PERFORMANCE ASSESSMENT

KAILI HAMILTON

Masters of Science in Data Analytics, Western Governors University

Course: D208 Predictive Modeling

Instructor: Dr. Daniel Smith

Program Mentor: Krissy Bryant

October, 2023

TABLE OF CONTENTS

A1

My research question is, “What factors predict how much data per year a customer uses on average?”

A2

The goals of the data analysis is to determine which factors influence how much data a customer uses per year on average and how strong the relationship is among the each factor and the response variable, “Bandwidth_GB_Year”. This will help the company optimize the services they offer to customers based on these factors and can thus save money. Also, the company may want to offer a data plan for their services with different thresholds of data.

B1

Assumptions of multiple linear regression include (Middleton, nd) (Nair, 2021):

1. Linearity between the explanatory and response variables,
2. Residuals of the regression are normally distributed,
3. Little to no multicollinearity exists in the data,
4. Homoscedasticity
5. Independence of observations

B2

Two benefits of using Python in support of various phases of the analysis include, but are not limited to, computing power to calculate and fit a regression model using vast amounts of data and creating myriad data visualizations to inform the analysis.

B3

Multiple linear regression is an appropriate technique to answer the research question (summarized in parts A1 and A2) because I am striving to find relationships among variables. Specifically, I want to predict which factors influence my continuous response variable “Bandwidth_GB_Year” (Van den Broeck, nd).

C1

My data cleaning goals are designed to help me answer my research question (Part A1). My data cleaning goals are to treat null values, remove extreme outliers, re-express categorical variables for use in multiple linear regression, and to determine which variables to use in my initial regression model (Middleton, nd).

Treated null values.

There were no null values in the data.

Check for null values

```
df.isna().sum()
```

```
Population             0
Age                    0
Income                 0
Churn                  0
Outage_sec_perweek     0
Yearly_equip_failure   0
Techie                 0
Port_modem             0
Tablet                 0
InternetService        0
Phone                  0
Multiple               0
OnlineSecurity         0
OnlineBackup           0
StreamingTV            0
StreamingMovies        0
Tenure                 0
MonthlyCharge          0
Bandwidth_GB_Year      0
dtype: int64
```

no null values

Removed outliers.

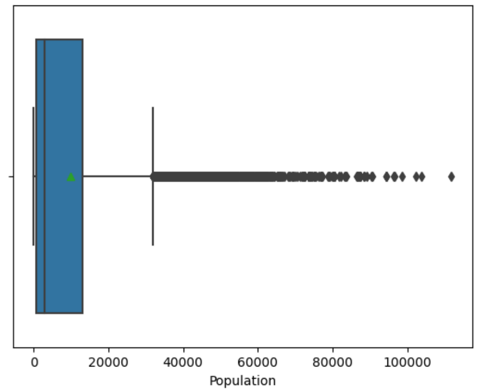
Regression is sensitive to outliers. However, due to the natural variation of the distribution of some variables, I decided to keep most outliers.

The variables “Population”, “Income”, “Outage_sec_perweek”, and “Yearly_equip_failure” all contain outliers. I created boxplots for all numerical variables, calculated the number of outliers for each, and graphed a scatterplot against the response variable, “Bandwidth_GB_Year” to inform whether to drop outliers. The regression plots helped to determine if the outliers were influencing the overall trend of the data. Also, I did not want to change the shape of the distribution so I opted to only drop the most extreme outliers. See the copy of the code for further details regarding code.

POPULATION

```
sns.boxplot(x='Population', data=df, showmeans=True)
```

```
<Axes: xlabel='Population'>
```




```
def find_outliers_IQR(df):

   q1=df.quantile(0.25)

   q3=df.quantile(0.75)

   IQR=q3-q1

   outliers = df[((df<(q1-1.5*IQR)) | (df>(q3+1.5*IQR)))]

   return outliers
```

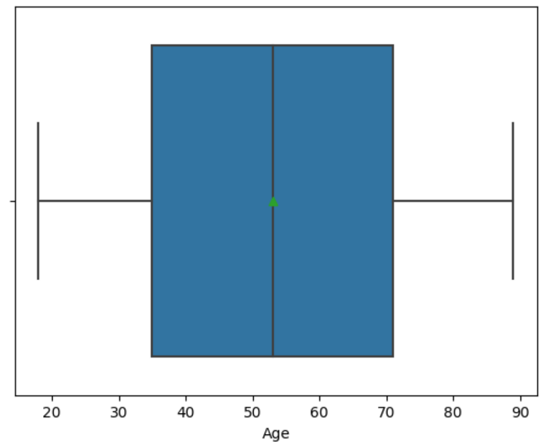
```
outliers = find_outliers_IQR(df['Population'])
print("number of outliers: " + str(len(outliers)))
print('max outlier value: ' + str(outliers.max()))
print('min outlier value: ' + str(outliers.min()))
```

```
number of outliers: 937
max outlier value: 111850
min outlier value: 31816
```

AGE

```
sns.boxplot(x='Age', data=df, showmeans=True)
```

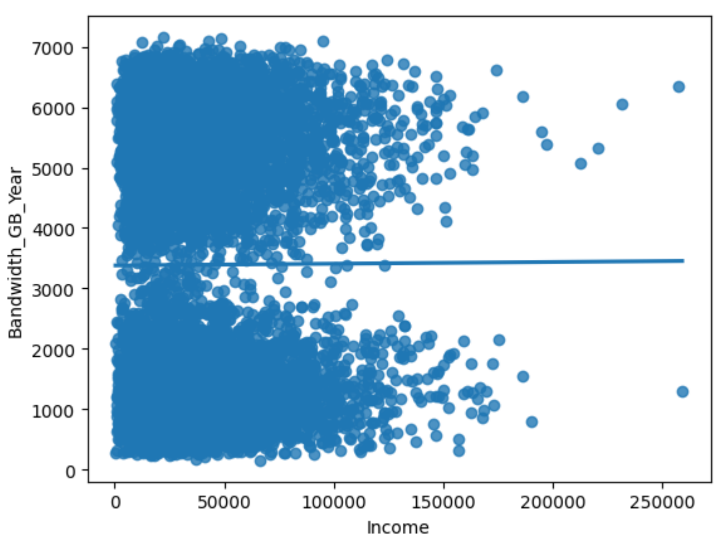
```
<Axes: xlabel='Age'>
```



no outliers in age

```
sns.regplot(x='Income', y='Bandwidth_GB_Year', data=df, ci=None)
```

```
<Axes: xlabel='Income', ylabel='Bandwidth_GB_Year'>
```



Outliers that were dropped:

- Population > 100,000

```
df.drop(df[df['Population'] > 100000].index, inplace=True)
```

```
df.shape
```

```
(9991, 19)
```

```
df['Population'].describe()
```

```
count     9991.000000
mean      9730.486037
std      14341.542493
min          0.000000
25%        737.500000
50%       2905.000000
75%      13161.000000
max      98660.000000
Name: Population, dtype: float64
```

- Income > 200,000

```
df['Income'].describe()
```

```
count      9994.000000
mean      39710.411833
std       27861.200412
min         348.670000
25%       19219.835000
50%       33156.205000
75%       53226.895000
max      196746.000000
Name: Income, dtype: float64
```

```
df.drop(df[df['Population'] > 100000].index, inplace=True)
```

```
df.shape
```

```
(9991, 19)
```

- Yearly_equip_failure = 6

```
df.drop(df[df['Yearly_equip_failure'] == 6].index, inplace=True)
```

```
df.shape
```

```
(9999, 19)
```

```
df['Yearly_equip_failure'].describe()
```

```
count    9999.000000
mean        0.397440
std         0.633512
min         0.000000
25%         0.000000
50%         0.000000
75%         1.000000
max         4.000000
Name: Yearly_equip_failure, dtype: float64
```

Re-expressed categorical data types.

First I re-expressed <u>ordinal</u> categorical data types using ordinal encoding. These included variables with a “yes” or “no” value. All “yes” values were replaced with “1” and all “no” values were replaced with “0”. Then I made sure the datatype changed from object to int64.

```
df['Churn'].replace({'No' : 0, 'Yes' : 1}, inplace=True)
```

```
df['Churn'].value_counts()
```

```
0    7341
1    2650
Name: Churn, dtype: int64
```

```
df['Techie'].replace({'No' : 0, 'Yes' : 1}, inplace=True)
df['Port_modem'].replace({'No' : 0, 'Yes' : 1}, inplace=True)
```

```
df['Tablet'].replace({'No' : 0, 'Yes' : 1}, inplace=True)
df['Phone'].replace({'No' : 0, 'Yes' : 1}, inplace=True)
```

```
df['Multiple'].replace({'No' : 0, 'Yes' : 1}, inplace=True)
df['OnlineSecurity'].replace({'No' : 0, 'Yes' : 1}, inplace=True)
df['OnlineBackup'].replace({'No' : 0, 'Yes' : 1}, inplace=True)
df['StreamingTV'].replace({'No' : 0, 'Yes' : 1}, inplace=True)
df['StreamingMovies'].replace({'No' : 0, 'Yes' : 1}, inplace=True)
```

```
df['StreamingMovies'].value_counts()
```

```
0    5104
1    4887
Name: StreamingMovies, dtype: int64
```

```
df.dtypes
```

```
Population               int64
Age                      int64
Income                 float64
Churn                    int64
Outage_sec_perweek     float64
Yearly_equip_failure     int64
Techie                   int64
Port_modem               int64
Tablet                   int64
InternetService         object
Phone                    int64
Multiple                 int64
OnlineSecurity           int64
OnlineBackup             int64
StreamingTV              int64
StreamingMovies          int64
Tenure                 float64
MonthlyCharge          float64
Bandwidth_GB_Year      float64
dtype: object
```

Next, I re-expressed <u>nominal</u> categorical data types for the variable “InternetService” using pd.getdummies(). “InternetService” has three values, with no inherent rank.

```
df['InternetService'].unique()
```

```
array(['Fiber Optic', 'DSL', 'None'], dtype=object)
```

```
df = pd.get_dummies(df, columns=['InternetService'])
```

```
df.head()
```

Port_modem	Tablet	Phone	...	OnlineSecurity	OnlineBackup	StreamingTV	StreamingMovies	Tenure	MonthlyCharge	Bandwidth_GB_Year	InternetService_DSL	InternetService_Fiber Optic	InternetService_None
1	1	1	...	1	1	0	1	6.795513	172.455519	904.536110	0	1	0
0	1	1	...	1	0	1	1	1.156681	242.632554	800.982766	0	1	0
1	0	1	...	0	0	0	1	15.754144	159.947583	2054.706961	1	0	0
0	0	1	...	1	0	1	0	17.087227	119.956840	2164.579412	1	0	0
1	0	0	...	0	0	1	0	1.670972	149.948316	271.493436	0	1	0

Determined which variables to use in my initial model.

I initially kept all variables that I thought would influence the response variable “Bandwidth_GB_Year”. “Job” was one of the variables I thought to include, but there 639 unique values, so I excluded it from the analysis. I also dropped “Martial” for the same reason, too many unique values. I also dropped “Children” so the focus is just the customer, not the household. This is also discussed in Part C4.

