D208- Predictive Modeling Performance Assessment

Task 1: Linear Regression Modeling

Western Governs University

Table of Contents

Part I: Research Question

A1. Research Question:

What factors contribute to the bandwidth_GB_year? This variable measures the amount of data used, in GB, in a year by the customer.

A2. Goals

The goal of this analysis is to determine which factors impact customer tenure in relation to the continuous variable bandwidth_GB_year. It is more cost-effective to retain current customers as opposed to gaining new customers. This analysis will utilize predictive and explanatory variables to examine the relationship between dependent and independent variables. Predictive variables are variables used to make predictions about an outcome of interest. Explanatory variables are variables used to explain the relationship between two other variables. This analysis will focus on the following independent variables: 'Population', 'Children', 'Age', 'Income', 'Outage_sec_perweek', 'Email', 'Contacts', 'Yearly_equip_failure','Tenure', 'MonthlyCharge', 'DummyChurn', 'DummyGender', 'DummyTechie', 'DummyTechSupport'.

Part II: Method Justification

B1. Summary of Assumptions

A multiple linear regression model is a statical model used to analyze the relationship between the dependent variables and independent variables, where the dependent variable is predicted based on predictor variables which are the independent variables (Massaron, 2016). The underline idea of a linear regression model is to fit a straight line through a set of data points. The line is represented by the equation $Y = a + bX$. Where Y is the dependent variable and X is the independent variable, a is the intercept, and b is the slope of the line (Massaron, 2016).

There are several assumptions when completing a linear regression model. One assumption is linearity. Linearity assumes the relationship between the dependent variables and each of the independent variables is linear and change in the dependent variable is proportional to the change in the independent variable. Another assumption would be homoscedasticity which assumes the variance of the error is constant across all levels of the independent variables (Z,

2021). This assumes that the errors are equally scattered across the range of the independent variables and that there is no pattern in the errors. When using a multiple linear regression model another assumption would be multivariate normality which means the residuals of the model are normally distributed. A fourth assumption would be that there is no multicollinearity. Multicollinearity occurs when two or more independent variables are highly correlated with each other which causes difficulty in estimating the effect of each independent variable on the dependent variable (Z, 2021).

B2. Tool Benefits

Data cleaning for the churn data set was completed utilizing Python. Python is an open-sourced programing language used for analysis and development. Python has a consistent syntax that makes coding and debugging user-friendly for beginners. Python is flexible and has the ability to import packages and to tailor data. The following packages were imported and used for their advantages. Below are the packages/libraries used for the assessment.

Packages/Libraries	Purpose
pandas	Main package for data uploading and manipulation
numpy	Main package for working with arrays
Matplotlib.pyplot	Visualization
Seaborn	Advanced visualization
scipy.stats	Normalization and statistics
sklearn.linear_model	Linear Regression
sklearn.preprocessing	to complete one hot encoding
sklearn.model_selection	split data into training and test sets
statsmodels.api	statistical model analysis (i.e. OLS)
statsmodels.stats.outliers_influence	variance inflation factor
statsmodels.stats.diagnostic import het_breuschpagan	Breusch-Pagan test

B3. Appropriate Technique

Multiple linear regression is an appropriate technique to use for analyzing the research question because it allows for the examination of the relationship between multiple independent variables to a single dependent variable (Massaron, 2016). The dependent variable in this analysis is 'bandwidth_BG_year' To answer the research question "What factors contribute to the bandwith_BG_year?" the independent variables that could influence bandwidth usage could include age, income, gender, location, population, etc. Using multiple linear regression you can examine the relationship between the dependent variable and each independent variable while controlling for the effects of other independent variables (Massaron, 2016). This allows for the identification of which independent variable is significantly impacting the dependent variable

and how much variation in the dependent variable can be explained by the independent variables. Multiple linear regression can also provide information about the strength and direction of the relationships between the variables and the goodness of the fit of the model (Massaron, 2016).

Part III: Data Preparation

C1. Data Cleaning

The methods discussed below are used to clean the raw data set of the churn data. Cleaning data is important in drawing accurate conclusions when analyzing data. It allows for different sorting options, filtering, and modification of the data set. Using the methods helps detect duplicates while maintaining the integrity of the data. It is necessary to detect and treat duplicate data because duplicate entries can lead to miscalculations or misrepresentations of the data. Python was used to detect duplicate data, missing values, outliers, and any other data quality issues in the churn data set. Utilizing the "import pandas as pd" command Panda was imported into Python. The read_cvs() function was used to read the churn data on my local hard drive. This file was assigned to the variable "df" for easy reference. To determine the data types included in the churn data the df.info(file_path). The data type of each variable is needed information due to certain functions working only with specific functions. This includes the column names and the number of non-null values for each column. Once datatypes are known the data could be cleaned. Cleaning data includes detecting duplicates, and identifying missing values, and outliers.

To determine if there were duplicate entries in the data the df.duplicated() function was completed. This function returns columns with TRUE or FALSE values. If the column returns a TRUE value there are duplicate records but if a FALSE value is returned there are no duplicates. The results of this function indicated all FALSE values meaning there were no duplicate values in the data set. To verify and count all entries that were FALSE the print(df.duplicated().value_counts()) was used. If duplicates were present the df.drop_duplicates() function could have been used to drop duplicate values.

The next step included determining if there were missing values. Missing values are usually represented in the form of nan, null, or none in the dataset. The df.isnull().sum() function was used. This function counted how many missing values were present in each column. There were no missing variables in the data set. Once duplicate and missing values were detected and

treated outliers were then determined for all quantitative variables. All the quantitative variables from the churn data were plotted on boxplots to visualize outliers. Outliers need to be detected and treated because outliers can provide incorrect/inconsistent collusions from the data. Outliers come from data entry errors, measurement errors, experimental errors, sampling errors, or novelties in the data (Lacrose & Lacrose, 2019). The scipy.stats function was used to calculate z-scores. There were two types of data from this data set, quantitative and qualitative data. Qualitative or categorical data (i.e. yes/no) requires re-expression or encoding of numbers to perform statical modeling (Lacrose & Lacrose, 2019). One hot encoding was thus used to transform categorical data into nominal data to be used in mathematical models like linear regressions. With one hot encoding, each categorical variable is represented as a binary vector where all elements are zero except the element corresponding to the category which is set to one. A loop was created to loop all independent variables to remove outliers and store the non-outlier data in a new data frame. For categorical columns label encoding was completed. The outliers were then filtered out and dropped based on z-score criterion. The non-outlier data for the variable was then stored in a new data frame (new_data[col]). One hot encoding was completed on the variable 'gender'. Dummy variables were created in this process and then dropped.

C2. Summary Statistics

Listed below is a description of the dependent variable (Bandwith_GB_year) and the independent variables used in this analysis.

Variable Name	Data Type	Description	Example
Population	Quantitative	Population of customer residence	8165
Children	Quantitative	Number of children of customer	5
Age	Quantitative	Age of customer	30
Income	Quantitative	Customer annual income reported	64256.81
Gender	Qualitative	Customer gender	Male
Outage_sec_perweek	Quantitative	Avg number of seconds per week of system outages in customer's neighborhood	12.63069124
Email	Quantitative	Number of emails sent to customer over past year	10
Contacts	Quantitative	Number of times customer contacted technical support	3
Yearly_equip_failure	Quantitative	Number of times customer's equipment failed and replaced	0
Techie	Qualitative	If customer considers themselves technically inclined	No
TechSupport	Qualitative	If customer has technical support add-on	Yes
Tenure	Quantitative	# of months customer has stayed with provider	10.06019902
MonthlyCharge	Quantitative	Amount charged to customer monthly	160.8055418
Bandwidth_GB_Year	Quantitative	Avg amount of data customer used (GB) in a year	1948.694497
Churn	Qualitative	If the customer discontinued services in the last month	Yes

Below there is a summary of the statistics for the dependent variable and the independent variables. It is beneficial to get the summary statics when running a logistic regression model because it provides information about the relationship between the independent variables and the dependent variable. This was done with code df.describe() and can be seen below. The dependent variable in this analysis was 'BandwidthGB_year'. The continuous independent variables include children, tenure, yearly_equip_failure, monthlycharge, age, income, email, contacts, outage_sec_perweek, population, and gender. The summary statistic:

df.describe():

	Population	Children	Age	Income	Outage_sec_perweek	Email	Contacts	Yearly_equip_failure	Tenure	MonthlyCharge	Bandwidth_GB_Year
count	8950.000000	8950.000000	8950.000000	8950.000000	8950.000000	8950.000000	8950.000000	8950.000000	8950.000000	8950.000000	8950.000000
mean	8508.409274	1.941453	53.161341	38329.400298	10.009065	12.021676	0.941676	0.374749	34.423473	172.78358	3379.459169
std	11759.988903	1.890668	20.634274	25123.528844	2.926500	3.011651	0.900798	0.582945	26.445276	42.99005	2185.204136
min	0.000000	0.000000	18.000000	348.670000	1.144796	3.000000	0.000000	0.000000	1.005104	79.97886	155.506715
25%	727.250000	0.000000	35.000000	19041.117500	8.031398	10.000000	0.000000	0.000000	7.892645	139.97923	1228.078013
50%	2750.000000	1.000000	53.000000	32778.475000	10.016014	12.000000	1.000000	0.000000	29.772986	167.48470	3120.633000
75%	11838.750000	3.000000	71.000000	52280.437500	11.961618	14.000000	2.000000	1.000000	61.389790	202.44330	5579.370794
max	52967.000000	8.000000	89.000000	124025.100000	18.851730	21.000000	3.000000	2.000000	71.999280	290.16041	7158.981530

To see the statistical breakdown df.describe() was used. This method is included in the data profiling states which is a type of data transformation that involves analyzing and summarizing data to gain insight into the characteristics and quality of the dataset. This is also useful for quickly getting a picture of the distribution of the data in a column, including its range, central tendency, and dispersion (Massaron, 2016). This method returns a series object that includes the following summary statics for each variable in the data frame:

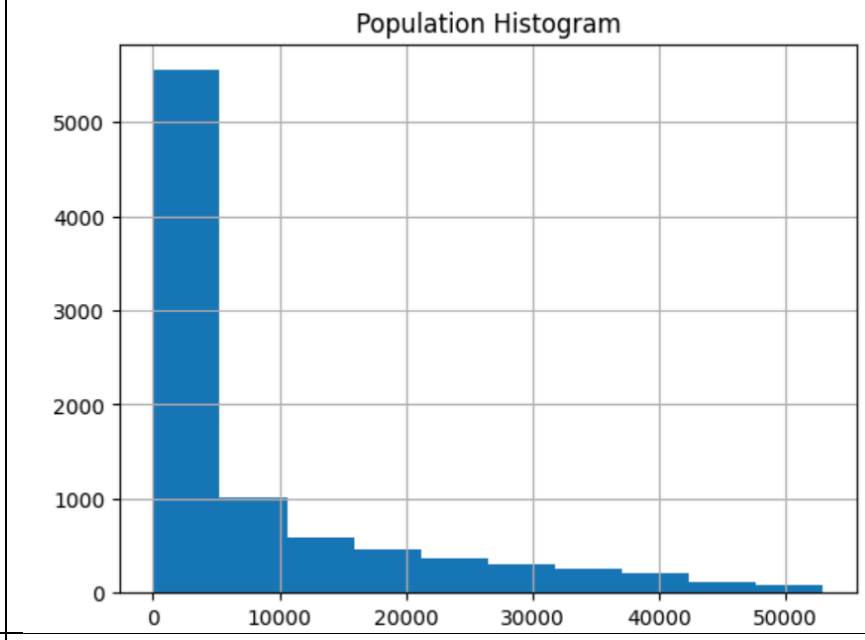
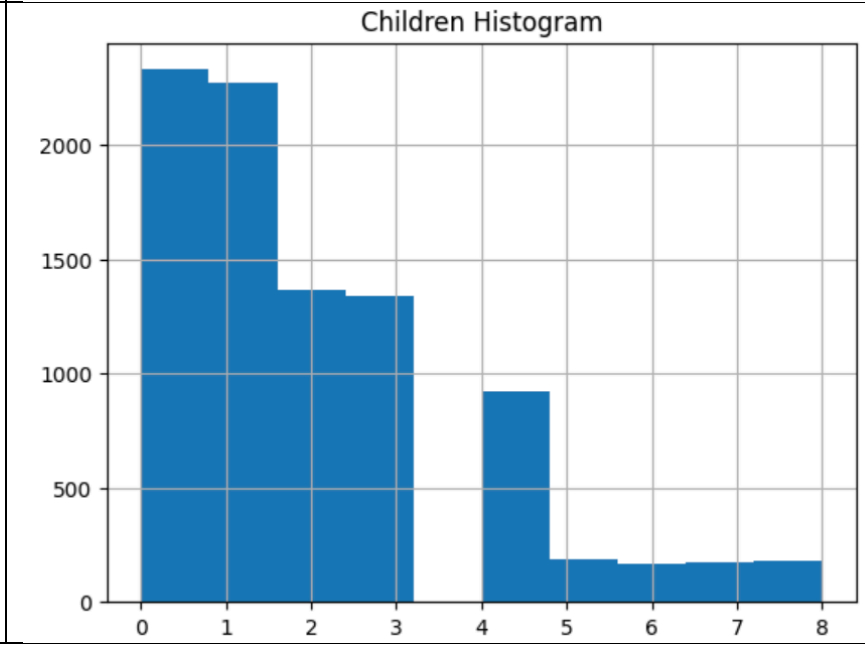
- count: the number of non-missing values in the column
- mean: the arithmetic mean of the values in the column
- std: the standard deviation of the values in the column
- min: the smallest value in the column
- 25% the 25th percentile of the values in the column
- 50% the 50th percentile (median) of the values in the column
- 75% the 75th percentile of the values in the column
- max: the largest value in the column

