Andreas Sebastian Dunn
Western Governors University, D208: Predictive Modeling
September 15, 2021

D208, Task 1

A multiple regression analysis on churn data in the telecom industry

Part 1

The dataset used in this analysis is customer churn data from the telecom industry. Churn is defined as the percentage of customers who stopped using a provider's service within a given time frame. Customer churn is a big deal in the telecom industry as it can be up to 10 times more expensive to acquire new customers vs retaining existing ones. This analysis will therefore be focused on a dataset centered around customer demographics and churn data.

Using the following five independent variables: churn, gender, income, age, and number of children, can we predict annual bandwidth usage for our customers in the data set? If so, how well do our independent variables predict annual bandwidth usage? Our analysis will focus on this question.

The objective of this analysis is to uncover and explain the relationships between our chosen data points and their influence on one another. Mainly, how we can use our predictor variables to explain bandwidth usage using multiple regression. Multiple linear regression calculates an equation that minimizes the distance between our predicted values on the regression line and the actual X values recorded in the data. For this analysis I will be using the ordinary least squares (OLS) regression which seeks to minimize the sum of the squared differences between our predicted value of Y (the regression line) and the actual value of Y (the sampled data).

Given the scope of our available data, our choice of predictor variables are limited and therefore our model may not be perfect. However, there are some important pieces of information of which internal stakeholders can benefit.

This analysis mainly benefits stakeholders within the telecom industry whom would like to know the relationship between bandwidth usage and other collected data points which include some data about demographics and customer churn. They can use this data to make better informed decisions for their business like changing the price/GB charged, implementing new technology to increase bandwidth availability, or changing pricing tiers.

Part 2

Multiple regression using the method of least squares, OLS, is the model I will be using in this paper. The OLS model has a few assumptions about the data that are worth discussing. Firstly, OLS assumes that there exists a linear relationship between X and Y (predictor variable and response variable), X are independent of one another, and the residuals have zero mean and are normally distributed (OLS model assumes zero mean in order to estimate the regression line). OLS is one of the most powerful formulas for understanding the linear relationship between our variables and is the most used regression estimator in multiple regression analysis.

I will be using Python + libraries and Jupyter notebook to perform the OLS regression. I will be using Tableau for the data visualization and presentation. Python is the most used language for data science; it offers a great selection of statistics libraries to use for data science while also being a general purpose programming language. Tableau is one of the leaders in data visualization and is widely used in the business intelligence industry.

Since our research question involves multiple predictor variables on one response variable, we need to use a tool that can predict the regression line using multiple inputs. Multiple regression using OLS is widely considered one of the best algorithms/formulas for doing so and is the most appropriate technique given our inputs. While we can certainly use more than 5 inputs, I fear that adding more predictors will not increase the strength of this analysis and is ultimately a waste of resources.

Part 3

For data preparation, my goal is to have data cleaned and outliers removed. Firstly, I will scrub the data by looking for missing or null values and I will use measures of central tendency such as mean or median to replace any I find. After this, I will use Python to look for outliers and drop them using Z-scores. I will estimate Z-scores for our 3 numeric independent variables (income, age, children) and drop any that exceed the absolute value of 3.

Next, I will show summary statistics and change our categorical variables, gender and churn, into dummy variables to perform the regression. The results from the regression should show five independent variables: gender_dummy, churn_dummy, age, income, and children.

[1]:
```
import pandas as pd
import numpy as np
from pandas import Series, DataFrame
import pylab
from pylab import rcParams
import statsmodels.api as sm
import statistics
from scipy import stats
import sklearn
from sklearn import preprocessing
from sklearn.linear_model import LinearRegression
from sklearn import metrics
from sklearn.metrics import classification_report
```

[2]:
```
df = pd.read_csv(r'C:\Users\dre2\Desktop\WGU\D208\churn_clean.csv')
```

[3]:
```
#Drop columns we dont need, look for nulls/replace as needed.
df = df.drop(columns=['City', 'State', 'County', 'Zip', 'Lat', 'Lng',
                      'Population', 'Area', 'TimeZone', 'Job', 'Marital', 'Outage_sec_perweek', 'Email', 'Contacts', 'Yearly_equip_failure', 'Techie', 'Contract', 'Port_modem', 'Tablet',
                      'InternetService', 'Phone', 'Multiple', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies', 'PaperlessBilling',
                      'Tenure', 'MonthlyCharge', 'Item1', 'Item2', 'Item3', 'Item4', 'Item5', 'Item6', 'Item7', 'Item8', 'PaymentMethod'])
nulls = df.isnull().any()
print(nulls)
```

```
CaseOrder          False
Customer_id        False
Interaction        False
UID                False
Children           False
Age                False
Income             False
Gender             False
Churn              False
Bandwidth_GB_Year  False
dtype: bool
```

[6]:
```
#summary statistics of cleaned df
df.describe()
```

[6]:

	CaseOrder	Children	Age	Income	Bandwidth_GB_Year
count	9960.000000	9960.000000	9960.000000	9960.000000	9960.000000
mean	5001.381426	2.088454	53.097892	39261.710241	3391.961345
std	2886.590495	2.147374	20.701925	26860.824875	2185.211493
min	1.000000	0.000000	18.000000	348.000000	155.000000
25%	2500.750000	0.000000	35.000000	19166.500000	1236.000000
50%	5001.500000	1.000000	53.000000	33028.000000	3287.000000
75%	7501.250000	3.000000	71.000000	52921.250000	5585.250000
max	10000.000000	10.000000	89.000000	152172.000000	7158.000000

[7]:
```
df['Gender'].value_counts()
```

[7]:
```
Female       5006
Male         4725
Nonbinary     229
Name: Gender, dtype: int64
```

[8]:
```
df['Churn'].value_counts()
```

[8]:
```
No     7326
Yes    2634
Name: Churn, dtype: int64
```

The summary statistics (shown above) on the cleaned data show that our mean for demographic data is 53 years old, $39,074 for annual income, 2 children, and a GB/year usage of 3392. The standard deviation is 2.148 years, $26495 in annual income, 2 children, and 2185 GB/year. Inner-quartile range is 35-71 years old, $19,153-$52824 in annual income, 0-3 children, and 1236-5584 GB/year. Using the value_counts() method we can get some summary of categorical data which show that most of our responses are female (5006 vs 4725 for males vs 229 for non-binary), and most of our responses are 'No' for Churn (7326 vs 2634).

The steps taken to clean the data are as follows: I imported the required data science libraries then dropped all the X variables we do not need for our analysis, then I checked for missing or null values and found none. I proceeded to check z-scores for our non-categorical X variables and dropped any that had a value > |3| (any variables with z-scores > |3| are defined as outliers for the purposes of this analysis). Below, I show the code used to prepare the data of the analysis, plots to visually show the distribution of our variables, and the results from the OLS regression. The results from the regression should show five independent variables: gender_dummy, churn_dummy, age, income, and children.

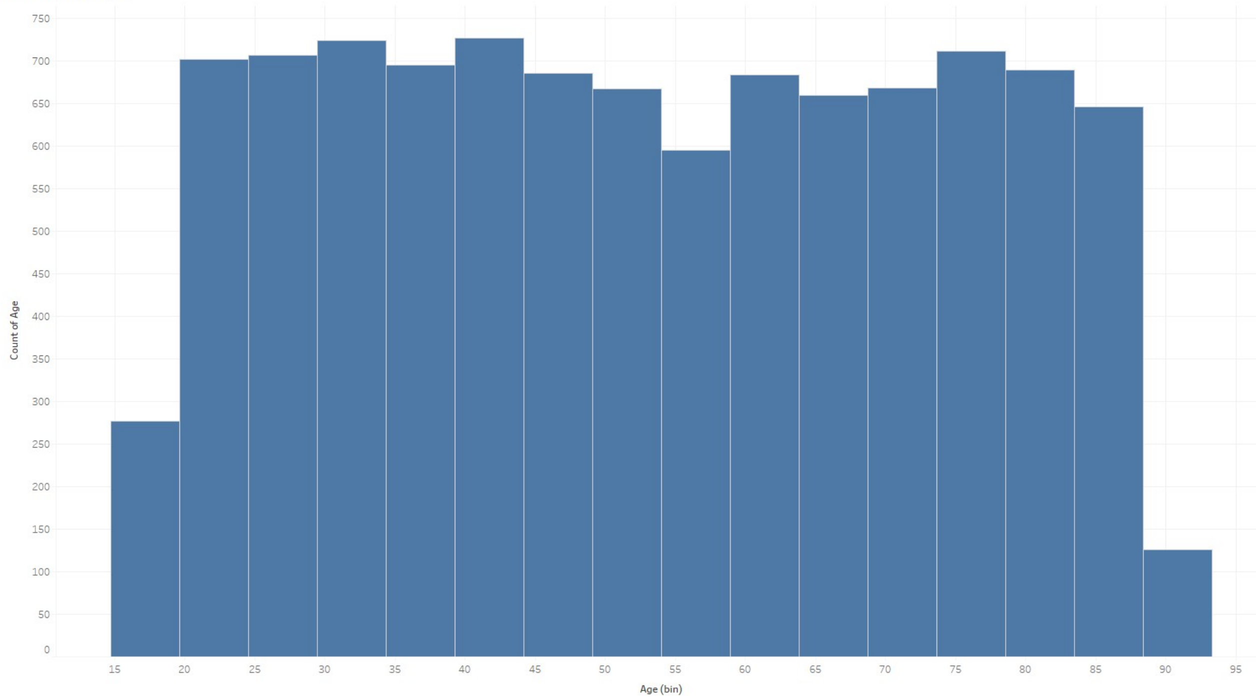
```
[4]: #Check z-scores for x-vars and drop if > |3|.
childrenZ = df.loc[ : , 'Children']
df['childrenZ'] = stats.zscore(childrenZ)

ageZ = df.loc[ : , 'Age']
df['ageZ'] = stats.zscore(ageZ)

incomeZ = df.loc[ : , 'Income']
df['incomeZ'] = stats.zscore(incomeZ)

#change data type from float to int64
df['childrenZ'] = df['childrenZ'].astype(np.int64)
df['ageZ'] = df['ageZ'].astype(np.int64)
df['incomeZ'] = df['incomeZ'].astype(np.int64)

#drop if > |3|
df.drop(df[df['childrenZ'] > 3].index, inplace = True)
df.drop(df[df['childrenZ'] < -3].index, inplace = True)
df.drop(df[df['ageZ'] > 3].index, inplace = True)
df.drop(df[df['ageZ'] < -3].index, inplace = True)
df.drop(df[df['incomeZ'] > 3].index, inplace = True)
df.drop(df[df['incomeZ'] < -3].index, inplace = True)

#delete z-score columns
df = df.drop(columns=['childrenZ', 'ageZ', 'incomeZ'])
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9960 entries, 0 to 9999
Data columns (total 10 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   CaseOrder          9960 non-null   int64
 1   Customer_id        9960 non-null   object
 2   Interaction        9960 non-null   object
 3   UID                9960 non-null   object
 4   Children           9960 non-null   int64
 5   Age                9960 non-null   int64
 6   Income             9960 non-null   float64
 7   Gender             9960 non-null   object
 8   Churn              9960 non-null   object
 9   Bandwidth_GB_Year  9960 non-null   float64
dtypes: float64(2), int64(3), object(5)
memory usage: 1.1+ MB
```

```
[5]: #Change data type of Bandwidth_GB_Year and Income to int64
df['Income'] = df['Income'].astype(np.int64)
df['Bandwidth_GB_Year'] = df['Bandwidth_GB_Year'].astype(np.int64)
df.info()
```