	Job
count	10000
unique	639
top	Occupational psychologist
freq	30

```
# drop "Job" variable - too many unique observations
df.drop(columns='Job', inplace=True)
```

```
df['Marital'].value_counts()
```

```
Divorced         2092
Widowed          2027
Separated        2014
Never Married    1956
Married          1911
Name: Marital, dtype: int64
```

```
# drop "Marital" variable - too many unique observations
df.drop(columns='Marital', inplace=True)
df.head()
```

```
df.drop(columns='Children', inplace=True)
df.head()
```

The explanatory variables used in my initial regression are:

- Population
- Age
- Income
- Churn
- Outage_sec_perweek
- Yearly_equip_failure
- Techie
- Port_modem
- Tablet
- InternetService
- Phone
- Multiple
- OnlineSecurity
- OnlineBackup
- StreamingTV
- StreamingMovies
- Tenure
- MonthlyCharge

C2

The dependent (aka response) variable for the data analysis is “Bandwidth_GB_Year”. From the summary statistics displayed below we see that there are 9991 values with a range of approximately 7003 GB. On average, customers use about 3391 GB per year with a median value of 3261 GB. The middle 50% of customers use between 1236 and 5585 GB per year.

```
df['Bandwidth_GB_Year'].describe()
```

```
count    9991.000000
mean     3390.795380
std      2185.252241
min       155.506715
25%      1236.046551
50%      3260.745232
75%      5584.704954
max      7158.981530
Name: Bandwidth_GB_Year, dtype: float64
```

The independent (aka explanatory) variables in this analysis are listed below along with their summary statistics. There are 9,991 entries for each variable.

- Population. The distribution of Population is highly skewed right as evidenced by the mean population of 9730 being much larger than the median population, 2905. The middle 50% of customers live in a one mile radius population size between 737 and 13,161 people.

```
df['Population'].describe()
```

```
count     9991.000000
mean      9730.486037
std      14341.542493
min          0.000000
25%        737.500000
50%       2905.000000
75%      13161.000000
max      98660.000000
Name: Population, dtype: float64
```

- Age. The mean and median age for customers is 53, with the oldest customer 89 years old and the youngest 20 years old.

```
df['Age'].describe()
```

```
count    9991.000000
mean       53.082174
std        20.702085
min        18.000000
25%        35.000000
50%        53.000000
75%        71.000000
max        89.000000
Name: Age, dtype: float64
```

- Income. On average, customers have a yearly income of about $39,716 (median value $33,169). The income of the middle 50% of customers ranges between $19,215 and $53,228.

```
df['Income'].describe()
```

```
count      9991.000000
mean      39715.501730
std       27863.826217
min         348.670000
25%       19214.740000
50%       33168.880000
75%       53227.795000
max      196746.000000
Name: Income, dtype: float64
```

- Churn. Out of 9991 customers in the dataset, 73.5% of customers churned within the last month, leaving 26.5% of customers staying with the company.

```
df['Churn'].value_counts()
```

```
Churn
0    7341
1    2650
Name: count, dtype: int64
```

- Outage_sec_perweek. On average, customers' neighborhood's experience 10 seconds of outage time per week.

```
df['Outage_sec_perweek'].describe()
```

```
count    9991.000000
mean       10.002278
std         2.976494
min         0.099747
25%         8.019310
50%        10.019720
75%        11.971418
max        21.207230
Name: Outage_sec_perweek, dtype: float64
```

- Yearly_equip_failure. The average number of times per year a customer's equipment failed and replaced was 0.4, with a median of 0 times. The maximum yearly equipment failure a customer experienced was 4.

```
df['Yearly_equip_failure'].describe()
```

```
count    9991.000000
mean        0.397157
std         0.633253
min         0.000000
25%         0.000000
50%         0.000000
75%         1.000000
max         4.000000
```

- Techie. Out of 9991 customers, only 16.8% of customers described themselves as technically inclined.

```
df['Techie'].value_counts()
```

```
Techie
0    8314
1    1677
Name: count, dtype: int64
```

- Port_modem. Out of 9991 customers, 51.6% has a portable modem.

```
df['Port_modem'].value_counts()
```

```
Port_modem
0    5158
1    4833
Name: count, dtype: int64
```

- Tablet. Out of 9991 customers, only 29.9% reported owning a tablet.

```
df['Tablet'].value_counts()
```

```
Tablet
0    7006
1    2985
Name: count, dtype: int64
```

- InternetService. Customer's internet service provider could be fiber optics, DSL, or none. 2128 out of 9991 customers (21.3%) have neither fiber optics nor DSL. More customers have fiber optics than DSL (44.1% vs. 34.6%).

```
df['InternetService_Fiber Optic'].value_counts()
```

```
InternetService_Fiber Optic
0    5587
1    4404
Name: count, dtype: int64
```

```
df['InternetService_DSL'].value_counts()
```

```
InternetService_DSL
0    6532
1    3459
Name: count, dtype: int64
```

- Phone. 90.7% of customers own a phone.

```
df['Phone'].value_counts()
```

```
Phone
1    9058
0     933
Name: count, dtype: int64
```

- Multiple. Less than half of customers have multiple phone lines, 46.1%.

```
df['Multiple'].value_counts()
```

```
Multiple
0    5387
1    4604
Name: count, dtype: int64
```

- OnlineSecurity. Just over a third of customers (35.8%) have an online security addon.

```
df['OnlineSecurity'].value_counts()
```

```
OnlineSecurity
0    6418
1    3573
Name: count, dtype: int64
```

- OnlineBackup. Just under half of all customers, 45.1%, have an addon for online backup.

```
df['OnlineBackup'].value_counts()
```

```
OnlineBackup
0    5489
1    4502
Name: count, dtype: int64
```

- StreamingTV. About half of all customers (49.3%) stream TV.

```
df['StreamingTV'].value_counts()
```

```
StreamingTV
0    5068
1    4923
Name: count, dtype: int64
```

- StreamingMovies. About half of all customers (48.9%) stream movies.

```
df['StreamingMovies'].value_counts()
```

```
StreamingMovies
0    5104
1    4887
```

- Tenure. The average tenure of customers is 34 months. The middle 50% of customers have been with the provider between 8 and 61 months. The most tenured customer has been with the provider for 72 months, while the least tenured is 1 month.

```
df['Tenure'].describe()
```

```
count    9991.000000
mean       34.508248
std        26.442933
min         1.000259
25%         7.916107
50%        33.196120
75%        61.473295
max        71.999280
Name: Tenure, dtype: float64
```

- <u>MonthlyCharge</u>. The average monthly charge per customer is $172.62 (median $167.48). The middle 50% of customers are charged between $139.98 and $200.17 monthly.

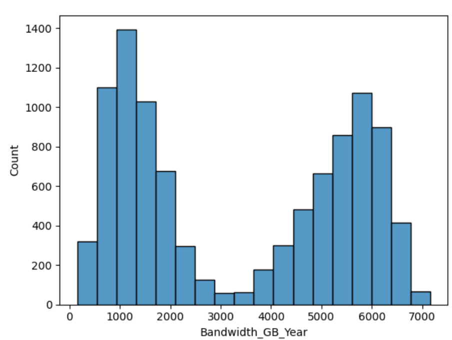
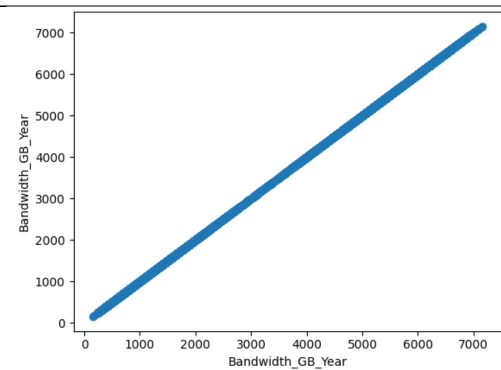
```
df['MonthlyCharge'].describe()
```

```
count    9991.000000
mean      172.618895
std        42.937793
min        79.978860
25%       139.979239
50%       167.484700
75%       200.165200
max       290.160419
Name: MonthlyCharge, dtype: float64
```

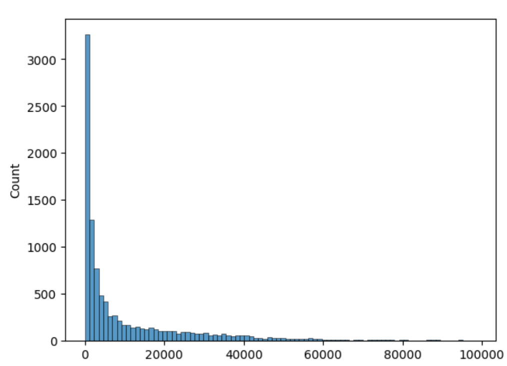
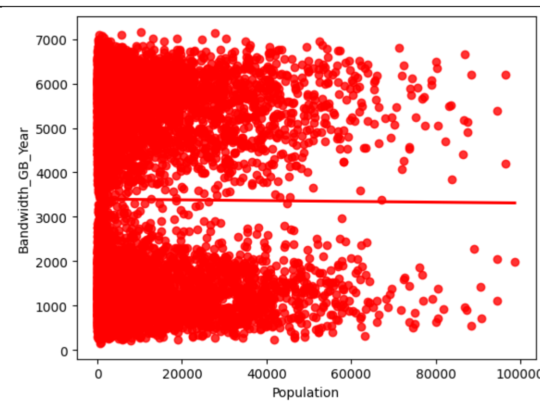
C3

Univariate and bivariate visualizations are for all variables are presented here. Bandwidth_GB_Year is the only dependent variable. Bivariate visualizations are plotted against Bandwidth_GB_Year

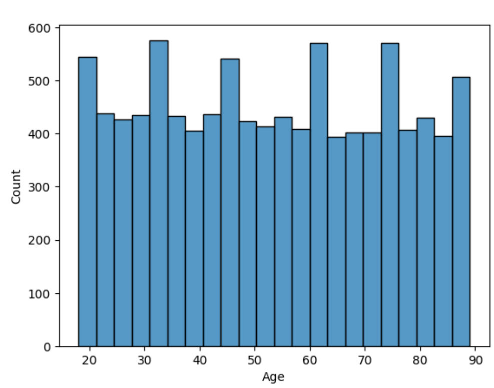
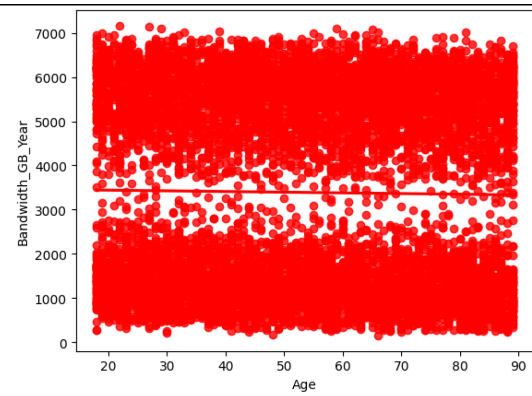
- Bandwidth_GB_Year (dependent variable).

Univariate Visualization	Bivariate Visualization.
	

