# D208: Predictive Modeling

**Scott Sullivan | 010547531** **June 6, 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: Four assumptions regarding multiple linear regression models

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

1. **Linear Relationship**: Our variables of interest must be exampled to have a linear relationship. This will be verified with a scatterplot.
2. **Target Variable is Continuous**: We must ensure that the dependant variable (our y-axis) is continuous.
3. **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.
4. **Independence**: Our fourth assumption is to assume that the observations are independent of each other.

### B2: Python Benefits Regarding This 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: What makes multiple linear regression appropriate for this 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: Preparation of Data

### 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. One note, for the first import, the author discovered that one of the three categorical values for Internet Services was "None" (meaning the customer had no internet service). As a result, during the import, pandas incorrectly classified it as a Null value. To correct this, the author excluded "None" as a possibile value to trigger the NaN flag.

import numpy as np  
import pandas as pd  
  
# Read the CSV  
na\_values = ['NaN', 'nan', '']  
csvfile = '\_data/churn\_clean.csv'  
df = pd.read\_csv(csvfile, keep\_default\_na=False, na\_values=na\_values)  
  
# 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(2)

CaseOrder Customer\_id Interaction \  
0 1 K409198 aa90260b-4141-4a24-8e36-b04ce1f4f77b   
1 2 S120509 fb76459f-c047-4a9d-8af9-e0f7d4ac2524   
  
 UID City State County \  
0 e885b299883d4f9fb18e39c75155d990 Point Baker AK Prince of Wales-Hyder   
1 f2de8bef964785f41a2959829830fb8a West Branch MI Ogemaw   
  
 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   
  
 Item2 Item3 Item4 Item5 Item6 Item7 Item8   
0 5 5 3 4 4 3 4   
1 4 3 3 4 3 4 4   
  
[2 rows x 50 columns]

#### Duplicates

We now examine the dataframe for duplicates:

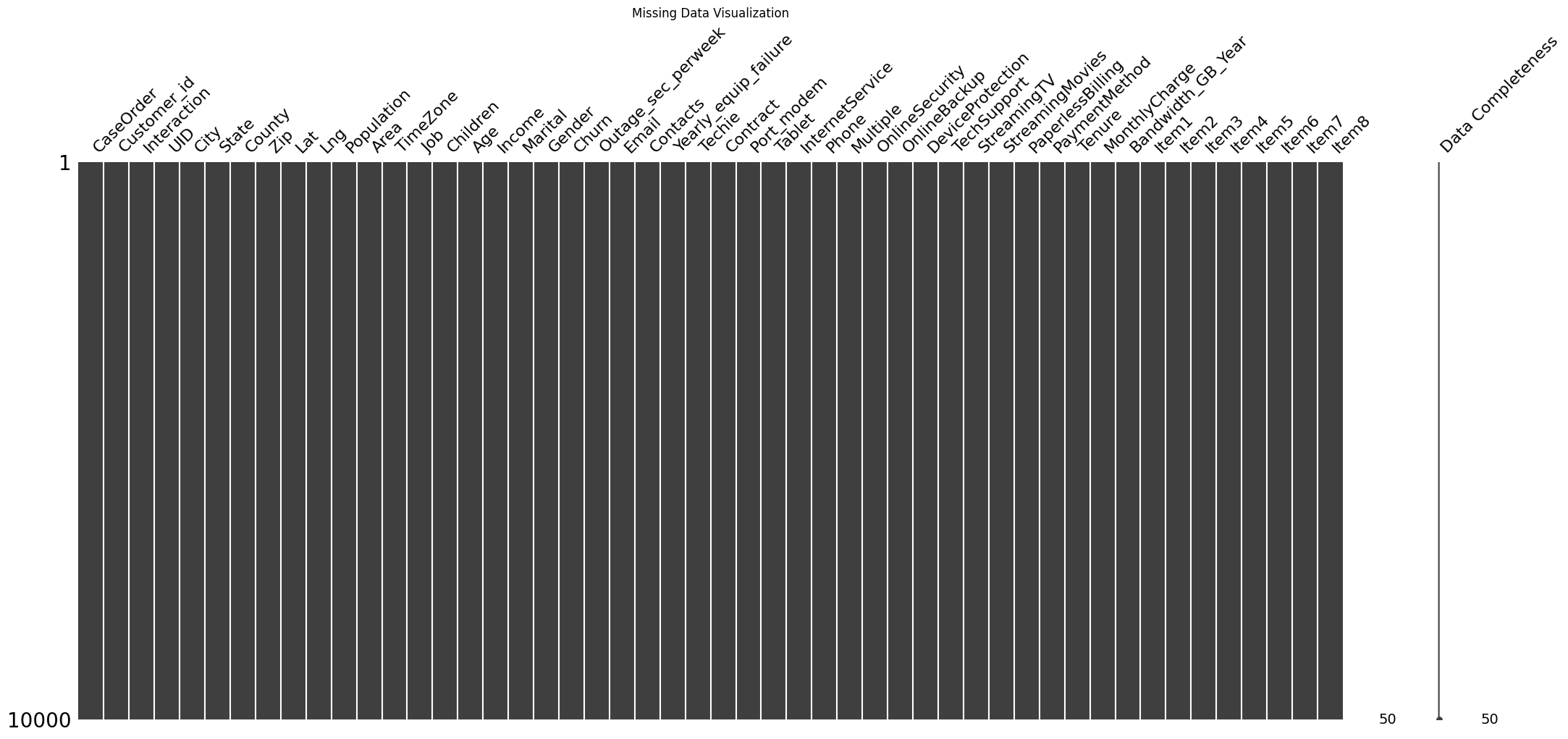
num\_duplicates = len(df) - len(df.duplicated())  
print(f"Duplicates: {num\_duplicates} of {len(df)} records.\n")

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()



As mentioned in the initial part of this document, the author had to import the CSV file using a special import which excluded "None" from appearing as a Null value. Initially, it appeared as if there were a significant number of null values for this attribute, but upon closer examplination, the realization was apparent that the value was, in fact, a third categorical option of "None."

