D208_Performance_Assessment_NBM2_Task_1

July 11, 2021

1 D208 Performance Assessment NBM2 Task 1

1.1 Multiple Regression for Predictive Modeling

Ryan L. Buchanan Student ID: 001826691 Masters Data Analytics (12/01/2020) Program Mentor: Dan Estes (385) 432-9281 (MST) rbuch49@wgu.edu

1.1.1 A1. Research Question:

How much many GBs of data will a customer use yearly? Can this be predicted accurately from a list of explanatory variables?

1.1.2 A2. Objectives & Goals:

Stakeholders in the company will benefit by knowing, with some measure of confidence, how much data a customer might predictably use. This will provide weight for decisions in whether or not to expand customer data limits, provide unlimited (or metered) media streaming & expand company cloud computing resources for increased bandwidth demands.

1.1.3 B1. Summary of Assumptions:

Assumptions of a multiple regression model include: * There is a linear relationship between the dependent variables & the independent variables. * The independent variables are not too highly correlated with each other. * yi observations are selected independently & randomly from the population. * Residuals should normally distributed with a mean of zero.

1.1.4 B2. Tool Benefits:

Python & IPython Jupyter notebooks will be used to support this analysis. Python offers very intuitive, simple & versatile programming style & syntax, as well as a large system of mature packages for data science & machine learning. Since, Python is cross-platform, it will work well whether consumers of the analysis are using Windows PCs or a MacBook laptop. It is fast when compared with other possible programming languages like R or MATLAB (Massaron, p. 8). Also, there is strong support for Python as the most popular data science programming language in popular literature & media (CBTNuggets)

1.1.5 B3. Appropriate Technique:

Multiple regression is an appropriate technique to analyze the research question because our target variable, predicting a real number of GBs per year, is a continuous variable (how much data is used). Also, perhaps there are several (versus simply one) explanatory variables (area type, job, children, age, income, etc.) that will add to our understanding when trying to predict how much data a customer will use in a given year. When adding or removing independent variables from our regression equation, we will find out whether or not they have a positive or negative relationship to our target variable & how that might affect company decisions on marketing segmentation.

1.1.6 Part III: Data Preparation

C. Summarize the data preparation process for multiple regression analysis by doing the following:

1. Describe your data preparation goals and the data manipulations that will be used to achieve the goals.
2. Discuss the summary statistics, including the target variable and all predictor variables that you will need to gather from the data set to answer the research question.
3. Explain the steps used to prepare the data for the analysis, including the annotated code.
4. Generate univariate and bivariate visualizations of the distributions of variables in the cleaned data set. Include the target variable in your bivariate visualizations.
5. Provide a copy of the prepared data set.

1.1.7 C1. Data Goals:

My approach will include: 1. Back up my data and the process I am following as a copy to my machine and, since this is a manageable dataset, to GitHub using command line and gitbash. 2. Read the data set into Python using Pandas' read_csv command. 3. Evaluate the data struture to better understand input data. 4. Naming the dataset as a the variable "churn_df" and subsequent useful slices of the dataframe as "df". 5. Examine potential misspellings, awkward variable naming & missing data. 6. Find outliers that may create or hide statistical significance using histograms. 7. Imputing records missing data with meaningful measures of central tendency (mean, median or mode) or simply remove outliers that are several standard deviations above the mean.

Most relevant to our decision making process is the dependent variable of "Bandwidth_GB_Year" (the average yearly amount of data used, in GB, per customer) which will be our continuous target variable. We need to train & then test our machine on our given dataset to develop a model that will give us an idea of how much data a customer may use given the amounts used by known customers given their respective data points for selected predictor variables.