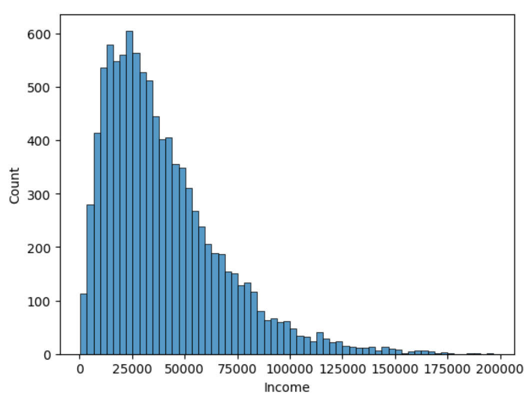
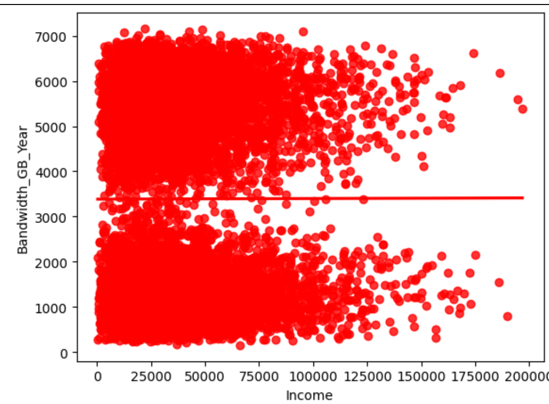
- Population

Univariate Visualization	Bivariate Visualization.
	

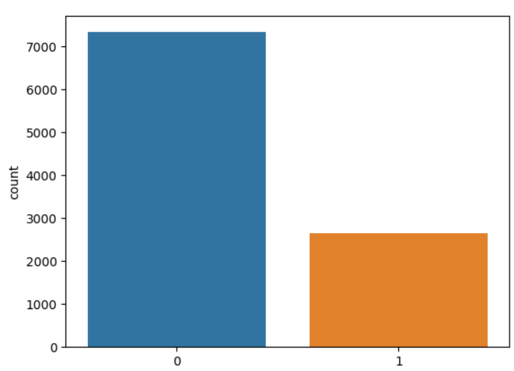
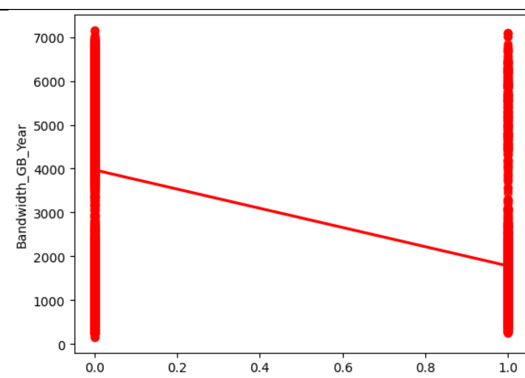
- Age

<table>
<tr><th>Univariate Visualization</th><th>Bivariate Visualization.</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

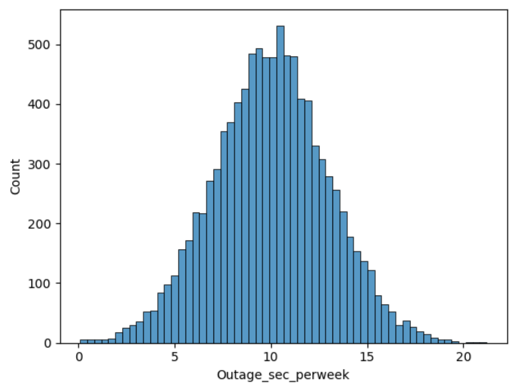
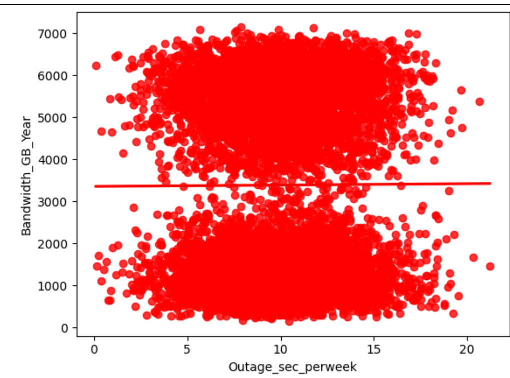
- Income

<table>
<tr><th>Univariate Visualization</th><th>Bivariate Visualization.</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

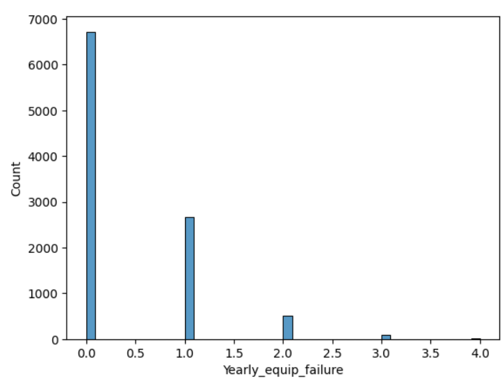
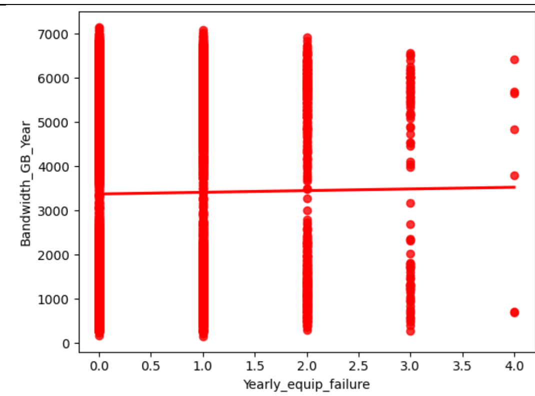
- Churn. 0 = No. 1 = Yes.

<table>
<tr><th>Univariate Visualization</th><th>Bivariate Visualization.</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

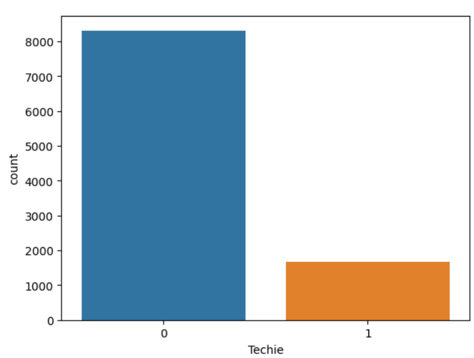
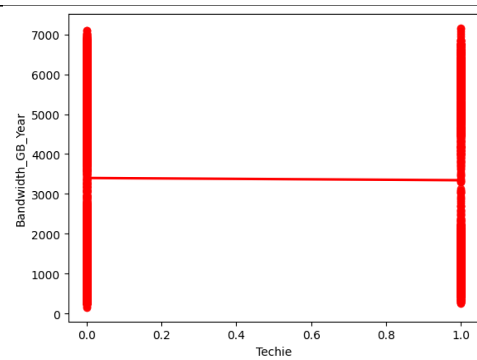
- Outage_sec_perweek

Univariate Visualization	Bivariate Visualization.
	

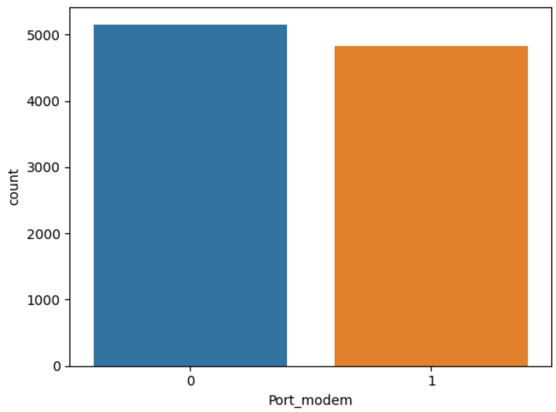
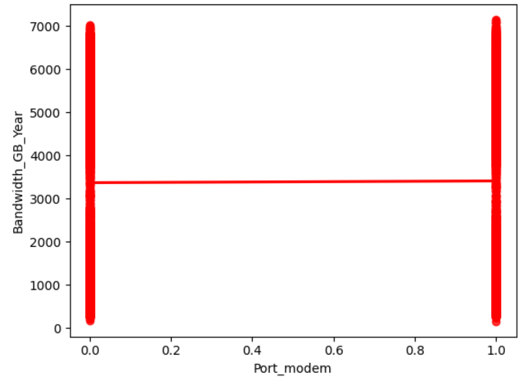
- Yearly_equip_failure

Univariate Visualization	Bivariate Visualization.
	

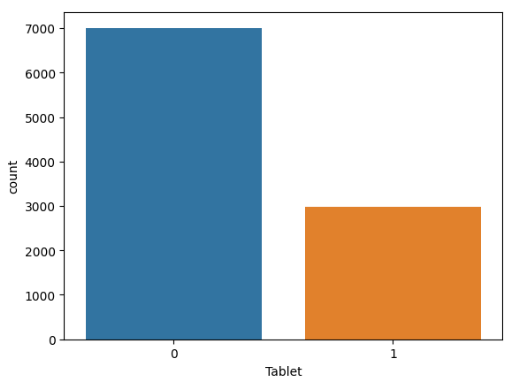
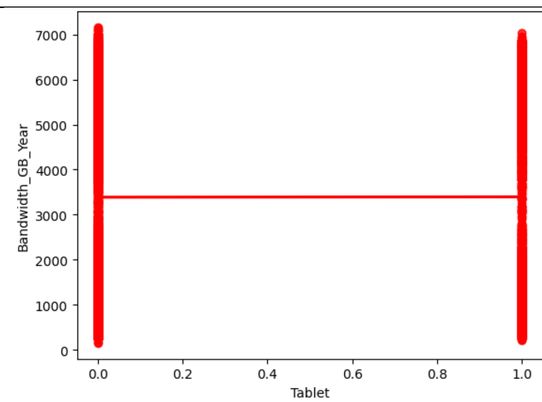
Techie. 0 = No. 1 = Yes.

Univariate Visualization	Bivariate Visualization.
	

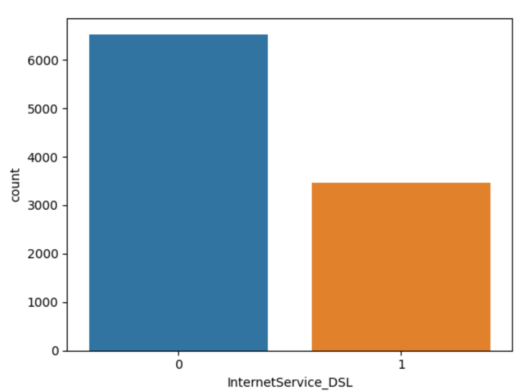
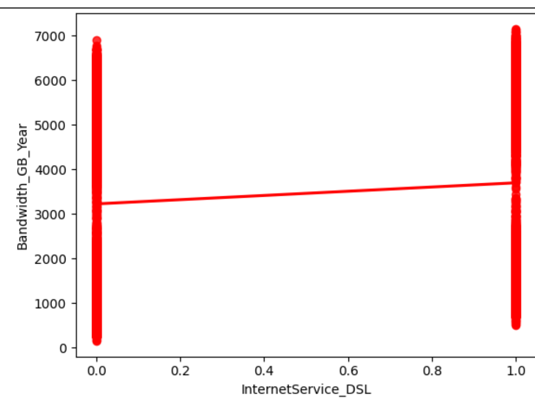
- Port_modem. 0 = No. 1 = Yes.