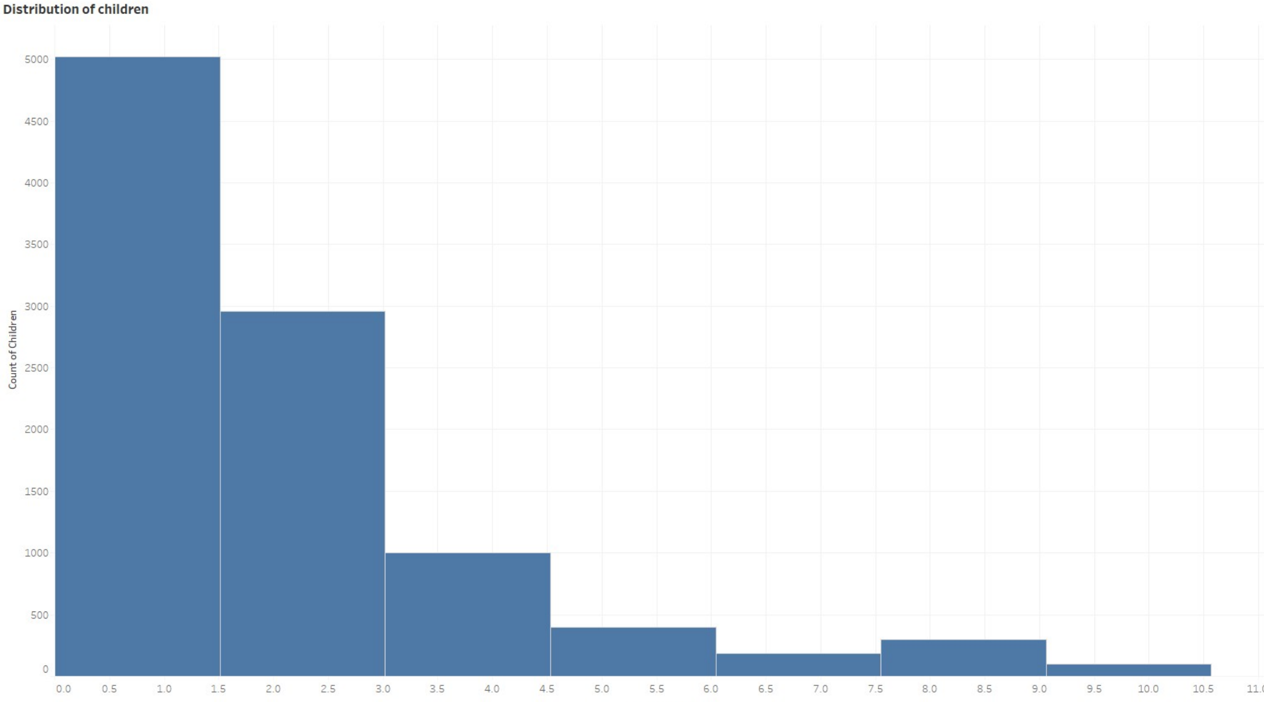
Univariate Distributions:

Distribution of age



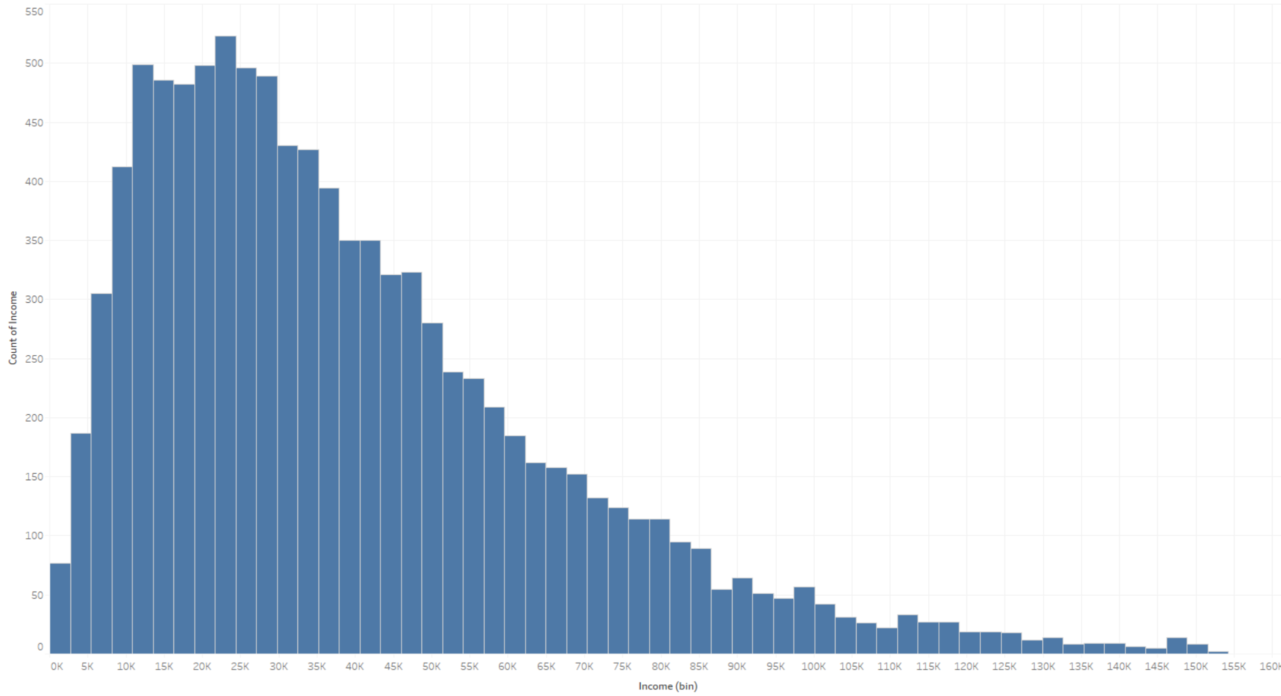
The trend of count of Age for Age (bin).

Distribution of children



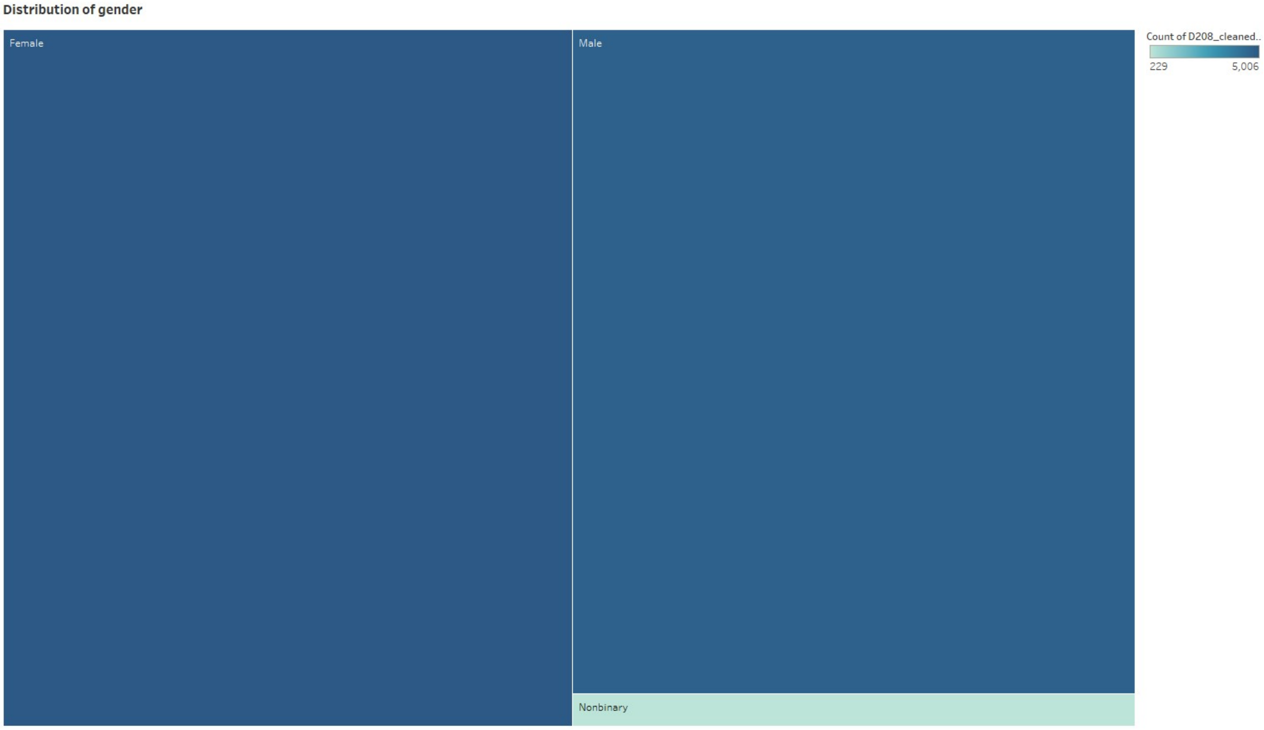
The trend of count of Children for Children (bin).