In cleaning the data, we may discover relevance of the continuous predictor variables: * Children * Income * Outage_sec_perweek * Yearly_equip_failure * Tenure (the number of months the customer has stayed with the provider) * MonthlyCharge * Bandwidth_GB_Year

Likewise, we may discover relevance of the categorical predictor variables (all binary categorical with only two values, "Yes" or "No", except where noted): * Techie: Whether the customer considers themselves technically inclined (based on customer questionnaire when they signed up for services) (yes, no) * Contract: The contract term of the customer (month-to-month, one year, two year) * Port_modem: Whether the customer has a portable modem (yes, no) * Tablet: Whether the customer owns a tablet such as iPad, Surface, etc. (yes, no) * InternetService: Customer's internet service provider (DSL, fiber optic, None) * Phone: Whether the customer has a phone service (yes, no) * Multiple: Whether the customer has multiple lines (yes, no) * OnlineSecurity: Whether the customer has an online security add-on (yes, no) * OnlineBackup: Whether the customer has an online backup add-on (yes, no) * DeviceProtection: Whether the customer has device protection add-on (yes, no) * TechSupport: Whether the customer has a technical support add-on (yes, no) * StreamingTV: Whether the customer has streaming TV (yes, no) * StreamingMovies: Whether the customer has streaming movies (yes, no)

Finally, discrete ordinal predictor variables from the survey responses from customers regarding various customer service features may be relevant in the decision-making process. In the surveys, customers provided ordinal numerical data by rating 8 customer service factors on a scale of 1 to 8 (1 = most important, 8 = least important):

- Item1: Timely response
- Item2: Timely fixes
- Item3: Timely replacements
- Item4: Reliability
- Item5: Options
- Item6: Respectful response
- Item7: Courteous exchange
- Item8: Evidence of active listening

1.1.8 C2. Summary Statistics:

Discuss the summary statistics, including the target variable and all predictor variables that you will need to gather from the data set to answer the research question.

1.1.9 C3. Steps to Prepare Data:

Explain the steps used to prepare the data for the analysis, including the annotated code.

- Imputing records missing data with meaningful measures of central tendency (mean, median or mode) or simply remove outliers that are several standard deviations above the mean.

*
*
*

- Finally, the prepared dataset will be extracted & provided as "churn_prepared.csv"

1.1.10 C4. Visualizations:

Generate univariate and bivariate visualizations of the distributions of variables in the cleaned data set. Include the target variable in your bivariate visualizations.

For univariate, use histograms. Make to show the statistics for each variable with describe() method. For categorical variables display with horizontal bar. For bivariate relationships, scatterplot.

1.1.11 C5. Prepared Dataset:

Provide a copy of the prepared data set.

[1]:
```
# Increase Jupyter display cell-width
from IPython.core.display import display, HTML
display(HTML("<style>.container { width:75% !important; }</style>"))
```

```
<IPython.core.display.HTML object>
```

[2]:
```
# Standard data science imports
import numpy as np
import pandas as pd
from pandas import Series, DataFrame

# Visualization libraries
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

# Statistics packages
import pylab
from pylab import rcParams
import statsmodels.api as sm
import statistics
from scipy import stats

# Scikit-learn
import sklearn
from sklearn import preprocessing
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import classification_report


# Import chisquare from SciPy.stats
from scipy.stats import chisquare
from scipy.stats import chi2_contingency

# Ignore Warning Code
import warnings
warnings.filterwarnings('ignore')
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19:
FutureWarning: pandas.util.testing is deprecated. Use the functions in the
public API at pandas.testing instead.
  import pandas.util.testing as tm
```

[3]:
```python
# Change color of Matplotlib font
import matplotlib as mpl

COLOR = 'white'
mpl.rcParams['text.color'] = COLOR
mpl.rcParams['axes.labelcolor'] = COLOR
mpl.rcParams['xtick.color'] = COLOR
mpl.rcParams['ytick.color'] = COLOR
```

[5]:
```python
# Load data set into Pandas dataframe
churn_df = pd.read_csv('churn_clean.csv')

# Rename last 8 survey columns for better description of variables
churn_df.rename(columns = {'Item1':'TimelyResponse',
                    'Item2':'Fixes',
                     'Item3':'Replacements',
                     'Item4':'Reliability',
                     'Item5':'Options',
                     'Item6':'Respectfulness',
                     'Item7':'Courteous',
                     'Item8':'Listening'},
          inplace=True)
```

