# D208: Predictive Modeling

**Scott Sullivan | 010547531** **June 5, 2023**

## Task 1

## Part I: Research Question

### A1: Summary Questions

What factors influence high bandwidth (as described by the attribute “Bandwidth\_GB\_Year”) used by customers during a year? The author proposes to find the factors (if any), in the dataset that could potentially have an impact on bandwidth (and by extension, the health and profitability of the network).

### A2: Define Goals

The first goal of this analysis is to understand customer behavior, specifically, factors that influence their bandwidth consumption. Through this, the author will better understand the customer base and usage patterns. These insights will be able to be used to develop more effective marketing strategies to increase customer satisfaction and retention.

Also, through the understanding of which factors influence 'Bandwidth\_GB\_Year', we can improve our product to better meet customer needs. For example, if certain attributes or behaviors correlate with higher bandwidth, the company could potentially develop new products or modify existing products or service bundles that would be aimed at this customer segment and increase revenue.

## Part II: Method Justification

### B1: Summarize four assumptions of a multiple linear regression model

There are multple assumptions the author is making in regards to the multiple linear regression model.

1. **Target Variable is Continuous**: We must ensure that the dependant variable (our y-axis) is continuous.
2. **Multiple independent variables**: While the dependent variable must be continuous, the independent varables may be continuous or categorical means, but we must have multple (2+) variables.
3. **Linear Relationship**: Our variables of interest must be exampled to have a linear relationship. This will be verified with a scatterplot.
4. **No Outliers**: To avoid skewed results caused by outliers, the author will ensure that the data is cleaned to remove outliers.

### B2: Benefits of using Python in support of various phases of the analysis

1. Python (via Jupyter Notebooks) has a broad range of libraries and frameworks that can be utilized by the author to analyze large datasets. Additionally, it is cross-platform, thereby allowing the code to be run on multiple platforms.
2. Python is also very powerful in visualization of datasets. Prior to analyzing and during the analysis, it will be beneficial to get a better view of the data, and Python allows us to graph and visualize our datasets to better extract information.

### B3: Why multiple linear regression is an appropriate technique

The author believes that making use of multiple linear regression will be appropriate for a number of reasons. Primarily, the dependent variable being used for this analysis (bandwidth) is continuous, which is a key requirement for this model.

Additionally, multiple linear regression excels because while it will identify the independent and dependent variable relationships, it is also capable of quantifying the relationships. As a result, the author will be able to determine the predicted change in 'Bandwidth\_GB\_Year' as a direct result of a change in one of the independent variables while maintaining the other variables constant. This can be useful when planning quantifying potential course of actions.

## Part III: Data Preparation

### C1: Data cleaning goals & steps to clean data (includes annotated code)

The author plans to clean the data by searching for Null values (using Pandas), outliers (using histograms), removing unnecessary data attributes, converting categorical values (such as yes or no) into numerical values (1 or 0 respectively) and finally relabeling attributes to be more descriptive.

Initial examination shows 10,000 records and 50 attributes. The code to import the data and show the initial attributes are as follows.

import numpy as np  
import pandas as pd  
  
# Read the CSV  
df = pd.read\_csv('\_data/churn\_clean.csv')  
  
# Copy the dataframe to RAW in case we need to come back to the raw data  
df\_raw = df  
df.head(5)

CaseOrder Customer\_id Interaction \  
0 1 K409198 aa90260b-4141-4a24-8e36-b04ce1f4f77b   
1 2 S120509 fb76459f-c047-4a9d-8af9-e0f7d4ac2524   
2 3 K191035 344d114c-3736-4be5-98f7-c72c281e2d35   
3 4 D90850 abfa2b40-2d43-4994-b15a-989b8c79e311   
4 5 K662701 68a861fd-0d20-4e51-a587-8a90407ee574   
  
 UID City State County \  
0 e885b299883d4f9fb18e39c75155d990 Point Baker AK Prince of Wales-Hyder   
1 f2de8bef964785f41a2959829830fb8a West Branch MI Ogemaw   
2 f1784cfa9f6d92ae816197eb175d3c71 Yamhill OR Yamhill   
3 dc8a365077241bb5cd5ccd305136b05e Del Mar CA San Diego   
4 aabb64a116e83fdc4befc1fbab1663f9 Needville TX Fort Bend   
  
 Zip Lat Lng ... MonthlyCharge Bandwidth\_GB\_Year Item1 \  
0 99927 56.25100 -133.37571 ... 172.455519 904.536110 5   
1 48661 44.32893 -84.24080 ... 242.632554 800.982766 3   
2 97148 45.35589 -123.24657 ... 159.947583 2054.706961 4   
3 92014 32.96687 -117.24798 ... 119.956840 2164.579412 4   
4 77461 29.38012 -95.80673 ... 149.948316 271.493436 4   
  
 Item2 Item3 Item4 Item5 Item6 Item7 Item8   
0 5 5 3 4 4 3 4   
1 4 3 3 4 3 4 4   
2 4 2 4 4 3 3 3   
3 4 4 2 5 4 3 3   
4 4 4 3 4 4 4 5   
  
[5 rows x 50 columns]

#### Duplicates

We now examine the dataframe for duplicates:

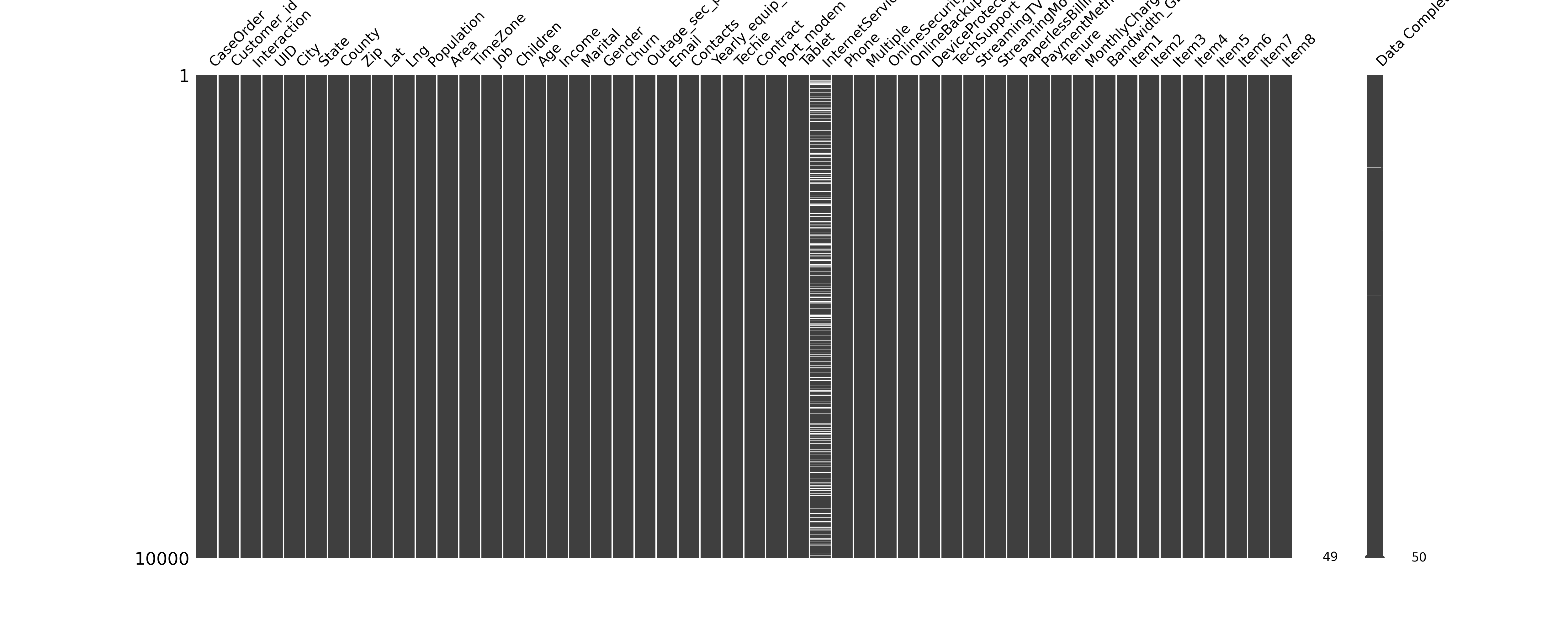
num\_duplicates = len(df) - len(df.duplicated())  
print(f"Duplicates: {num\_duplicates} of {len(df)} records.\n")

Duplicates: 0 of 10000 records.

#### Nulls

And now we use missingno to check for nulls.

import missingno as msno  
%matplotlib inline  
import matplotlib.pyplot as plt  
msno.matrix(df.sample(10000), labels=True)  
plt.title('Missing Data Visualization')  
plt.show()



We can see here that InternetService is the only attribute with Null values. To prevent significant data loss by dropping the large number of InternetService values, the author decided to perform a percentage analysis to impute the Null values with a proportional distribution matching that of the existing dataset.