Distribution of income



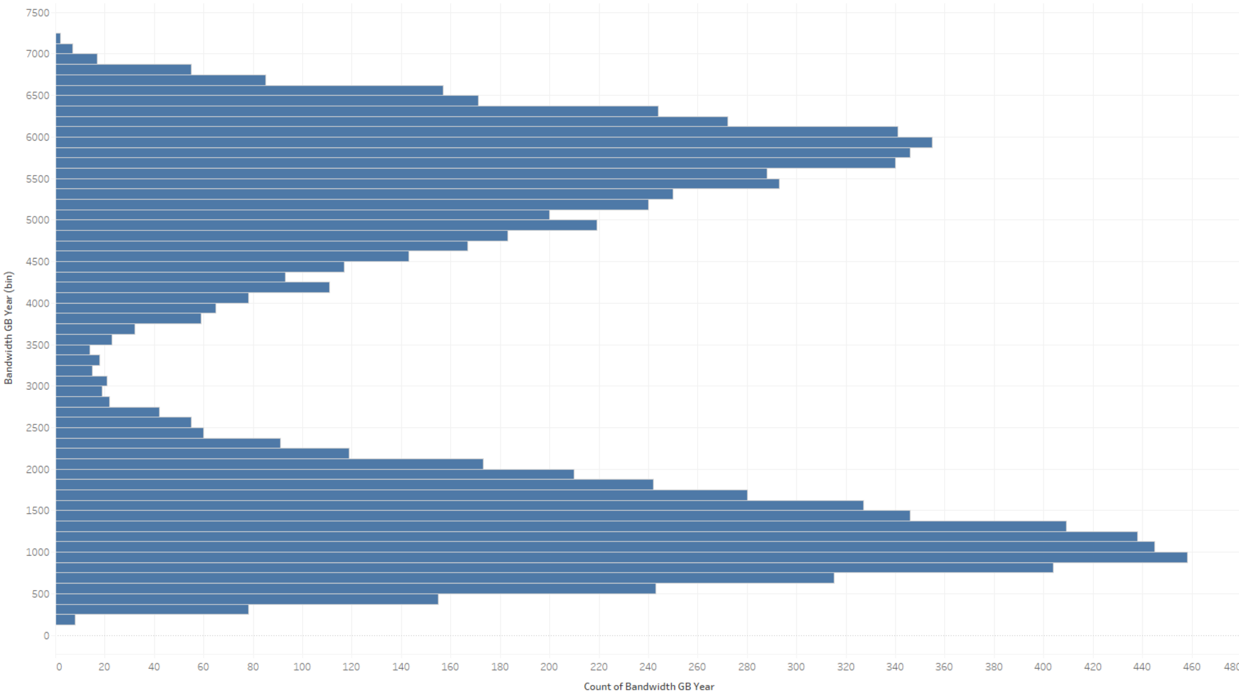
The trend of count of Income for Income (bin).