[6]:
```python
# Display Churn dataframe
churn_df
```

[6]:
```
      CaseOrder Customer_id  ... Courteous Listening
0             1     K409198  ...         3         4
1             2     S120509  ...         4         4
2             3     K191035  ...         3         3
3             4      D90850  ...         3         3
4             5     K662701  ...         4         5
...         ...         ...  ...       ...       ...
9995       9996     M324793  ...         2         3
9996       9997     D861732  ...         2         5
9997       9998     I243405  ...         4         5
9998       9999     I641617  ...         5         4
9999      10000      T38070  ...         4         1

[10000 rows x 50 columns]
```

[7]:
```python
# List of Dataframe Columns
df = churn_df.columns
print(df)
```

```
Index(['CaseOrder', 'Customer_id', 'Interaction', 'UID', 'City', 'State',
       'County', 'Zip', 'Lat', 'Lng', 'Population', 'Area', 'TimeZone', 'Job',
       'Children', 'Age', 'Income', 'Marital', 'Gender', 'Churn',
       'Outage_sec_perweek', 'Email', 'Contacts', 'Yearly_equip_failure',
       'Techie', 'Contract', 'Port_modem', 'Tablet', 'InternetService',
       'Phone', 'Multiple', 'OnlineSecurity', 'OnlineBackup',
       'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies',
       'PaperlessBilling', 'PaymentMethod', 'Tenure', 'MonthlyCharge',
       'Bandwidth_GB_Year', 'TimelyResponse', 'Fixes', 'Replacements',
       'Reliability', 'Options', 'Respectfulness', 'Courteous', 'Listening'],
      dtype='object')
```

[8]:
```
# Find number of records and columns of dataset
churn_df.shape
```

[8]:
```
(10000, 50)
```

[9]:
```
# Describe Churn dataset statistics
churn_df.describe()
```

[9]:
```
         CaseOrder           Zip  ...     Courteous     Listening
count  10000.00000  10000.000000  ...  10000.000000  10000.000000
mean    5000.50000  49153.319600  ...      3.509500      3.495600
std     2886.89568  27532.196108  ...      1.028502      1.028633
min        1.00000    601.000000  ...      1.000000      1.000000
25%     2500.75000  26292.500000  ...      3.000000      3.000000
50%     5000.50000  48869.500000  ...      4.000000      3.000000
75%     7500.25000  71866.500000  ...      4.000000      4.000000
max    10000.00000  99929.000000  ...      7.000000      8.000000

[8 rows x 23 columns]
```

[10]:
```
# Remove less meaningful demographic variables from statistics description
churn_df = churn_df.drop(columns=['CaseOrder', 'Customer_id', 'Interaction',␣
 ↪'UID', 'City',
                                  'State', 'County', 'Zip', 'Lat', 'Lng',␣
 ↪'Population',
                                  'Area', 'TimeZone', 'Job', 'Marital'])
churn_df.describe()
```

[10]:
```
         Children           Age  ...     Courteous     Listening
count  10000.0000  10000.000000  ...  10000.000000  10000.000000
mean       2.0877     53.078400  ...      3.509500      3.495600
std        2.1472     20.698882  ...      1.028502      1.028633
min        0.0000     18.000000  ...      1.000000      1.000000
25%        0.0000     35.000000  ...      3.000000      3.000000
50%        1.0000     53.000000  ...      4.000000      3.000000
75%        3.0000     71.000000  ...      4.000000      4.000000
max       10.0000     89.000000  ...      7.000000      8.000000
```

```
[8 rows x 18 columns]
```

[11]:
```
# Discover missing data points within dataset
data_nulls = churn_df.isnull().sum()
print(data_nulls)
```

```
Children                0
Age                     0
Income                  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
TimelyResponse          0
Fixes                   0
Replacements            0
Reliability             0
Options                 0
Respectfulness          0
Courteous               0
Listening               0
dtype: int64
```