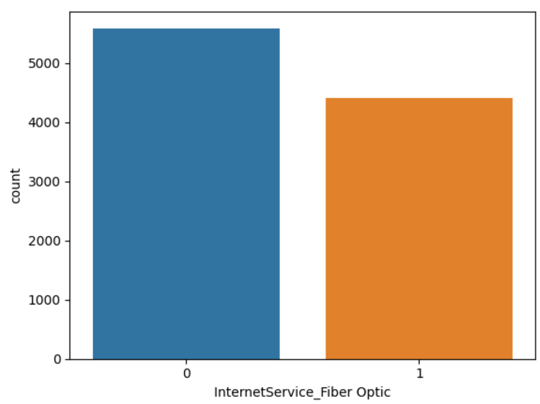
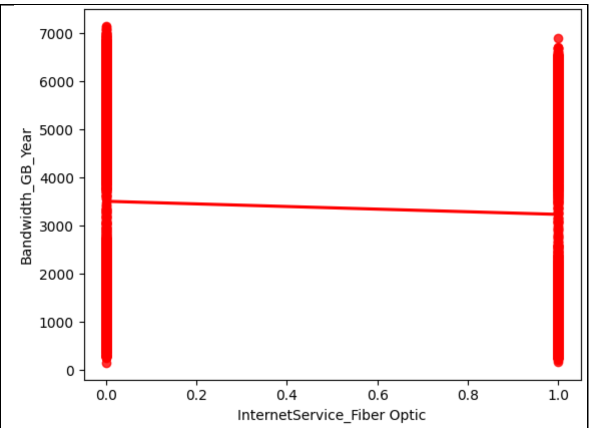
Univariate Visualization	Bivariate Visualization.
	

- Tablet. 0 = No. 1 = Yes.

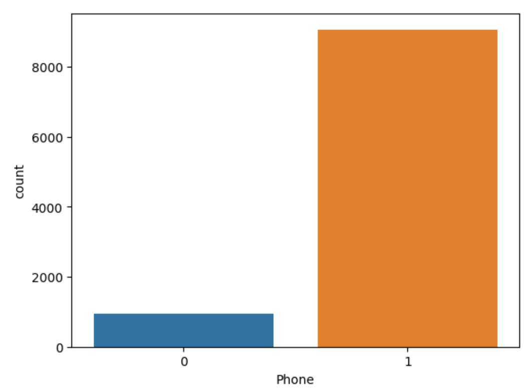
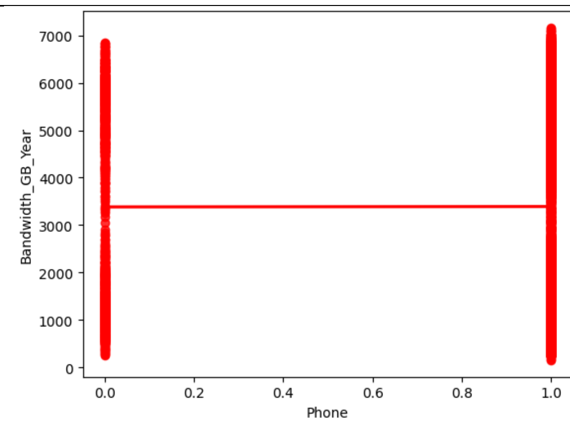
Univariate Visualization	Bivariate Visualization.
	

- InternetService. 0 = No. 1 = Yes.

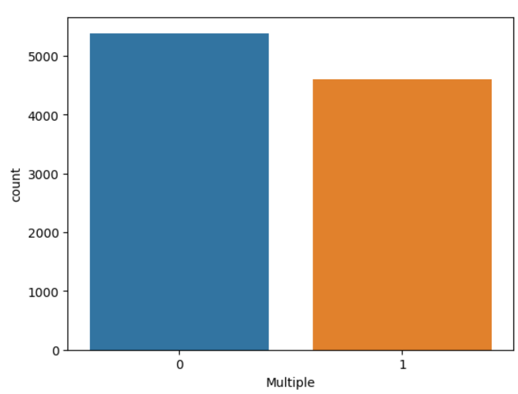
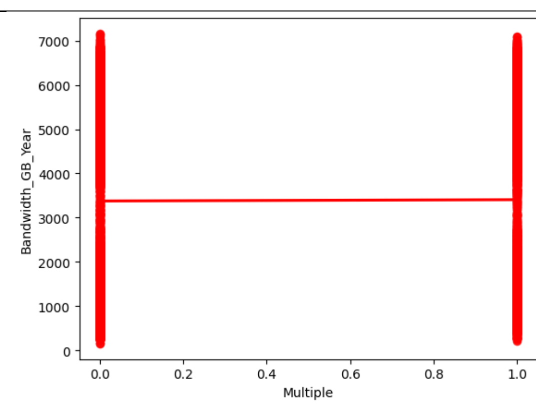
Univariate Visualization	Bivariate Visualization.
	

<table>
<tr>
<td>
</td>
<td>
</td>
</tr>
</table>

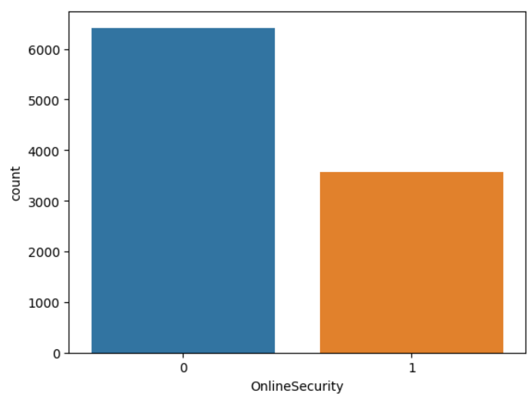
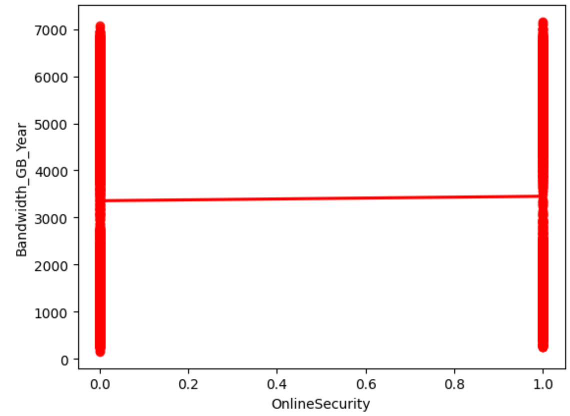
- Phone. 0 = No. 1 = Yes.

<table>
<tr><td>Univariate Visualization</td><td>Bivariate Visualization.</td></tr>
<tr>
<td>
</td>
<td>
</td>
</tr>
</table>

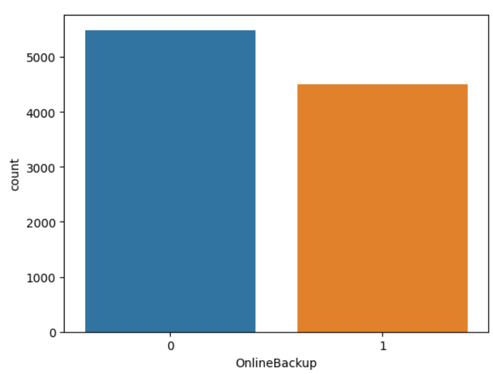
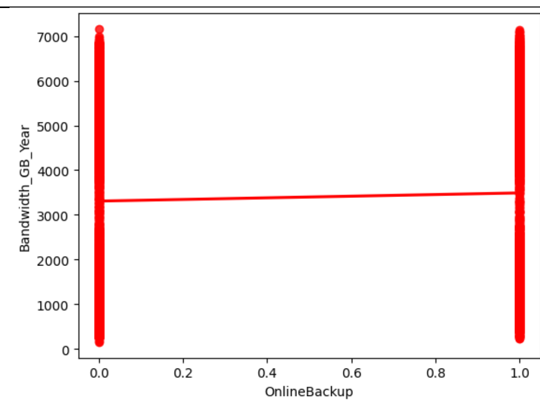
- Multiple. 0 = No. 1 = Yes.

<table>
<tr><td>Univariate Visualization</td><td>Bivariate Visualization.</td></tr>
<tr>
<td>
</td>
<td>
</td>
</tr>
</table>

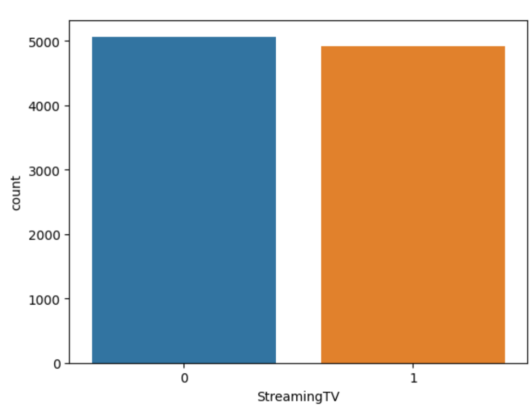
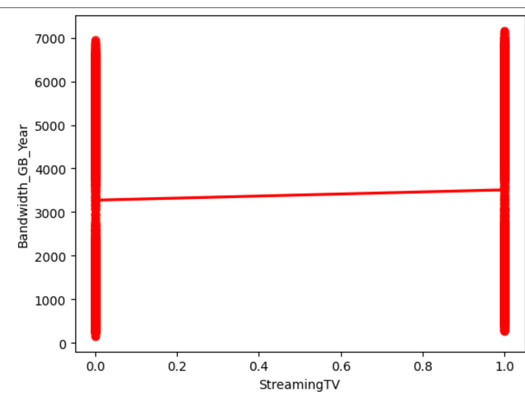
- OnlineSecurity. 0 = No. 1 = Yes.

Univariate Visualization	Bivariate Visualization.
	

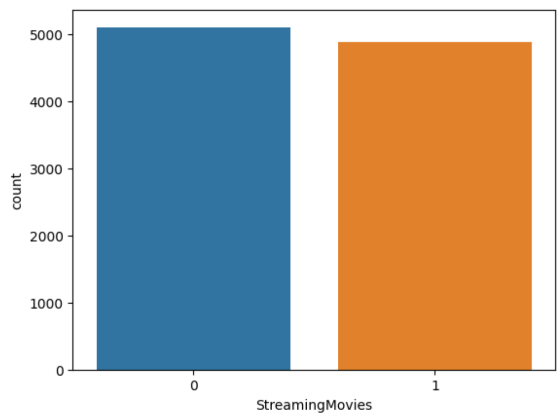
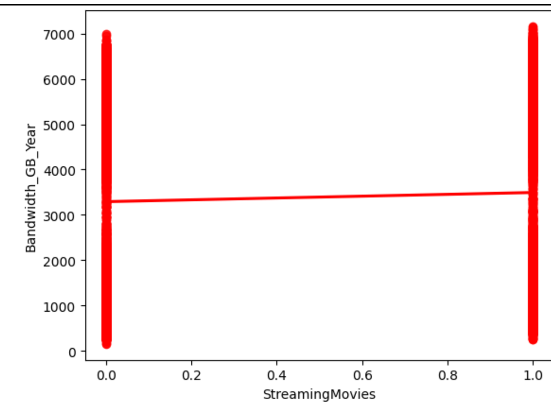
- OnlineBackup. 0 = No. 1 = Yes.

Univariate Visualization	Bivariate Visualization.
	