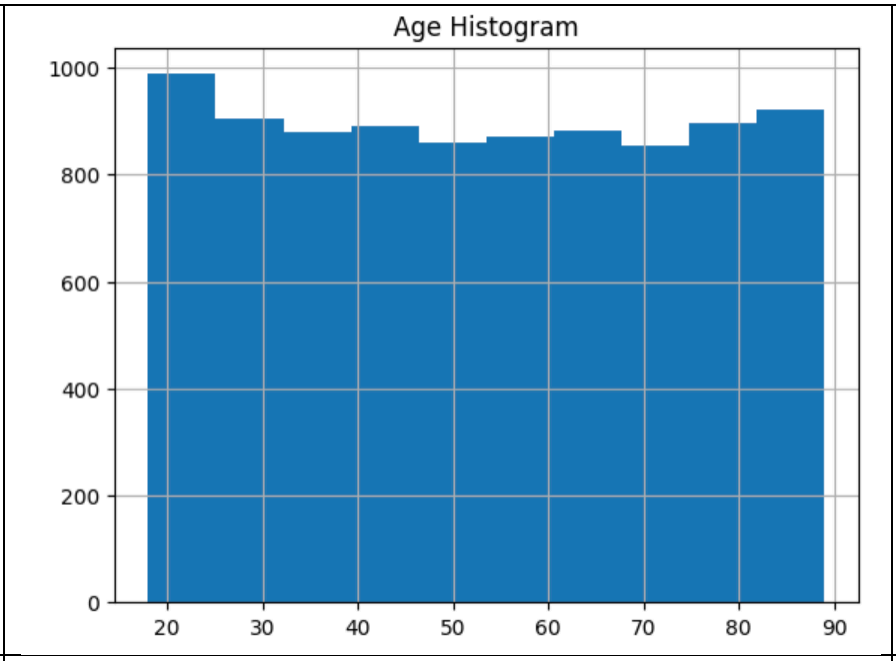
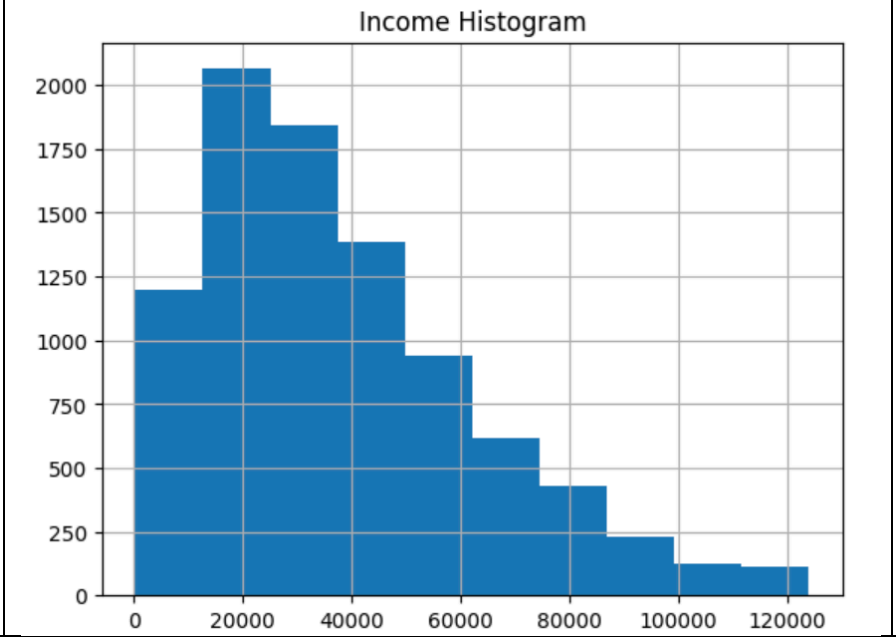
The count for the data set was 8950 observations for each variable. For the variable 'population' the minimum value is 0 which suggests that there may be some missing or errors in the dataset. The average population is 8508.41. When looking at the 'children' variable the mean value is 1.941453 meaning that on average each customer had 2 children. The standard deviation of 1.890668 suggests that there is considerable variation in the number of children among the customers. The mean age is 53.1 with a minimum age of 18 years old and maximum age is 89 years old. 75% of the customers were 71 years or less. The average income is 38329.40 with a minimum income 348.67 and the maximum income 124025.10. The average for outage_sec_perweek was 10.009065. The variable email has a mean of 10.0090065. The minimum was 3 and the maximum was 21. The majority of customers received 8 to 14 emails per week. For the variable contacts, there was an average number of contacts per customer of around 12. 50% of the customers have between 10 and 14 contacts. For the variable yearly_equip_failure the average yearly equipment failure is 0.37. Most customers did not experience any equipment failure and 75% of customers experiences failures once or less. The average tenure is 34.42 months. The minimum tenure is 1.01 months and the maximum is 72 months. The average monthly change for customers was $173 (172.783589). Monthly charges ranged from $80 (79.978860) and a maximum of $290 (290.160419). The average bandwidth used per year was 3379.46 GB. The 50th percentile of Bandwidth_GB_year was 3120.63 which means that half of the customers used less than this amount of bandwidth per year, while the other half used more per year.

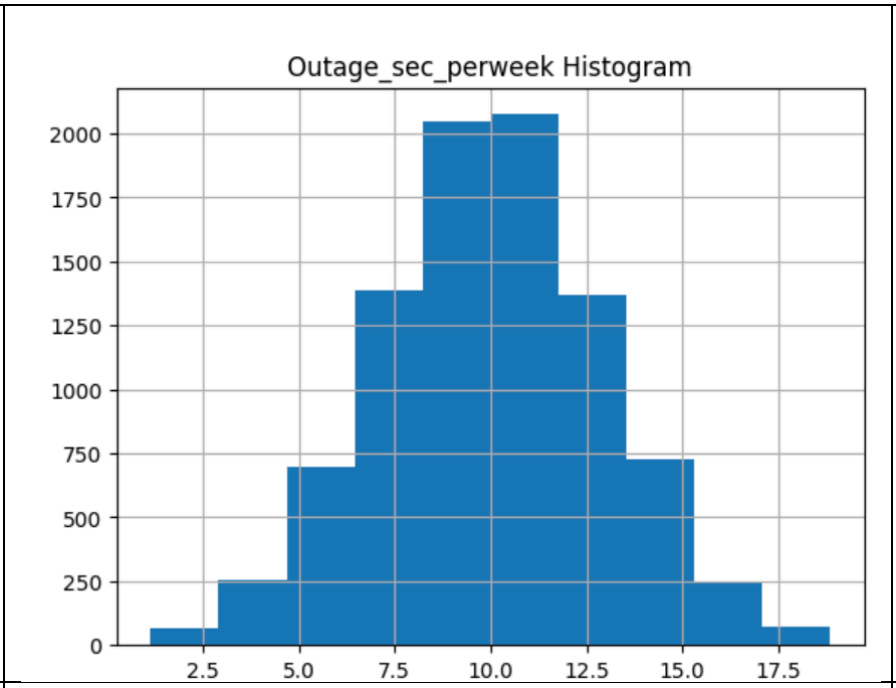
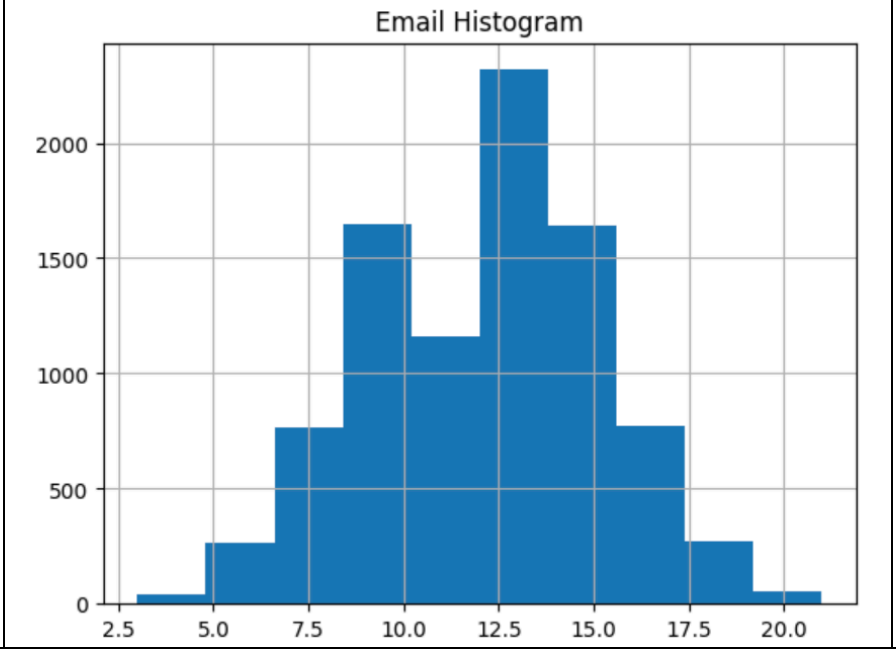
An analysis of variance (ANOVA) test can be performed before running a linear regression model to obtain information about the relationship between the dependent and independent variables when the relationship is not linear. This can help identify which variables are most likely to be useful in predicting the response variable. If there is a significant relationship but the relationship is not linear, a linear regression model may not be appropriate. An ANOVA test can also be used to test whether the assumptions of the linear regression model are met (Bruce, 2020). Welch's ANOVA is a modification of the traditional ANOVA that does not assume equal variances across groups and takes into account unequal variances (Bruce, 2020). A Welch's ANOVA was completed for each independent variable against the dependent variable 'Bandwith_GB_Year', shown below. The following analysis provided a p-value (p-unc) that gives an indication of the statistical significance of each independent variable. Further bivariant analyses are shown by visualization below.

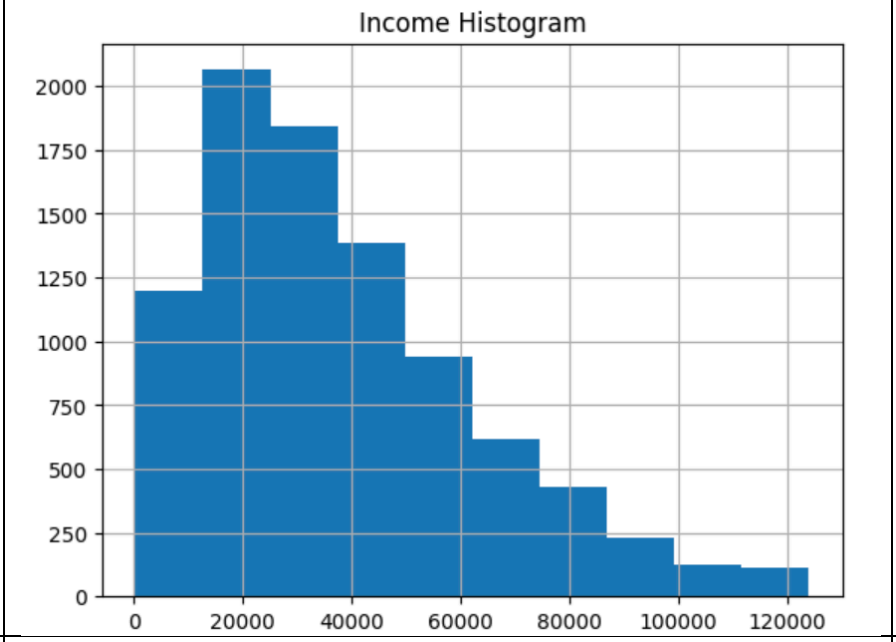
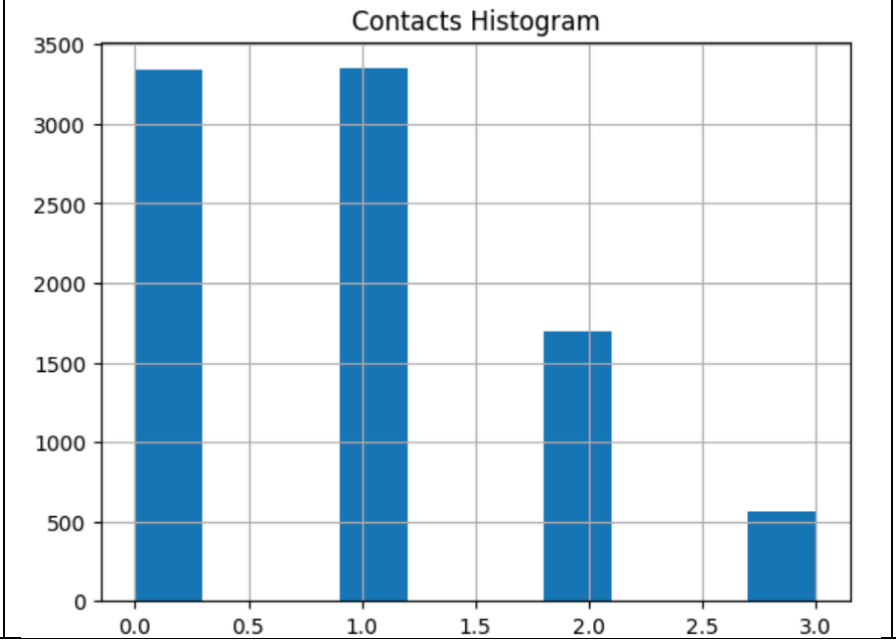
Code	*Output*
pg.welch_anova(dv='Bandwidth_GB_Year', between='Children', data=df)	Source ddof1 ddof2 F p-unc np2 0 Children 10 939.557131 1.545685 0.118271 0.001597
pg.welch_anova(dv='Bandwidth_GB_Year', between='Tenure', data=df)	Source ddof1 ddof2 F p-unc np2 0 Tenure 9995 1.119296e+07 0.024786 1.0 0.999997
pg.welch_anova(dv='Bandwidth_GB_Year', between='Yearly_equip_failure', data=df)	Source ddof1 ddof2 F p-unc np2 0 Yearly_equip_failure 2 4.646048 0.302717 0.752349 0.35
pg.welch_anova(dv='Bandwidth_GB_Year', between='MonthlyCharge', data=df)	Source ddof1 ddof2 F p-unc np2 0 MonthlyCharge 4 inf 0.0 1.0 1.0
pg.welch_anova(dv='Bandwidth_GB_Year', between='Age', data=df)	Source ddof1 ddof2 F p-unc np2 0 Age 4 inf 0.0 1.0 1.0
pg.welch_anova(dv='Bandwidth_GB_Year', between='Income', data=df)	Source ddof1 ddof2 F p-unc np2 0 Income 4 inf 0.0 1.0 1.0
pg.welch_anova(dv='Bandwidth_GB_Year', between='Email', data=df)	Source ddof1 ddof2 F p-unc np2 0 Email 4 inf 0.0 1.0 1.0
pg.welch_anova(dv='Bandwidth_GB_Year', between='Contacts', data=df)	Source ddof1 ddof2 F p-unc np2 0 Contacts 4 inf 0.0 1.0 1.0
pg.welch_anova(dv='Bandwidth_GB_Year', between='Outage_sec_perweek', data=df)	Source ddof1 ddof2 F p-unc np2 0 Outage_sec_perweek 4 inf 0.0 1.0 1.0
pg.welch_anova(dv='Bandwidth_GB_Year', between='Population', data=df)	Source ddof1 ddof2 F p-unc np2 0 Population 4 inf 0.0 1.0 1.0
pg.welch_anova(dv='Bandwidth_GB_Year', between='Phone', data=df)	Source ddof1 ddof2 F p-unc np2 0 Phone 1 2.882353 3.618915e-31 1.0 0.0
pg.welch_anova(dv='Bandwidth_GB_Year', between='TechSupport', data=df)	Source ddof1 ddof2 F p-unc np2 0 TechSupport 1 2.882353 3.618915e-31 1.0 0.0
pg.welch_anova(dv='Bandwidth_GB_Year', between='Techie', data=df)	Source ddof1 ddof2 F p-unc np2 0 Techie 1 2.882353 3.618915e-31 1.0 0.0
pg.welch_anova(dv='Bandwidth_GB_Year', between='Gender', data=df)	Source ddof1 ddof2 F p-unc np2 0 Gender 1 2.882353 3.618915e-31 1.0 0.0