Distribution of gender



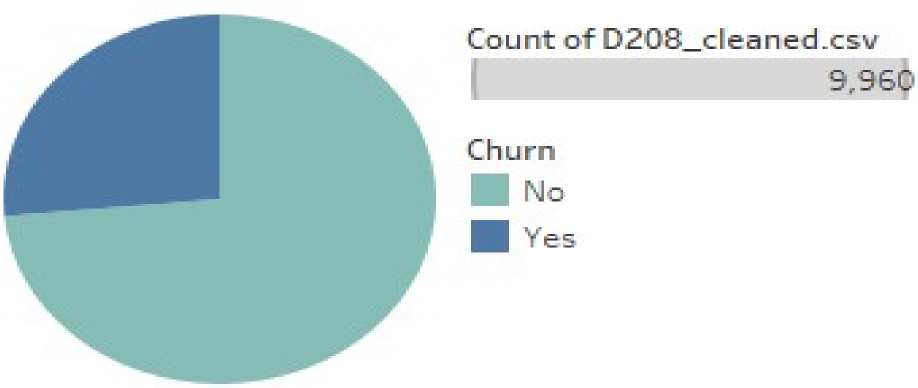
Gender. Color shows count of D208_cleaned.csv. Size shows count of D208_cleaned.csv. The marks are labeled by Gender.

Distribution of GB/year



The trend of count of Bandwidth GB Year for Bandwidth GB Year (bin).

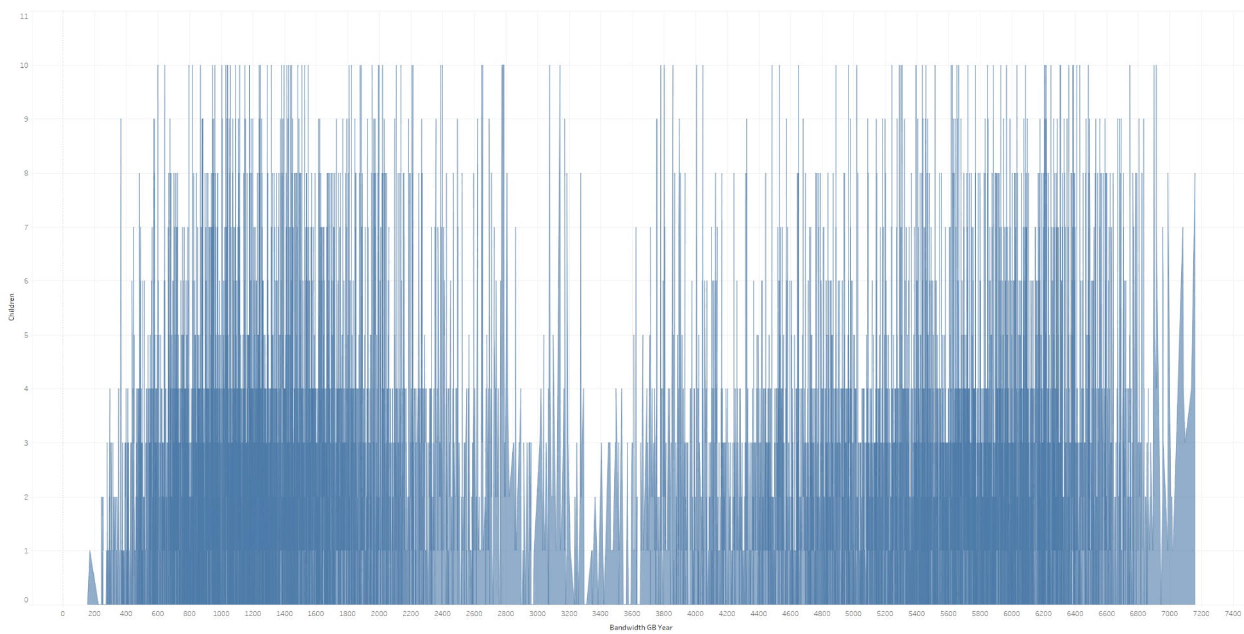
Distribution of churn



Churn (color) and count of D208_cleaned.csv (size).