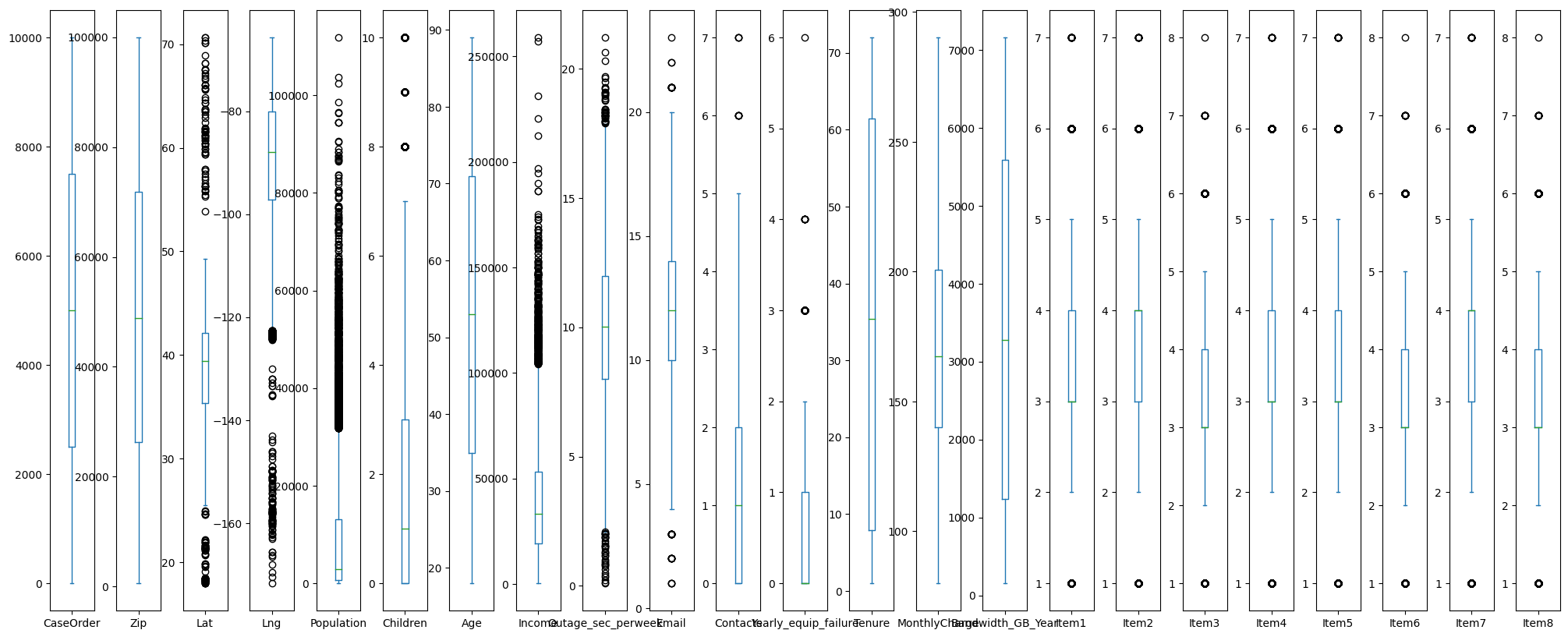
# Verify there are no null values  
df.isnull().sum()

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:  
 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(2))

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   
  
 UID City State County \  
0 e885b299883d4f9fb18e39c75155d990 Point Baker AK Prince of Wales-Hyder   
1 f2de8bef964785f41a2959829830fb8a West Branch MI Ogemaw   
  
 Zip Lat Lng ... Contract\_Month-to-month Contract\_One year \  
0 99927 56.25100 -133.37571 ... False True   
1 48661 44.32893 -84.24080 ... True False   
  
 Contract\_Two Year InternetService\_DSL InternetService\_Fiber Optic \  
0 False False True   
1 False False True   
  
 InternetService\_None PaymentMethod\_Bank Transfer(automatic) \  
0 False False   
1 False True   
  
 PaymentMethod\_Credit Card (automatic) PaymentMethod\_Electronic Check \  
0 True False   
1 False False   
  
 PaymentMethod\_Mailed Check   
0 False   
1 False   
  
[2 rows x 59 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.

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

Updating: Techie   
Updating: Churn   
Updating: Port\_modem   
Updating: Phone   
Updating: Tablet   
Updating: Multiple   
Updating: OnlineSecurity   
Updating: DeviceProtection   
Updating: OnlineBackup   
Updating: StreamingTV   
Updating: TechSupport   
Updating: StreamingMovies   
Updating: 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 ... Contract\_Month-to-month \  
0 1 0 1 ... False   
1 1 1 0 ... True   
2 1 1 1 ... False   
3 0 1 0 ... False   
4 1 0 1 ... 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 InternetService\_None \  
0 True False   
1 True False   
2 False False   
3 False False   
4 True False   
  
 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 44 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   
  
InternetService\_None bool 0   
  
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 (see section A1) will be "Bandwidth\_GB\_Year", 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, while at the same time, dropping the attributes that are administrative and would not have any impact on bandwidth. For example, "CaseOrder", "Customer\_id", etc. are removed as attributes.

Once those attributes are removed, the following attributes are used as independent variables: 'Children': number of children of the customer, 'Age': age of the customer, 'Churn': customers who left in the last month, 'Income': income of the customer (annual), 'Outage\_sec\_perweek': average seconds per week the system was out, 'Email': how many emails sent to the customer (in last year), 'Contacts': how many times the customer was contacted (in last year), 'Yearly\_equip\_failure': number of customer equipment fails in the past year, 'Techie': does the customer feel techie?, 'Port\_modem': do they have a port modem?, 'Tablet': do they own an iPad/surface?, 'Phone': does the customer have phone service, 'Multiple': multple phone lines?, 'OnlineSecurity' does the customer have online security as an add-on service, 'DeviceProtection': did they sign up for device protection service?, 'OnlineBackup': did the customer have online backup service, 'TechSupport': whether the customer has signed up for tech support service, 'StreamingTV': did the customer sign up for streaming tv, 'StreamingMovies': did they sign up for streaming movies, 'PaperlessBilling': did they opt into paperless billing, 'Tenure': how many months has the customer been with the company, 'MonthlyCharge': amount charged each month, 'Timely\_Response': from survey, 'Timely\_Fixes': from survey, 'Timely\_Replacements': from survey, 'Reliability': from survey, 'Options': from survey, 'Respectful': from survey, 'Courteous',: from survey 'Active\_Listening': from survey. 'Gender': customer's gender, 'Contract': the customer's contract term (monthly, annual, or biannual)

Below is the code and summary statistic describing all of the independent variables that are necessary for answering the research question from A1.