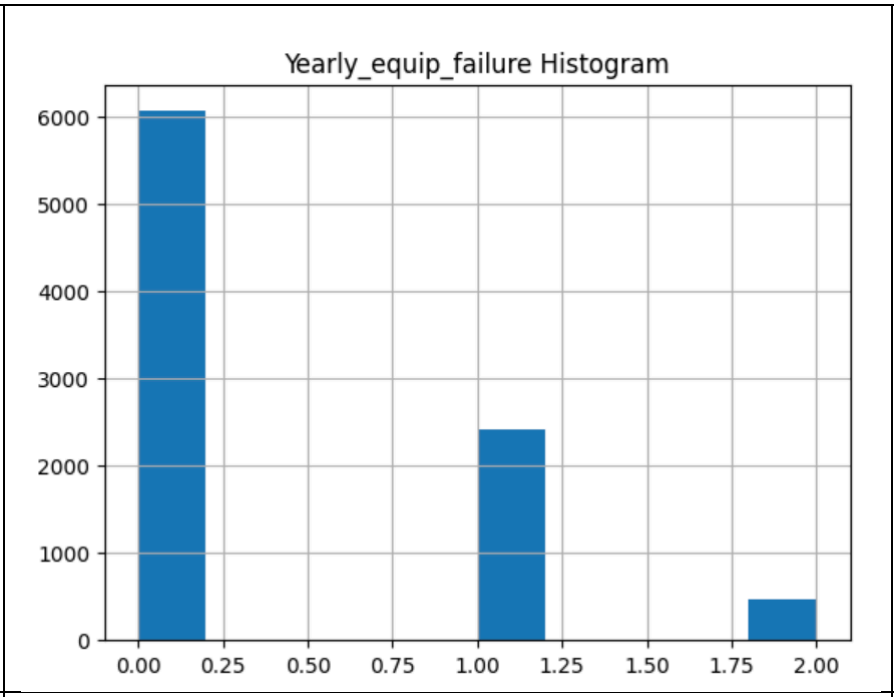
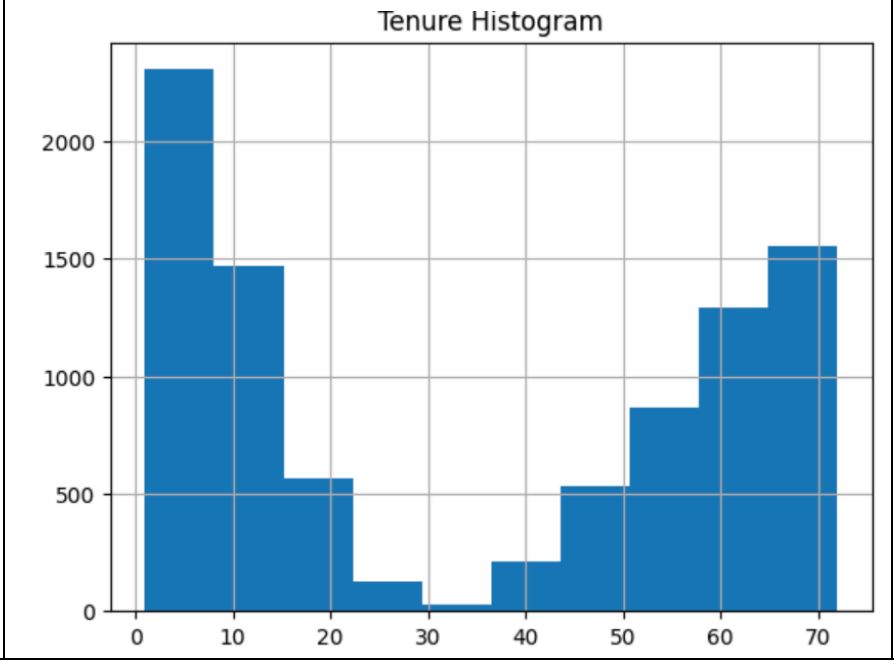
Univariate statistics is the statical analysis of a single variable at one time (Bruce et al., 2020). Below is the distribution of all independent variables in this analysis. A histogram and boxplot was used to visualize the distribution of each variable. The distribution of each independent variable is also listed below (LabXchange, n.d.).

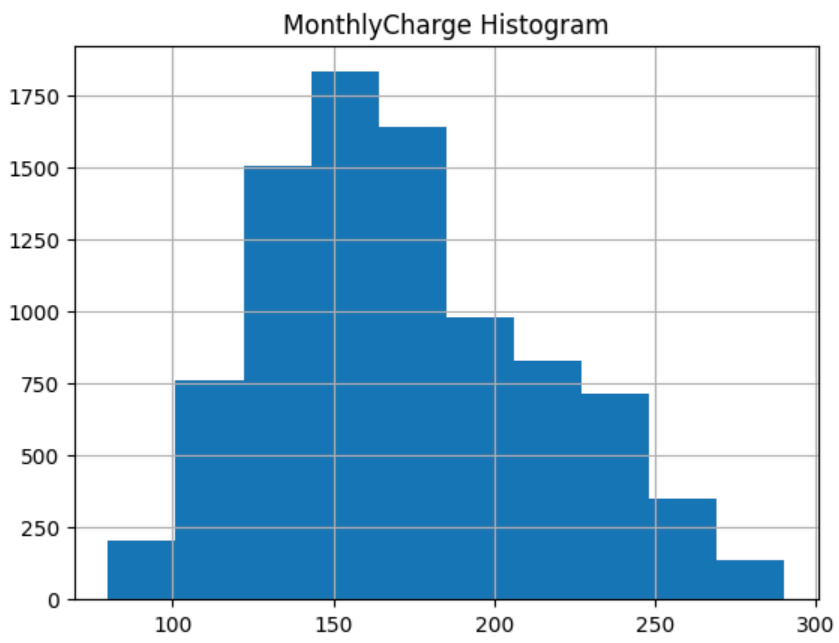
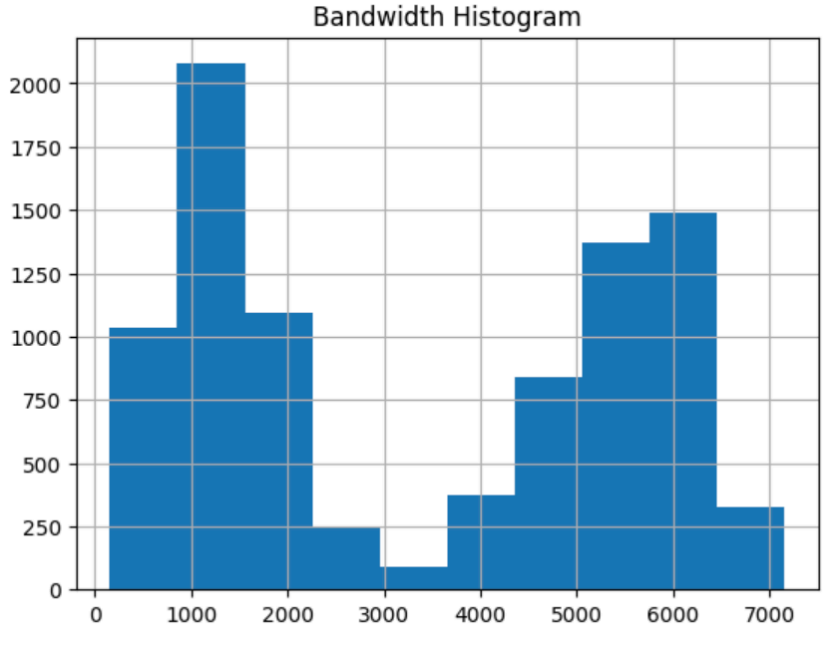
Variable	***Visuals (Holtz, n.d.)***	***Distribution***
Population		Right-Skewed
Children		Right-Skewed

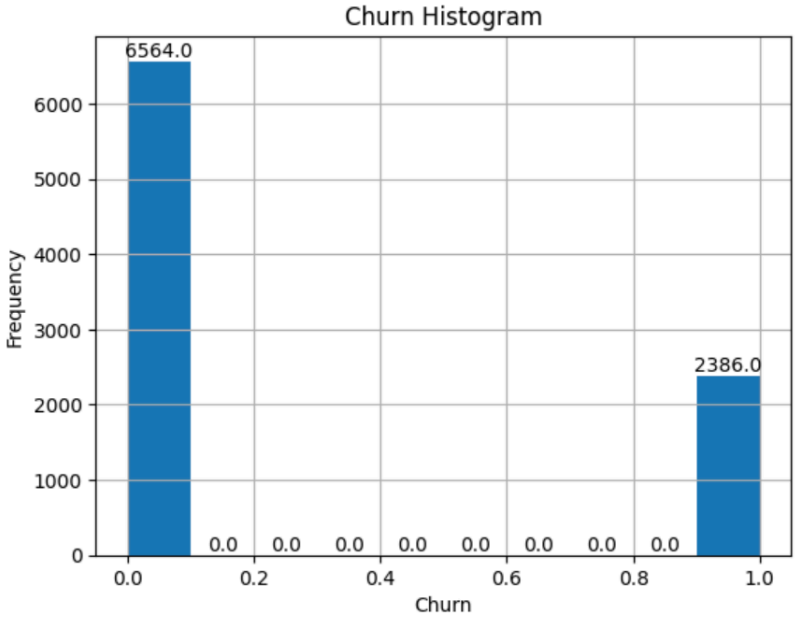
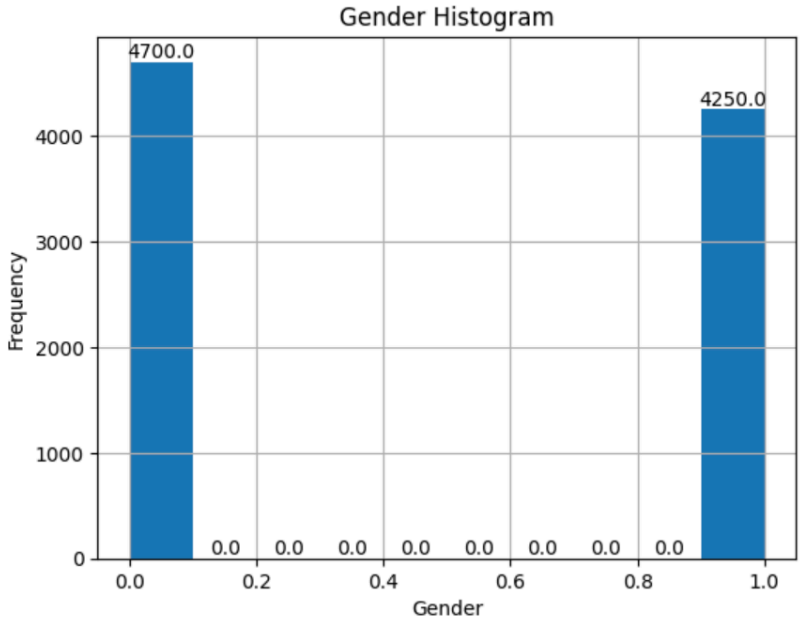
Age		Non-symmetric bimodal
Income		Right-Skewed

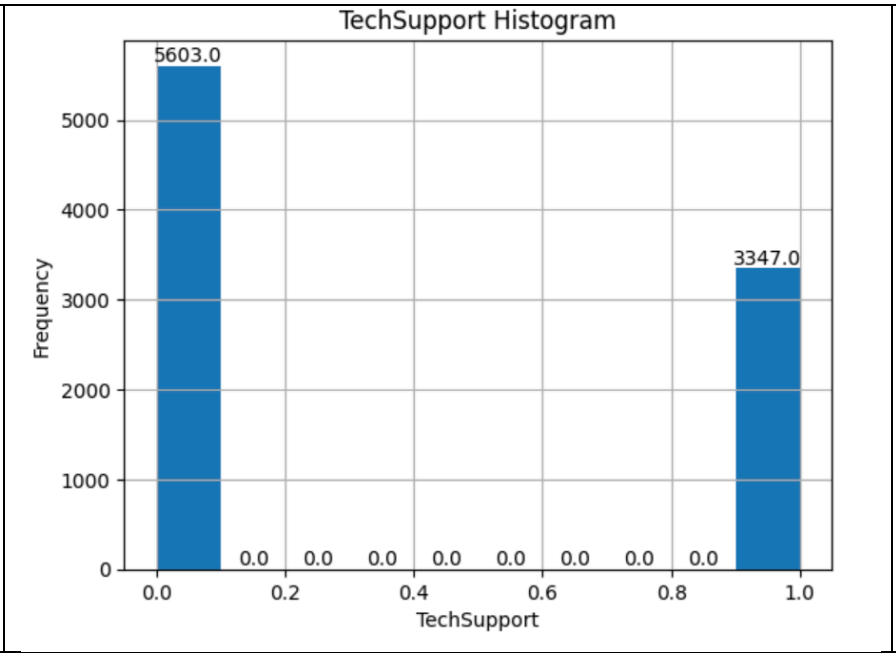
Outage_sec_perweek		Normal-unimodal
Email		Uniform