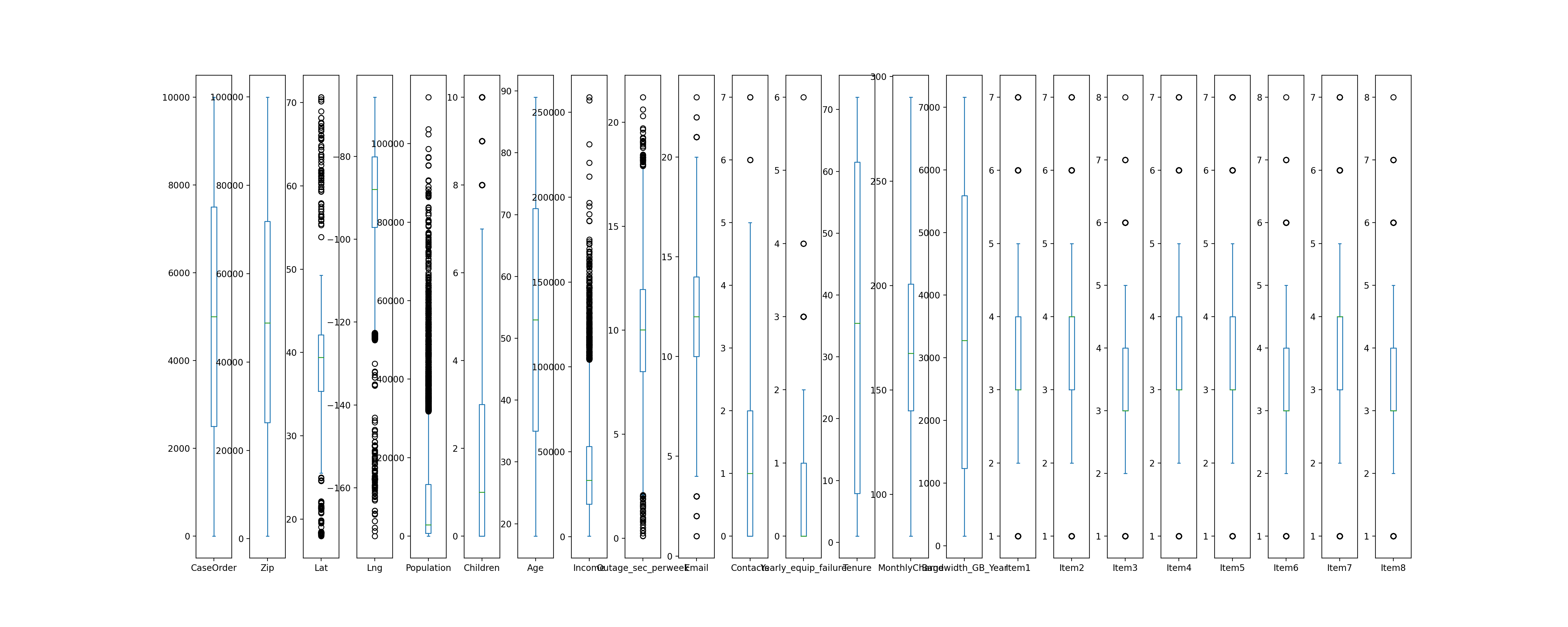
# Calculate distribution of known values  
dist = df['InternetService'].value\_counts(normalize=True)  
  
# Count Nulls to impute based on the distribution   
null\_count = df['InternetService'].isnull().sum()   
impute\_counts = (dist \* null\_count).round().astype(int)  
  
# Impute missing values based on the distribution  
missing\_indices = df['InternetService'].isnull()  
df.loc[missing\_indices, 'InternetService'] = np.random.choice(dist.index, size=null\_count, p=dist.values)  
  
# Verify there are no more null values  
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

#### Outliers

The author now looked at the outliers by printing a boxplot of the appropriate variables.

# Plot boxplot for each attribute.   
# NOTE: this code was previously written and used by the author for an earlier class, D206  
import plotnine as p9  
  
df.plot(  
 kind='box',  
 subplots=True,  
 sharey=False,  
 figsize=(25, 10)  
)  
# increase spacing between subplots  
plt.subplots\_adjust(wspace=0.5)  
plt.show()



Regarding the outliers, the author was found that although there were several attributes that may have outliers. That said, on a closer look, only "Children" and "Outages per week" had what were actual outliers. The former outlier included customers with 7-10 children. Outliers for the latter showed a cluster of outliers between 35 and 45 outages per week.

Therefore, the author decided to keep the outlier data since it was likely that either one or both may play a role in finding insights with our bandwidth analysis.

#### One Hot Encoding

For the attributes that were not a simple boolean Yes/No, the author performed a One Hot Encoding to each of the attributes, such as Gender, to quantify the values. These newly encoded columns were added to the dataframe and the originals dropped.

# One Hot Encoding  
one\_hot\_attr = ['Gender', 'Contract', 'InternetService', 'PaymentMethod']  
  
for attr in one\_hot\_attr:  
 if attr in df.columns: # if the attribute column exists error checking  
 y = pd.get\_dummies(df[attr], prefix=attr)  
 df.drop(attr, axis=1, inplace=True)  
 print(f"Dropped: {attr}\t")  
 df = pd.concat([df, y], axis="columns")  
  
print(df.head())

Dropped: Gender   
Dropped: Contract   
Dropped: InternetService   
Dropped: PaymentMethod   
 CaseOrder Customer\_id Interaction \  
0 1 K409198 aa90260b-4141-4a24-8e36-b04ce1f4f77b   
1 2 S120509 fb76459f-c047-4a9d-8af9-e0f7d4ac2524   
2 3 K191035 344d114c-3736-4be5-98f7-c72c281e2d35   
3 4 D90850 abfa2b40-2d43-4994-b15a-989b8c79e311   
4 5 K662701 68a861fd-0d20-4e51-a587-8a90407ee574   
  
 UID City State County \  
0 e885b299883d4f9fb18e39c75155d990 Point Baker AK Prince of Wales-Hyder   
1 f2de8bef964785f41a2959829830fb8a West Branch MI Ogemaw   
2 f1784cfa9f6d92ae816197eb175d3c71 Yamhill OR Yamhill   
3 dc8a365077241bb5cd5ccd305136b05e Del Mar CA San Diego   
4 aabb64a116e83fdc4befc1fbab1663f9 Needville TX Fort Bend   
  
 Zip Lat Lng ... Gender\_Nonbinary Contract\_Month-to-month \  
0 99927 56.25100 -133.37571 ... False False   
1 48661 44.32893 -84.24080 ... False True   
2 97148 45.35589 -123.24657 ... False False   
3 92014 32.96687 -117.24798 ... False False   
4 77461 29.38012 -95.80673 ... False True   
  
 Contract\_One year Contract\_Two Year InternetService\_DSL \  
0 True False False   
1 False False False   
2 False True True   
3 False True True   
4 False False False   
  
 InternetService\_Fiber Optic PaymentMethod\_Bank Transfer(automatic) \  
0 True False   
1 True True   
2 False False   
3 False False   
4 True False   
  
 PaymentMethod\_Credit Card (automatic) PaymentMethod\_Electronic Check \  
0 True False   
1 False False   
2 True False   
3 False False   
4 False False   
  
 PaymentMethod\_Mailed Check   
0 False   
1 False   
2 False   
3 True   
4 True   
  
[5 rows x 58 columns]

#### YES / NO and Boolean attributes:

For each of the attributes with a Yes or No value, the author wrote the following python code to replace each string with the respective integer value with No being replaced with 0, and Yes being replaced with 1. Additionally, the attributes that are NOT pertaining to any correlation will be dropped. These include attributes such as Customer\_ID, Interaction, UID, etc.

#### Item1-8 Renaming

Items1-8 have also been renamed here for more user-friendly analysis. The replacement names are pulled from the data dictionary provided by the instructor.

attributes\_to\_change = ['Churn', 'Techie', 'Port\_modem', 'Tablet','Phone','Multiple','OnlineSecurity', 'DeviceProtection',  
 'OnlineBackup','TechSupport','StreamingTV', 'StreamingMovies', 'PaperlessBilling']  
# 'Gender\_Male', 'Gender\_Female', 'Gender\_Nonbinary', 'Contract\_Month-to-month', 'Contract\_One year', 'Contract\_Two Year',  
# 'InternetService\_DSL', 'InternetService\_Fiber Optic']  
attributes\_to\_drop = ['CaseOrder', 'Customer\_id', 'Interaction', 'UID', 'Population', 'City', 'State', 'County',   
 'Zip', 'Lat', 'Lng', 'Marital', 'Area', 'TimeZone', 'Job']  
attributes\_to\_rename = {'Item1':'Timely\_Response', 'Item2':'Timely\_Fixes','Item3':'Timely\_Replacements','Item4':'Reliability','Item5':'Options',  
 'Item6':'Respectful','Item7':'Courteous','Item8':'Active\_Listening'}  
  
for attr in attributes\_to\_change:  
 print(f"Updated Strings to Numerical: {attr}\t")  
 df[attr] = df[attr].replace({"Yes": 1, "No": 0, "True": 1, "False": 0})  
 df[attr] = df[attr].astype(int) # Ensure data is saved as a 1 and 0 int, not a boolean "true"/"false"  
  
  
for attr in attributes\_to\_drop:  
 if attr in df.columns: # if the attribute column exists error checking  
 df.drop(attr, axis=1, inplace=True)  
 print(f"Dropped: {attr}\t")  
  
df = df.rename(columns=attributes\_to\_rename)  
  
df.head(5)

Updated Strings to Numerical: Churn   
Updated Strings to Numerical: Techie   
Updated Strings to Numerical: Port\_modem   
Updated Strings to Numerical: Tablet   
Updated Strings to Numerical: Phone   
Updated Strings to Numerical: Multiple   
Updated Strings to Numerical: OnlineSecurity   
Updated Strings to Numerical: DeviceProtection   
Updated Strings to Numerical: OnlineBackup   
Updated Strings to Numerical: TechSupport   
Updated Strings to Numerical: StreamingTV   
Updated Strings to Numerical: StreamingMovies   
Updated Strings to Numerical: PaperlessBilling   
Dropped: CaseOrder   
Dropped: Customer\_id   
Dropped: Interaction   
Dropped: UID   
Dropped: Population   
Dropped: City   
Dropped: State   
Dropped: County   
Dropped: Zip   
Dropped: Lat   
Dropped: Lng   
Dropped: Marital   
Dropped: Area   
Dropped: TimeZone   
Dropped: Job