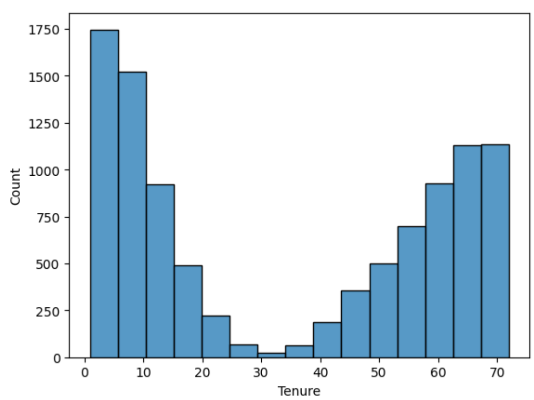
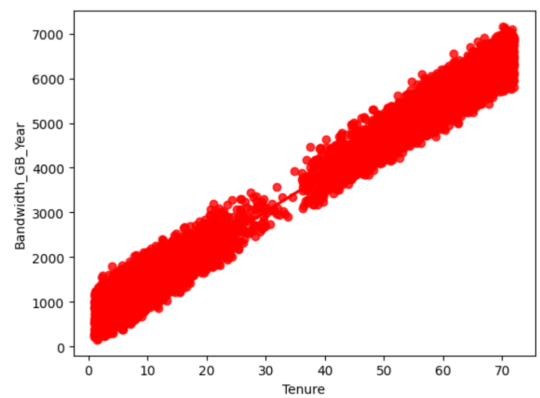
- StreamingTV. 0 = No. 1 = Yes.

Univariate Visualization	Bivariate Visualization.
	

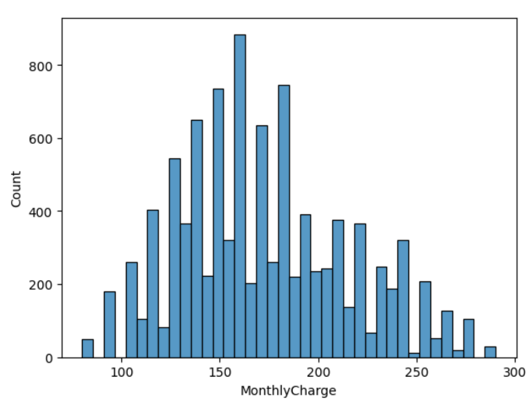
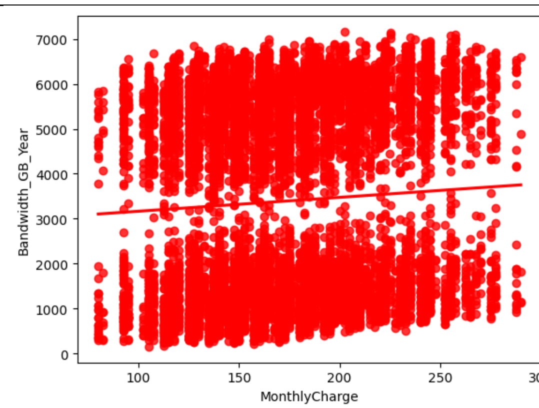
- StreamingMovies. 0 = No. 1 = Yes.

Univariate Visualization	Bivariate Visualization.
	

- Tenure

Univariate Visualization	Bivariate Visualization.
	

- MonthlyCharge

Univariate Visualization	Bivariate Visualization.
	

C4

My data transformation goals included determining which variables to keep. I initially kept all variables that I thought would influence the response variable “Bandwidth_GB_Year”. “Job” was one of the variables I thought to include, but there 639 unique values, so I excluded it from the analysis. I also dropped “Martial” for the same reason, too many unique values. I also dropped “Children” so the focus is just the customer, not the household.

	Job
count	10000
unique	639
top	Occupational psychologist
freq	30

```
# drop "Job" variable - too many unique observations
df.drop(columns='Job', inplace=True)
```

```
df['Marital'].value_counts()
```

```
Divorced         2092
Widowed          2027
Separated        2014
Never Married    1956
Married          1911
Name: Marital, dtype: int64
```

```
# drop "Marital" variable - too many unique observations
df.drop(columns='Marital', inplace=True)
df.head()
```

```
df.drop(columns='Children', inplace=True)
df.head()
```

The explanatory variables used in my initial regression are:

- Population
- Age
- Income
- Churn
- Outage_sec_perweek
- Yearly_equip_failure

- Techie
- Port_modem
- Tablet
- InternetService
- Phone
- Multiple
- OnlineSecurity
- OnlineBackup
- StreamingTV
- StreamingMovies
- Tenure
- MonthlyCharge

Other data transformation goals are to re-express categorical variables for use in the regression models. First I re-expressed ordinal categorical data types using ordinal encoding. These included variables with a "yes" or "no" value. All "yes" values were replaced with "1" and all "no" values were replaced with "0". Then I made sure the datatype changed from object to int64.

```
df['Churn'].replace({'No' : 0, 'Yes' : 1}, inplace=True)
```

```
df['Churn'].value_counts()
```

```
0    7341
1    2650
Name: Churn, dtype: int64
```

```
df['Techie'].replace({'No' : 0, 'Yes' : 1}, inplace=True)
df['Port_modem'].replace({'No' : 0, 'Yes' : 1}, inplace=True)
```

```
df['Tablet'].replace({'No' : 0, 'Yes' : 1}, inplace=True)
df['Phone'].replace({'No' : 0, 'Yes' : 1}, inplace=True)
```

```
df['Multiple'].replace({'No' : 0, 'Yes' : 1}, inplace=True)
df['OnlineSecurity'].replace({'No' : 0, 'Yes' : 1}, inplace=True)
df['OnlineBackup'].replace({'No' : 0, 'Yes' : 1}, inplace=True)
df['StreamingTV'].replace({'No' : 0, 'Yes' : 1}, inplace=True)
df['StreamingMovies'].replace({'No' : 0, 'Yes' : 1}, inplace=True)
```

```
df['StreamingMovies'].value_counts()
```

```
0    5104
1    4887
Name: StreamingMovies, dtype: int64
```

```
df.dtypes
```

```
Population              int64
Age                     int64
Income                float64
Churn                   int64
Outage_sec_perweek    float64
Yearly_equip_failure    int64
Techie                  int64
Port_modem              int64
Tablet                  int64
InternetService        object
Phone                   int64
Multiple                int64
OnlineSecurity          int64
OnlineBackup            int64
StreamingTV             int64
StreamingMovies         int64
Tenure                float64
MonthlyCharge         float64
Bandwidth_GB_Year     float64
dtype: object
```

Next, I re-expressed nominal categorical data types for the variable "InternetService" using pd.getdummies(). "InternetService" has three values, with no inherent rank.

```
df['InternetService'].unique()
```

```
array(['Fiber Optic', 'DSL', 'None'], dtype=object)
```

```
df = pd.get_dummies(df, columns=['InternetService'])
```

```
df.head()
```

Port_modem	Tablet	Phone	...	OnlineSecurity	OnlineBackup	StreamingTV	StreamingMovies	Tenure	MonthlyCharge	Bandwidth_GB_Year	InternetService_DSL	InternetService_Fiber Optic	InternetService_None
1	1	1	...	1	1	0	1	6.795513	172.455519	904.536110	0	1	0
0	1	1	...	1	0	1	1	1.156681	242.632554	800.982766	0	1	0
1	0	1	...	0	0	0	1	15.754144	159.947583	2054.706961	1	0	0
0	0	1	...	1	0	1	0	17.087227	119.956840	2164.579412	1	0	0
1	0	0	...	0	0	1	0	1.670972	149.948316	271.493436	0	1	0

C5

A copy of the prepared data set is submitted as a CSV file titled 'prepared_dataset_churn_D208.csv'.

D1

The initial multiple linear regression model from all independent variables that were identified in part C2 is expressed below.

$$
\begin{aligned}
\boldsymbol{Bandwidth\_GB\_Year} = \mathbf{13.59250} \\
&+ -0.00003875838 * \ Population \\
&+ \ -3.359748 \ * \ Age \\
&+ 0.0000002482354 \ * \ Income \\
&+ 1.678670 \ * \ Churn \\
&+ 0.375506 \ * \ Outage\_sec\_perweek \\
&+ \ -0.4325477 * \ Yearly\_equip\_failure \\
&+ \ -1.330278 * \ Techie \\
&+ 1.828037 \ * \ Port\_modem \\
&+ \ -0.4262595 \ * \ Tablet \\
&+ 01.906635 \ * \ Phone \\
&+ \ -0.3745406 \ * \ Multiple \\
&+ 69.28261 * \ OnlineSecurity \\
&+ 20.06130 \ * \ OnlineBackup \\
&+ 89.90295 \ * \ StreamingTV \\
&+ 41.04283 \ * \ StreamingMovies \\
&+ 41.04283 \ * \ Tenure \\
&+ 3.272196 * \ MonthlyCharge \\
&+ 370.03850 \ * \ InternetService\_DSL \\
&+ \ -107.8163 \ * \ InternetService\_FiberOptic
\end{aligned}
$$

D2

I reduced the initial model, represented in part D1, to better align with the research question by engaging in the following tasks.

First, I checked for multicollinearity among the variables. This is not a feature selection procedure, but instead is a method for reducing multicollinearity, an assumption of multiple linear regression. One variable, “MonthlyCharge”, had a VIF greater than 10. This variable was removed due to a high VIF factor.

Next, I ran the model without “MonthlyCharge”. On this step, I engaged in a statistically based feature selection procedure by selecting variables using an iterative backwards elimination method until all variables are statistically significant with a $p - value \ < \ 0.05$. I removed the variable with the highest p-value first,”Yearly_equip_failure”, with $p = 0.9820$. Then I ran the model again. “Techie” now had the highest p-value of $p = 0.767$ so it was removed. After running the model again, “Port_modem” had the largest p-value of $p = 0.537$.

Next, all variables in the model are statistically significant with $p < 0.05$. The reduced model is provided in part D3.

D3

As a result of the feature selection process described in part D2, the reduced linear regression model is as follows:

$$
\begin{aligned}
\boldsymbol{Bandwidth\_GB\_Year} = \ & 278.3015 \\
& + -0.00005178 * Population \\
& + \ -3.3205 \ * \ Age \\
& 0.00002559 * \mathrm{Income} \\
& + 15.9373 * \ Churn \\
& + 0.4132 * \ Outage\_sec\_perweek \\
& + \ -1.6094 \ * \ Tablet \\
& + \ -5.6398 \ * \ Phone \\
& + 67.3540 * \ Multiple \\
& + 79.2940 \ * \ OnlineSecurity \\
& + 93.3939 \ * \ OnlineBackup \\
& + 225.0095 \ * \ StreamingTV \\
& + 208.8257 \ * \ StreamingMovies \\
& + 81.9902 \ * \ Tenure \\
& + 411.1733 \ * \ InternetService\_DSL \\
& + \ -1.6165 * \ InternetService\_FiberOptic
\end{aligned}
$$

- OLS Results for the Initial Linear Regression Model:

OLS Regression Results

Dep. Variable:	Bandwidth_GB_Year	R-squared:	0.999
Model:	OLS	Adj. R-squared:	0.999
Method:	Least Squares	F-statistic:	3.906e+05
Date:	Sun, 19 Nov 2023	Prob (F-statistic):	0.00
Time:	17:12:39	Log-Likelihood:	-57962.
No. Observations:	9991	AIC:	1.160e+05
Df Residuals:	9971	BIC:	1.161e+05
Df Model:	19		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	13.5925	7.451	1.824	0.068	-1.012	28.197
MonthlyCharge	3.2722	0.066	49.955	0.000	3.144	3.401
Population	-3.876e-05	5.59e-05	-0.693	0.488	-0.000	7.09e-05
Age	-3.3597	0.039	-86.693	0.000	-3.436	-3.284
Income	2.482e-07	2.88e-05	0.009	0.993	-5.62e-05	5.67e-05
Churn	1.6787	2.377	0.706	0.480	-2.980	6.337
Outage_sec_perweek	0.3755	0.269	1.394	0.163	-0.153	0.904
Yearly_equip_failure	0.4325	1.267	0.342	0.733	-2.050	2.915
Techie	-1.3303	2.153	-0.618	0.537	-5.551	2.891
Port_modem	1.8280	1.605	1.139	0.255	-1.318	4.974
Tablet	-0.4263	1.753	-0.243	0.808	-3.863	3.011
Phone	-1.9066	2.759	-0.691	0.489	-7.314	3.501
Multiple	-37.4541	2.657	-14.097	0.000	-42.662	-32.246
OnlineSecurity	69.2826	1.686	41.095	0.000	65.978	72.587
OnlineBackup	20.0613	2.184	9.186	0.000	15.780	24.342
StreamingTV	89.9030	3.182	28.250	0.000	83.665	96.141
StreamingMovies	41.0428	3.771	10.882	0.000	33.650	48.436
Tenure	81.9007	0.036	2280.237	0.000	81.830	81.971
InternetService_DSL	370.3950	2.365	156.628	0.000	365.759	375.030
InternetService_FiberOptic	-107.8163	3.000	-35.942	0.000	-113.696	-101.936