Income		Right-Skewed
Contacts		Normal-unimodal

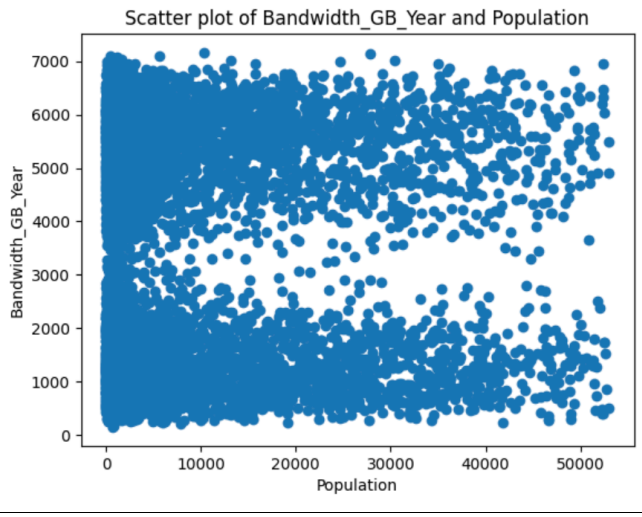
Yearly_equip_failure		Right-Skewed
Tenure		Normal-unimodal

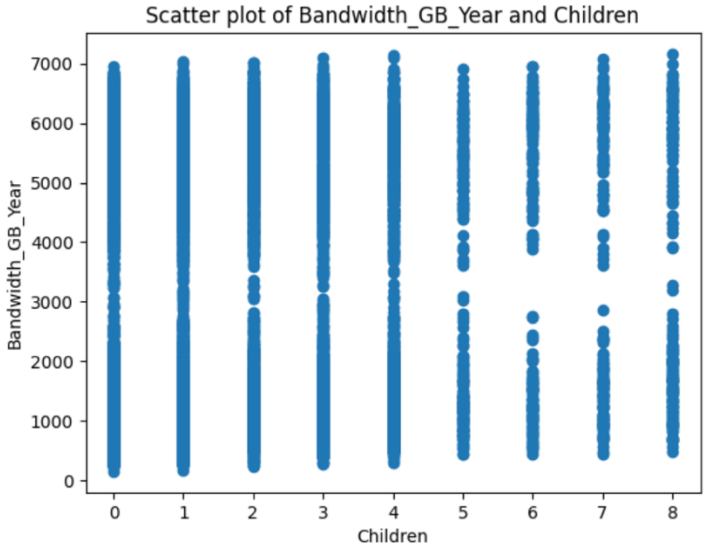
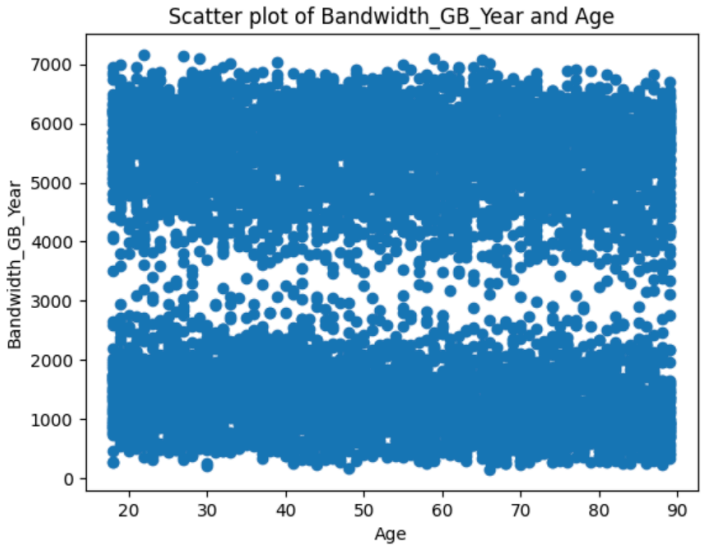
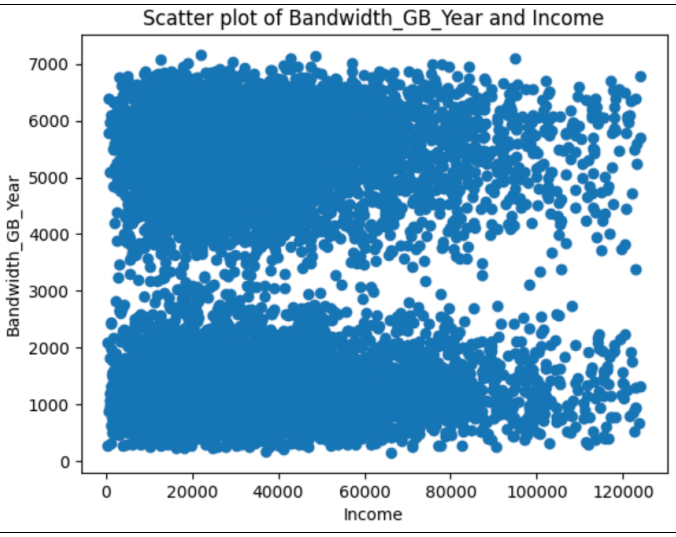
MonthlyCharge		Right-Skewed
Bandwidth_GB_Year		Non-symmetric bimodal

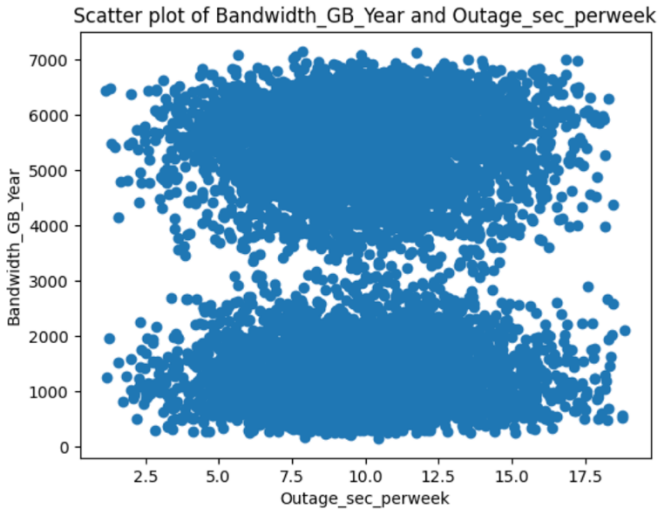
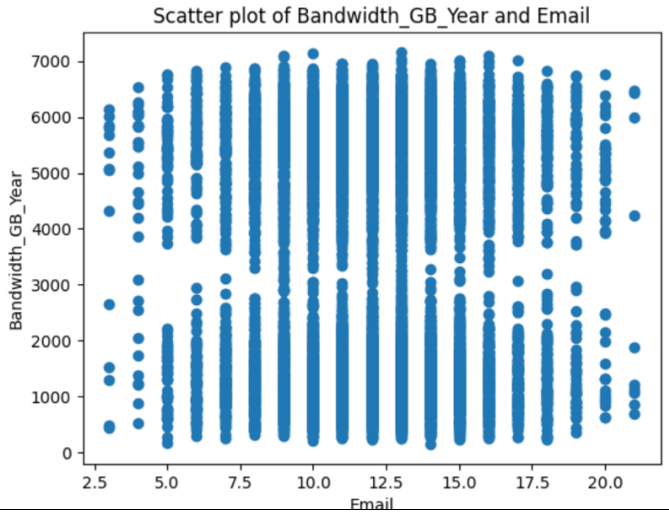
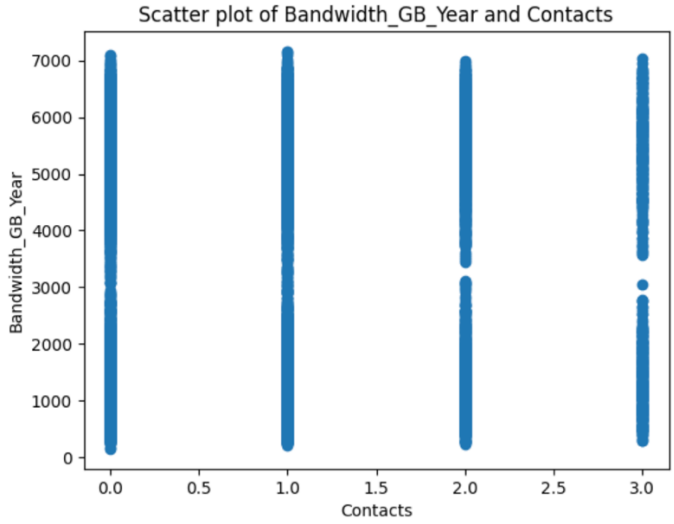
Churn		Right-Skewed
Gender		Right-Skewed
TechSupport		Right-Skewed

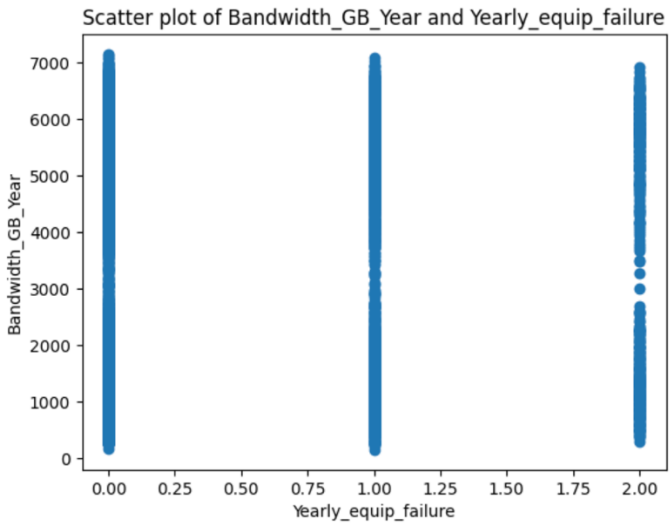
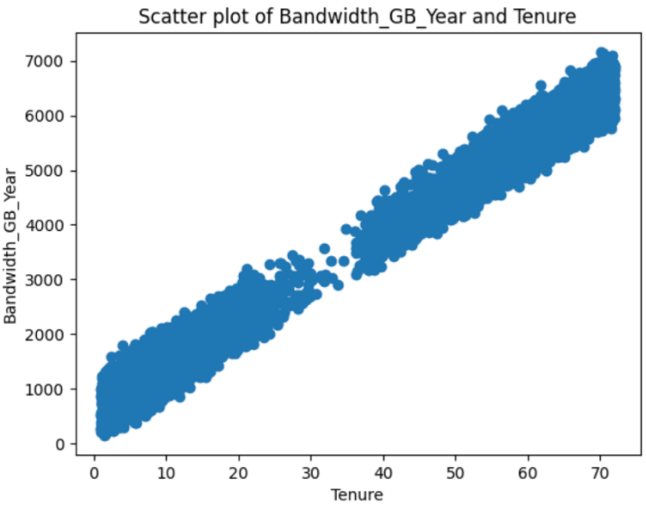
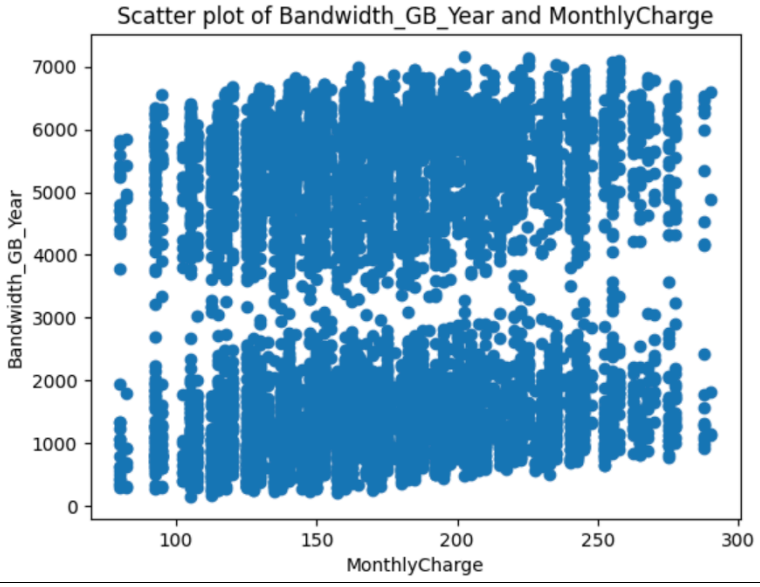