Children Age Income Churn Outage\_sec\_perweek Email Contacts \  
0 0 68 28561.99 0 7.978323 10 0   
1 1 27 21704.77 1 11.699080 12 0   
2 4 50 9609.57 0 10.752800 9 0   
3 1 48 18925.23 0 14.913540 15 2   
4 0 83 40074.19 1 8.147417 16 2   
  
 Yearly\_equip\_failure Techie Port\_modem ... Gender\_Nonbinary \  
0 1 0 1 ... False   
1 1 1 0 ... False   
2 1 1 1 ... False   
3 0 1 0 ... False   
4 1 0 1 ... False   
  
 Contract\_Month-to-month Contract\_One year Contract\_Two Year \  
0 False True False   
1 True False False   
2 False False True   
3 False False True   
4 True False False   
  
 InternetService\_DSL InternetService\_Fiber Optic \  
0 False True   
1 False True   
2 True False   
3 True False   
4 False True   
  
 PaymentMethod\_Bank Transfer(automatic) \  
0 False   
1 True   
2 False   
3 False   
4 False   
  
 PaymentMethod\_Credit Card (automatic) PaymentMethod\_Electronic Check \  
0 True False   
1 False False   
2 True False   
3 False False   
4 False False   
  
 PaymentMethod\_Mailed Check   
0 False   
1 False   
2 False   
3 True   
4 True   
  
[5 rows x 43 columns]

#### Print sample row to verify values for each column have been updated

column\_names = df.columns  
dtypes = df.dtypes  
first\_row = df.iloc[1].values  
  
column\_names = df.columns  
dtypes = df.dtypes  
first\_row = df.iloc[1].values  
print(f"\033[1mName\t\tType\tValue\033[0m")  
for column\_name, dtype, value in zip(column\_names, dtypes, first\_row):  
 print(f"{column\_name:<15}\t{dtype}\t{value:<18}\n")

Name Type Value  
Children int64 1   
  
Age int64 27   
  
Income float64 21704.77   
  
Churn int64 1   
  
Outage\_sec\_perweek float64 11.69907956   
  
Email int64 12   
  
Contacts int64 0   
  
Yearly\_equip\_failure int64 1   
  
Techie int64 1   
  
Port\_modem int64 0   
  
Tablet int64 1   
  
Phone int64 1   
  
Multiple int64 1   
  
OnlineSecurity int64 1   
  
OnlineBackup int64 0   
  
DeviceProtection int64 0   
  
TechSupport int64 0   
  
StreamingTV int64 1   
  
StreamingMovies int64 1   
  
PaperlessBilling int64 1   
  
Tenure float64 1.156680997   
  
MonthlyCharge float64 242.632554   
  
Bandwidth\_GB\_Year float64 800.9827661   
  
Timely\_Response int64 3   
  
Timely\_Fixes int64 4   
  
Timely\_Replacements int64 3   
  
Reliability int64 3   
  
Options int64 4   
  
Respectful int64 3   
  
Courteous int64 4   
  
Active\_Listening int64 4   
  
Gender\_Female bool 1   
  
Gender\_Male bool 0   
  
Gender\_Nonbinary bool 0   
  
Contract\_Month-to-month bool 1   
  
Contract\_One year bool 0   
  
Contract\_Two Year bool 0   
  
InternetService\_DSL bool 0   
  
InternetService\_Fiber Optic bool 1   
  
PaymentMethod\_Bank Transfer(automatic) bool 1   
  
PaymentMethod\_Credit Card (automatic) bool 0   
  
PaymentMethod\_Electronic Check bool 0   
  
PaymentMethod\_Mailed Check bool 0

### C2: Describe dependent and independent variables using summary statistics

The **dependent variable** for our question from section A1 is "Bandwidth\_GB\_Year", which is a continuous variable that captures the total bandwidth used by the customer for the year (in gigabytes). By having this as our dependent variable, the author will analyze the other attributes (or independent variables) in order to find which have a correlation affect the dependent variable.

Our **independent variables** are the remainder of the attributes in our dataframe (after cleaning). Specifically, we have a number of categorical, boolean, and continuous variables. Included below is a summary statistic describing all of the independent variables that are necessary for answering the research question from A1.

Below is the code and output for the summary statistics:

# Independent variables  
independent\_variables = ['Children', 'Age', 'Income', 'Churn', 'Outage\_sec\_perweek', 'Email', 'Contacts',   
 'Yearly\_equip\_failure', 'Techie', 'Port\_modem', 'Tablet', 'Phone', 'Multiple',   
 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',  
 'StreamingMovies', 'PaperlessBilling', 'Tenure', 'MonthlyCharge',  
 'Timely\_Response', 'Timely\_Fixes', 'Timely\_Replacements', 'Reliability', 'Options',  
 'Respectful', 'Courteous', 'Active\_Listening']   
boolean\_variables = ['Gender\_Female', 'Gender\_Male', 'Gender\_Nonbinary',  
 'Contract\_Month-to-month', 'Contract\_One year','Contract\_Two Year',  
 'InternetService\_DSL','InternetService\_Fiber Optic','PaymentMethod\_Bank Transfer(automatic)',  
 'PaymentMethod\_Credit Card (automatic)','PaymentMethod\_Electronic Check',  
 'PaymentMethod\_Mailed Check']  
variables = independent\_variables + boolean\_variables  
import plotnine as p9  
  
for var in variables:  
 print(f"Summary statistics for {var}:")  
 print(df[var].describe())  
 print("\n")

Summary statistics for Children:  
count 10000.0000  
mean 2.0877  
std 2.1472  
min 0.0000  
25% 0.0000  
50% 1.0000  
75% 3.0000  
max 10.0000  
Name: Children, dtype: float64  
  
  
Summary statistics for Age:  
count 10000.000000  
mean 53.078400  
std 20.698882  
min 18.000000  
25% 35.000000  
50% 53.000000  
75% 71.000000  
max 89.000000  
Name: Age, dtype: float64  
  
  
Summary statistics for Income:  
count 10000.000000  
mean 39806.926771  
std 28199.916702  
min 348.670000  
25% 19224.717500  
50% 33170.605000  
75% 53246.170000  
max 258900.700000  
Name: Income, dtype: float64  
  
  
Summary statistics for Churn:  
count 10000.000000  
mean 0.265000  
std 0.441355  
min 0.000000  
25% 0.000000  
50% 0.000000  
75% 1.000000  
max 1.000000  
Name: Churn, dtype: float64  
  
  
Summary statistics for Outage\_sec\_perweek:  
count 10000.000000  
mean 10.001848  
std 2.976019  
min 0.099747  
25% 8.018214  
50% 10.018560  
75% 11.969485  
max 21.207230  
Name: Outage\_sec\_perweek, dtype: float64  
  
  
Summary statistics for Email:  
count 10000.000000  
mean 12.016000  
std 3.025898  
min 1.000000  
25% 10.000000  
50% 12.000000  
75% 14.000000  
max 23.000000  
Name: Email, dtype: float64  
  
  
Summary statistics for Contacts:  
count 10000.000000  
mean 0.994200  
std 0.988466  
min 0.000000  
25% 0.000000  
50% 1.000000  
75% 2.000000  
max 7.000000  
Name: Contacts, dtype: float64  
  
  
Summary statistics for Yearly\_equip\_failure:  
count 10000.000000  
mean 0.398000  
std 0.635953  
min 0.000000  
25% 0.000000  
50% 0.000000  
75% 1.000000  
max 6.000000  
Name: Yearly\_equip\_failure, dtype: float64  
  
  
Summary statistics for Techie:  
count 10000.000000  
mean 0.167900  
std 0.373796  
min 0.000000  
25% 0.000000  
50% 0.000000  
75% 0.000000  
max 1.000000  
Name: Techie, dtype: float64  
  
  
Summary statistics for Port\_modem:  
count 10000.000000  
mean 0.483400  
std 0.499749  
min 0.000000  
25% 0.000000  
50% 0.000000  
75% 1.000000  
max 1.000000  
Name: Port\_modem, dtype: float64  
  
  
Summary statistics for Tablet:  
count 10000.000000  
mean 0.299100  
std 0.457887  
min 0.000000  
25% 0.000000  
50% 0.000000  
75% 1.000000  
max 1.000000  
Name: Tablet, dtype: float64  
  
  
Summary statistics for Phone:  
count 10000.000000  
mean 0.906700  
std 0.290867  
min 0.000000  
25% 1.000000  
50% 1.000000  
75% 1.000000  
max 1.000000  
Name: Phone, dtype: float64  
  
  
Summary statistics for Multiple:  
count 10000.000000  
mean 0.460800  
std 0.498486  
min 0.000000  
25% 0.000000  
50% 0.000000  
75% 1.000000  
max 1.000000  
Name: Multiple, dtype: float64  
  
  
Summary statistics for OnlineSecurity:  
count 10000.000000  
mean 0.357600  
std 0.479317  
min 0.000000  
25% 0.000000  
50% 0.000000  
75% 1.000000  
max 1.000000  
Name: OnlineSecurity, dtype: float64  
  
  
Summary statistics for OnlineBackup:  
count 10000.000000  
mean 0.450600  
std 0.497579  
min 0.000000  
25% 0.000000  
50% 0.000000  
75% 1.000000  
max 1.000000  
Name: OnlineBackup, dtype: float64  
  
  
Summary statistics for DeviceProtection:  
count 10000.000000  
mean 0.438600  
std 0.496241  
min 0.000000  
25% 0.000000  
50% 0.000000  
75% 1.000000  
max 1.000000  
Name: DeviceProtection, dtype: float64  
  