# Independent variables  
ind\_vars = ['Children', 'Age', 'Churn', 'Income', 'Outage\_sec\_perweek', 'Email', 'Contacts',   
 'Yearly\_equip\_failure', 'Techie', 'Port\_modem', 'Tablet', 'Phone', 'Multiple',   
 'OnlineSecurity', 'DeviceProtection', 'OnlineBackup', 'TechSupport', 'StreamingTV',  
 'StreamingMovies', 'PaperlessBilling', 'Tenure', 'MonthlyCharge',  
 'Timely\_Response', 'Timely\_Fixes', 'Timely\_Replacements', 'Reliability', 'Options',  
 'Respectful', 'Courteous', 'Active\_Listening']   
bool\_vars = ['Gender\_Female', 'Gender\_Male', 'Gender\_Nonbinary',  
 'Contract\_Month-to-month', 'Contract\_One year','Contract\_Two Year',  
 'InternetService\_DSL','InternetService\_Fiber Optic',  
 'InternetService\_None', 'PaymentMethod\_Bank Transfer(automatic)',  
 'PaymentMethod\_Credit Card (automatic)','PaymentMethod\_Electronic Check',  
 'PaymentMethod\_Mailed Check']  
vars = ind\_vars + bool\_vars  
import plotnine as p9  
  
# Convert boolean values to integers for model fitting later  
for var in vars:  
 if var in bool\_vars:  
 print(var + " Is boolean")  
 df[var] = df[var].astype('int64')  
  
print(df[vars].describe())

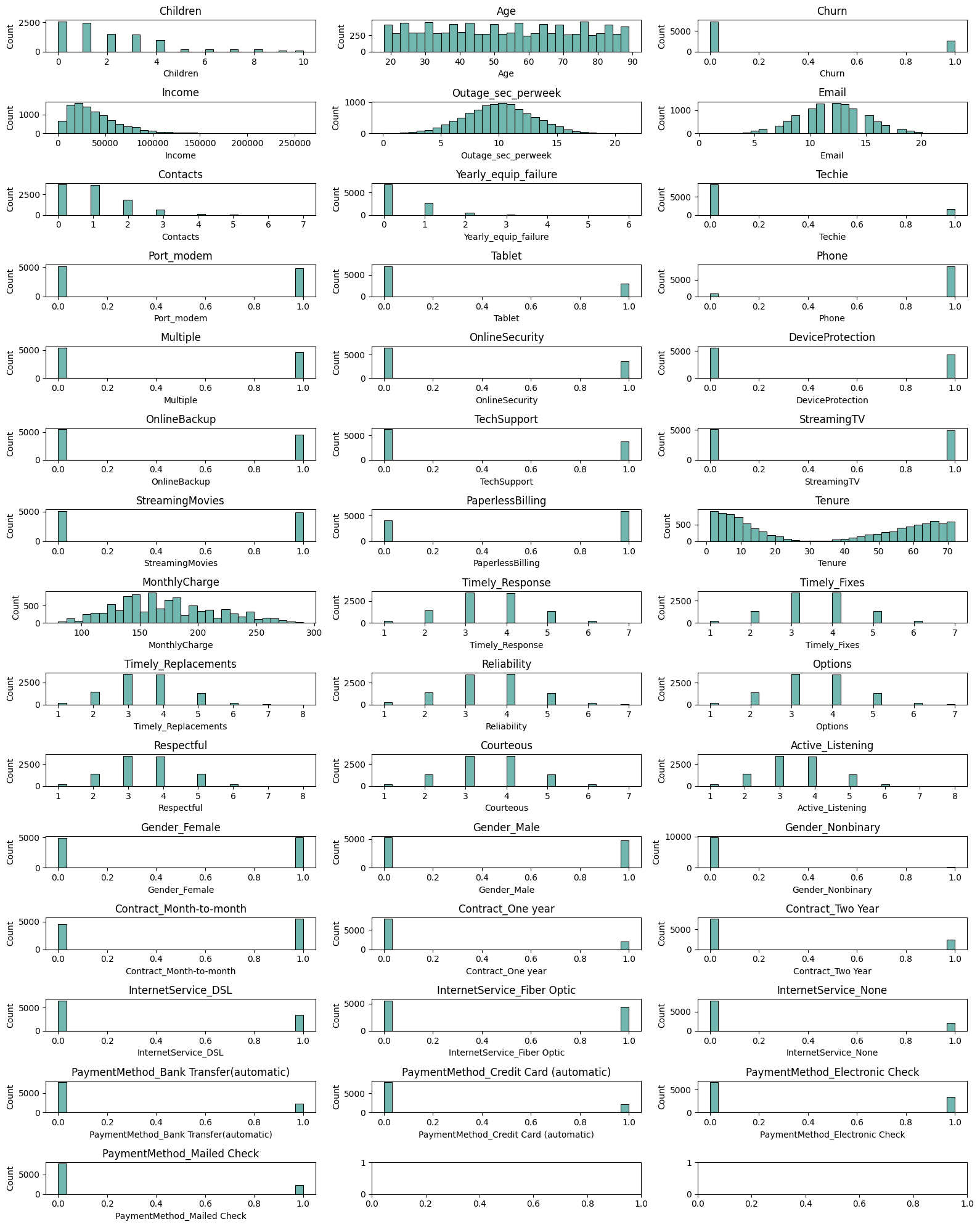
Gender\_Female Is boolean  
Gender\_Male Is boolean  
Gender\_Nonbinary Is boolean  
Contract\_Month-to-month Is boolean  
Contract\_One year Is boolean  
Contract\_Two Year Is boolean  
InternetService\_DSL Is boolean  
InternetService\_Fiber Optic Is boolean  
InternetService\_None Is boolean  
PaymentMethod\_Bank Transfer(automatic) Is boolean  
PaymentMethod\_Credit Card (automatic) Is boolean  
PaymentMethod\_Electronic Check Is boolean  
PaymentMethod\_Mailed Check Is boolean  
 Children Age Churn Income \  
count 10000.0000 10000.000000 10000.000000 10000.000000   
mean 2.0877 53.078400 0.265000 39806.926771   
std 2.1472 20.698882 0.441355 28199.916702   
min 0.0000 18.000000 0.000000 348.670000   
25% 0.0000 35.000000 0.000000 19224.717500   
50% 1.0000 53.000000 0.000000 33170.605000   
75% 3.0000 71.000000 1.000000 53246.170000   
max 10.0000 89.000000 1.000000 258900.700000   
  