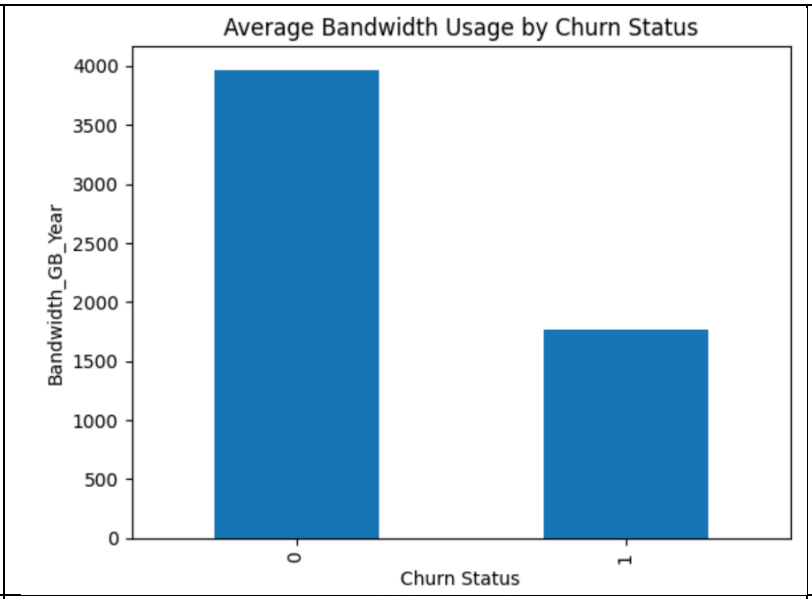
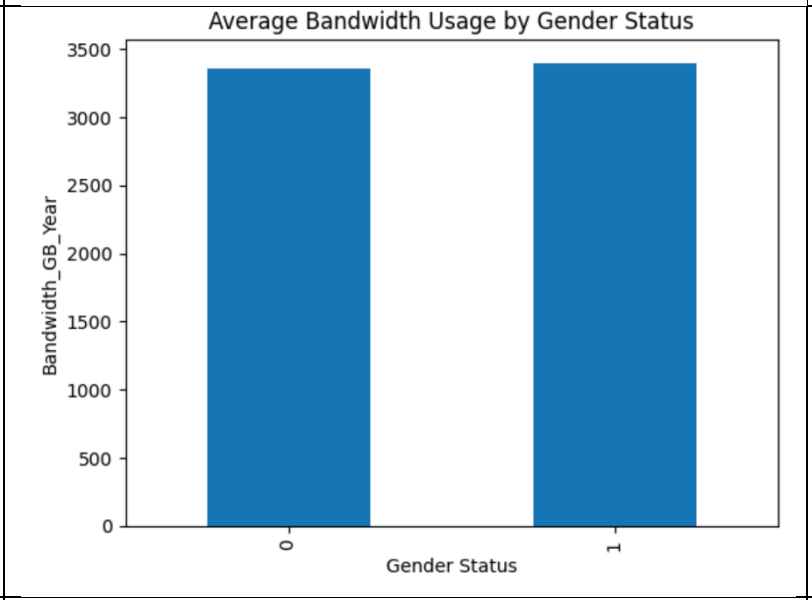
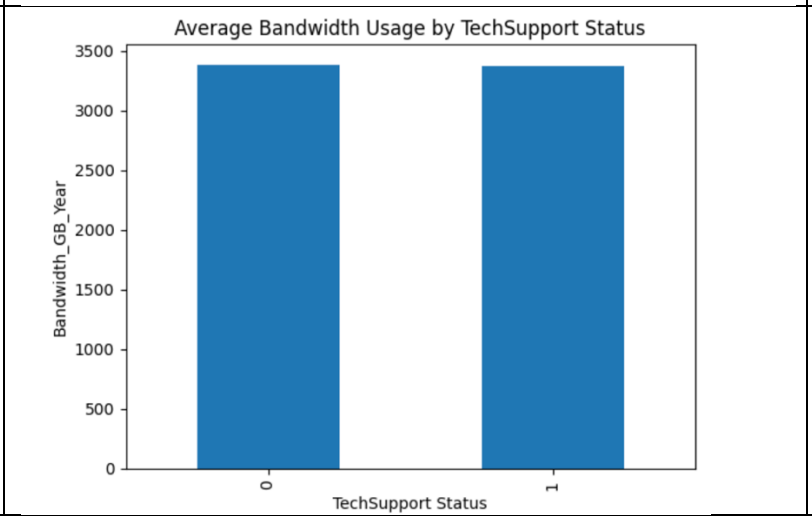
Bivariate statistical analysis refers to the statistical analysis of two variables at once (Bruce et al., 2020). A scatterplot was created for each independent variable to examine the relationship with the dependent variable. The distribution of each independent variable is also listed below (LabXchange, n.d.).

Independent Variable	***Bivariate Analysis (Holtz, n.d.)***	***Distribution***
Population		No relationship

Children		
Age		No relationship
Income		No relationship

Outage_sec_perweek		No relationship
Email		No relationship
Contacts		No relationship

Yearly_equip_failure		No relationship
Tenure		Linear, positive
MonthlyCharge		No relationship

Churn		
Gender		
TechSupport		

C4. Data Transformation

To determine which factors impact customer tenure in relation to the continuous dependent variable bandwidth_GB_year a multiple linear regression model was created. The regression model measures the independent variable in correlation to the dependent variable to see which has the most impact. Data wrangling was completed to examine the data for further analysis. Data wrangling is the process of cleaning, transforming, and organizing raw data from various sources into a consistent format for analysis (Lacrose & Lacrose, 2019). An important step in the data wrangling process is data transformation. Data transformation is a part of the data cleaning process that involves converting data into a standard format and/or applying calculations to generate new variables (Lacrose & Lacrose, 2019).

The data cleaning process began by determining the data types with the function df.info(file_path). The following data types were present in the churn data set:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 52 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Unnamed: 0            10000 non-null  int64
 1   CaseOrder             10000 non-null  int64
 2   Customer_id           10000 non-null  object
 3   Interaction           10000 non-null  object
 4   City                  10000 non-null  object
 5   State                 10000 non-null  object
 6   County                10000 non-null  object
 7   Zip                   10000 non-null  int64
 8   Lat                   10000 non-null  float64
 9   Lng                   10000 non-null  float64
 10  Population            10000 non-null  int64
 11  Area                  10000 non-null  object
 12  Timezone              10000 non-null  object
 13  Job                   10000 non-null  object
 14  Children              7505 non-null   float64
 15  Age                   7525 non-null   float64
 16  Education             10000 non-null  object
 17  Employment            10000 non-null  object
 18  Income                7510 non-null   float64
 19  Marital               10000 non-null  object
 20  Gender                10000 non-null  object
 21  Churn                 10000 non-null  object
 22  Outage_sec_perweek    10000 non-null  float64
 23  Email                 10000 non-null  int64
 24  Contacts              10000 non-null  int64
 25  Yearly_equip_failure  10000 non-null  int64
 26  Techie                7523 non-null   object
 27  Contract              10000 non-null  object
 28  Port_modem            10000 non-null  object
 29  Tablet                10000 non-null  object
```

```
 30  InternetService       10000 non-null  object
 31  Phone                 8974 non-null   object
 32  Multiple              10000 non-null  object
 33  OnlineSecurity        10000 non-null  object
 34  OnlineBackup          10000 non-null  object
 35  DeviceProtection      10000 non-null  object
 36  TechSupport           9009 non-null   object
 37  StreamingTV           10000 non-null  object
 38  StreamingMovies       10000 non-null  object
 39  PaperlessBilling      10000 non-null  object
 40  PaymentMethod         10000 non-null  object
 41  Tenure                9069 non-null   float64
 42  MonthlyCharge         10000 non-null  float64
 43  Bandwidth_GB_Year     8979 non-null   float64
 44  item1                 10000 non-null  int64
 45  item2                 10000 non-null  int64
 46  item3                 10000 non-null  int64
 47  item4                 10000 non-null  int64
 48  item5                 10000 non-null  int64
 49  item6                 10000 non-null  int64
 50  item7                 10000 non-null  int64
 51  item8                 10000 non-null  int64
dtypes: float64(9), int64(15), object(28)
```

There were 9 floats, 15 integers, and 28 strings present in the data set. Different data types have different methods for handling missing values (Lacrose & Lacrose, 2019). Duplicates in the data set could then be determined with function df.duplicated(). This function provided TRUE/FALSE values, where TRUE means duplicates were present and FALSE means duplicates were not present (Lacrose & Lacrose, 2019). The results below indicate there were no duplicate data entries in the data set.

```
0       False
1       False
2       False
3       False
4       False
       ...
9995    False
9996    False
9997    False
9998    False
9999    False
Length: 10000, dtype: bool
```

Missing values were then detected using the df.isnull().sum(). There were no missing values present in the data set as indicated below using df.isnull().sum().

```
CaseOrder             0
Customer_id           0
```

```
Interaction             0
UID                     0
City                    0
State                   0
County                  0
Zip                     0
Lat                     0
Lng                     0
Population              0
Area                    0
TimeZone                0
Job                     0
Children                0
Age                     0
Income                  0
Marital                 0
Gender                  0
Churn                   0
Outage_sec_perweek      0
Email                   0
Contacts                0
Yearly_equip_failure    0
Techie                  0
Contract                0
Port_modem              0
Tablet                  0
InternetService         0
Phone                   0
Multiple                0
OnlineSecurity          0
OnlineBackup            0
DeviceProtection        0
TechSupport             0
StreamingTV             0
StreamingMovies         0
PaperlessBilling        0
PaymentMethod           0
Tenure                  0
MonthlyCharge           0
Bandwidth_GB_Year       0
Item1                   0
Item2                   0
Item3                   0
Item4                   0
Item5                   0
Item6                   0
Item7                   0
Item8                   0
dtype: int64
```

The next step in the data transformation process was getting outliers. To detect outliers boxplots were first used as a visualization of outliers for each continuous variable using code:

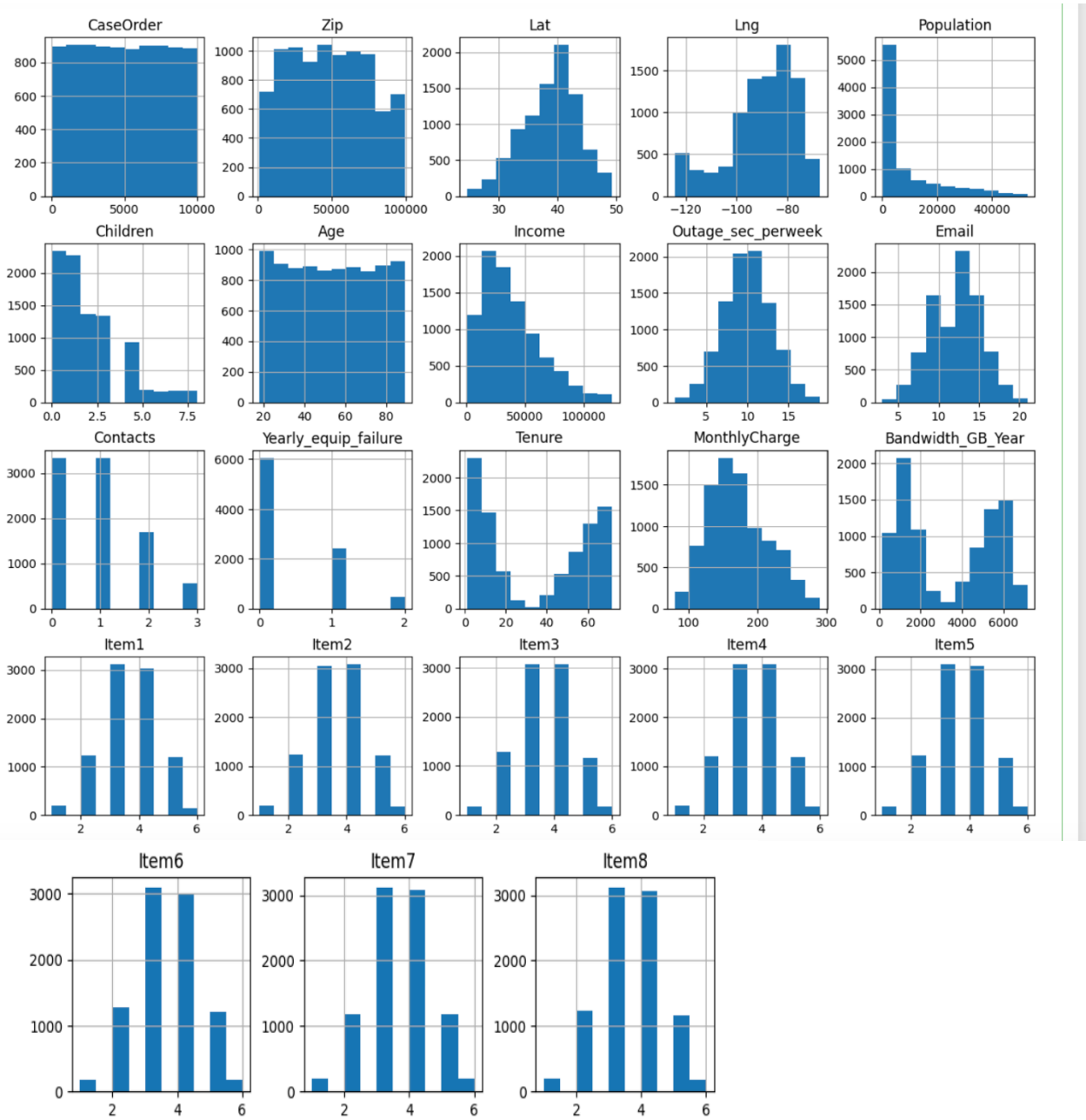
boxplot=seaborn.boxplot(x='Children',data=df)

To remove outliers and to ensure outliers were removed the following code was used :

 # check for outliers and remove
print(df.shape)
df = df[(np.abs(stats.zscore(df.select_dtypes(include=np.number))) < 3).all(axis=1)]
print(df.shape

The output:
```
(10000, 50)
(8950, 50)
```

df.hist(figsize = (15,15)) was used to visual columns as histograms after outliers had been removed.



Next, the least meaningful columns in the data set were dropped:

```
#Drop the less meaningful columns
df = df.drop(columns=['CaseOrder', 'Customer_id', 'Interaction', 'UID', 'City', 'State', 'County',
'Zip', 'Lat', 'Lng',
                'TimeZone', 'Job','Marital', 'Contract', 'Port_modem', 'Tablet', 'InternetService',
'Phone', 'Multiple',
                'OnlineSecurity', 'OnlineBackup', 'Area','DeviceProtection', 'StreamingTV',
'StreamingMovies', 'PaperlessBilling',
                'PaymentMethod', 'Item1', 'Item2',
                'Item3', 'Item4', 'Item5', 'Item6', 'Item7', 'Item8'])
```

The shape was verified again to ensure same number of data points/observations remained:

```
# Display the dimension of dataframe
df.shape
```

Output:

```
(8950, 15)
```

Next to visualize all the data in the first five rows of the data set:

```
# display data set with all the columns
df.head()
```

	Population	Children	Age	Income	Gender	Churn	Outage_sec_perweek	Email	Contacts	Yearly_equip_failure	Techie	TechSupport	Tenure	MonthlyCharge	Bandwidth_GB_Year
1	10446	1	27	21704.77	Female	Yes	11.699080	12	0	1	Yes	No	1.156681	242.632554	800.982766
2	3735	4	50	9609.57	Female	No	10.752800	9	0	1	Yes	No	15.754144	159.947583	2054.706961
3	13863	1	48	18925.23	Male	No	14.913540	15	2	0	Yes	No	17.087227	119.956840	2164.579412
4	11352	0	83	40074.19	Male	Yes	8.147417	16	2	1	No	Yes	1.670972	149.948316	271.493436
5	17701	3	83	22660.20	Female	No	8.420993	15	3	1	No	No	7.000994	185.007692	1039.357983

df.describe() was used to examine the statistical output of the quantitative independent variables.

Data was ensured with no missing values again with the remaining columns.

 # Validate there are no nulls

df.isnull().sum()