  
Summary statistics for TechSupport:  
count 10000.000000  
mean 0.375000  
std 0.484147  
min 0.000000  
25% 0.000000  
50% 0.000000  
75% 1.000000  
max 1.000000  
Name: TechSupport, dtype: float64  
  
  
Summary statistics for StreamingTV:  
count 10000.000000  
mean 0.492900  
std 0.499975  
min 0.000000  
25% 0.000000  
50% 0.000000  
75% 1.000000  
max 1.000000  
Name: StreamingTV, dtype: float64  
  
  
Summary statistics for StreamingMovies:  
count 10000.000000  
mean 0.489000  
std 0.499904  
min 0.000000  
25% 0.000000  
50% 0.000000  
75% 1.000000  
max 1.000000  
Name: StreamingMovies, dtype: float64  
  
  
Summary statistics for PaperlessBilling:  
count 10000.000000  
mean 0.588200  
std 0.492184  
min 0.000000  
25% 0.000000  
50% 1.000000  
75% 1.000000  
max 1.000000  
Name: PaperlessBilling, dtype: float64  
  
  
Summary statistics for Tenure:  
count 10000.000000  
mean 34.526188  
std 26.443063  
min 1.000259  
25% 7.917694  
50% 35.430507  
75% 61.479795  
max 71.999280  
Name: Tenure, dtype: float64  
  
  
Summary statistics for MonthlyCharge:  
count 10000.000000  
mean 172.624816  
std 42.943094  
min 79.978860  
25% 139.979239  
50% 167.484700  
75% 200.734725  
max 290.160419  
Name: MonthlyCharge, dtype: float64  
  
  
Summary statistics for Timely\_Response:  
count 10000.000000  
mean 3.490800  
std 1.037797  
min 1.000000  
25% 3.000000  
50% 3.000000  
75% 4.000000  
max 7.000000  
Name: Timely\_Response, dtype: float64  
  
  
Summary statistics for Timely\_Fixes:  
count 10000.000000  
mean 3.505100  
std 1.034641  
min 1.000000  
25% 3.000000  
50% 4.000000  
75% 4.000000  
max 7.000000  
Name: Timely\_Fixes, dtype: float64  
  
  
Summary statistics for Timely\_Replacements:  
count 10000.000000  
mean 3.487000  
std 1.027977  
min 1.000000  
25% 3.000000  
50% 3.000000  
75% 4.000000  
max 8.000000  
Name: Timely\_Replacements, dtype: float64  
  
  
Summary statistics for Reliability:  
count 10000.000000  
mean 3.497500  
std 1.025816  
min 1.000000  
25% 3.000000  
50% 3.000000  
75% 4.000000  
max 7.000000  
Name: Reliability, dtype: float64  
  
  
Summary statistics for Options:  
count 10000.000000  
mean 3.492900  
std 1.024819  
min 1.000000  
25% 3.000000  
50% 3.000000  
75% 4.000000  
max 7.000000  
Name: Options, dtype: float64  
  
  
Summary statistics for Respectful:  
count 10000.000000  
mean 3.497300  
std 1.033586  
min 1.000000  
25% 3.000000  
50% 3.000000  
75% 4.000000  
max 8.000000  
Name: Respectful, dtype: float64  
  
  
Summary statistics for Courteous:  
count 10000.000000  
mean 3.509500  
std 1.028502  
min 1.000000  
25% 3.000000  
50% 4.000000  
75% 4.000000  
max 7.000000  
Name: Courteous, dtype: float64  
  
  
Summary statistics for Active\_Listening:  
count 10000.000000  
mean 3.495600  
std 1.028633  
min 1.000000  
25% 3.000000  
50% 3.000000  
75% 4.000000  
max 8.000000  
Name: Active\_Listening, dtype: float64  
  
  
Summary statistics for Gender\_Female:  
count 10000  
unique 2  
top True  
freq 5025  
Name: Gender\_Female, dtype: object  
  
  
Summary statistics for Gender\_Male:  
count 10000  
unique 2  
top False  
freq 5256  
Name: Gender\_Male, dtype: object  
  
  
Summary statistics for Gender\_Nonbinary:  
count 10000  
unique 2  
top False  
freq 9769  
Name: Gender\_Nonbinary, dtype: object  
  
  
Summary statistics for Contract\_Month-to-month:  
count 10000  
unique 2  
top True  
freq 5456  
Name: Contract\_Month-to-month, dtype: object  
  
  
Summary statistics for Contract\_One year:  
count 10000  
unique 2  
top False  
freq 7898  
Name: Contract\_One year, dtype: object  
  
  
Summary statistics for Contract\_Two Year:  
count 10000  
unique 2  
top False  
freq 7558  
Name: Contract\_Two Year, dtype: object  
  
  
Summary statistics for InternetService\_DSL:  
count 10000  
unique 2  
top False  
freq 5597  
Name: InternetService\_DSL, dtype: object  
  
  
Summary statistics for InternetService\_Fiber Optic:  
count 10000  
unique 2  
top True  
freq 5597  
Name: InternetService\_Fiber Optic, dtype: object  
  
  
Summary statistics for PaymentMethod\_Bank Transfer(automatic):  
count 10000  
unique 2  
top False  
freq 7771  
Name: PaymentMethod\_Bank Transfer(automatic), dtype: object  
  
  
Summary statistics for PaymentMethod\_Credit Card (automatic):  
count 10000  
unique 2  
top False  
freq 7917  
Name: PaymentMethod\_Credit Card (automatic), dtype: object  
  
  
Summary statistics for PaymentMethod\_Electronic Check:  
count 10000  
unique 2  
top False  
freq 6602  
Name: PaymentMethod\_Electronic Check, dtype: object  
  
  
Summary statistics for PaymentMethod\_Mailed Check:  
count 10000  
unique 2  
top False  
freq 7710  
Name: PaymentMethod\_Mailed Check, dtype: object

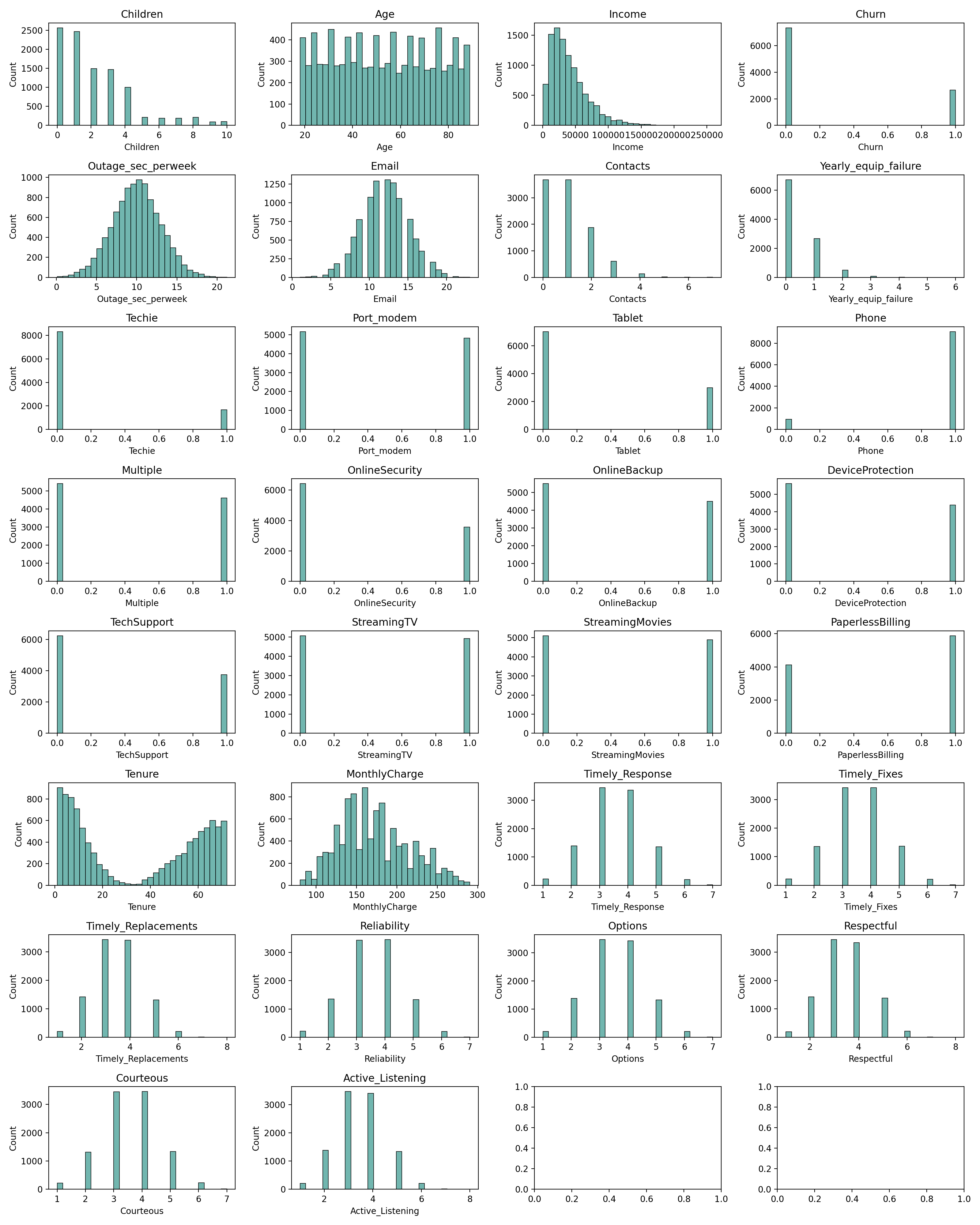
### C3. Univariate and bivariate visualizations (dependent and independent variables)

including the dependent variable in your bivariate visualizations.