 Outage\_sec\_perweek Email Contacts Yearly\_equip\_failure \  
count 10000.000000 10000.000000 10000.000000 10000.000000   
mean 10.001848 12.016000 0.994200 0.398000   
std 2.976019 3.025898 0.988466 0.635953   
min 0.099747 1.000000 0.000000 0.000000   
25% 8.018214 10.000000 0.000000 0.000000   
50% 10.018560 12.000000 1.000000 0.000000   
75% 11.969485 14.000000 2.000000 1.000000   
max 21.207230 23.000000 7.000000 6.000000   
  
 Techie Port\_modem ... Contract\_Month-to-month \  
count 10000.000000 10000.000000 ... 10000.000000   
mean 0.167900 0.483400 ... 0.545600   
std 0.373796 0.499749 ... 0.497941   
min 0.000000 0.000000 ... 0.000000   
25% 0.000000 0.000000 ... 0.000000   
50% 0.000000 0.000000 ... 1.000000   
75% 0.000000 1.000000 ... 1.000000   
max 1.000000 1.000000 ... 1.000000   
  
 Contract\_One year Contract\_Two Year InternetService\_DSL \  
count 10000.000000 10000.000000 10000.000000   
mean 0.210200 0.244200 0.346300   
std 0.407471 0.429633 0.475814   
min 0.000000 0.000000 0.000000   
25% 0.000000 0.000000 0.000000   
50% 0.000000 0.000000 0.000000   
75% 0.000000 0.000000 1.000000   
max 1.000000 1.000000 1.000000   
  
 InternetService\_Fiber Optic InternetService\_None \  
count 10000.000000 10000.000000   
mean 0.440800 0.212900   
std 0.496508 0.409378   
min 0.000000 0.000000   
25% 0.000000 0.000000   
50% 0.000000 0.000000   
75% 1.000000 0.000000   
max 1.000000 1.000000   
  
 PaymentMethod\_Bank Transfer(automatic) \  
count 10000.000000   
mean 0.222900   
std 0.416213   
min 0.000000   
25% 0.000000   
50% 0.000000   
75% 0.000000   
max 1.000000   
  
 PaymentMethod\_Credit Card (automatic) PaymentMethod\_Electronic Check \  
count 10000.000000 10000.000000   
mean 0.208300 0.339800   
std 0.406113 0.473665   
min 0.000000 0.000000   
25% 0.000000 0.000000   
50% 0.000000 0.000000   
75% 0.000000 1.000000   
max 1.000000 1.000000   
  
 PaymentMethod\_Mailed Check   
count 10000.00000   
mean 0.22900   
std 0.42021   
min 0.00000   
25% 0.00000   
50% 0.00000   
75% 0.00000   
max 1.00000   
  
[8 rows x 43 columns]

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

The univariate and bivariate visualizations were created by breaking down each into a separate list and then graphic with histograms or scatterplots.

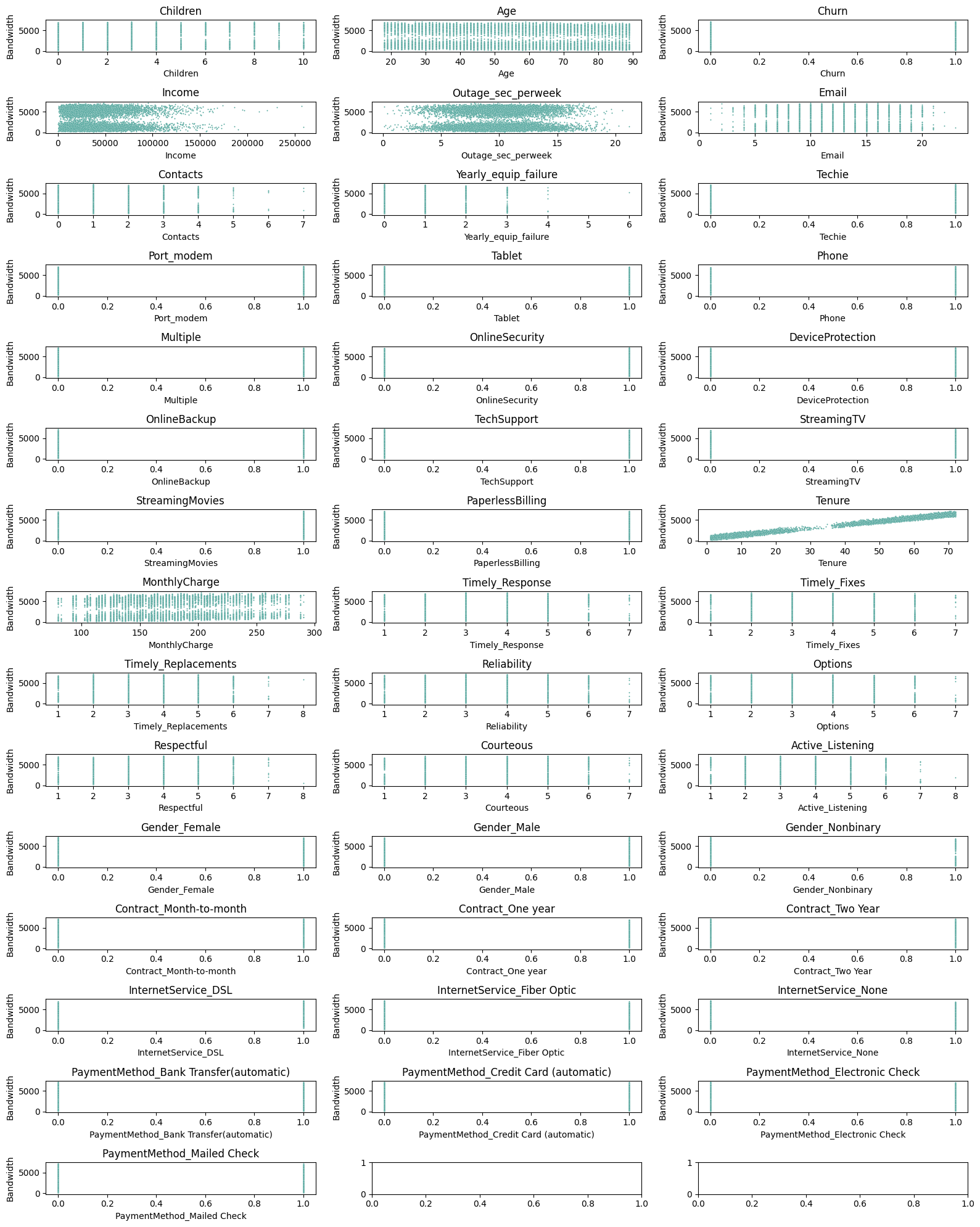
import plotnine as p9  
import seaborn as sns  
import matplotlib.pyplot as plt  
  
