with the intercept as B_0. In this dataset, the variable "const" is the intercept. Afterwards the regression coefficients for every explanatory variable in the reduced model was added.

The coefficients provided a linear relationship between the target variable and the explanatory variable. The “Churn” data had a coefficient of 3.57. This was interpreted as for every one unit increase in “Churn”, there was a positive 3.57 unit increase in “MonthlyCharge”. The explanatory variables “Multiple”, “OnlineSecurity”, “OnlineBackup”, “DeviceProtection”, “TechSupport”, “StreamingTV”, “StreamingMovies”, “Tenure”, “Contract One year”, “Contract Two Year”, and “InternetService Fiber Optic” also had a positive relationship with the target variable. For every unit increase in those variables, there was a positive increase in the “MonthlyCharge”. This was represented for each value accordingly. For every one unit increase in “Multiple” there was an associated 32.18 unit increase in “MonthlyCharge”. For every one unit increase in “Multiple”, there was an associated 32.18 unit increase in “MonthlyCharge”. For every one unit increase in “OnlineSecurity”, there was an associated 2.71 unit increase in “MonthlyCharge”. For every one unit increase in “OnlineBackup”, there was an associated 22.37 unit increase in “MonthlyCharge”. For every one unit increase in “OnlineBackup”, there was an associated 22.37 unit increase in “MonthlyCharge”. For every one unit increase in “DeviceProtection”, there was an associated 12.40 unit increase in “MonthlyCharge”. For every one unit increase in “TechSupport”, there was an associated 12.49 unit increase in “MonthlyCharge”. For every one unit increase in “StreamingTV”, there was an associated 41.42 unit increase in “MonthlyCharge”. For every one unit increase in “StreamingMovies”, there was an associated 51.41 unit increase in “MonthlyCharge”. For every one unit increase in “Tenure”, there was an associated 0.02 unit increase in “MonthlyCharge”. For every one unit increase in “Contract One year”, there was an associated 1.05 unit increase in

“MonthlyCharge”. For every one unit increase in “Contract Two Year”, there was an associated 1.00 unit increase in “MonthlyCharge”. For every one unit increase in “InternetService Fiber Optic”, there was an associated 20.16 unit increase in “MonthlyCharge”. The intercept or “const” is the average expected value for the target variable when all explanatory variables are equal to zero (Vestuto, n.d.). With that explained, for every unit increase of the “const” there will be an associated 82.47 unit increase in “MonthlyCharge”. There was only one variable that had a negative relationship with the target, which was “InternetService None”. For this variable every one unit increase as associated with a 12.55 unit decrease in “MonthlyCharge”.

The reduced model was statistically significant. The model was significant because the F-statistic and Prob(F-Statistic) found in the regression summary was less than the significance level of 0.05. This assumed that the sample data provided sufficient evidence to the fit of the data better than a model without those independent variables. The R-Squared value also contributed to the goodness of fit for the model. The R-squared value was relatively high as it was over 0.95 or 95%. This value represented how well the model explained the variability of the dependent variable ‘MonthlyCharge”.

The model had practical significance. The model results on the telecommunication dataset have real world significance to the stakeholders. The coefficient estimations within the regression model provided an idea of what variables would impact the range of the monthly charges. There was an observation of the contract terms and the monthly charges having a one to one ratio of change with increase. Whereas tenure had minor change to the increase of monthly charges as it increased. StreamingTV and StreamingMovies had the largest impact on the monthly charges. All of this information provided the stakeholders with insight on which services should be further explored in terms of retention and marketability (Expert Panel, Forbes Agency Council, 2019). There should be some caution utilizing the regression modal though. The residuals found were noted to have a bimodal distribution which violated the regression assumption of normality of residuals (Bobbitt, 2021). There was an unknown underlying factor that was causing this. The telecommunication dataset did not provide the best fit to evaluate this further.

There are various limitations to the data analysis. Regression is sensitive to outliers and multicollinearity. The presence of this can provide unreliable results. Linear regression must have the assumptions verified to know if the model is the right predictive modeling technique. There are other limitations noted as well in regard to the data preparation step. The treatment of outliers could be a limitation. Capping was utilized in this data preparation method and can affect the results of further analysis. Capping is imputing the upper and lower outliers to the upper and lower values. If another method were used the results could reflect differently. Another limitation would be the data wrangling stage. The type of encoding utilized can affect how the categorical variables are represented in numerical values. For example, if ordinal encoding were used on gender, it would provide inaccurate results for male, female, and nonbinary. Lastly, model reduction method limitations are related to the multicollinearity and p-values of the independent variables. The VIF value is sensitive to the variables being assessed. Including a variable that is directly related to another variable can increase the multicollinearity and the VIF value. The presence of multicollinearity can affect the p-value of the independent variables in the regression model. It can note unclear and inaccurate results. All of the above mentioned points are examples of the various limitations of data analysis.

F2. Recommendation of Action

The result of the research question deemed some variables had an impact on the spread or range of the monthly charges. The variables which had the most impact was streaming tv or streaming movies as both had a larger increase per unit associated with monthly charge. Other variables were noted such as online backup, device protection, tech support, and whether a customer had fiber optic internet service. The recommended course of action based on the results of the research analysis was to suggest the telecommunication company put efforts into its sales and pricing evaluation of streaming tv and movies. The significant increase in pricing for this specification should be analyzed to aid in future sales as well as retention. The stakeholders should consider decreasing pricing of these features in the long term for client retention (Expert Panel, Forbes Agency Council, 2019).

Part VI. Demonstration

G. Panopto Video

Please see attached Panopto video link. Link Found here:

[redacted]

This is a video providing an overview of the Python code used to discover anomalies, the data cleaning process, univariate and bivariate statistics, and regression modeling with coefficients. The recording will demonstrate the code's warning and error-free functionality as well as provide an overview of the programming environment used, Jupyter Notebook.

H. Third-Party Web Sources

There were no web sources used to acquire data nor segments of third-party code to support the application. All code and segments are original work.

I. References

Bobbitt, Z. (2021, Nov 16). *The Five Assumptions of Multiple Linear Regression*. Retrieved from Statology: https://www.statology.org/multiple-linear-regression-assumptions/

Broeck, M. V. (n.d.). *Intermediate Regression with statsmodels in Python*. Retrieved from Datacamp: https://app.datacamp.com/learn/courses/intermediate-regression-with-statsmodels-in-python

Broeck, M. V. (n.d.). *Introduction to Regression with statsmodels in Python*. Retrieved from Datacamp: https://app.datacamp.com/learn/courses/introduction-to-regression-with-statsmodels-in-python

Expert Panel, Forbes Agency Council. (2019, December 20). *How To Increase Client Retention: 10 Effective Strategies*. Retrieved from Forbes: https://www.forbes.com/sites/forbesagencycouncil/2019/12/20/how-to-increase-client-retention-10-effective-strategies/?sh=3f663798390b

Vestuto, J. (n.d.). *Introduction to Linear Modeling in Python*. Retrieved from Datacamp: https://app.datacamp.com/learn/courses/introduction-to-linear-modeling-in-python