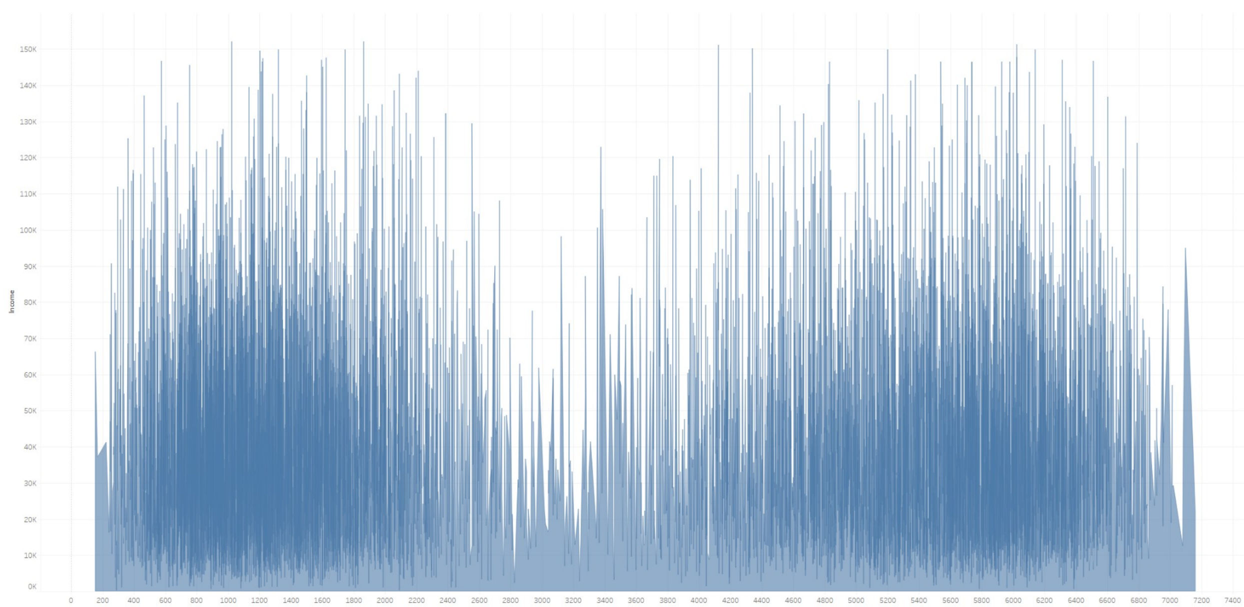
Bivariate Distributions:

Bivariate: GB/year + children



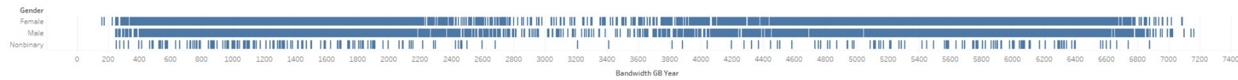
The plot of sum of Children for Bandwidth GB Year.

Bivariate: GB/year + income



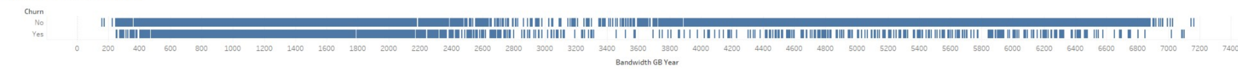
The plot of sum of Income for Bandwidth GB Year.

Bivariate: GB/year + gender



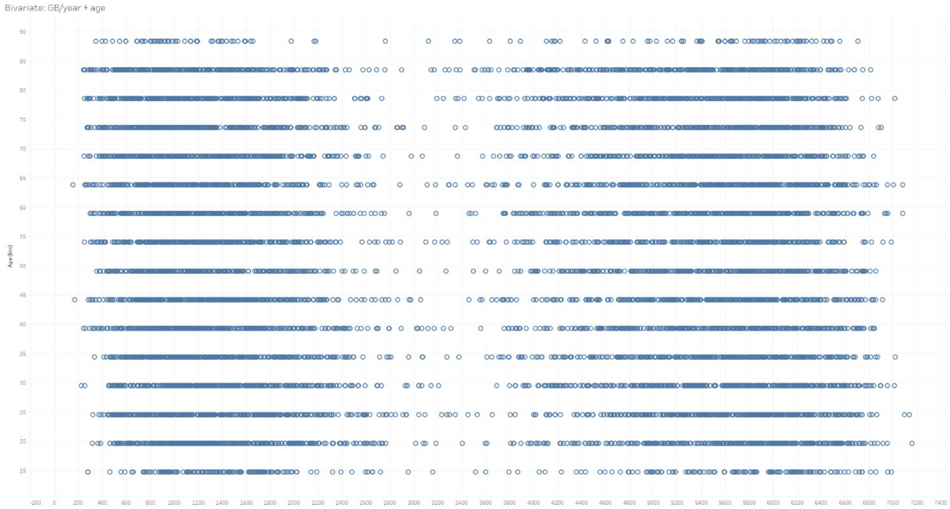
Bandwidth GB Year for each Gender.

Bivariate: churn & GB/year



Bandwidth GB Year for each Churn.

Bivariate: GB/year + age



Bandwidth GB Year vs. Age (bin).