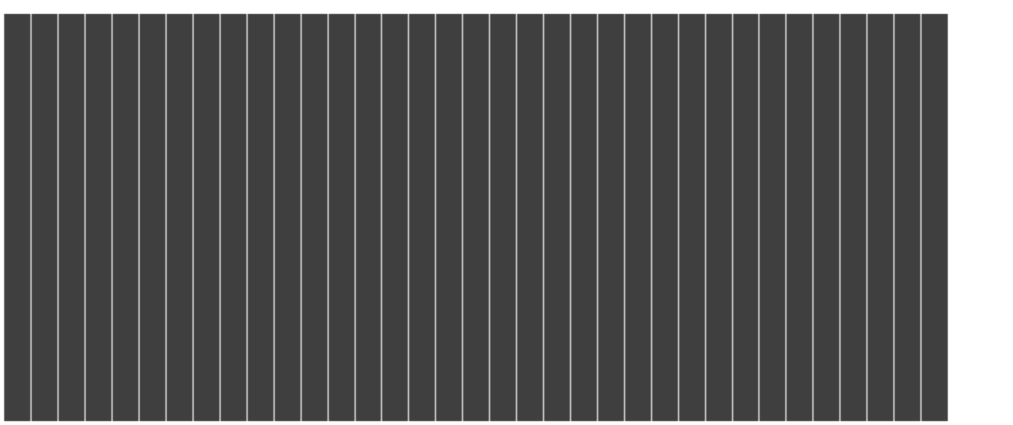
[12]:
```
# Visualize missing values in dataset

# Install appropriate library
!pip install missingno
```

```python
# Importing the libraries
import missingno as msno

# Visualize missing values as a matrix
msno.matrix(churn_df);
```

```
Requirement already satisfied: missingno in /usr/local/lib/python3.7/dist-
packages (0.4.2)
Requirement already satisfied: seaborn in /usr/local/lib/python3.7/dist-packages
(from missingno) (0.11.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages
(from missingno) (1.19.5)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-
packages (from missingno) (3.2.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages
(from missingno) (1.4.1)
Requirement already satisfied: pandas>=0.23 in /usr/local/lib/python3.7/dist-
packages (from seaborn->missingno) (1.1.5)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in
/usr/local/lib/python3.7/dist-packages (from matplotlib->missingno) (2.4.7)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7
/dist-packages (from matplotlib->missingno) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-
packages (from matplotlib->missingno) (0.10.0)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7
/dist-packages (from matplotlib->missingno) (2.8.1)
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.7/dist-
packages (from pandas>=0.23->seaborn->missingno) (2018.9)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages
(from cycler>=0.10->matplotlib->missingno) (1.15.0)
```