# Independent variables  
independent\_variables = ['Children', 'Age', 'Income', 'Churn', 'Outage\_sec\_perweek', 'Email', 'Contacts',   
 'Yearly\_equip\_failure', 'Techie', 'Port\_modem', 'Tablet', 'Phone', 'Multiple',   
 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',  
 'StreamingMovies', 'PaperlessBilling', 'Tenure', 'MonthlyCharge',  
 'Timely\_Response', 'Timely\_Fixes', 'Timely\_Replacements', 'Reliability', 'Options',  
 'Respectful', 'Courteous', 'Active\_Listening']   
boolean\_variables = ['Gender\_Female', 'Gender\_Male', 'Gender\_Nonbinary',  
 'Contract\_Month-to-month', 'Contract\_One year','Contract\_Two Year',  
 'InternetService\_DSL','InternetService\_Fiber Optic','PaymentMethod\_Bank Transfer(automatic)',  
 'PaymentMethod\_Credit Card (automatic)','PaymentMethod\_Electronic Check',  
 'PaymentMethod\_Mailed Check']

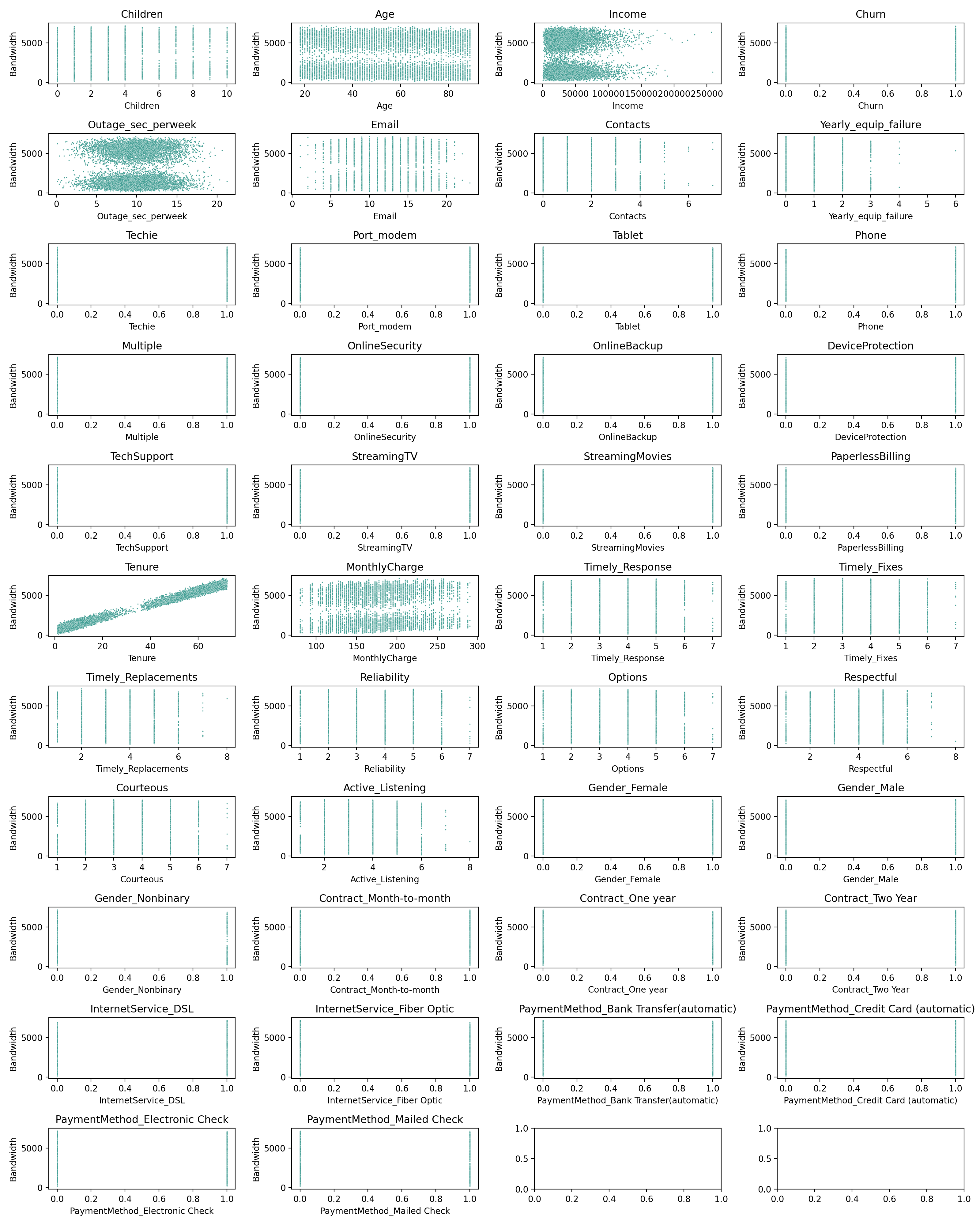
import seaborn as sns  
import matplotlib.pyplot as plt  
import plotnine as p9  
  
# UNIVARIATE Histograms  
print("Univariate Visualizations\n")  
  
variables = independent\_variables  
fig, axs = plt.subplots(nrows=8, ncols=4, figsize=(16,20))  
for ax, var in zip(axs.flatten(), variables):  
 sns.histplot(data=df, x=var, bins=30, color='#429e94', ax=ax)  
 ax.set\_title(var)  
 ax.set\_xlabel(var)  
 ax.set\_ylabel('Count')  
plt.tight\_layout()  
plt.show()

Univariate Visualizations



import seaborn as sns  
import matplotlib.pyplot as plt  
import plotnine as p9  
  
# BIVARIATE Histograms  
print("Bivariate Visualizations\n")  
variables = independent\_variables + boolean\_variables  
fig, axs = plt.subplots(nrows=11, ncols=4, figsize=(16,20))  
for ax, var in zip(axs.flatten(), variables):  
 sns.scatterplot(data=df, x=var, y='Bandwidth\_GB\_Year', s=2, color='#429e94', ax=ax)  
 ax.set\_title(var)  
 ax.set\_xlabel(var)  
 ax.set\_ylabel('Bandwidth')  
plt.tight\_layout()  
plt.show()

Bivariate Visualizations



### C4. Data Transformation Goals and Transformation Steps Taken

The author focused his attention around the bandwidth attribute (Bandwidth\_GB\_Year) as the dependent variable (it is the continuous target variable) to explore and analyze which factors (from the independent variables) showed a correlation with the bandwidth.

**Data was transformed using the following steps.** First, it needed to be cleaned in preperation for the training and testing the dataset model. To clean the data, null values were removed, "yes/no" and "true"/"false" values were converted to 1/0, respectively. This was done by replacing null values with appropriate default vales for the former case. For the latter cases, the boolean strings were converted to numerical booleans using a replace on each column with said data values.

Additionally, categorical data was encoded via one-hot encoding to multiple new attributes that were boolean values. This was accomplished using the pandas pd.get\_dummies(df[attribute], prefix=attribute) code from Section C1 above.

**This goal of data transformation** left us with a complete set of data that was prepared for proper analysis on the dependent variable. The author ensured that all required data for the analysis was in numerical form (for later regression analysis). Specifically, this included converting categorical data into numerical form (ie. one-hot encoding mentioned above). The author searched for missing values and depending on the nature of each data attribute, resolved these issues. Finally, the author checked for outliers (though the author determined that the nature of the outliters did not require they be removed due to the data insights they might hold).

All annotated code is illustrated in section C1, labeled according to each transformation. Due to this being a jupter notebook, the actual code will be kept in section C1.

### C5. Export Prepared Data Set as CSV File

Below is the code written by the author to export the prepared dataframe as a CSV file.

# Export Prepared Data to CSV  
file\_name = '\_data/cleaned\_and\_prepared\_dataframe.csv'  
df.to\_csv(file\_name, index=False)  
print(f"Exported to {file\_name}")

Exported to \_data/cleaned\_and\_prepared\_dataframe.csv

## Part IV: Model Comparison and Analysis

### D1: Construct Initial Multiple Linear Regression Model from All Independent Variables from C2

For clarification, the independent variables from C2 are being copied here. This is the exact same list given in C2:

# Independent variables  
independent\_variables = ['Children', 'Age', 'Income', 'Churn', 'Outage\_sec\_perweek', 'Email', 'Contacts',   
 'Yearly\_equip\_failure', 'Techie', 'Port\_modem', 'Tablet', 'Phone', 'Multiple',   
 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',  
 'StreamingMovies', 'PaperlessBilling', 'Tenure', 'MonthlyCharge',  
 'Timely\_Response', 'Timely\_Fixes', 'Timely\_Replacements', 'Reliability', 'Options',  
 'Respectful', 'Courteous', 'Active\_Listening']   
boolean\_variables = ['Gender\_Female', 'Gender\_Male', 'Gender\_Nonbinary',  
 'Contract\_Month-to-month', 'Contract\_One year','Contract\_Two Year',  
 'InternetService\_DSL','InternetService\_Fiber Optic','PaymentMethod\_Bank Transfer(automatic)',  
 'PaymentMethod\_Credit Card (automatic)','PaymentMethod\_Electronic Check',  
 'PaymentMethod\_Mailed Check']  
variables = independent\_variables + boolean\_variables

At this point, the author will create an initial multiple lindear regression model that uses all the above independent variables. The author chose to use Scikit Learn due to its popularity and the fact that the author has used this framework previously. The LinearRegression model is imported, and the attributes listed immediately above are loaded as the X (independent) axis and our bandwith (Bandwith\_GB\_Year) is loaded as our dependent variable.