```
Population              0
Children                0
Age                     0
Income                  0
Gender                  0
Churn                   0
Outage_sec_perweek      0
Email                   0
Contacts                0
Yearly_equip_failure    0
Techie                  0
TechSupport             0
Tenure                  0
MonthlyCharge           0
Bandwidth_GB_Year       0
```

To transform categorical variables into numeric variables the encoding process was completed. This can help identify potential issues with the data and evaluate the effectiveness of different transformation methods. Label encoding was applied to the categorical variables. constructing a logistic regression model from all predictors that were identified above. Dummy variables are used to represent categorical variables as numerical values. Categorical variables, such as gender, cannot be included directly in a logistic regression model because they are not numerical. Categorical variables can be converted into dummy variables, which are numerical variables that represent a category (Massaron, 2016). Dummy variables are useful because they allow one to compare the effect of different categories on the probability of a certain outcome.

Create dummy variables in order to encode categorical, yes/no data points into 1/0 numerical values.

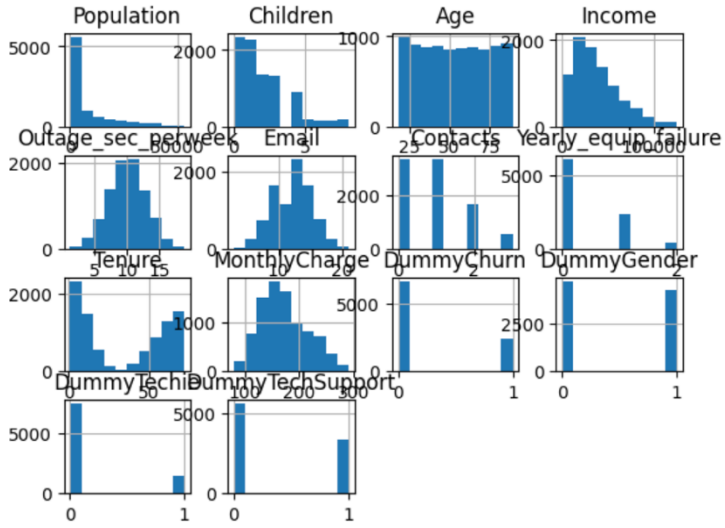
```
df['DummyChurn'] = [1 if v == 'Yes' else 0 for v in df['Churn']]
df['DummyGender'] = [1 if v == 'Male' else 0 for v in df['Gender']]
df['DummyTechie'] = [1 if v == 'Yes' else 0 for v in df['Techie']]
df['DummyTechSupport'] = [1 if v == 'Yes' else 0 for v in df['TechSupport']]
```

Drop original categorical features from dataframe

```
df = df.drop(columns=['Gender', 'Churn', 'Techie','TechSupport'])
```

Histograms are a useful way to understand the distribution of a set of numeric data. A histogram is a graph that displays the frequency of data points in a set of data, divided into intervals. Histograms for each of the columns were visualized to quickly be able to see the range of values that the data set covers and how frequently each value occurs.

df.hist():



To check the remaining data set and variables:
df.columns

```
Index(['Population', 'Children', 'Age', 'Income', 'Outage_sec_perweek',
   'Email', 'Contacts', 'Yearly_equip_failure', 'Tenure', 'MonthlyCharge',
       'DummyChurn', 'DummyGender', 'DummyTechie', 'DummyTechSupport'],
      dtype='object')
```

To gain further insights into the distribution of the data visualization techniques shown above were used. To complete the data transformation process data profiling using code df.nunique() that involved calculating the number of unique values. By calculating the unique values one can see the diversity and distribution of the data and identify any potential issues that need to be addressed before the data can be used for analysis or modeling (Massaron, 2016).

```
Population              5414
Children                   9
Age                       72
Income                  8945
Outage_sec_perweek      8940
Email                     19
Contacts                   4
Yearly_equip_failure       3
Tenure                  8948
```

```
MonthlyCharge            748
Bandwidth_GB_Year       8950
DummyChurn                 2
DummyGender                2
DummyTechie                2
DummyTechSupport           2
dtype: int64
```

C5. Prepared Data Set

The new data frame was saved to a new file and attached as a csv file.

```
# Prepared dataset saved to new file
df.to_csv('D208_task1_revision1.csv', index=False)
```

Part IV: Model Comparison and Analysis

D1. Initial Model

A multiple linear regression model is a statistical model used to predict a dependent variable based on two or more independent variables. The model assumes that there is a linear relationship between the independent and dependent variables. Below is the initial linear regression model utilizing the independent and dependent variables listed above.

Initial Linear Regression Model (Zach, 2020):

```
# Set up your independent and dependent variables
X = df[['Population', 'Children', 'Age', 'Income', 'Outage_sec_perweek', 'Email', 'Contacts',
'Yearly_equip_failure',
      'Tenure', 'MonthlyCharge', 'DummyChurn', 'DummyGender', 'DummyTechie',
'DummyTechSupport']]
y = df['Bandwidth_GB_Year']

# Calculate VIFs for each independent variable
vifs = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]

# Print the VIFs for each variable
for i, col in enumerate(X.columns):
   print(f'{col}: {vifs[i]:.2f}')


# Add a constant term to the independent variable matrix
X = sm.add_constant(X)

# Fit an OLS regression model on the data
model = sm.OLS(y, X).fit()
```

 # Print the model summary
print(model.summary())

```
Population: 1.52
Children: 2.02
Age: 6.81
Income: 3.18
Outage_sec_perweek: 10.32
Email: 12.39
Contacts: 2.06
Yearly_equip_failure: 1.40
Tenure: 3.68
MonthlyCharge: 16.51
DummyChurn: 2.20
DummyGender: 1.89
DummyTechie: 1.20
DummyTechSupport: 1.63
```

OLS Regression Results

Dep. Variable:	Bandwidth_GB_Year	R-squared:	0.990
Model:	OLS	Adj. R-squared:	0.990
Method:	Least Squares	F-statistic:	6.187e+04
Date:	Wed, 22 Mar 2023	Prob (F-statistic):	0.00
Time:	14:53:20	Log-Likelihood:	-61005.
No. Observations:	8950	AIC:	1.220e+05
Df Residuals:	8935	BIC:	1.221e+05
Df Model:	14		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	84.7034	17.931	4.724	0.000	49.555	119.851
Population	-1.691e-05	0.000	-0.085	0.932	-0.000	0.000
Children	31.3210	1.237	25.326	0.000	28.897	33.745
Age	-3.3467	0.113	-29.530	0.000	-3.569	-3.125
Income	0.0002	9.3e-05	1.973	0.049	1.17e-06	0.000
Outage_sec_perweek	-0.1608	0.799	-0.201	0.840	-1.727	1.405
Email	-0.4887	0.776	-0.629	0.529	-2.010	1.033
Contacts	0.4100	2.595	0.158	0.874	-4.678	5.498
Yearly_equip_failure	1.2333	4.009	0.308	0.758	-6.625	9.091
Tenure	83.0339	0.104	798.654	0.000	82.830	83.238
MonthlyCharge	2.8665	0.061	47.327	0.000	2.748	2.985
DummyChurn	126.4408	6.719	18.818	0.000	113.270	139.612
DummyGender	72.6334	4.684	15.508	0.000	63.452	81.815
DummyTechie	-6.0648	6.276	-0.966	0.334	-18.367	6.237
DummyTechSupport	-24.5396	4.872	-5.037	0.000	-34.089	-14.990

Omnibus:	10516.227	Durbin-Watson:	1.979
Prob(Omnibus):	0.000	Jarque-Bera (JB):	891.402
Skew:	0.470	Prob(JB):	2.72e-194
Kurtosis:	1.772	Cond. No.	3.57e+05

D2. Justification of Model Reduction

To answer the research question what factors contribute to the bandwidth_GB_year? model evaluation metrics can be used to justify a reduced initial model and improve the accuracy in predicting the dependent variable. One possible selection procedure is the backward elimination method. This involves removing independent variables from the model one at a time based on their statical significance until only statically significant variables remain (Massaron, 2016). This approach is justified since the goal is to identify the most important independent variables while minimizing the number of irrelevant variables. The r-squared, adjusted r-squared, and root mean squared error can be used as model evaluation metrics.

The initial model had 14 independent variables. The initial model has an r-squared value of 0.990 meaning that 99% of the variance in the dependent variable is explained by the independent variables in the initial model (Massaron, 2016). The high r-squared value also means the model fits the data well and its a good predictor of the dependent variable, Bandwidth_GB_year. The p-value is associated with the prob F-statistic. The Prob F-statistic is 0.000 and the F-statistic is high at 6.187e+04 indicating the model is statistically significant. When looking at the coefficients of the independent variables some independent variables were large and others were relatively small. 'Tenure' had a large coefficient 83.0339 indicating a large effect on the independent variable.

The p-values of the independent variables also indicated significate with some of the variables while others were not significant. Independent variables with p-values greater than 0.05 were: 'population', 'outage_sec_week', 'email', 'contacts', 'yearly_equip_failure, 'dummytechie'. The model indicated a condition number of 3.57 e+05. This might indicate that there is multicollinearity among the predictor variables meaning some of the independent variables may be highly correlated with one another.

To check for multicollinearity among the independent variables the variance inflation factor for each variable can be calculated (Massaron, 2016). The VIF measures how much the variance of the estimated regression coefficient is increased due to collinearity. A VIF greater than 5 or 10 indicates that the variable is highly collinear with other variables and may need to be removed from the model (Massaron, 2016). The independent variables with the highest VIF were 'outage_sec_perweek' (10.32), 'email '(12.39), 'monthlycharge' (16.51.

Reduced Model with VIF (removal of 'Outage_sec_perweek'):

#reduced model with removal of Outage_sec_per due to p-value of 0.840 and VIF of 10.29

```
Children: 2.01
Age: 6.63
Income: 3.15
Email: 11.24
Contacts: 2.05
Yearly_equip_failure: 1.40
Tenure: 3.67
MonthlyCharge: 15.08
DummyChurn: 2.20
DummyGender: 1.88
DummyTechie: 1.20
DummyTechSupport: 1.62
```

OLS Regression Results

Dep. Variable:	Bandwidth_GB_Year	R-squared:	0.990
Model:	OLS	Adj. R-squared:	0.990
Method:	Least Squares	F-statistic:	7.220e+04
Date:	Wed, 22 Mar 2023	Prob (F-statistic):	0.00
Time:	18:46:57	Log-Likelihood:	-61005.
No. Observations:	8950	AIC:	1.220e+05
Df Residuals:	8937	BIC:	1.221e+05
Df Model:	12		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	83.0031	16.078	5.162	0.000	51.486	114.520
Children	31.3168	1.236	25.328	0.000	28.893	33.741
Age	-3.3466	0.113	-29.540	0.000	-3.569	-3.124
Income	0.0002	9.3e-05	1.976	0.048	1.46e-06	0.000
Email	-0.4898	0.776	-0.631	0.528	-2.011	1.031
Contacts	0.3992	2.595	0.154	0.878	-4.687	5.485
Yearly_equip_failure	1.2314	4.008	0.307	0.759	-6.625	9.088
Tenure	83.0341	0.104	798.843	0.000	82.830	83.238
MonthlyCharge	2.8662	0.061	47.339	0.000	2.748	2.985
DummyChurn	126.4560	6.718	18.824	0.000	113.287	139.625
DummyGender	72.6240	4.682	15.510	0.000	63.446	81.802
DummyTechie	-6.0624	6.275	-0.966	0.334	-18.363	6.238
DummyTechSupport	-24.5201	4.870	-5.035	0.000	-34.066	-14.974

Omnibus:	10538.832	Durbin-Watson:	1.979
Prob(Omnibus):	0.000	Jarque-Bera (JB):	891.516
Skew:	0.470	Prob(JB):	2.57e-194
Kurtosis:	1.772	Cond. No.	3.16e+05

Reduced Model with VIF (removal of Email'):

reduced model with removal of 'Email' due to p-value of 0.528 and VIF of 11.24

```
Children: 1.98
Age: 6.12
Income: 3.07
Contacts: 2.03
Yearly_equip_failure: 1.40
Tenure: 3.64
MonthlyCharge: 11.93
DummyChurn: 2.19
DummyGender: 1.86
DummyTechie: 1.20
DummyTechSupport: 1.62
```

OLS Regression Results

Dep. Variable:	Bandwidth_GB_Year	R-squared:	0.990
Model:	OLS	Adj. R-squared:	0.990
Method:	Least Squares	F-statistic:	7.220e+04
Date:	Wed, 22 Mar 2023	Prob (F-statistic):	0.00
Time:	18:46:57	Log-Likelihood:	-61005.
No. Observations:	8950	AIC:	1.220e+05
Df Residuals:	8937	BIC:	1.221e+05
Df Model:	12		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	83.0031	16.078	5.162	0.000	51.486	114.520
Children	31.3168	1.236	25.328	0.000	28.893	33.741
Age	-3.3466	0.113	-29.540	0.000	-3.569	-3.124
Income	0.0002	9.3e-05	1.976	0.048	1.46e-06	0.000
Email	-0.4898	0.776	-0.631	0.528	-2.011	1.031
Contacts	0.3992	2.595	0.154	0.878	-4.687	5.485
Yearly_equip_failure	1.2314	4.008	0.307	0.759	-6.625	9.088
Tenure	83.0341	0.104	798.843	0.000	82.830	83.238
MonthlyCharge	2.8662	0.061	47.339	0.000	2.748	2.985
DummyChurn	126.4560	6.718	18.824	0.000	113.287	139.625
DummyGender	72.6240	4.682	15.510	0.000	63.446	81.802
DummyTechie	-6.0624	6.275	-0.966	0.334	-18.363	6.238
DummyTechSupport	-24.5201	4.870	-5.035	0.000	-34.066	-14.974

Omnibus:	10538.832	Durbin-Watson:	1.979
Prob(Omnibus):	0.000	Jarque-Bera (JB):	891.516
Skew:	0.470	Prob(JB):	2.57e-194
Kurtosis:	1.772	Cond. No.	3.16e+05

Reduced Model with VIF (removal of 'MonthlyCharge'):

reduced model with removal of 'MonthlyCharge due to p-value of 0.000 but VIF of 11.93