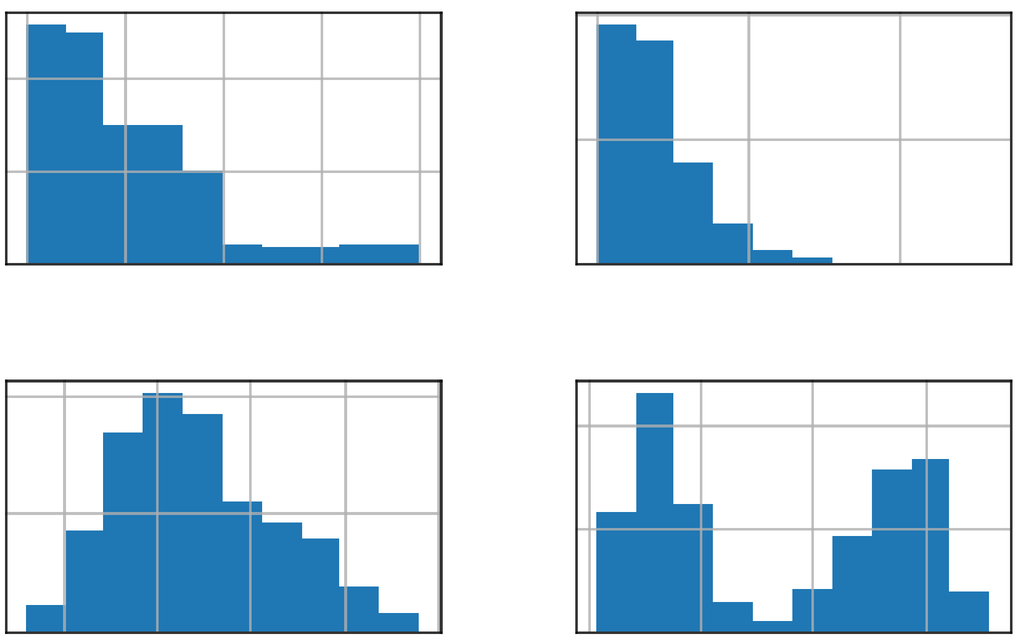


[13]:
```
'''No need to impute an missing values as the dataset appears complete/
↪cleaned'''
# Impute missing fields for variables Children, Age, Income, Tenure and␣
↪Bandwidth_GB_Year with median or mean
# churn_df['Children'] = churn_df['Children'].fillna(churn_df['Children'].
↪median())
# churn_df['Age'] = churn_df['Age'].fillna(churn_df['Age'].median())
# churn_df['Income'] = churn_df['Income'].fillna(churn_df['Income'].median())
# churn_df['Tenure'] = churn_df['Tenure'].fillna(churn_df['Tenure'].median())
# churn_df['Bandwidth_GB_Year'] = churn_df['Bandwidth_GB_Year'].
↪fillna(churn_df['Bandwidth_GB_Year'].median())
```

[13]:
```
'No need to impute an missing values as the dataset appears complete/cleaned'
```

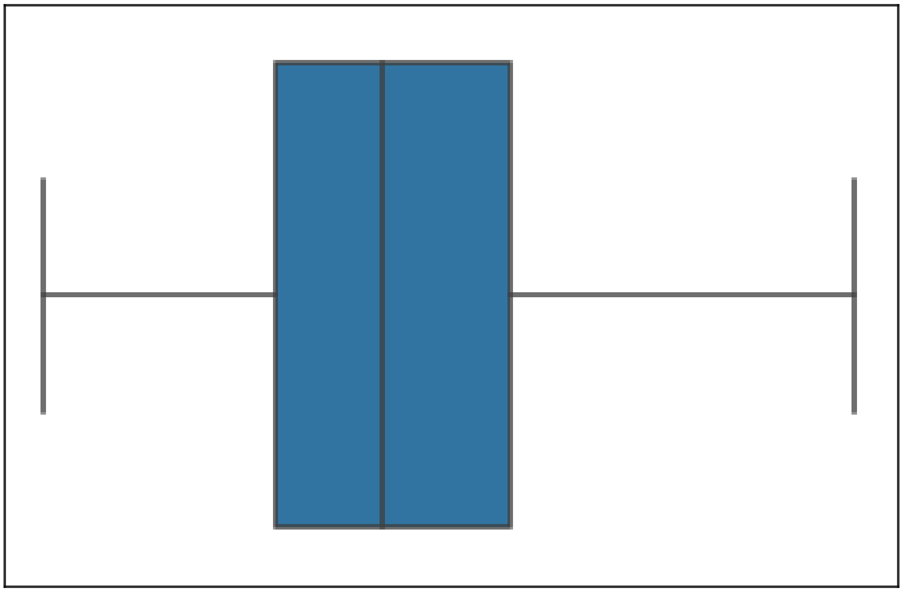
1.2 Univariate Statistics

[14]:
```
# Create histograms of contiuous & categorical variables
churn_df[['Children', 'Income', 'MonthlyCharge', 'Bandwidth_GB_Year']].hist()
plt.savefig('churn_pyplot.jpg')
plt.tight_layout()
```