Omnibus:	1037.448	Durbin-Watson:	2.005
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1455.422
Skew:	0.820	Prob(JB):	0.00
Kurtosis:	3.897	Cond. No.	5.27e+05

- OLS Results Reduced Linear Regression Model:

OLS Regression Results

Dep. Variable:	Bandwidth_GB_Year	**R-squared:**	0.998
Model:	OLS	**Adj. R-squared:**	0.998
Method:	Least Squares	**F-statistic:**	3.957e+05
Date:	Sun, 19 Nov 2023	**Prob (F-statistic):**	0.00
Time:	17:18:42	**Log-Likelihood:**	-59079.
No. Observations:	9991	**AIC:**	1.182e+05
Df Residuals:	9975	**BIC:**	1.183e+05
Df Model:	15		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	278.3015	5.771	48.226	0.000	266.990	289.613
Population	-5.178e-05	6.25e-05	-0.828	0.407	-0.000	7.08e-05
Age	-3.3205	0.043	-76.661	0.000	-3.405	-3.236
Income	2.559e-05	3.22e-05	0.795	0.426	-3.75e-05	8.87e-05
Churn	15.9373	2.629	6.062	0.000	10.784	21.091
Outage_sec_perweek	0.4132	0.301	1.372	0.170	-0.177	1.004
Tablet	-1.6094	1.960	-0.821	0.412	-5.451	2.232
Phone	-5.6398	3.083	-1.829	0.067	-11.683	0.403
Multiple	67.3540	1.822	36.971	0.000	63.783	70.925
OnlineSecurity	79.2940	1.871	42.380	0.000	75.626	82.962
OnlineBackup	93.3939	1.808	51.667	0.000	89.851	96.937
StreamingTV	225.0095	1.874	120.068	0.000	221.336	228.683
StreamingMovies	208.8257	1.916	108.979	0.000	205.070	212.582
Tenure	81.9902	0.040	2045.848	0.000	81.912	82.069
InternetService_DSL	411.1733	2.481	165.717	0.000	406.310	416.037
InternetService_FiberOptic	-1.6165	2.366	-0.683	0.495	-6.254	3.022

Omnibus:	606.095	**Durbin-Watson:**	1.982
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	729.668
Skew:	0.619	**Prob(JB):**	3.59e-159
Kurtosis:	3.468	**Cond. No.**	3.36e+05

E1

In my data analysis process I developed a multiple linear regression model to determine what factors influence how much bandwidth customers use. I will use the adjusted R^2 value as a model evaluation metric to make a comparison of the initial and reduced multiple linear regression model. For the initial model adjusted $R^2 = 0.999$ and for the reduced model adjusted $R^2 = 0.998$. Both models explain over 99% of the variation, while taking into account the number of predictor variables (Van den Broeck, nd).

Initial Model Calculations

R-squared:	0.999
Adj. R-squared:	0.999
F-statistic:	3.906e+05
Prob (F-statistic):	0.00
Log-Likelihood:	-57962.
AIC:	1.160e+05
BIC:	1.161e+05

Reduced Model Calculations

R-squared:	0.998
Adj. R-squared:	0.998
F-statistic:	3.957e+05
Prob (F-statistic):	0.00
Log-Likelihood:	-59079.
AIC:	1.182e+05
BIC:	1.183e+05

E2

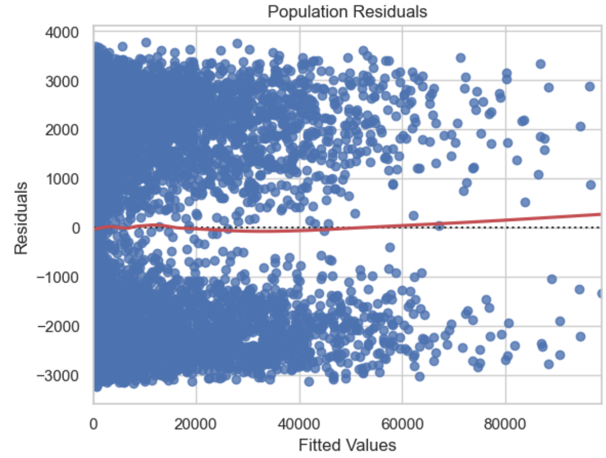
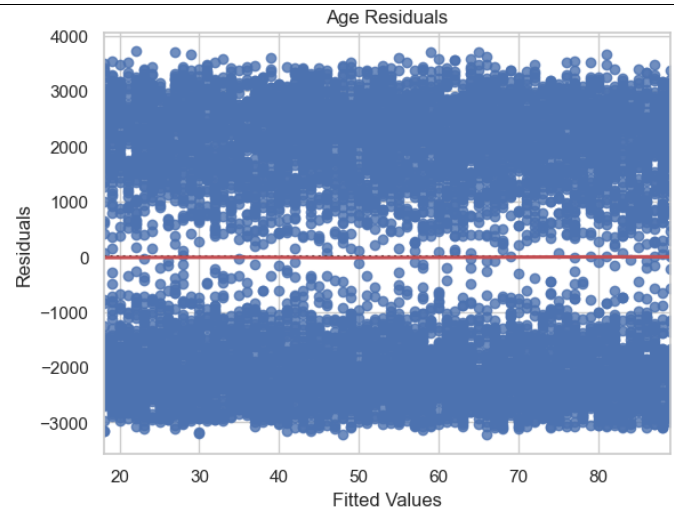
The output and all calculations of the analysis I performed for the reduced model are outlined below as well as a residual plot for each independent variable vs. Bandwidth_GB_Year

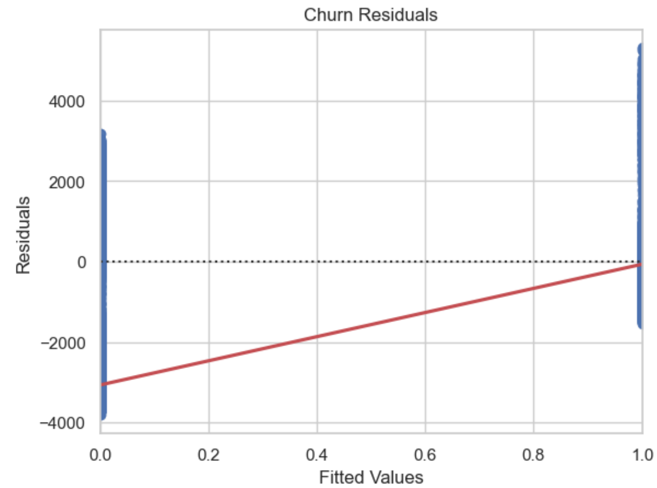
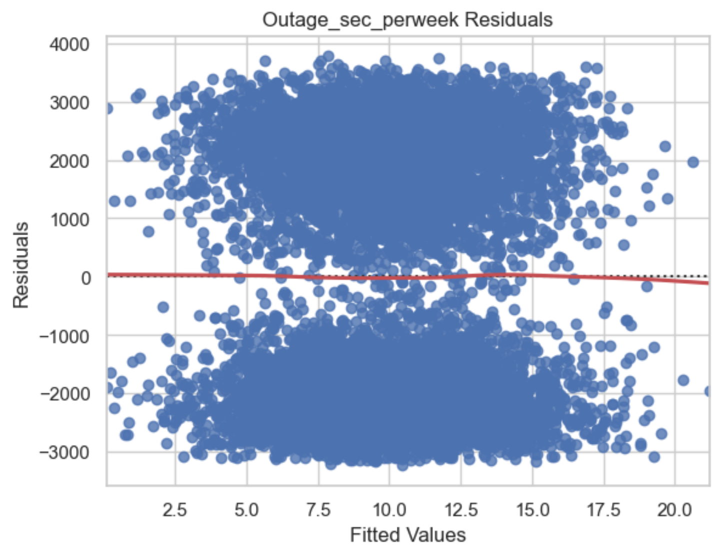
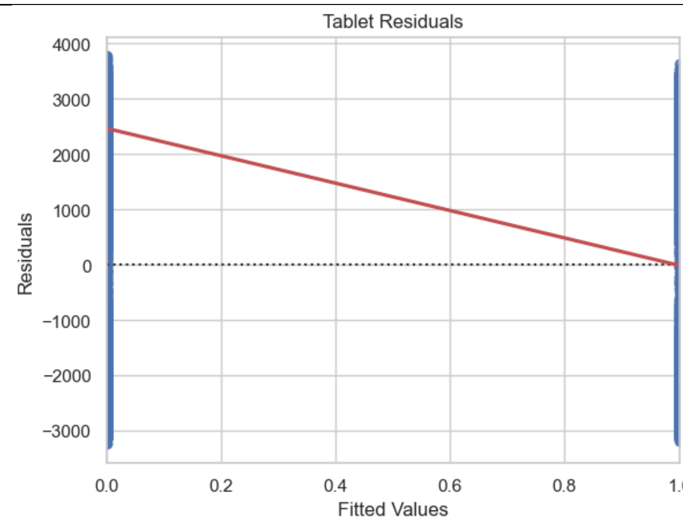
Reduced Model Calculations