```
Children: 1.92
Age: 4.87
Income: 2.88
Contacts: 1.97
Yearly_equip_failure: 1.38
Tenure: 2.92
DummyChurn: 1.63
DummyGender: 1.81
DummyTechie: 1.20
DummyTechSupport: 1.56
```

OLS Regression Results

Dep. Variable:	Bandwidth_GB_Year	R-squared:	0.987
Model:	OLS	Adj. R-squared:	0.987
Method:	Least Squares	F-statistic:	6.910e+04
Date:	Wed, 22 Mar 2023	Prob (F-statistic):	0.00
Time:	18:50:16	Log-Likelihood:	-62006.
No. Observations:	8950	AIC:	1.240e+05
Df Residuals:	8939	BIC:	1.241e+05
Df Model:	10		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	488.1134	10.957	44.549	0.000	466.636	509.591
Children	31.5758	1.383	22.837	0.000	28.865	34.286
Age	-3.3163	0.127	-26.177	0.000	-3.565	-3.068
Income	0.0001	0.000	1.380	0.168	-6.04e-05	0.000
Contacts	0.3530	2.902	0.122	0.903	-5.335	6.041
Yearly_equip_failure	1.0502	4.482	0.234	0.815	-7.735	9.835
Tenure	84.1320	0.113	742.531	0.000	83.910	84.354
DummyChurn	262.0220	6.794	38.564	0.000	248.703	275.341
DummyGender	72.5860	5.235	13.864	0.000	62.323	82.849
DummyTechie	-11.8630	7.014	-1.691	0.091	-25.613	1.887
DummyTechSupport	4.4452	5.402	0.823	0.411	-6.144	15.034

Omnibus:	464.899	Durbin-Watson:	1.989
Prob(Omnibus):	0.000	Jarque-Bera (JB):	319.423
Skew:	0.352	Prob(JB):	4.35e-70
Kurtosis:	2.399	Cond. No.	2.00e+05

Reduced Model with VIF (removal of 'Contacts'):

reduced model with removal of 'Contacts' due to p-value of 0.903

```
Children: 1.92
Age: 4.67
Income: 2.85
Yearly_equip_failure: 1.38
Tenure: 2.88
DummyChurn: 1.62
DummyGender: 1.80
DummyTechie: 1.20
DummyTechSupport: 1.55
```

OLS Regression Results

Dep. Variable:	Bandwidth_GB_Year	R-squared:	0.987
Model:	OLS	Adj. R-squared:	0.987
Method:	Least Squares	F-statistic:	6.910e+04
Date:	Wed, 22 Mar 2023	Prob (F-statistic):	0.00
Time:	18:50:16	Log-Likelihood:	-62006.
No. Observations:	8950	AIC:	1.240e+05
Df Residuals:	8939	BIC:	1.241e+05
Df Model:	10		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	488.1134	10.957	44.549	0.000	466.636	509.591
Children	31.5758	1.383	22.837	0.000	28.865	34.286
Age	-3.3163	0.127	-26.177	0.000	-3.565	-3.068
Income	0.0001	0.000	1.380	0.168	-6.04e-05	0.000
Contacts	0.3530	2.902	0.122	0.903	-5.335	6.041
Yearly_equip_failure	1.0502	4.482	0.234	0.815	-7.735	9.835
Tenure	84.1320	0.113	742.531	0.000	83.910	84.354
DummyChurn	262.0220	6.794	38.564	0.000	248.703	275.341
DummyGender	72.5860	5.235	13.864	0.000	62.323	82.849
DummyTechie	-11.8630	7.014	-1.691	0.091	-25.613	1.887
DummyTechSupport	4.4452	5.402	0.823	0.411	-6.144	15.034

Omnibus:	464.899	Durbin-Watson:	1.989
Prob(Omnibus):	0.000	Jarque-Bera (JB):	319.423
Skew:	0.352	Prob(JB):	4.35e-70
Kurtosis:	2.399	Cond. No.	2.00e+05

Reduced Model with VIF (removal of 'Yearly_equip_failure'):

reduced model with removal of 'Yearly_equip_failure' due to p-value of 0.815

```
Children: 1.91
Age: 4.60
Income: 2.84
Tenure: 2.86
DummyChurn: 1.61
DummyGender: 1.80
DummyTechie: 1.20
DummyTechSupport: 1.55
```

OLS Regression Results

Dep. Variable:	Bandwidth_GB_Year	R-squared:	0.987
Model:	OLS	Adj. R-squared:	0.987
Method:	Least Squares	F-statistic:	8.639e+04
Date:	Wed, 22 Mar 2023	Prob (F-statistic):	0.00
Time:	18:54:32	Log-Likelihood:	-62006.
No. Observations:	8950	AIC:	1.240e+05
Df Residuals:	8941	BIC:	1.241e+05
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	488.8215	10.497	46.569	0.000	468.246	509.397
Children	31.5763	1.382	22.852	0.000	28.868	34.285
Age	-3.3162	0.127	-26.181	0.000	-3.564	-3.068
Income	0.0001	0.000	1.381	0.167	-6.02e-05	0.000
Tenure	84.1322	0.113	742.639	0.000	83.910	84.354
DummyChurn	262.0267	6.793	38.571	0.000	248.710	275.343
DummyGender	72.5815	5.234	13.867	0.000	62.321	82.842
DummyTechie	-11.8712	7.014	-1.693	0.091	-25.619	1.877
DummyTechSupport	4.4452	5.401	0.823	0.410	-6.141	15.032

Omnibus:	465.271	Durbin-Watson:	1.989
Prob(Omnibus):	0.000	Jarque-Bera (JB):	319.490
Skew:	0.352	Prob(JB):	4.20e-70
Kurtosis:	2.399	Cond. No.	1.93e+05

Reduced Model with VIF (removal of 'TechSupport'):
reduced model with removal of 'TechSupport' due to p-value of 0.410

```
Children: 1.90
Age: 4.46
Income: 2.82
Tenure: 2.83
DummyChurn: 1.60
DummyGender: 1.80
DummyTechie: 1.20
```

OLS Regression Results

Dep. Variable:	Bandwidth_GB_Year	R-squared:	0.987
Model:	OLS	Adj. R-squared:	0.987
Method:	Least Squares	F-statistic:	9.874e+04
Date:	Wed, 22 Mar 2023	Prob (F-statistic):	0.00
Time:	18:56:54	Log-Likelihood:	-62006.
No. Observations:	8950	AIC:	1.240e+05
Df Residuals:	8942	BIC:	1.241e+05
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	490.3040	10.341	47.414	0.000	470.034	510.574
Children	31.5757	1.382	22.852	0.000	28.867	34.284
Age	-3.3137	0.127	-26.169	0.000	-3.562	-3.066
Income	0.0001	0.000	1.384	0.166	-5.99e-05	0.000
Tenure	84.1329	0.113	742.675	0.000	83.911	84.355
DummyChurn	262.1494	6.792	38.599	0.000	248.836	275.463
DummyGender	72.5395	5.234	13.860	0.000	62.280	82.799
DummyTechie	-11.8311	7.013	-1.687	0.092	-25.579	1.916

Omnibus:	465.606	Durbin-Watson:	1.989
Prob(Omnibus):	0.000	Jarque-Bera (JB):	319.968
Skew:	0.352	Prob(JB):	3.31e-70
Kurtosis:	2.399	Cond. No.	1.90e+05

Reduced Model with VIF (removal of 'DummyTechie'):

reduced model with removal of 'DummyTechie' due to p-value of 0.092

```
Children: 1.90
Age: 4.44
Income: 2.82
Tenure: 2.82
DummyChurn: 1.58
DummyGender: 1.80
```

OLS Regression Results

Dep. Variable:	Bandwidth_GB_Year	R-squared:	0.987
Model:	OLS	Adj. R-squared:	0.987
Method:	Least Squares	F-statistic:	1.152e+05
Date:	Wed, 22 Mar 2023	Prob (F-statistic):	0.00
Time:	18:57:53	Log-Likelihood:	-62008.
No. Observations:	8950	AIC:	1.240e+05
Df Residuals:	8943	BIC:	1.241e+05
Df Model:	6		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	488.5588	10.290	47.479	0.000	468.388	508.730
Children	31.5984	1.382	22.867	0.000	28.890	34.307
Age	-3.3128	0.127	-26.159	0.000	-3.561	-3.065
Income	0.0001	0.000	1.383	0.167	-6e-05	0.000
Tenure	84.1287	0.113	742.737	0.000	83.907	84.351
DummyChurn	261.3584	6.776	38.570	0.000	248.076	274.641
DummyGender	72.6012	5.234	13.870	0.000	62.341	82.862

Omnibus:	465.337	Durbin-Watson:	1.989
Prob(Omnibus):	0.000	Jarque-Bera (JB):	320.168
Skew:	0.353	Prob(JB):	3.00e-70
Kurtosis:	2.399	Cond. No.	1.89e+05

D3. Reduced Linear Regression Model

Final Reduced Model:

reduced model with removal of ' Income' due to p-value of 0.167

```
Children: 1.87
Age: 3.87
Tenure: 2.68
DummyChurn: 1.55
DummyGender: 1.79
```

OLS Regression Results

Dep. Variable:	Bandwidth_GB_Year	R-squared:	0.987
Model:	OLS	Adj. R-squared:	0.987
Method:	Least Squares	F-statistic:	1.382e+05
Date:	Wed, 22 Mar 2023	Prob (F-statistic):	0.00
Time:	18:58:43	Log-Likelihood:	-62009.
No. Observations:	8950	AIC:	1.240e+05
Df Residuals:	8944	BIC:	1.241e+05
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	494.1545	9.462	52.226	0.000	475.607	512.702
Children	31.6139	1.382	22.878	0.000	28.905	34.323
Age	-3.3135	0.127	-26.164	0.000	-3.562	-3.065
Tenure	84.1289	0.113	742.701	0.000	83.907	84.351
DummyChurn	261.3743	6.777	38.571	0.000	248.091	274.658
DummyGender	72.4215	5.233	13.839	0.000	62.164	82.679

Omnibus:	465.649	Durbin-Watson:	1.989
Prob(Omnibus):	0.000	Jarque-Bera (JB):	320.424
Skew:	0.353	Prob(JB):	2.64e-70
Kurtosis:	2.399	Cond. No.	260.

E1. Model Comparison:

The initial regression model and reduced regression model can be compared for a full analysis of the independent variables on the dependent variable. The initial model had 14 independent variables, which were used to predict the association with the dependent variable, 'bandwidth_GB_year'. The reduced model had 5 variables: 'children', 'age', tenure', 'churn', and 'gender'. The r-squared values of both models indicate there is a significant amount of variation in the response variables. the initial model had an r-squared value of 0.990 and the reduced model had an r-squared value of 0.987. This means that model explains 98.7% of the variance in the dependent variable. There was slightly less variation in the reduced model compared to the initial model due to a decrease in r-squared values. The initial model had many variables with a

p-value greater than 0.05. In the reduced model the independent variables show significance. The AIC is a measure of the relative quality of the model given the data set. The AIC of the initial model was 122,000 and the reduced model was 124,000. Since the AIC had a lower value in the initial model this indicated the initial model was a better fit for the data. However, the difference in the AIC values between the two models is relatively small and may not need to take other statical factors into account as well. The F-statistic measures the overall significance of the model. A higher F-statistic indicates the model is a better fit. The reduced model had a lower F-statistic than the initial model. the initial model's f-statistic was 8.639e+04 and in the reduced model was 1.382e+05. Therefore, the reduced model has a better fit to the data compared to the initial model. There was a slight difference in the log-likelihood of the initial and reduced model. The initial model had a log-likelihood of -61005 and the reduced model was -62009 showing the initial model to be a better fit. However, since the difference was so small other factors have to be taken into account. The condition number of the initial model was 3.16+05 and in the reduced model it was 260 which suggests the initial model had a high degree of multicollinearity. The reduced model had a much lower condition number of 206 indicating that multicollinearity may have been less of an issue in the reduced model.

A metric to analyze the reduced regression model is the mean squared error (MSE). The MSE is a measure of the average squared difference between the predicted and actual values (Massaron, 2016). The lower the MSE the better the model performs. The MSE of the 61013.54 suggests that the reduced model may not be a good fit for the data

MSE code and output (GeeksforGeeks, 2023):

```
X = df[['Children', 'Age','Tenure', 'DummyChurn', 'DummyGender']]
y = df['Bandwidth_GB_Year']

# Fit reduced model
model = sm.OLS(y, sm.add_constant(X[['Children', 'Age','Tenure', 'DummyChurn',
'DummyGender']])).fit()

# Get predicted values and actual values
y_pred = model.predict(sm.add_constant(X[['Children', 'Age','Tenure', 'DummyChurn',
'DummyGender']]))
y_true = y

# Calculate MSE
mse = mean_squared_error(y_true, y_pred)
print("Mean Squared Error: ", mse)
```

MSE: `Mean Squared Error:  61013.544358285566`

E2. Output and Calculation

Cross-validation was used to validate the performance of the model. Cross-validation helps to reduce the risk of overfitting by providing a more reliable estimate of a model's performance on the data. The RMSE below indicates that the difference between the predicted and actual values in the dataset is very small leading to a better-fit model. The standard deviation of the RMSE is also small and the model has a consistent level of accuracy across the dataset. The cross-validation scores below are the accuracy scores obtained for each fold of the cross-validation along with the mean scores (GeeksforGeeks, 2023). The cross-validation scores indicate how well the logistic model performed. The mean score is the average accuracy score across the folds. The higher the mean score, the better the model is expected to perform with new data.

Cross-validation code (GeeksforGeeks, 2023):

```
# Define the logistic regression model
linreg = LinearRegression()

# Define the independent and dependent variables
X = df[[ 'Children', 'Age','Tenure', 'DummyChurn', 'DummyGender']]
y = df['Bandwidth_GB_Year']

# Perform cross-validation on the logistic regression model
scores = cross_val_score(linreg, X, y, cv=5)

# Print the cross-validation scores
print("Cross-validation scores:", scores)
print("Mean score:", scores.mean())
```

```
Cross-validation scores: [0.82146388 0.8350788  0.98691488 0.88445365 0.88
818804]
Mean score: 0.8832198488973741
```

The cross-validation scores range from 0.821 to 0.986 indicating varying degrees of accuracy across the data set. The mean score is 0.88 and provides an estimate of the model's performance. This suggests the model has a good performance on the data.

Standardized coefficients were calculated due to the measured variables being on different scales. Getting the standardized coefficient allows for comparison of the relative importance of variables in predicting the outcome variable, regardless of the unit of measurement.

standardized coefficents:

```
# Fit the OLS model
X = sm.add_constant(X)
model = sm.OLS(y, X).fit()

# Get the coefficient estimates and standard errors
coef = model.params.values[1:]
std_err = model.bse[1:]

# Calculate the standard deviation of each predictor variable
std_dev = np.std(X.iloc[:, 1:], axis=0)

# Standardize the coefficients
coef_std = coef / std_dev

# Print the standardized coefficients
print(coef_std)
```

```
Children        16.721955
Age             -0.160593
Tenure           3.181423
DummyChurn     591.107400
DummyGender    145.026497
```

The coefficients and intercept were obtained from the code below. The coefficients repr esent the estimated values of the slopes of the regression line for each independent variable. The 5 coefficients below represent the slope of the five independent variables and the intercept, 0.394 5, represents with a y-intercept of the slope.

Coefficients:

```
# create some sample data
X = np.random.rand(100, 5)
y = np.random.rand(100)

# fit the linear regression model
reg = LinearRegression().fit(X, y)

# get the coefficients and intercept
coefficients = reg.coef_
```

```
intercept = reg.intercept_

print('Coefficients:', coefficients)
print('Intercept:', intercept)
```

```
Coefficients: [ 0.06505519 0.0086776  -0.0304789 0.1511595 0.1051755 ]
Intercept: 0.3945925040540118
```

The residual standard error (RSE) is a measure of the variability of the errors in a linear regression model. It measures the average amount that the response variable deviates from the dependent variable. The RSE will show the goodness of fit of a linear regression model. A small RSE suggests the model is a good fit for the data and the predicted values are very close to the actual values. However, this doesn't mean the model is perfect and there may be some variability. A RSE of 247 and an r-squared value of 0.987 indicates less error in the model's prediction.

RSE code & output:

```
# Load your data into X and y variables
X = df[['Children', 'Age','Tenure', 'DummyChurn', 'DummyGender']]
y = df['Bandwidth_GB_Year']

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create a linear regression model
model = LinearRegression()

# Fit the model to the training data
model.fit(X_train, y_train)

# Use the model to make predictions on the test data
y_pred = model.predict(X_test)

# Calculate the mean squared error and residual standard error
mse = np.mean((y_test - y_pred) ** 2)
rse = np.sqrt(mse / (len(y_test) - 2))

# Print the RSE
print("Residual Standard Error:", rse)
```

RSE:

```
Residual Squared Error:  247.00919893454488
```

A residual plot is a graphical representation of the errors. A well-fitting model will have residuals that are randomly distributed around zero, meaning the model is able to explain the variation in the dependent variable based on the independent variables (Massaron, 2016). Based on the residual plot below the model is not linear.

Residual Plot (Holtz, n.d.):

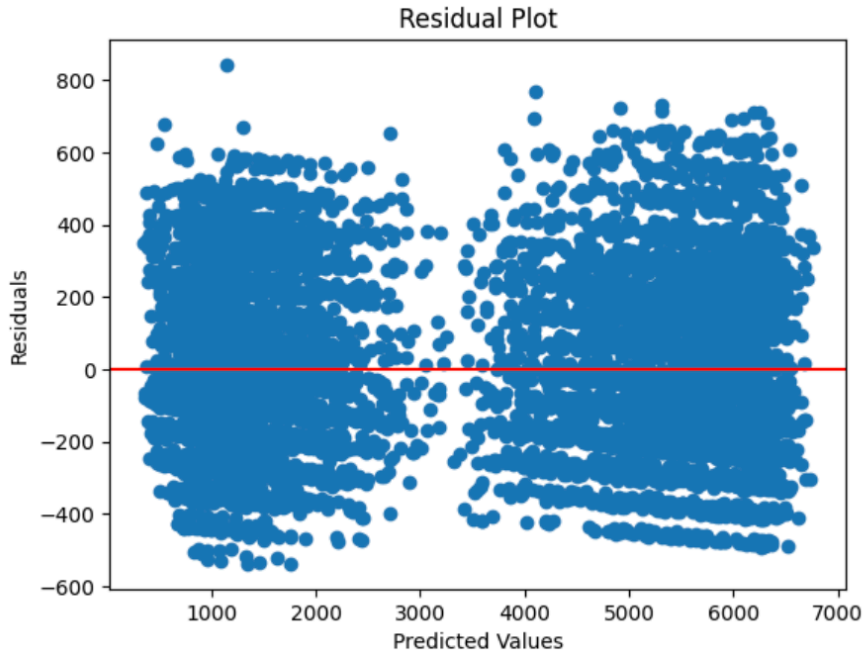
```
# Define the predictor variables
X = df[['Children', 'Age','Tenure', 'DummyChurn', 'DummyGender']]
y = df['Bandwidth_GB_Year']

# Fit the model
model = sm.OLS(y, sm.add_constant(X)).fit()

# Generate the predicted values
y_pred = model.predict(sm.add_constant(X))

# Calculate the residuals
residuals = y - y_pred

# Create the residual plot
plt.scatter(y_pred, residuals)
plt.axhline(y=0, color='r', linestyle='-')
plt.xlabel('Predicted Values')
plt.ylabel('Residuals')
plt.title('Residual Plot')
plt.show()
```



Cook's distance is a measure used to assess the influence of individual data points on the fitted regression model by measuring the difference between the estimated regression coefficients with and without the inclusion of each observation (Massaron, 2016). A large cook's distance means the corresponding data points has a large influence on the regression coefficients and may be an outlier driving the data.

Cook's Distance:

```
# create some sample data
X = np.random.rand(100, 5)
y = np.random.rand(100)

# fit the linear regression model
model = sm.OLS(y, sm.add_constant(X)).fit()

# get the Cook's distance
influence = model.get_influence()
cook_distance = influence.cooks_distance[0]

print('Cook\'s distance:', cook_distance)
```

```
Cook's distance: [2.32850470e-03 3.02479147e-03 4.16434330e-02 4.71189898e-03
 5.96763813e-03 2.67363160e-02 2.00701025e-04 6.92057500e-03
 1.08103039e-02 3.56868459e-03 1.91776710e-02 3.06294894e-03
 8.73739614e-03 5.69597626e-04 2.39578906e-03 7.79597480e-03
 1.33750719e-02 7.38020795e-04 6.45927702e-02 2.46355531e-05
 1.39731188e-04 3.07069621e-02 1.76967883e-03 2.70937805e-03
 1.71649351e-02 3.11462538e-03 4.25834229e-02 2.28030235e-03
 1.37174226e-02 3.75261792e-03 1.23331320e-02 8.08375580e-04
 3.13233604e-03 3.61342899e-03 1.89819349e-03 1.10292812e-02
 2.12988067e-02 3.00518845e-03 6.54047763e-04 6.81054894e-03
 2.42876833e-02 2.18301815e-04 1.29351205e-02 1.74916176e-02
 4.14653693e-04 3.81805376e-02 1.92670975e-03 4.89714242e-03
 2.09945783e-02 1.67052841e-02 1.36089120e-03 3.21362621e-02
 1.17114520e-02 8.73546186e-04 2.39405951e-03 5.75640188e-02
 1.93615894e-02 3.97049509e-03 2.25332469e-04 1.02621223e-02
 2.89575992e-02 2.47480134e-02 8.45790593e-03 1.40926039e-02
 2.70035644e-02 1.67323391e-03 2.01842933e-02 1.12286384e-02
 8.41183784e-02 5.44937542e-04 8.58338019e-03 1.20925746e-02
 1.43829458e-03 9.30255927e-03 1.31185458e-03 2.21633420e-04
 1.07365824e-05 1.08218251e-03 1.60726058e-03 2.00303722e-02
 5.88424814e-04 2.13013687e-02 6.14491389e-04 4.36523773e-03
 1.10631827e-02 5.86076758e-03 6.18955877e-04 1.03252193e-03
 2.24480863e-02 2.42692136e-02 2.30754609e-03 9.28917989e-05
 7.74225298e-03 2.74540162e-04 1.44468718e-03 3.16086567e-02
 4.21323958e-03 3.47902802e-04 8.40916017e-04 1.68243397e-03]
```

E3. Code

An error-free copy of the code used to support the implementation of the linear regression model using python is attached.

Part V: Data Summary and Implications

F1. Results

a. A regression equation for the reduced model:

A regression equation is a mathematical equation that expressed the relationship between a dependent variable and the independent variables(s). The equation takes the form of a line. The equation below is a regression equation for the reduced model for the dependent variable Bandwidth_GB_Year and the independent variables 'children', 'age', 'tenure', 'dummychurn', and 'dummygender':

Bandwidth_GB_Year = 494.1545 + 31.6139 * Children - 3.3135 * Age + 84.1289 * Tenure + 261.3743 * DummyChurn + 72.4215 * DummyGender

The intercept is 494.1545 represents the predicted value of bandwidth_gb_year when all the other predictors are equal to zero. Thus, the intercept represents the expected bandwidth usage for a customer with zero tenure, no children, and is not a techie or churned.

b. An interpretation of the coefficients of the reduced model:

The coefficients represent the change in the dependent variable per unit increase of each of the independent variables while holding all other variables constant. The coefficient can be used to make predictions on new data using the same model. The constant term is 494.1545 representing the predicted value of the dependent variable when the independent variables are equal to zero.

The coefficient 'children' is 31.6139 signifying that a rise of one unit in 'children' is associated with an increase of bandwidth_GB-year by 31.6139. This means that having more children is associated with a higher bandwidth usage.

The coefficient 'age' is -3.315 signifying that a rise of one unit in 'age is associated with a decrease of bandwidth_GB-year by 3.315. This means that an increase in age is associated with a decrease in bandwidth by 3.31 per year and that older customers use less bandwidth.

The coefficient for 'tenure' is 84.1289 indicating that a one-unit increase in 'tenure' is associated with an increase in bandwidth_GB_year by 84.1289. This suggests that for each additional year of tenure with the internet service provider, the bandwidth usage is expected to increase by 84.13 per year, holding all other predictors constant.

The coefficient for 'dummychurn' (binary variable representing churn) is 261.3743, indicating that if ' dummychurn' increases by one unit, the bandwidth_GB_year_ will also increase by 261.3743. This suggests that customers who churn use 261.37 GB more bandwidth per year than customers do not churn.

The coefficient for DummyGender (binary variable representing gender) is 72.4215 indicating that a one-unit increase in 'dummygender' is associated with an increase in bandwidth_GB_year by 72.4215. This indicates female customers tend to use more bandwidth_GB_year.

c. The statistical and practical significance of the reduced model:

Statistically, the reduced model has a higher statistical significance as indicated by an r-squared value of 0.987, lower residual standard error, and significant coefficients with p-values less than 0.005. These findings suggest that the model is able to explain a large portion of the variance in the dependent variable and the relationship between the independent and dependent variables is statistically significant. The model has a high F-statistic of 1.382e+05 and a low probability (Prob (f-statistic)) indicating statistical significance.

The reduced model is practical in that the interpretation of the coefficients suggests that each independent variable has an impact on the dependent variable. The magnitude of the coeffects indicates the strengths of each independent variable's effect and provides insight into the factors that drive variation in bandwidth usage. This model is practical in that it could have an important implication for internet service providers and their customers. The independent variables (children, age, tenure, churn, and gender) can be used to explain variations in the

amount of bandwidth used by customers. For example, the coefficient for 'Tenure' suggests that for each additional year of tenure, bandwidth usage increased by 84.13. GB per year. Similarly, the coefficient for 'children' suggest customers with more children use more bandwidth which has implications for marketing and product development among families.

d. The limitations of the data analysis:

There are several limitations to the reduced logistic model. One limitation of this model is the small data size of the churn data set. A larger data set could provide more data and a better-fit model. The limited number of independent variables used in the analysis may not fully capture all the favors that influence customer churn. This analysis only indicates correlations and not causation between the dependent and independent variables. It is possible that other factors not included in the analysis may be responsible for churn rates. More investigation is required.

F2. Recommendations

It is recommended that model be expanded with more predictor variables. For example, if the service provider offers different types of internet plans with varying speeds, the speed of the customer's plan could be an important predictor of bandwidth usage. Additional data collection could be useful. The current analysis does not include context regarding factors such as the time period in which the data was collected, demographic information of customers, or the geographic location of the customers. Another recommendation would be to investigate outliers further prior to removal. A cook's distance of greater than 0.5 or 1 may indicate a significant impact on the model's results. Some values obtained from completing cook's distance on the linear regression suggest that some observations have a large impact on the regression model than others and may need to be investigated further to determine whether they are outliers or have some other issue that needing to be addressed. Thus, further investigation of the data is recommended.

Part VI: Demonstration

G. Panopto Demonstration

Link to video: https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=587707a0-b0c8-43b9-92ae-afcf00e01b16

H. Sources of Third-Party Code

GeeksforGeeks. (2023, January 11). *ML: Label encoding of datasets in Python*. GeeksforGeeks. Retrieved March 15, 2023, from https://www.geeksforgeeks.org/ml-label-encoding-of-datasets-in-python/

Holtz, Y. (n.d.). Python Graph Gallery | The Python Graph Gallery. The Python Graph Gallery. https://www.python-graph-gallery.com/

Roualdes, R. D. E. (n.d.). 2.4 Bivariate Visualizations | Applied Statistics. https://norcalbiostat.github.io/AppliedStatistics_notes/bivariate-visualizations.html

Zach. (2020, October 29). *A complete guide to linear regression in python*. Statology. Retrieved March 15, 2023, from https://www.statology.org/linear-regression-python/

I. Sources

Bruce, P., Bruce, A., & Gedeck, P. (2020a). Practical Statistics for Data Scientists: 50+ Essential Concepts Using R and Python (2nd ed.). O'Reilly Media.

Daniel T. Larose, & Chantal D. Larose. (2019). Data Science Using Python and R. Wiley.

LabXchange.(n.d.). https://www.labxchange.org/library/items/lb:LabXchange:10d3270e:html:1

Massaron, L. (2016). *Regression analysis with python: Learn the art of regression analysis with python*. Packt Publishing.

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