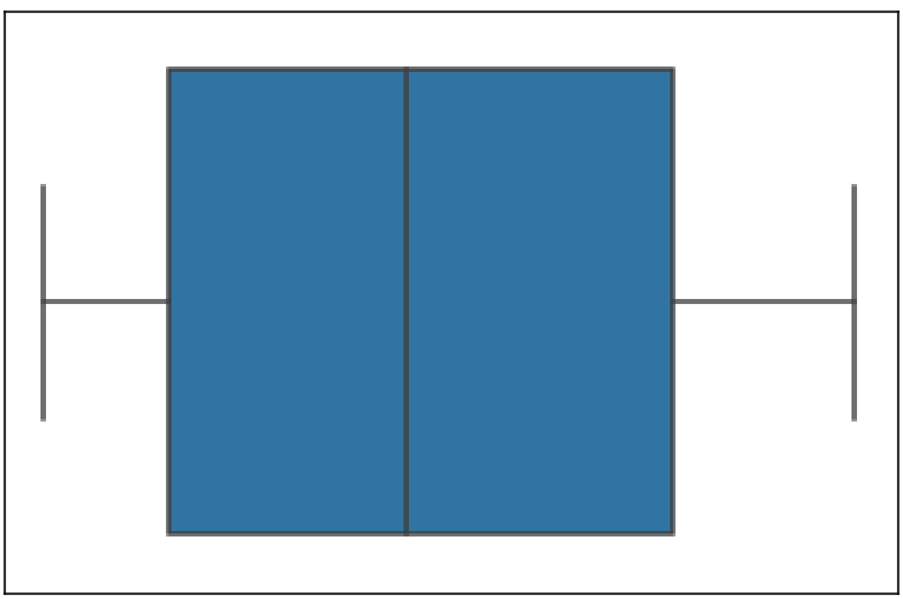


[15]:
```
# Create Seaborn boxplots for continuous & categorical variables
sns.boxplot('MonthlyCharge', data = churn_df)
```

```
plt.show()
```



[16]:
```
sns.boxplot('Bandwidth_GB_Year', data = churn_df)
plt.show()
```



1.2.1 It appears that anomolies have been removed from the dataset present "churn_clean.csv" as there are no remaining outliers.