# UNIVARIATE Histograms Plotting  
print("Univariate Histograms\n")  
  
variables = vars  
fig, axs = plt.subplots(nrows=15, ncols=3, figsize=(16,20))  
for ax, var in zip(axs.flatten(), variables):  
 sns.histplot(data=df, 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 Histograms



import plotnine as p9  
import seaborn as sns  
import matplotlib.pyplot as plt  
  
# BIVARIATE Histograms  
print("Bivariate Visualizations\n")  
variables = vars  
fig, axs = plt.subplots(nrows=15, ncols=3, 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 / Analysis

### D1: Initial Multiple Linear Regression Model (Using All Independent Variables from C2)

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 statistic evaluation. The author will use p-values as a criterion for feature selection and will use a normally used threshold where p < 0.05.

df.dtypes

Children int64  
Age int64  
Income float64  
Churn int64  
Outage\_sec\_perweek float64  
Email int64  
Contacts int64  
Yearly\_equip\_failure int64  
Techie int64  
Port\_modem int64  
Tablet int64  
Phone int64  
Multiple int64  
OnlineSecurity int64  
OnlineBackup int64  
DeviceProtection int64  
TechSupport int64  
StreamingTV int64  
StreamingMovies int64  
PaperlessBilling int64  
Tenure float64  
MonthlyCharge float64  
Bandwidth\_GB\_Year float64  
Timely\_Response int64  
Timely\_Fixes int64  
Timely\_Replacements int64  
Reliability int64  
Options int64  
Respectful int64  
Courteous int64  
Active\_Listening int64  
Gender\_Female int64  
Gender\_Male int64  
Gender\_Nonbinary int64  
Contract\_Month-to-month int64  
Contract\_One year int64  
Contract\_Two Year int64  
InternetService\_DSL int64  
InternetService\_Fiber Optic int64  
InternetService\_None int64  
PaymentMethod\_Bank Transfer(automatic) int64  
PaymentMethod\_Credit Card (automatic) int64  
PaymentMethod\_Electronic Check int64  
PaymentMethod\_Mailed Check int64  
dtype: object