The Ordinary Least Squares was performed as the multiple linear regression. This model was chosen due to its ability to estimate the relationship between multiple independent variables (by minimizing the sum of squares). Since scikit-learn does not directly provide the required statistics for this, the author used Statsmodels for the statistics.

from sklearn.linear\_model import LinearRegression  
import statsmodels.api as sm  
  
# independent variables from above and the target variable (bandwidth used by the customer)  
X = df[independent\_variables]  
y = df['Bandwidth\_GB\_Year']  
  
# Create a Linear Regression model  
model = sm.OLS(y,X)  
results = model.fit()  
print(results.summary())

OLS Regression Results   
=======================================================================================  
Dep. Variable: Bandwidth\_GB\_Year R-squared (uncentered): 0.998  
Model: OLS Adj. R-squared (uncentered): 0.998  
Method: Least Squares F-statistic: 1.359e+05  
Date: Sun, 04 Jun 2023 Prob (F-statistic): 0.00  
Time: 23:48:12 Log-Likelihood: -67136.  
No. Observations: 10000 AIC: 1.343e+05  
Df Residuals: 9970 BIC: 1.345e+05  
Df Model: 30   
Covariance Type: nonrobust   
========================================================================================  
 coef std err t P>|t| [0.025 0.975]  
----------------------------------------------------------------------------------------  
Children 32.2981 0.927 34.833 0.000 30.481 34.116  
Age -2.9576 0.095 -31.179 0.000 -3.144 -2.772  
Income 0.0003 7.04e-05 3.619 0.000 0.000 0.000  
Churn 91.8733 5.837 15.741 0.000 80.432 103.314  
Outage\_sec\_perweek 2.9839 0.646 4.618 0.000 1.717 4.251  
Email 4.1234 0.628 6.567 0.000 2.893 5.354  
Contacts 4.7884 2.017 2.374 0.018 0.834 8.742  
Yearly\_equip\_failure 5.6342 3.136 1.796 0.072 -0.514 11.782  
Techie 0.5970 5.359 0.111 0.911 -9.908 11.102  
Port\_modem 2.5816 3.988 0.647 0.517 -5.236 10.399  
Tablet 4.1497 4.362 0.951 0.341 -4.401 12.700  
Phone 32.1947 6.659 4.835 0.000 19.142 45.247  
Multiple 122.9794 5.601 21.957 0.000 112.000 133.958  
OnlineSecurity 89.0391 4.189 21.257 0.000 80.828 97.250  
OnlineBackup 128.2174 4.832 26.537 0.000 118.746 137.689  
DeviceProtection 104.3405 4.312 24.196 0.000 95.887 112.794  
TechSupport 33.4189 4.370 7.647 0.000 24.852 41.985  
StreamingTV 281.7902 6.405 43.998 0.000 269.236 294.345  
StreamingMovies 272.8372 7.355 37.096 0.000 258.420 287.254  
PaperlessBilling 1.2286 4.045 0.304 0.761 -6.700 9.158  
Tenure 82.8945 0.089 935.243 0.000 82.721 83.068  
MonthlyCharge -1.5214 0.113 -13.405 0.000 -1.744 -1.299  
Timely\_Response -6.9245 2.862 -2.420 0.016 -12.534 -1.315  
Timely\_Fixes 5.7051 2.682 2.127 0.033 0.449 10.962  
Timely\_Replacements 3.7956 2.446 1.552 0.121 -1.000 8.591  
Reliability 16.3423 2.037 8.022 0.000 12.349 20.336  
Options 25.7709 1.979 13.023 0.000 21.892 29.650  
Respectful 7.3437 2.328 3.154 0.002 2.780 11.908  
Courteous 8.2076 2.194 3.741 0.000 3.907 12.508  
Active\_Listening 10.9887 2.089 5.259 0.000 6.893 15.084  
==============================================================================  
Omnibus: 9779.697 Durbin-Watson: 1.975  
Prob(Omnibus): 0.000 Jarque-Bera (JB): 1048.003  
Skew: 0.514 Prob(JB): 2.69e-228  
Kurtosis: 1.792 Cond. No. 2.54e+05  
==============================================================================  
  
Notes:  
[1] R² is computed without centering (uncentered) since the model does not contain a constant.  
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.  
[3] The condition number is large, 2.54e+05. This might indicate that there are  
strong multicollinearity or other numerical problems.

The author repeated this with the encoded categorical variables.

X = df[boolean\_variables]  
y = df['Bandwidth\_GB\_Year']  
  
# Create a Linear Regression model  
model = sm.OLS(y,X)  
results = model.fit()  
print(results.summary())

OLS Regression Results   
==============================================================================  
Dep. Variable: Bandwidth\_GB\_Year R-squared: 0.008  
Model: OLS Adj. R-squared: 0.007  
Method: Least Squares F-statistic: 10.07  
Date: Sun, 04 Jun 2023 Prob (F-statistic): 4.29e-14  
Time: 23:49:50 Log-Likelihood: -91044.  
No. Observations: 10000 AIC: 1.821e+05  
Df Residuals: 9991 BIC: 1.822e+05  
Df Model: 8   
Covariance Type: nonrobust   
==========================================================================================================  
 coef std err t P>|t| [0.025 0.975]  
----------------------------------------------------------------------------------------------------------  
Gender\_Female 3.944e+15 3.39e+15 1.162 0.245 -2.71e+15 1.06e+16  
Gender\_Male 3.944e+15 3.39e+15 1.162 0.245 -2.71e+15 1.06e+16  
Gender\_Nonbinary 3.944e+15 3.39e+15 1.162 0.245 -2.71e+15 1.06e+16  
Contract\_Month-to-month 2.884e+15 2.48e+15 1.162 0.245 -1.98e+15 7.75e+15  
Contract\_One year 2.884e+15 2.48e+15 1.162 0.245 -1.98e+15 7.75e+15  
Contract\_Two Year 2.884e+15 2.48e+15 1.162 0.245 -1.98e+15 7.75e+15  
InternetService\_DSL -9.172e+15 7.89e+15 -1.162 0.245 -2.46e+16 6.3e+15  
InternetService\_Fiber Optic -9.172e+15 7.89e+15 -1.162 0.245 -2.46e+16 6.3e+15  
PaymentMethod\_Bank Transfer(automatic) 2.343e+15 2.02e+15 1.162 0.245 -1.61e+15 6.3e+15  
PaymentMethod\_Credit Card (automatic) 2.343e+15 2.02e+15 1.162 0.245 -1.61e+15 6.3e+15  
PaymentMethod\_Electronic Check 2.343e+15 2.02e+15 1.162 0.245 -1.61e+15 6.3e+15  
PaymentMethod\_Mailed Check 2.343e+15 2.02e+15 1.162 0.245 -1.61e+15 6.3e+15  
==============================================================================  
Omnibus: 42569.760 Durbin-Watson: 0.182  
Prob(Omnibus): 0.000 Jarque-Bera (JB): 1239.009  
Skew: 0.067 Prob(JB): 8.97e-270  
Kurtosis: 1.281 Cond. No. 3.56e+15  
==============================================================================  
  
Notes:  
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.  
[2] The smallest eigenvalue is 1.32e-27. This might indicate that there are  
strong multicollinearity problems or that the design matrix is singular.

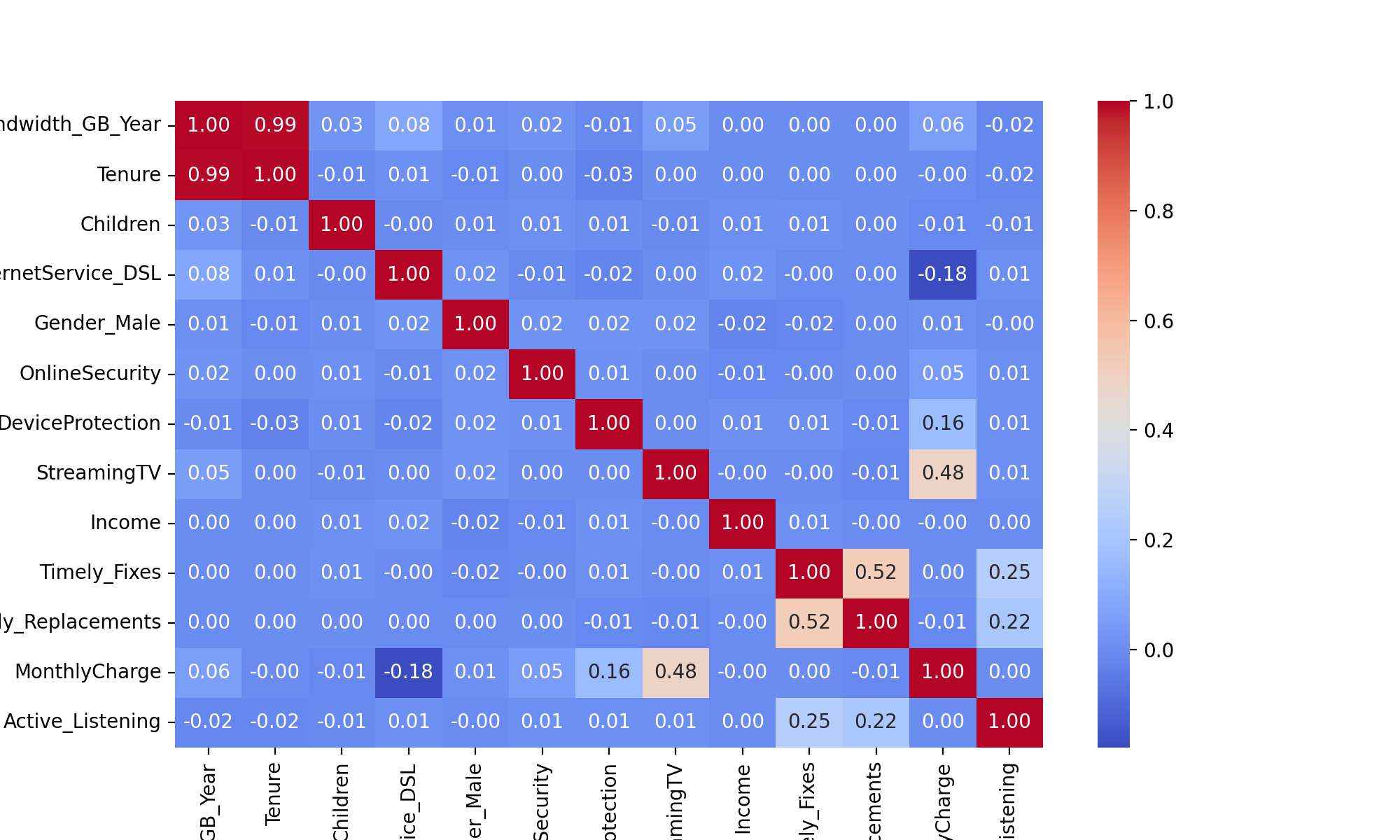
### D2: Statistically Based Model Reduction Justification

Based on the metrics analyzed above, the author decided to reduce the model to some key parameters in addition to a few 'placebo' parameters to ensure that the correlation was present.