[17]:
```
# Develop the estimated regression equation that could be used to predict the␣
↪Bandwidth_GB_Year,
# given the continuous variables
churn_df['intercept'] = 1
lm_bandwidth = sm.OLS(churn_df['Bandwidth_GB_Year'], churn_df[['Children',␣
↪'Age',
                                                              'Income',
                                                             ␣
↪'Outage_sec_perweek',
                                                             ␣
↪'Yearly_equip_failure',
                                                              'Tenure',␣
↪'MonthlyCharge',
                                                              'intercept']]).
↪fit()
print(lm_bandwidth.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:                 1.293e+05
Date:                Sun, 11 Jul 2021   Prob (F-statistic):               0.00
Time:                        14:02:15   Log-Likelihood:                -68496.
No. Observations:               10000   AIC:                         1.370e+05
Df Residuals:                    9992   BIC:                         1.371e+05
Df Model:                           7
Covariance Type:            nonrobust
================================================================================
========
                           coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
--------
Children                30.8584      1.064     28.992      0.000      28.772
32.945
Age                     -3.3110      0.110    -29.983      0.000      -3.528
-3.095
Income                   0.0001    8.1e-05      1.277      0.202   -5.53e-05
0.000
Outage_sec_perweek      -0.2570      0.768     -0.335      0.738      -1.762
```

```
1.248
Yearly_equip_failure      0.6729      3.592      0.187      0.851      -6.369