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

OLS Regression Results   
==============================================================================  
Dep. Variable: Bandwidth\_GB\_Year R-squared: 1.000  
Model: OLS Adj. R-squared: 1.000  
Method: Least Squares F-statistic: 1.579e+07  
Date: Tue, 06 Jun 2023 Prob (F-statistic): 0.00  
Time: 19:17:12 Log-Likelihood: -35923.  
No. Observations: 10000 AIC: 7.193e+04  
Df Residuals: 9960 BIC: 7.222e+04  
Df Model: 39   
Covariance Type: nonrobust   
==========================================================================================================  
 coef std err t P>|t| [0.025 0.975]  
----------------------------------------------------------------------------------------------------------  
Children 30.7688 0.041 748.801 0.000 30.688 30.849  
Age -3.2761 0.004 -768.266 0.000 -3.284 -3.268  
Churn -0.2511 0.280 -0.898 0.369 -0.799 0.297  
Income -1.225e-06 3.13e-06 -0.391 0.695 -7.36e-06 4.91e-06  
Outage\_sec\_perweek -0.0237 0.030 -0.801 0.423 -0.082 0.034  
Email 0.0053 0.029 0.181 0.856 -0.052 0.062  
Contacts 0.0500 0.089 0.560 0.576 -0.125 0.225  
Yearly\_equip\_failure -0.0129 0.139 -0.093 0.926 -0.285 0.259  
Techie -0.0031 0.237 -0.013 0.990 -0.467 0.461  
Port\_modem -0.2192 0.176 -1.243 0.214 -0.565 0.127  
Tablet -0.0543 0.193 -0.282 0.778 -0.432 0.324  
Phone -0.1416 0.304 -0.466 0.641 -0.737 0.453  
Multiple -21.9977 0.372 -59.134 0.000 -22.727 -21.269  
OnlineSecurity 68.1127 0.186 365.869 0.000 67.748 68.478  
DeviceProtection 48.9096 0.218 224.576 0.000 48.483 49.336  
OnlineBackup 29.0958 0.288 101.108 0.000 28.532 29.660  
TechSupport -31.5559 0.222 -142.276 0.000 -31.991 -31.121  
StreamingTV 106.3829 0.458 232.033 0.000 105.484 107.282  
StreamingMovies 59.2110 0.554 106.933 0.000 58.126 60.296  
PaperlessBilling 0.3060 0.179 1.707 0.088 -0.045 0.657  
Tenure 81.9322 0.004 2.04e+04 0.000 81.924 81.940  
MonthlyCharge 2.8854 0.010 285.273 0.000 2.866 2.905  
Timely\_Response -0.1895 0.126 -1.499 0.134 -0.437 0.058  
Timely\_Fixes 0.0599 0.118 0.506 0.613 -0.172 0.292  
Timely\_Replacements -0.0888 0.109 -0.817 0.414 -0.302 0.124  
Reliability -0.0376 0.097 -0.388 0.698 -0.228 0.153  
Options 0.0498 0.101 0.494 0.621 -0.148 0.247  
Respectful 0.1046 0.104 1.008 0.314 -0.099 0.308  
Courteous 0.0874 0.098 0.890 0.373 -0.105 0.280  
Active\_Listening -0.0057 0.093 -0.061 0.951 -0.189 0.177  
Gender\_Female -2.0670 0.398 -5.188 0.000 -2.848 -1.286  
Gender\_Male 62.8498 0.396 158.754 0.000 62.074 63.626  
Gender\_Nonbinary -23.5261 0.565 -41.669 0.000 -24.633 -22.419  
Contract\_Month-to-month 12.5278 0.374 33.491 0.000 11.795 13.261  
Contract\_One year 12.4098 0.392 31.631 0.000 11.641 13.179  
Contract\_Two Year 12.3192 0.391 31.512 0.000 11.553 13.086  
InternetService\_DSL 295.2343 0.368 802.968 0.000 294.514 295.955  
InternetService\_Fiber Optic -176.2580 0.506 -348.362 0.000 -177.250 -175.266  
InternetService\_None -81.7195 0.323 -252.918 0.000 -82.353 -81.086  
PaymentMethod\_Bank Transfer(automatic) 9.5678 0.314 30.508 0.000 8.953 10.183  
PaymentMethod\_Credit Card (automatic) 9.4027 0.317 29.671 0.000 8.781 10.024  
PaymentMethod\_Electronic Check 9.3241 0.303 30.816 0.000 8.731 9.917  
PaymentMethod\_Mailed Check 8.9622 0.313 28.658 0.000 8.349 9.575  
==============================================================================  
Omnibus: 34755.510 Durbin-Watson: 2.003  
Prob(Omnibus): 0.000 Jarque-Bera (JB): 1638.722  
Skew: 0.034 Prob(JB): 0.00  
Kurtosis: 1.018 Cond. No. 9.02e+19  
==============================================================================  
  
Notes:  
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.  
[2] The smallest eigenvalue is 2.93e-27. This might indicate that there are  
strong multicollinearity problems or that the design matrix is singular.

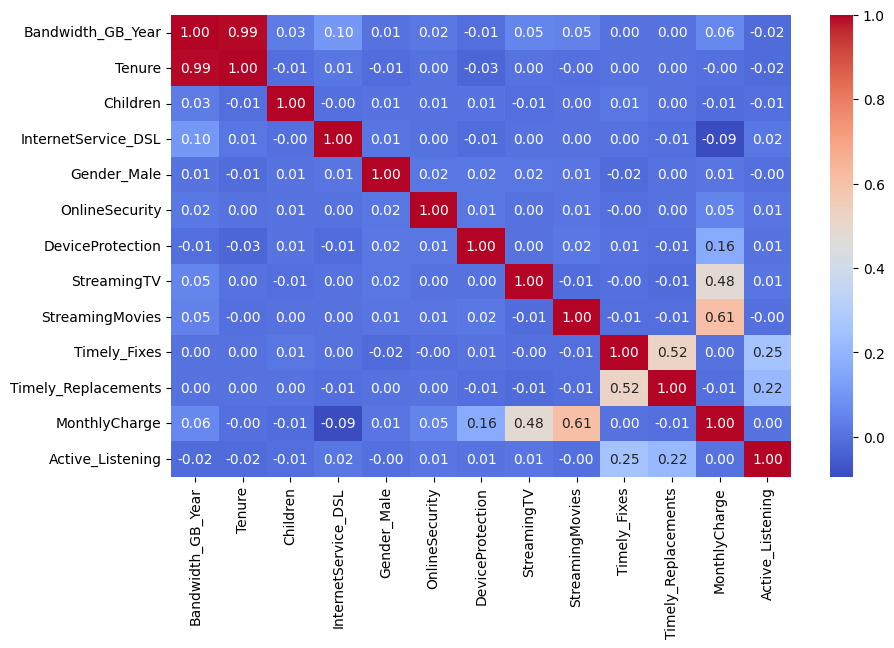
### D2: Statistically Based Model Reduction Justification

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 (68), DeviceProtection(48), StreamingTV(106), and StreamingMovies(59). InternetService seems to be extrememly high (possibly an outlier due to the one hot encoding conversion that was done on that attribute earlier.

Additionally, as mentioned previously as part of the author's criterea, attributes with a p-value that are not statistically significant (values larger than 5% or 0.05) are to be 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', 'StreamingMovies',   
 '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 higher than normal correlation that appeared 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 high in our correlation chart and a coefficient of 82 in our previously fitted model (for Tenure). For good measure, the author reduced the model to include only 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','InternetService\_DSL','Children','Gender\_Male', 'OnlineSecurity','StreamingTV','StreamingMovies','MonthlyCharge']]  
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): 0.999  
Model: OLS Adj. R-squared (uncentered): 0.999  
Method: Least Squares F-statistic: 2.455e+06  
Date: Tue, 06 Jun 2023 Prob (F-statistic): 0.00  
Time: 19:21:15 Log-Likelihood: -59297.  
No. Observations: 10000 AIC: 1.186e+05  
Df Residuals: 9992 BIC: 1.187e+05  
Df Model: 8   
Covariance Type: nonrobust   
=======================================================================================  
 coef std err t P>|t| [0.025 0.975]  
---------------------------------------------------------------------------------------  
Tenure 81.5305 0.033 2437.956 0.000 81.465 81.596  
InternetService\_DSL 413.7832 1.871 221.192 0.000 410.116 417.450  
Children 28.5299 0.416 68.501 0.000 27.714 29.346  
Gender\_Male 52.9104 1.797 29.448 0.000 49.388 56.432  
OnlineSecurity 65.5017 1.899 34.492 0.000 61.779 69.224  
StreamingTV 170.7744 2.127 80.298 0.000 166.606 174.943  
StreamingMovies 142.9372 2.237 63.886 0.000 138.551 147.323  
MonthlyCharge 1.0358 0.017 61.377 0.000 1.003 1.069  
==============================================================================  
Omnibus: 144.239 Durbin-Watson: 1.957  
Prob(Omnibus): 0.000 Jarque-Bera (JB): 86.488  
Skew: 0.044 Prob(JB): 1.66e-19  
Kurtosis: 2.553 Cond. No. 503.  
==============================================================================  
  