Tenure had a high coefficient (82), as did OnlineSecurity (69), DeviceProtection(44), StreamingTV(90), and Gender\_Male(52). InternetService seems to be extrememly high (possibly an outlier due to the one hot encoding conversion that was done on that attribute earlier.

Additionally, attributes with a p-value that was not statistically significant (values larger than 5% or 0.05) were removed. This has the benefit of reducing the likelihood of overfitting the data.

reduced\_df = df[['Bandwidth\_GB\_Year', 'Tenure', 'Children', 'InternetService\_DSL', 'Gender\_Male',  
 'OnlineSecurity', 'DeviceProtection', 'StreamingTV', 'Income',  
 'Timely\_Fixes', 'Timely\_Replacements', 'MonthlyCharge','Active\_Listening']]  
plt.figure(figsize=(10,6))  
sns.heatmap(reduced\_df.corr(), annot=True, fmt=".2f", cmap='coolwarm')  
plt.show()



Upon looking at the heatmap, the author determined there was a very high correlation between Tenure and Bandwidth.

### D3: Reduced Linear Regression Model

Based on the analysis in D2, the author has reduced the model to include only the positive correlated attributes. The primmary attribute, Tenure and StreamingTV, are the largest one, showing a 0.99 in our correlation chart and a coefficient of 82 in our previously fitted model (for Tenure). For good measure, the author decided to include most of the positively correlated attributes to verify the R-squared result.

#### Significance of the Coefficients

The **coefficients** are examined as they signify the extent of impact on the dependent variable from that of the of each of the independent variables.

Below is the creation of the redcued linear model with these attributes.

# Reduced linear regression model  
reduced\_independent = df[['Tenure','MonthlyCharge','Children', 'StreamingTV']]  
reduced\_dataframe = sm.OLS(df['Bandwidth\_GB\_Year'], reduced\_independent).fit()  
print(reduced\_dataframe.summary())

OLS Regression Results   
=======================================================================================  
Dep. Variable: Bandwidth\_GB\_Year R-squared (uncentered): 0.997  
Model: OLS Adj. R-squared (uncentered): 0.997  
Method: Least Squares F-statistic: 7.507e+05  
Date: Sun, 04 Jun 2023 Prob (F-statistic): 0.00  
Time: 23:51:36 Log-Likelihood: -68675.  
No. Observations: 10000 AIC: 1.374e+05  
Df Residuals: 9996 BIC: 1.374e+05  
Df Model: 4   
Covariance Type: nonrobust   
=================================================================================  
 coef std err t P>|t| [0.025 0.975]  
---------------------------------------------------------------------------------  
Tenure 81.9394 0.084 973.562 0.000 81.774 82.104  
MonthlyCharge 2.5339 0.029 87.463 0.000 2.477 2.591  
Children 31.6861 1.056 30.004 0.000 29.616 33.756  
StreamingTV 122.0037 5.156 23.660 0.000 111.896 132.111  
==============================================================================  
Omnibus: 3847.007 Durbin-Watson: 1.972  
Prob(Omnibus): 0.000 Jarque-Bera (JB): 792.357  
Skew: 0.430 Prob(JB): 8.75e-173  
Kurtosis: 1.922 Cond. No. 402.  
==============================================================================  
  
Notes:  
[1] R² is computed without centering (uncentered) since the model does not contain a constant.  
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

As we can see in the above OLS Regression Results, the R-squared is still 0.997 (compared to 0.998 in our full dataset). So while we have eliminated almost all of the previous attributes, we have retained the correlation almost exactly, showing the strong correlation between Tenure and bandwidth used (StreamingTV was also a large coefficient, but also had a much higher standard error).

### E1: Data Analysis

The author's initial multiple linear regression model, given above, included a large number of predictors from the initial dataset. The author begain with this model with the goal of understanding the inter-relationship between the various attributes which had a potential to drive the dependent attribute, Bandwidth\_GB\_Year. By using this large a number of attributes, though, increases the likelihood of overfitting the data and not being able to be a proper predictor with future data, thus making it a poor model.

To remedy this, the author reduced negatively correlated attributes and only included positively correlated ones. This reduced dataset would reduce the complexity of our model and, thus, reduce the chance of overfitting. The author accomplished this by comparing coefficients from the regression model, p-values, and correlation values. Promising candidates for attribute removal were low coefficients, high p-values, and low/negative correlation values.

This reduced model became simpler and more likely to fit unknown data. As a final check, the R-squared value of the original- and reduced-model were compared and found to be statistically similar while reducing the model size. In both linear regression models, the R-squared models were 0.997 or higher (illustrating a higher probability of more variability in the data).

### E2: Output of all calculations

Below is the data summary from both the full dataset model and the reduced dataset model.

print("Multiple Linear Regression Summary - Full Set")  
print(results.summary())  
print("\n\nReduced Linear Regression Summary - Reduced Set")  
print(reduced\_dataframe.summary())

Multiple Linear Regression Summary - Full Set  
 OLS Regression Results   
==============================================================================  
Dep. Variable: Bandwidth\_GB\_Year R-squared: 0.008  
Model: OLS Adj. R-squared: 0.007  
Method: Least Squares F-statistic: 10.07  
Date: Sun, 04 Jun 2023 Prob (F-statistic): 4.29e-14  
Time: 23:52:45 Log-Likelihood: -91044.  
No. Observations: 10000 AIC: 1.821e+05  
Df Residuals: 9991 BIC: 1.822e+05  
Df Model: 8   
Covariance Type: nonrobust   
==========================================================================================================  
 coef std err t P>|t| [0.025 0.975]  
----------------------------------------------------------------------------------------------------------  
Gender\_Female 3.944e+15 3.39e+15 1.162 0.245 -2.71e+15 1.06e+16  
Gender\_Male 3.944e+15 3.39e+15 1.162 0.245 -2.71e+15 1.06e+16  
Gender\_Nonbinary 3.944e+15 3.39e+15 1.162 0.245 -2.71e+15 1.06e+16  
Contract\_Month-to-month 2.884e+15 2.48e+15 1.162 0.245 -1.98e+15 7.75e+15  
Contract\_One year 2.884e+15 2.48e+15 1.162 0.245 -1.98e+15 7.75e+15  
Contract\_Two Year 2.884e+15 2.48e+15 1.162 0.245 -1.98e+15 7.75e+15  
InternetService\_DSL -9.172e+15 7.89e+15 -1.162 0.245 -2.46e+16 6.3e+15  
InternetService\_Fiber Optic -9.172e+15 7.89e+15 -1.162 0.245 -2.46e+16 6.3e+15  
PaymentMethod\_Bank Transfer(automatic) 2.343e+15 2.02e+15 1.162 0.245 -1.61e+15 6.3e+15  
PaymentMethod\_Credit Card (automatic) 2.343e+15 2.02e+15 1.162 0.245 -1.61e+15 6.3e+15  
PaymentMethod\_Electronic Check 2.343e+15 2.02e+15 1.162 0.245 -1.61e+15 6.3e+15  
PaymentMethod\_Mailed Check 2.343e+15 2.02e+15 1.162 0.245 -1.61e+15 6.3e+15  
==============================================================================  
Omnibus: 42569.760 Durbin-Watson: 0.182  
Prob(Omnibus): 0.000 Jarque-Bera (JB): 1239.009  
Skew: 0.067 Prob(JB): 8.97e-270  
Kurtosis: 1.281 Cond. No. 3.56e+15  
==============================================================================  
  
Notes:  
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.  
[2] The smallest eigenvalue is 1.32e-27. This might indicate that there are  
strong multicollinearity problems or that the design matrix is singular.  
  