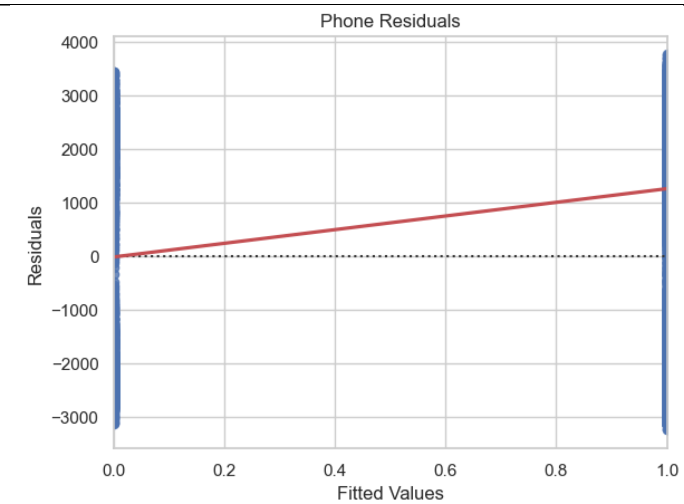
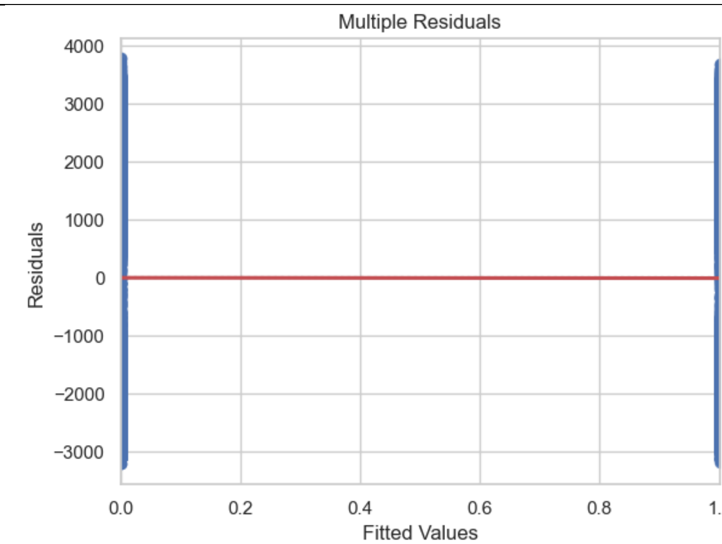
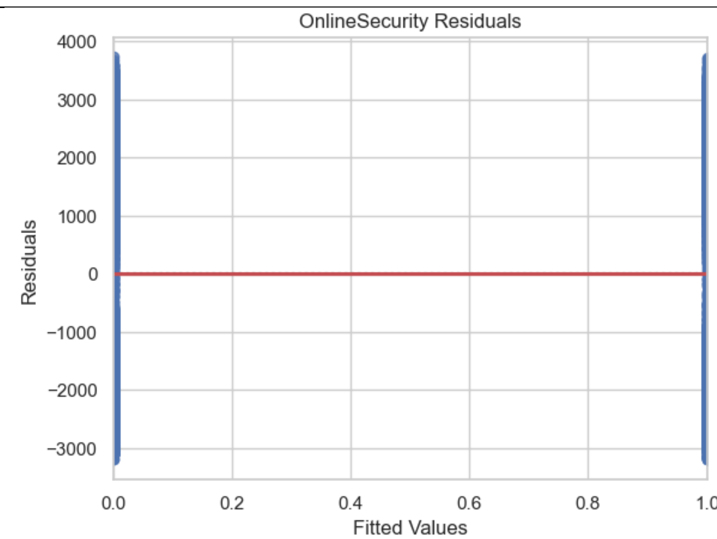
```
mse = model_reduced4.mse_resid
print('mse: ' , mse)

rse = np.sqrt(mse)
print('rse: ', rse)
```

```
mse:  8023.015882801472
rse:  89.57128938896365
```

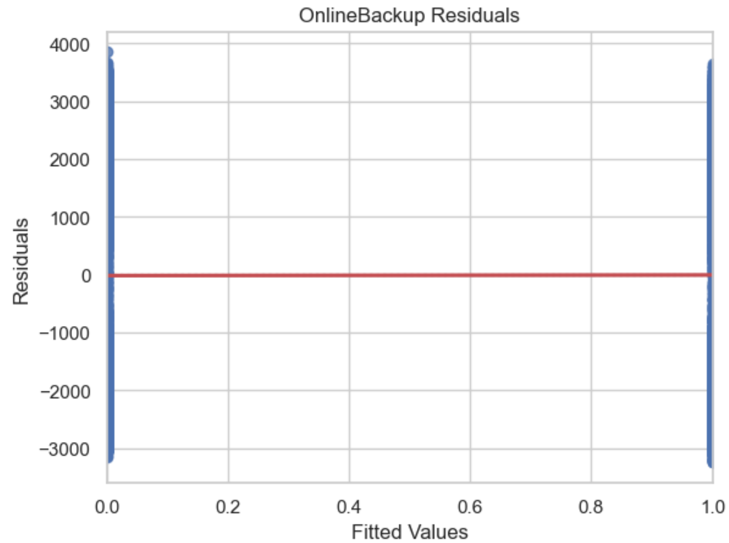
R-squared:	0.998
Adj. R-squared:	0.998
F-statistic:	3.957e+05
Prob (F-statistic):	0.00
Log-Likelihood:	-59079.
AIC:	1.182e+05
BIC:	1.183e+05

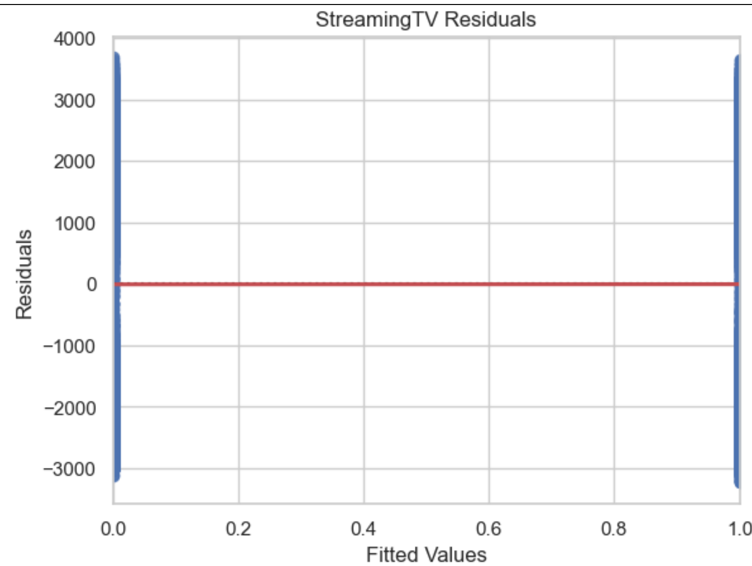
Residual Plot Population vs. Bandwidth_GB_Year	
Residual Plot Age vs. Bandwidth_GB_Year	
Residual Plot Income vs. Bandwidth_GB_Year	

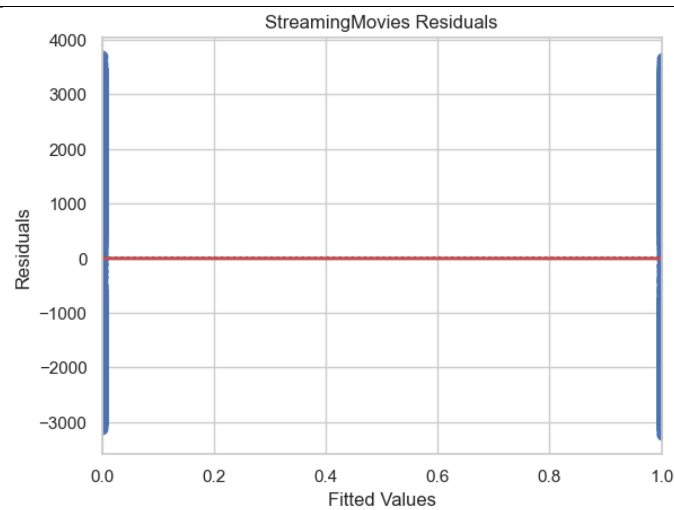
Residual Plot Churn vs. Bandwidth_GB_Year	
Residual Plot Outage_sec_perweek vs. Bandwidth_GB_Year	
Residual Plot Tablet vs. Bandwidth_GB_Year	

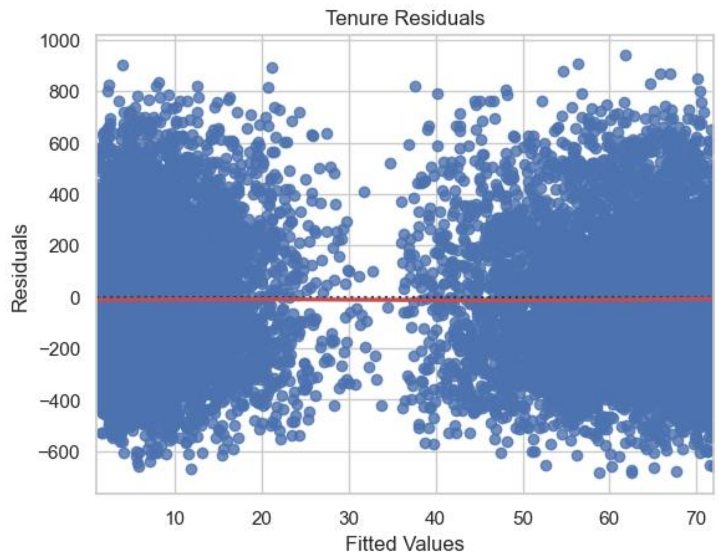
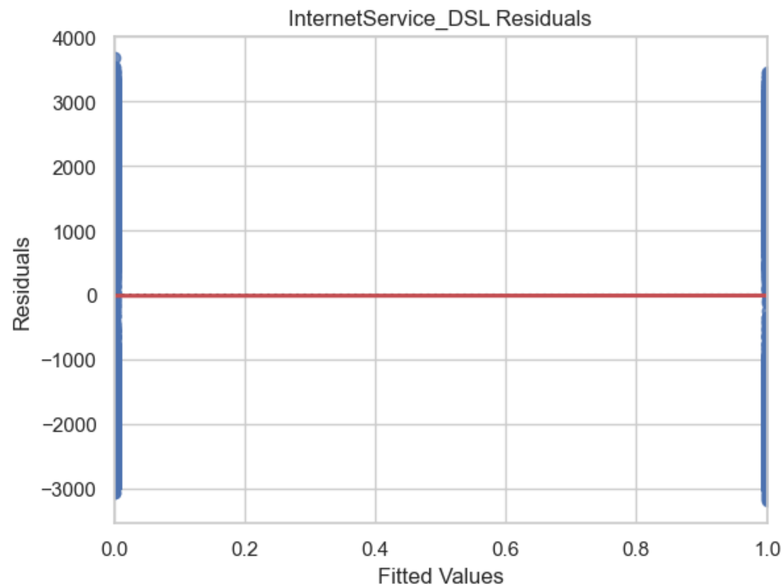
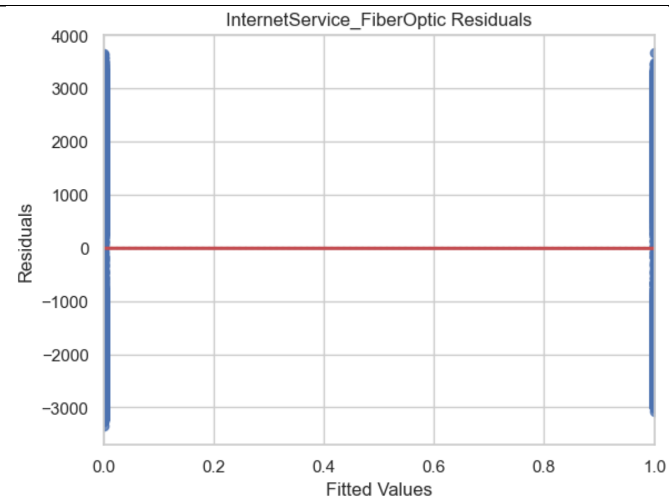
Residual Plot Phone vs. Bandwidth_GB_Year	
Residual Plot Multiple vs. Bandwidth_GB_Year	
Residual Plot OnlineSecurity vs. Bandwidth_GB_Year	

<table>
<tr>
<td>Residual Plot

OnlineBackup vs.
Bandwidth_GB_Year</td>
<td>
</td>
</tr>
<tr>
<td>Residual Plot

StreamingTV vs.
Bandwidth_GB_Year</td>
<td>
</td>
</tr>
<tr>
<td>Residual Plot

StreamingMovies vs.
Bandwidth_GB_Year</td>
<td>
</td>
</tr>
</table>

Residual Plot Tenure vs. Bandwidth_GB_Year	
Residual Plot InternetService_DSL vs. Bandwidth_GB_Year	
Residual Plot InternetService_FiberOptic vs. Bandwidth_GB_Year	

E3

A copy of the code used to support the implementation of the linear regression model is submitted. There are two code files, “D208 PA Task 1 – Code File Part 1” and “D208 PA Task 1 – Code File Part 3”. The Part 3 file uses the cleaned data set created from Part 1 to perform multiple linear regression.