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 0.999. So while we have eliminated almost all of the previous attributes, we have retained the correlation almost exactly, showing the strong correlation between both MontlyCharge and Tenure and bandwidth used (InternetService\_DSL, StreamingTV, and StreamingMovies in the reduced model also had large coefficients, 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 0.999 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: 1.000  
Model: OLS Adj. R-squared: 1.000  
Method: Least Squares F-statistic: 1.579e+07  
Date: Tue, 06 Jun 2023 Prob (F-statistic): 0.00  
Time: 19:22:43 Log-Likelihood: -35923.  
No. Observations: 10000 AIC: 7.193e+04  
Df Residuals: 9960 BIC: 7.222e+04  
Df Model: 39   
Covariance Type: nonrobust   
==========================================================================================================  
 coef std err t P>|t| [0.025 0.975]  
----------------------------------------------------------------------------------------------------------  
Children 30.7688 0.041 748.801 0.000 30.688 30.849  
Age -3.2761 0.004 -768.266 0.000 -3.284 -3.268  
Churn -0.2511 0.280 -0.898 0.369 -0.799 0.297  
Income -1.225e-06 3.13e-06 -0.391 0.695 -7.36e-06 4.91e-06  
Outage\_sec\_perweek -0.0237 0.030 -0.801 0.423 -0.082 0.034  
Email 0.0053 0.029 0.181 0.856 -0.052 0.062  
Contacts 0.0500 0.089 0.560 0.576 -0.125 0.225  
Yearly\_equip\_failure -0.0129 0.139 -0.093 0.926 -0.285 0.259  
Techie -0.0031 0.237 -0.013 0.990 -0.467 0.461  
Port\_modem -0.2192 0.176 -1.243 0.214 -0.565 0.127  
Tablet -0.0543 0.193 -0.282 0.778 -0.432 0.324  
Phone -0.1416 0.304 -0.466 0.641 -0.737 0.453  
Multiple -21.9977 0.372 -59.134 0.000 -22.727 -21.269  
OnlineSecurity 68.1127 0.186 365.869 0.000 67.748 68.478  
DeviceProtection 48.9096 0.218 224.576 0.000 48.483 49.336  
OnlineBackup 29.0958 0.288 101.108 0.000 28.532 29.660  
TechSupport -31.5559 0.222 -142.276 0.000 -31.991 -31.121  
StreamingTV 106.3829 0.458 232.033 0.000 105.484 107.282  
StreamingMovies 59.2110 0.554 106.933 0.000 58.126 60.296  
PaperlessBilling 0.3060 0.179 1.707 0.088 -0.045 0.657  
Tenure 81.9322 0.004 2.04e+04 0.000 81.924 81.940  
MonthlyCharge 2.8854 0.010 285.273 0.000 2.866 2.905  
Timely\_Response -0.1895 0.126 -1.499 0.134 -0.437 0.058  
Timely\_Fixes 0.0599 0.118 0.506 0.613 -0.172 0.292  
Timely\_Replacements -0.0888 0.109 -0.817 0.414 -0.302 0.124  
Reliability -0.0376 0.097 -0.388 0.698 -0.228 0.153  
Options 0.0498 0.101 0.494 0.621 -0.148 0.247  
Respectful 0.1046 0.104 1.008 0.314 -0.099 0.308  
Courteous 0.0874 0.098 0.890 0.373 -0.105 0.280  
Active\_Listening -0.0057 0.093 -0.061 0.951 -0.189 0.177  
Gender\_Female -2.0670 0.398 -5.188 0.000 -2.848 -1.286  
Gender\_Male 62.8498 0.396 158.754 0.000 62.074 63.626  
Gender\_Nonbinary -23.5261 0.565 -41.669 0.000 -24.633 -22.419  
Contract\_Month-to-month 12.5278 0.374 33.491 0.000 11.795 13.261  
Contract\_One year 12.4098 0.392 31.631 0.000 11.641 13.179  
Contract\_Two Year 12.3192 0.391 31.512 0.000 11.553 13.086  
InternetService\_DSL 295.2343 0.368 802.968 0.000 294.514 295.955  
InternetService\_Fiber Optic -176.2580 0.506 -348.362 0.000 -177.250 -175.266  
InternetService\_None -81.7195 0.323 -252.918 0.000 -82.353 -81.086  
PaymentMethod\_Bank Transfer(automatic) 9.5678 0.314 30.508 0.000 8.953 10.183  
PaymentMethod\_Credit Card (automatic) 9.4027 0.317 29.671 0.000 8.781 10.024  
PaymentMethod\_Electronic Check 9.3241 0.303 30.816 0.000 8.731 9.917  
PaymentMethod\_Mailed Check 8.9622 0.313 28.658 0.000 8.349 9.575  
==============================================================================  
Omnibus: 34755.510 Durbin-Watson: 2.003  
Prob(Omnibus): 0.000 Jarque-Bera (JB): 1638.722  
Skew: 0.034 Prob(JB): 0.00  
Kurtosis: 1.018 Cond. No. 9.02e+19  
==============================================================================  
  