Part 4

```
[9]: df['gender_dummy'] = [1 if v == 'Female' else 0 for v in df['Gender']]
     df['churn_dummy'] = [1 if v == 'Yes' else 0 for v in df['Churn']]

     df['intercept'] = 1
     model = sm.OLS(df['Bandwidth_GB_Year'], df[['gender_dummy', 'churn_dummy','Children', 'Age', 'Income', 'intercept']]).fit()
     print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:      Bandwidth_GB_Year   R-squared:                       0.196
Model:                            OLS   Adj. R-squared:                  0.196
Method:                 Least Squares   F-statistic:                     485.7
Date:                Sat, 11 Sep 2021   Prob (F-statistic):               0.00
Time:                        10:57:26   Log-Likelihood:                -89632.
No. Observations:                9960   AIC:                         1.793e+05
Df Residuals:                    9954   BIC:                         1.793e+05
Df Model:                           5
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
gender_dummy   -59.9693     39.301     -1.526      0.127    -137.007      17.069
churn_dummy  -2189.9948     44.541    -49.168      0.000   -2277.304   -2102.686
Children        24.2064      9.150      2.646      0.008       6.271      42.142
Age             -1.1886      0.949     -1.252      0.210      -3.049       0.672
Income           0.0003      0.001      0.437      0.662      -0.001       0.002
intercept     4001.2641     68.324     58.563      0.000    3867.336    4135.192
==============================================================================
Omnibus:                     2245.415   Durbin-Watson:                   0.617
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              496.702
Skew:                          -0.232   Prob(JB):                    1.39e-108
Kurtosis:                       2.010   Cond. No.                     1.70e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.7e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

As you can see from the regression results, our five variables and the intercept are shown below the coef column. Our OLS regression shows a relatively low adj. R-squared which suggest that our Y variable is not very good at explaining the variance in our predictors, meaning that our goodness-of-fit is weak; the regression line is not reliable at predicting new Y values. Our beta coefficient for Children is high, which suggests higher bandwidth usage for those who have more children, which makes intuitive sense and is further backed by a low P-value. The beta coefficient for Age is low and negative however, with a high P-value we cannot consider it significant.

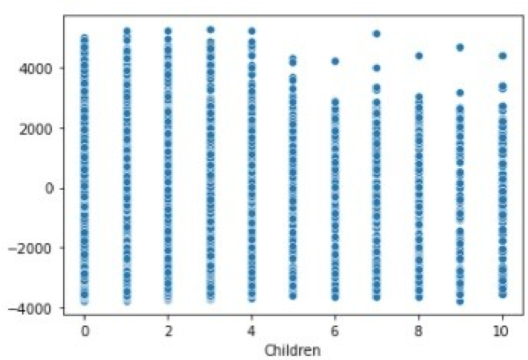
For the reduced model, I will drop the X variables that have a high P-value. We can only include those variables that have statistical significance and we will drop the variables that have a P-value above the 0.05 significance level. Our reduced model will include Children and churn_dummy.

```
[11]: df['intercept'] = 1
reduced_model = sm.OLS(df['Bandwidth_GB_Year'], df[['churn_dummy','Children', 'intercept']]).fit()
print(reduced_model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:      Bandwidth_GB_Year   R-squared:                       0.196
Model:                            OLS   Adj. R-squared:                  0.196
Method:                 Least Squares   F-statistic:                     1212.
Date:                Sat, 11 Sep 2021   Prob (F-statistic):               0.00
Time:                        11:48:51   Log-Likelihood:                -89634.
No. Observations:                9960   AIC:                         1.793e+05
Df Residuals:                    9957   BIC:                         1.793e+05
Df Model:                           2
Covariance Type:            nonrobust
===============================================================================
                  coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------
churn_dummy -2188.4209     44.526    -49.150      0.000   -2275.700   -2101.141
Children       24.6801      9.146      2.699      0.007       6.753      42.607
intercept    3919.1632     29.855    131.275      0.000    3860.642    3977.684
==============================================================================
Omnibus:                     2233.173   Durbin-Watson:                   0.619
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              495.333
Skew:                          -0.232   Prob(JB):                    2.75e-108
Kurtosis:                       2.011   Cond. No.                         7.37
==============================================================================

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

Our reduced model still has a pretty low adj. R-squared, so our regression model has not improved that much even though we have left out some unnecessary data. The coefficients show that for a 1 unit increase in Children we will have a 24 point increase in Bandwidth_GB_Year and a 1 unit increase in Churn will result in a decrease of 2188 Bandwidth_GB_Year. Evaluating the model we find that our R-squared and adj. R-squared show a poor goodness of fit, generally an R-squared higher than 0.7 would be required to say that this model has a reliable regression line and therefore fits the data well. Below is a residual plot for the reduced model.

```
[14]: df['intercept'] = 1
residuals = df['Bandwidth_GB_Year'] - reduced_model.predict(df[['churn_dummy', 'Children', 'intercept']])
sns.scatterplot(x=df['Children'],y=residuals)
plt.show();
```



Part 5

The regression equation for the reduced model is:

y = 3919.16 + (24.68*Children) + (-2188.42*churn_dummy)

Our coefficients while being statistically significant are not very good at predicting Y values due to the R-squared of the overall model. This suggests that our data analysis is somewhat limited and using more data points or different data points will make this analysis stronger. I recommend, based on this OLS regression, that the business in question should focus it's efforts on marketing to families and create a new pricing tiers for families that have higher numbers of children. This will likely result in higher revenues for the business since bigger families will lead to more bandwidth usage.

All references and code are from course lectures + videos, official docs from Python libraries, or from *https://stackoverflow.com/*