  
Reduced Linear Regression Summary - Reduced Set  
 OLS Regression Results   
=======================================================================================  
Dep. Variable: Bandwidth\_GB\_Year R-squared (uncentered): 0.997  
Model: OLS Adj. R-squared (uncentered): 0.997  
Method: Least Squares F-statistic: 7.507e+05  
Date: Sun, 04 Jun 2023 Prob (F-statistic): 0.00  
Time: 23:52:45 Log-Likelihood: -68675.  
No. Observations: 10000 AIC: 1.374e+05  
Df Residuals: 9996 BIC: 1.374e+05  
Df Model: 4   
Covariance Type: nonrobust   
=================================================================================  
 coef std err t P>|t| [0.025 0.975]  
---------------------------------------------------------------------------------  
Tenure 81.9394 0.084 973.562 0.000 81.774 82.104  
MonthlyCharge 2.5339 0.029 87.463 0.000 2.477 2.591  
Children 31.6861 1.056 30.004 0.000 29.616 33.756  
StreamingTV 122.0037 5.156 23.660 0.000 111.896 132.111  
==============================================================================  
Omnibus: 3847.007 Durbin-Watson: 1.972  
Prob(Omnibus): 0.000 Jarque-Bera (JB): 792.357  
Skew: 0.430 Prob(JB): 8.75e-173  
Kurtosis: 1.922 Cond. No. 402.  
==============================================================================  
  
Notes:  
[1] R² is computed without centering (uncentered) since the model does not contain a constant.  
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

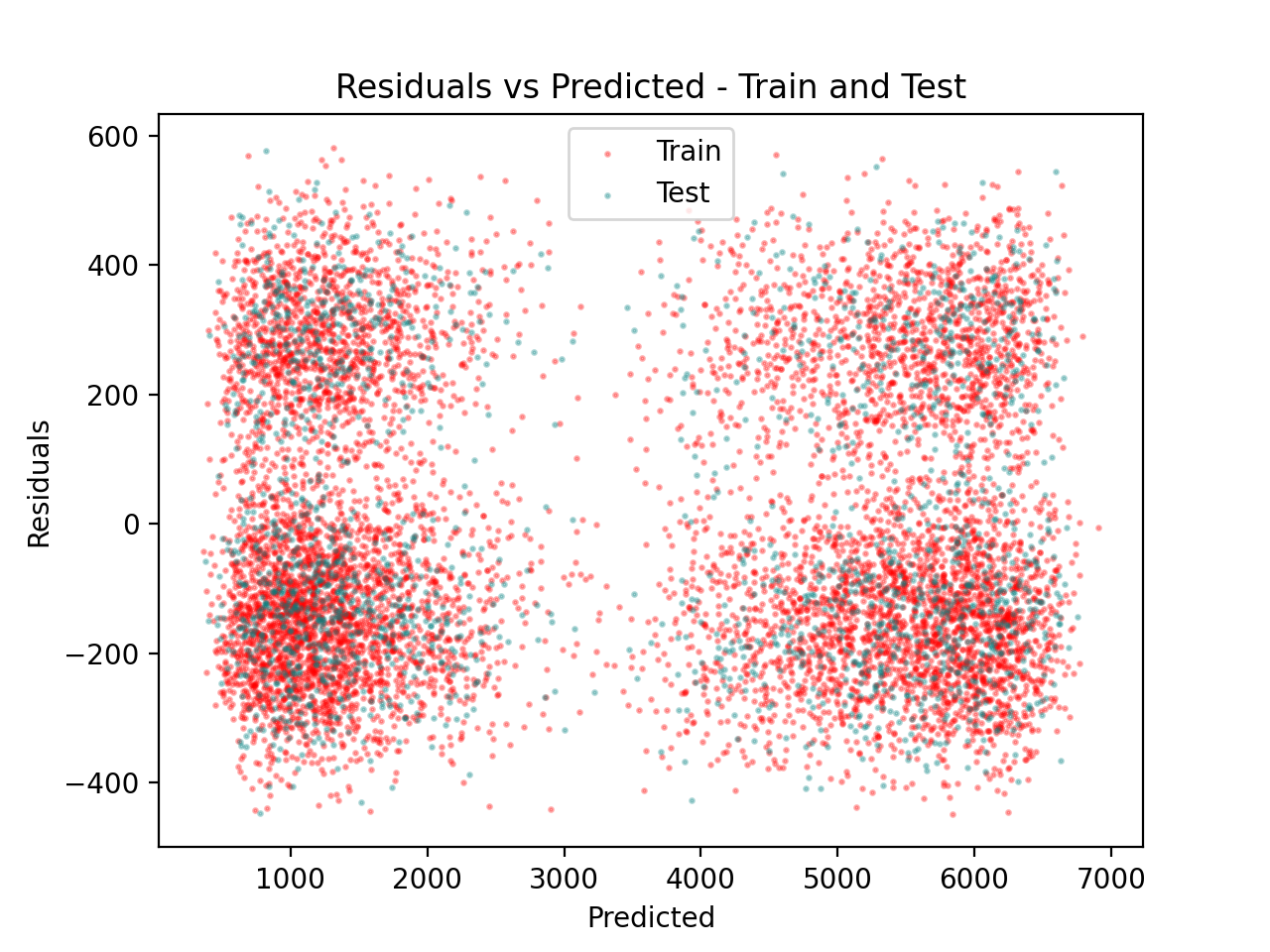
Below is a residual plot of the reduced linear regression model followed by the residual standard error:

from sklearn.model\_selection import train\_test\_split  
import statsmodels.api as sm  
  
reduced\_independent = ['Tenure','MonthlyCharge','Children', 'StreamingTV']  
  
# Training and test set creation  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(df[reduced\_independent], df['Bandwidth\_GB\_Year'], test\_size=0.2, random\_state=42)  
  
X\_train = sm.add\_constant(X\_train)  
X\_test = sm.add\_constant(X\_test)  
model = sm.OLS(y\_train, X\_train)  
results = model.fit()  
  
print(results.summary())

OLS Regression Results   
==============================================================================  
Dep. Variable: Bandwidth\_GB\_Year R-squared: 0.989  
Model: OLS Adj. R-squared: 0.989  
Method: Least Squares F-statistic: 1.755e+05  
Date: Sun, 04 Jun 2023 Prob (F-statistic): 0.00  
Time: 23:53:05 Log-Likelihood: -54928.  
No. Observations: 8000 AIC: 1.099e+05  
Df Residuals: 7995 BIC: 1.099e+05  
Df Model: 4   
Covariance Type: nonrobust   
=================================================================================  
 coef std err t P>|t| [0.025 0.975]  
---------------------------------------------------------------------------------  
const -8.7239 11.922 -0.732 0.464 -32.094 14.646  
Tenure 81.9959 0.098 835.688 0.000 81.804 82.188  
MonthlyCharge 2.5760 0.069 37.350 0.000 2.441 2.711  
Children 31.4600 1.217 25.852 0.000 29.074 33.846  
StreamingTV 120.2630 5.912 20.343 0.000 108.674 131.852  
==============================================================================  
Omnibus: 2999.085 Durbin-Watson: 2.003  
Prob(Omnibus): 0.000 Jarque-Bera (JB): 630.364  
Skew: 0.429 Prob(JB): 1.31e-137  
Kurtosis: 1.926 Cond. No. 838.  
==============================================================================  
  
Notes:  
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

from sklearn.metrics import mean\_squared\_error  
import matplotlib.pyplot as plt  
import numpy as np  
import seaborn as sns  
  
#y\_pred = results.predict(X\_test)  
#residuals = y\_test - y\_pred  
  
y\_train\_pred = results.predict(X\_train)  
y\_test\_pred = results.predict(X\_test)  
residuals\_train = y\_train - y\_train\_pred  
residuals\_test = y\_test - y\_test\_pred  
  
plt.scatter(y\_train\_pred, residuals\_train, color="red", s=2, alpha=0.3, label='Train')  
plt.scatter(y\_test\_pred, residuals\_test, color="teal", s=2, alpha=0.3, label='Test')  
plt.legend()  
plt.title("Residuals vs Predicted - Train and Test")  
plt.xlabel("Predicted")  
plt.ylabel("Residuals")  
  
mean\_squared\_error = mean\_squared\_error(y\_test, y\_test\_pred)  
p = X\_test.shape[1]  
residual\_standard\_error = np.sqrt(mean\_squared\_error / (X\_test.shape[0] - p - 1))  
print(f"Residual Standard Error: {residual\_standard\_error}")

Residual Standard Error: 5.235684194666419



### E3: Working code for linear regression models using a Python

The author has included the error-free, working code in the entire section E1 and E2 as seen above.

## Part V: Data Summmary and Implications

### F1: Summary of Findings of Data Analysis

Upon analysis of the data analysis, the author found that Tenure is the most significant correlated attribute with the amount of bandwidth used.

**A regression equation for the reduced model** would follow the equation Y = b0 + b1X1 + b2X2 + ... + bn\*Xn + e. For the specific reduced model calculated by the author, the values to substitute in would be b1 (tenure) = 81.9959. While there were three other attributes, the author is not including them as part of the equation due to their respective insignificance in the calculation. Our constant from our OLS analysis (intercept) was -8.7239. This would result in our Linear Regression equation as follows:

Bandwidth\_GB\_Year = -8.7239 + 81.9959\*Tenure

As the author **interprets the coefficients of the reduced model** (again, mostly for Tenure since the other three did not significantly impact the bandwidth), is that the Tenure coefficient of 82.8945 shows there is a large impact on bandwidth from Tenure.

The **statistical and practical significance of the reduced model** was shown to be a large reduction of the model size (and reduction in likelihood of overfitting) while the practical element shows that the linear regression equation could be drastically reduced to a single variable (Tenure).

Still, **the limitations of the data analysis** is seen in that the relationship must be a linear one. Thus, a more complicated relationship may not be able to be captured and modelled as accurately.

### F2: Recommended Course of Action Based on Analysis

The author recommends to the stakeholders work with marketers in creating a program among customers focused on bandwidth because of the positive correlation with tenure. An initial course of action could be to add a tiered program for long-term customers. For example, offering a higher-priced plan that offers higher internet speeds or higher bandwidth.

## Part VI: Demonstration

### G: Panopto Video

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=307a2900-ba18-4a9f-ad79-b018003a652a>

### H: Web Sources

"Getting Started with D208" Part I, Dr. Keiona Middleton. Webinar video. "Introduction to Predictive Analytics in Python", Nele Verbiest. Datacamp Interactive Course

### I: In-text citations and references

No specific citations were used "in-text" within this document, but both web sources from section H were heavily used as instructional content in creating the required Python code for this task.