F1

Here I provide a summary of my findings and assumptions by discussing the regression equation for the reduced model, an interpretation of the coefficients of the reduced model, the statistical and practical significance of the recued model, and the limitations of the data analysis.

- The regression equation is given by:

$$
\begin{aligned}
\boldsymbol{Bandwidth\_GB\_Year} = \ & 278.3015 \\
& -0.00005178 * Population \\
& + -3.3205 * Age \\
& 0.00002559 * \text{Income} \\
& + 15.9373 * Churn \\
& + 0.4132 * Outage\_sec\_perweek \\
& + -1.6094 * Tablet \\
& + -5.6398 * Phone \\
& + 67.3540 * Multiple \\
& + 79.2940 * OnlineSecurity \\
& + 93.3939 * OnlineBackup \\
& + 225.0095 * StreamingTV \\
& + 208.8257 * StreamingMovies \\
& + 81.9902 * Tenure \\
& + 411.1733 * InternetService\_DSL \\
& + -1.6165 * InternetService\_FiberOptic
\end{aligned}
$$

- An interpretation for the coefficients of the reduced model is given below:

Coefficient * Variable	**Interpretation**
$Intercept = 278.3015$	The y-intercept represents when all other independent variables are 0, the amount of bandwidth used per year on average is 278.3015.
$-0.00005178 * Population$	For every one person increase in population within a mile radius of the customer, the customer uses about 0 more GB in bandwidth per year on average.

$-3.3205 * Age$	For every one year increase in age, the customer uses about 3.3 less GB in bandwidth per year on average.
$0.00002559 * Income$	For every one dollar increase in income, the customer uses about 0 more GB in bandwidth per year on average.
$15.9373 * Churn$	Customers who churn use about 16 more GB in bandwidth per year on average than customers who don't.
$0.4132 * Outage\_sec\_perweek$	For every one second increase in average weekly outages, customers use about 0.4 more GB per year on average.
$-1.6094 * Tablet$	Customers who have a tablet use about 1.6 less GB in bandwidth per year on average than customers who don't.
$-5.6398 * Phone$	Customers who have a phone service use about 5.6 less GB in bandwidth per year on average than customers who don't.
$67.3540 * Multiple$	Customers who have multiple lines use about 67.4 more GB in bandwidth per year on average than customers who don't.
$79.2940 * OnlineSecurity$	Customers who have an online security add-on use about 79.3 more GB in bandwidth per year on average than customers who don't.
$93.3939 * OnlineBackup$	Customers who have an online backup add-on use about 93.4 more GB in bandwidth per year on average than customers who don't.
$225.0095 * StreamingTV$	Customers who have streaming TV use about 225 more GB in bandwidth per year on average than customers who don't.
$208.8257 * StreamingMovies$	Customers who have streaming movies use about 208.8 more GB in bandwidth per year on average than customers who don't.
$81.9902 * Tenure$	For every one month increase in tenure, customers use about 82 more GB in bandwidthper year on average.
$411.1733 * InternetService\_DSL$	Customers who have DSL internet use about 411.2 more GB in bandwidth per year on average than customers who don't.
$-1.6165 * InternetService\_FiberOptic$	Customers who have Fiber Optic internet use about 1.6 less GB in bandwidth per year on average than customers who don't.

- The statistical and practical significance of the reduced model are discussed here.

  Each coefficient in the reduced model is statistically significant because the p-value is less than 0.05 for each. RSE is 89.6, meaning the difference between predicted values and observed values is typically 89.6.

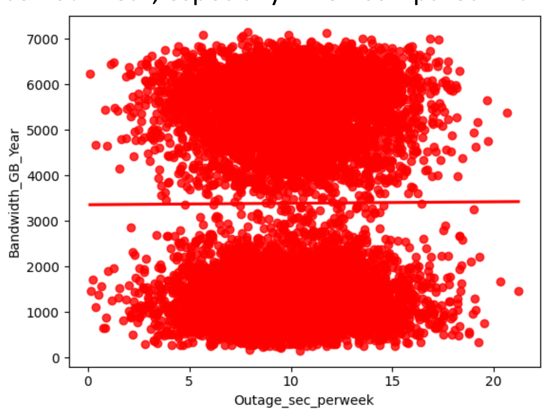
  The practical significance of the model could be lacking due to the higher number of independent variables used—there are fifteen independent variables. This can make the model hard to interpret and potentially less practical to use in the real world. Also, population and income have a coefficient essentially equal to 0, so these factors have hardly any influence on how much bandwidth a customer is using. Also, "Outage_sec_perweek" might not be a very useful variable for practical applications of the model.

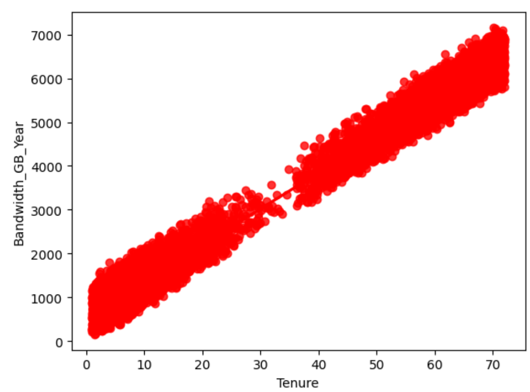
- The limitations in the analysis are as follows (Middleton, nd):

  First, not all assumptions of linear regression are met.

  I assume that the observations in this data set are independent and random.

  Multiple linear regression assumes that each independent variable is linearly related to the dependent variable. In part C3, a scatterplot was made to visualize if such a linear relationship existed. Some data when paired with "Bandwidth_GB_Year", such as "Outage_sec_perweek", was not linear, especially when compared with "Tenure".





  Independent variables should not be highly correlated with one another. As described in part D2, a VIF was calculated for each variable to check for multicollinearity and only one factor was removed as a result.

  Normality of residuals is an additional assumption of linear regression. The Prob(Omnibus) Test yields a value of 0, and since it is smaller than 0.05, this indicates that the residuals are not normally disributed. Furthermore, Prob(Jarque-Bera) Test also concludes that the residuals are not normally distributed because it also yielded a value less 0.05 ("Assumptions of Multiple Linear Regression", 2023).

Omnibus:	606.095	**Durbin-Watson:**	1.982
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	729.668
Skew:	0.619	**Prob(JB):**	3.59e-159
Kurtosis:	3.468	**Cond. No.**	3.36e+05

The last assumption is homoscedasticity. The residual plots, found in part E2, indicate that data are homoscedastic.

Additionally, the data in the reduced model still contains outliers, which affects the model. The model has a very high R^2 value, which means this model is overfitting the data.

F2

My recommended course of action based on the results my analysis is as follows:

Continue to refine the model to make it more practically significant by reducing the number of independent variables, using interactions between variables, transforming variables to make data more linear, and removing more outliers.

However, the reduced linear regression model does offer some insight into which factors are influencing customers' use of bandwidth. The aim of the research question was to help the company optimize the services they offer to customers based on factors that influence bandwidth. We can see that population and income have very little influence on the amount of bandwidth used due to their near 0 coefficients. Also, customers who have a DSL internet service use quite a lot more data on average than customers who don't.

Additionally, the company may consider selling a data plan add-on for their services with different thresholds of data. For example, if a customer streams TV or movies, they are using 209-225 more GB per year on average than customers who don't. The company also sells a data package to the customer that is equipped to handle the streaming services.

G

A Panopto video recording is provided in the submission of this performance assessment: https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=eb06ab32-ff8e-4e1f-b834-b0c00023105f

H

Web sources used to acquire segments of code to support the application:

"Data Science – Statistics Correlation Matrix." W3 Schools. www.w3schools.com/datascience/ds_stat_correlation_matrix.asp

"Detecting Multicollinearity with VIF – Python." Geeks for Geeks. www.geeksforgeeks.org/detecting-multicollinearity-with-vif-python/

"seaborn.residplot." seaborn. seaborn.pydata.org/generated/seaborn.residplot.html

Van den Broeck, Maarten. "Introduction to Regression with statsmodels in Python." Datacamp, app.datacamp.com/learn/courses/introduction-to-regression-with-statsmodels-in-python

Van den Broeck, Maarten. "Intermediate Regression with statsmodels in Python." Datacamp, app.datacamp.com/learn/courses/intermediate-regression-with-statsmodels-in-python

I

I acknowledged sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized.

References:

"Assumptions of Multiple Linear Regression". Complete Dissertation by Statistics Solutions. 2023. www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/assumptions-of-multiple-linear-regression/#:~:text=The%20last%20assumption%20of%20multiple,)%2C%20the%20data%20is%20heteroscedastic.

Middleton, Keiona. "Getting Started with D208 Part I." Predictive Modeling – D208, nd, westerngovernorsuniversity.sharepoint.com/sites/DataScienceTeam/Shared%20Documents/For ms/AllItems.aspx?csf=1&web=1&e=9ccodm&cid=3911c4ec%2D7d83%2D4ba9%2Da54b%2D7da9cbb5d38f&FolderCTID=0x01200022092E63FD85A64A8ABFB4F5AEA4839A&id=%2Fsites%2FDataScienceTeam%2FShared%20Documents%2FGraduate%20Team%2FD208%2FStudent%20Facing%20Resources%2FDr%2E%20Middleton%20Getting%20Started%20with%20D208%28Part%20I%29COIT%2Epdf&parent=%2Fsites%2FDataScienceTeam%2FShared%20Documents%2FGraduate%20Team%2FD208%2FStudent%20Facing%20Resources

Middleton, Keiona. "Getting Started with D208 Part II." Predictive Modeling – D208, nd, westerngovernorsuniversity.sharepoint.com/sites/DataScienceTeam/Shared%20Documents/Forms/AllItems.aspx?csf=1&web=1&e=9ccodm&cid=3911c4ec%2D7d83%2D4ba9%2Da54b%2D7da9cbb5d38f&FolderCTID=0x01200022092E63FD85A64A8ABFB4F5AEA4839A&id=%2Fsites%2FDataScienceTeam%2FShared%20Documents%2FGraduate%20Team%2FD208%2FStudent%20Facing%20Resources%2FDr%2E%20Middleton%20Getting%20Started%20with%20D208%20%28Part%20II%29COIT%2Epdf&parent=%2Fsites%2FDataScienceTeam%2FShared%20Documents%2FGraduate%20Team%2FD208%2FStudent%20Facing%20Resources

Nair, Aashish. "Targeting Multicollinearity With Python." Medium. December 6, 2021, towardsdatascience.com/targeting-multicollinearity-with-python-3bd3b4088d0b

Van den Broeck, Maarten. "Introduction to Regression with statsmodels in Python." Datacamp, app.datacamp.com/learn/courses/introduction-to-regression-with-statsmodels-in-python

Van den Broeck, Maarten. "Intermediate Regression with statsmodels in Python." Datacamp, app.datacamp.com/learn/courses/intermediate-regression-with-statsmodels-in-python

J

Professional communication is demonstrated in the content and presentation of my Performance Assessment.