Notes:  
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.  
[2] The smallest eigenvalue is 2.93e-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): 0.999  
Model: OLS Adj. R-squared (uncentered): 0.999  
Method: Least Squares F-statistic: 2.455e+06  
Date: Tue, 06 Jun 2023 Prob (F-statistic): 0.00  
Time: 19:22:43 Log-Likelihood: -59297.  
No. Observations: 10000 AIC: 1.186e+05  
Df Residuals: 9992 BIC: 1.187e+05  
Df Model: 8   
Covariance Type: nonrobust   
=======================================================================================  
 coef std err t P>|t| [0.025 0.975]  
---------------------------------------------------------------------------------------  
Tenure 81.5305 0.033 2437.956 0.000 81.465 81.596  
InternetService\_DSL 413.7832 1.871 221.192 0.000 410.116 417.450  
Children 28.5299 0.416 68.501 0.000 27.714 29.346  
Gender\_Male 52.9104 1.797 29.448 0.000 49.388 56.432  
OnlineSecurity 65.5017 1.899 34.492 0.000 61.779 69.224  
StreamingTV 170.7744 2.127 80.298 0.000 166.606 174.943  
StreamingMovies 142.9372 2.237 63.886 0.000 138.551 147.323  
MonthlyCharge 1.0358 0.017 61.377 0.000 1.003 1.069  
==============================================================================  
Omnibus: 144.239 Durbin-Watson: 1.957  
Prob(Omnibus): 0.000 Jarque-Bera (JB): 86.488  
Skew: 0.044 Prob(JB): 1.66e-19  
Kurtosis: 2.553 Cond. No. 503.  
==============================================================================  
  
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.

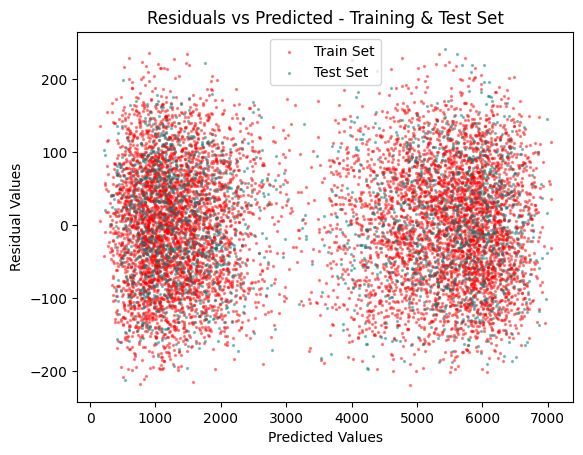
Below is a residual plot of the reduced linear regression model followed by the residual standard error:

from sklearn.model\_selection import train\_test\_split  
import statsmodels.api as sm  
  
reduced\_independent = ['Tenure','InternetService\_DSL','Children','Gender\_Male', 'OnlineSecurity','StreamingTV','StreamingMovies','MonthlyCharge']  
  
# 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()  
  
# Now that the model is created and fit, print the summary  
print(results.summary())

OLS Regression Results   
==============================================================================  
Dep. Variable: Bandwidth\_GB\_Year R-squared: 0.999  
Model: OLS Adj. R-squared: 0.999  
Method: Least Squares F-statistic: 6.933e+05  
Date: Tue, 06 Jun 2023 Prob (F-statistic): 0.00  
Time: 19:23:02 Log-Likelihood: -46698.  
No. Observations: 8000 AIC: 9.341e+04  
Df Residuals: 7991 BIC: 9.348e+04  
Df Model: 8   
Covariance Type: nonrobust   
=======================================================================================  
 coef std err t P>|t| [0.025 0.975]  
---------------------------------------------------------------------------------------  
const -214.6420 5.146 -41.709 0.000 -224.730 -204.554  
Tenure 81.8814 0.035 2333.581 0.000 81.813 81.950  
InternetService\_DSL 434.1457 1.975 219.771 0.000 430.273 438.018  
Children 32.0457 0.435 73.642 0.000 31.193 32.899  
Gender\_Male 67.7133 1.859 36.424 0.000 64.069 71.358  
OnlineSecurity 71.0637 1.945 36.529 0.000 67.250 74.877  
StreamingTV 131.8502 2.358 55.921 0.000 127.228 136.472  
StreamingMovies 90.3776 2.607 34.667 0.000 85.267 95.488  
MonthlyCharge 2.2932 0.035 65.701 0.000 2.225 2.362  
==============================================================================  
Omnibus: 209.424 Durbin-Watson: 2.005  
Prob(Omnibus): 0.000 Jarque-Bera (JB): 106.535  
Skew: 0.051 Prob(JB): 7.35e-24  
Kurtosis: 2.444 Cond. No. 1.07e+03  
==============================================================================  
  
Notes:  
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.  
[2] The condition number is large, 1.07e+03. This might indicate that there are  
strong multicollinearity or other numerical problems.

import numpy as np  
import seaborn as sns  
from sklearn.metrics import mean\_squared\_error  
import matplotlib.pyplot as plt  
  
y\_train\_pred = results.predict(X\_train)  
y\_test\_pred = results.predict(X\_test)  
res\_train = y\_train - y\_train\_pred  
res\_test = y\_test - y\_test\_pred  
  
plt.scatter(y\_train\_pred, res\_train, color="red", s=2, alpha=0.4, label='Train Set')  
plt.scatter(y\_test\_pred, res\_test, color="teal", s=2, alpha=0.4, label='Test Set')  
plt.legend()  
plt.title("Residuals vs Predicted - Training & Test Set")  
plt.xlabel("Predicted Values")  
plt.ylabel("Residual Values")  
  
mean\_squared\_err = mean\_squared\_error(y\_test, y\_test\_pred)  
p = X\_test.shape[1]  
residual\_standard\_err = np.sqrt(mean\_squared\_err / (X\_test.shape[0] - p - 1))  
print(f"Model's Residual standard error: {residual\_standard\_err}")

Model's Residual standard error: 1.8197228022913154



### 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 Along With Implications

### F1: Findings Summary 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 format: 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) = 81.88. 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 -214.6420. This would result in our Linear Regression equation as follows:

Bandwidth\_GB\_Year = -214.6 + 81.88\*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 81.88 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 Presentation

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=9e1d212a-e0fa-4e39-9f06-b019017c0cab>

### 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.