7.715
Tenure                   82.0128      0.086    949.221      0.000      81.843
82.182
MonthlyCharge             3.2753      0.053     61.557      0.000       3.171
3.380
intercept               104.8529     14.383      7.290      0.000      76.659
133.047
==============================================================================
Omnibus:                12845.406   Durbin-Watson:                   1.979
Prob(Omnibus):              0.000   Jarque-Bera (JB):              973.247
Skew:                       0.450   Prob(JB):                    4.59e-212
Kurtosis:                   1.764   Cond. No.                     3.07e+05
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 3.07e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

1.2.2 Based on an R2 value = 0.989. So, 99% of the variation is explained by this model.

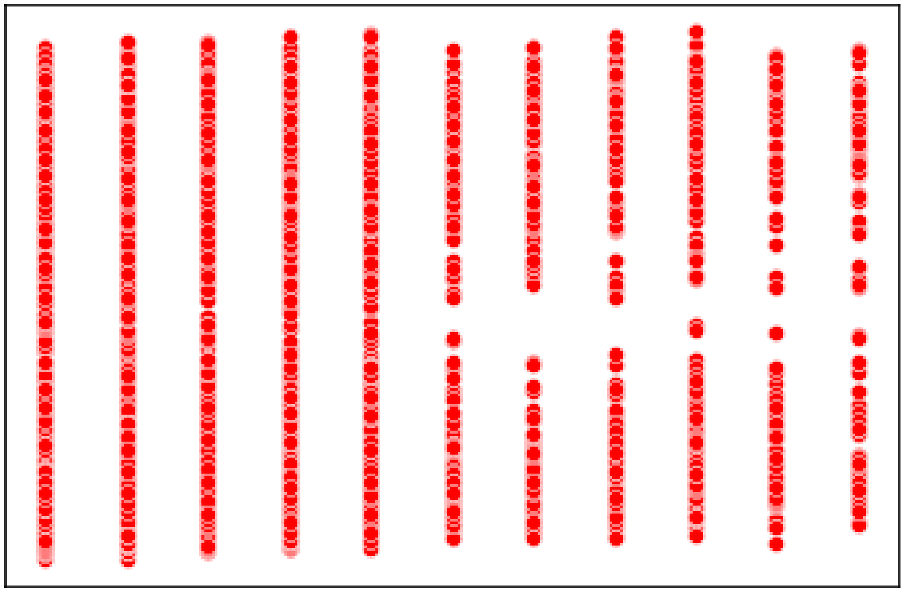
1.2.3 Initial Multiple Linear Regression Model

With seven indpendent variables: y = 104.85 + 30.86 * Children - 3.31 * Age + 0.00 * Income - 0.26 * Outage_sec_perweek + 0.67 * Yearly_equip_failure + 82.01 * Tenure + 3.28 * MonthlyCharge

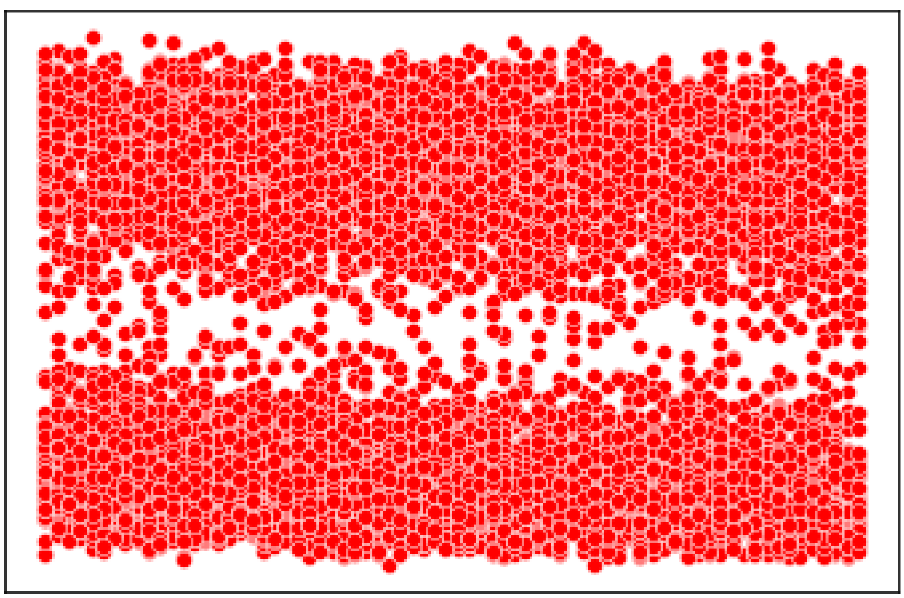
1.3 Bivariate Statistics

1.3.1 Let's run some scatterplots to get an idea of our linear relationships with bandwidth usage & some of the respective predictor variables.

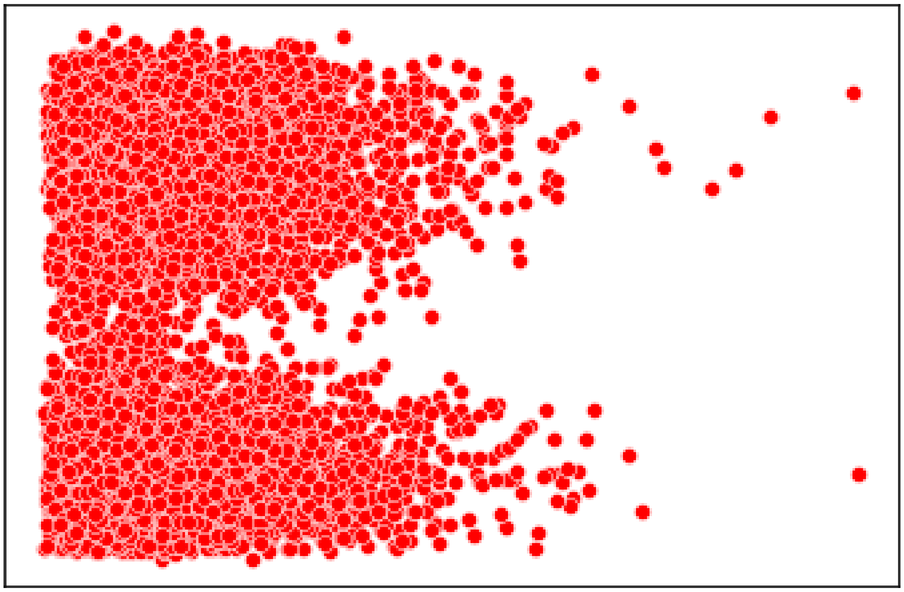
[18]:
```python
# Run scatterplots to show direct or inverse relationships between target &
 ↪independent variables
sns.scatterplot(x=churn_df['Children'], y=churn_df['Bandwidth_GB_Year'],
 ↪color='red')
plt.show();
```



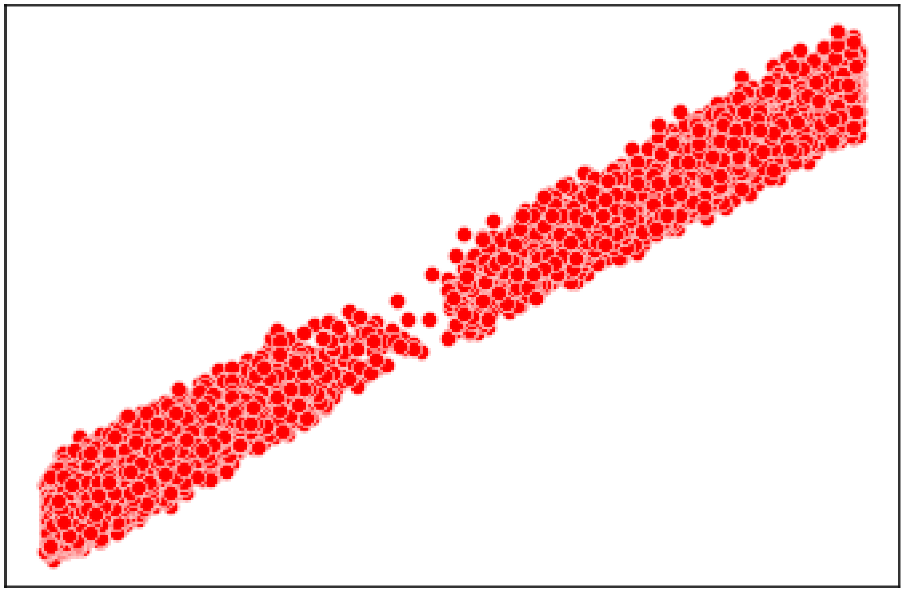
[19]:
```
sns.scatterplot(x=churn_df['Age'], y=churn_df['Bandwidth_GB_Year'], color='red')
plt.show();
```



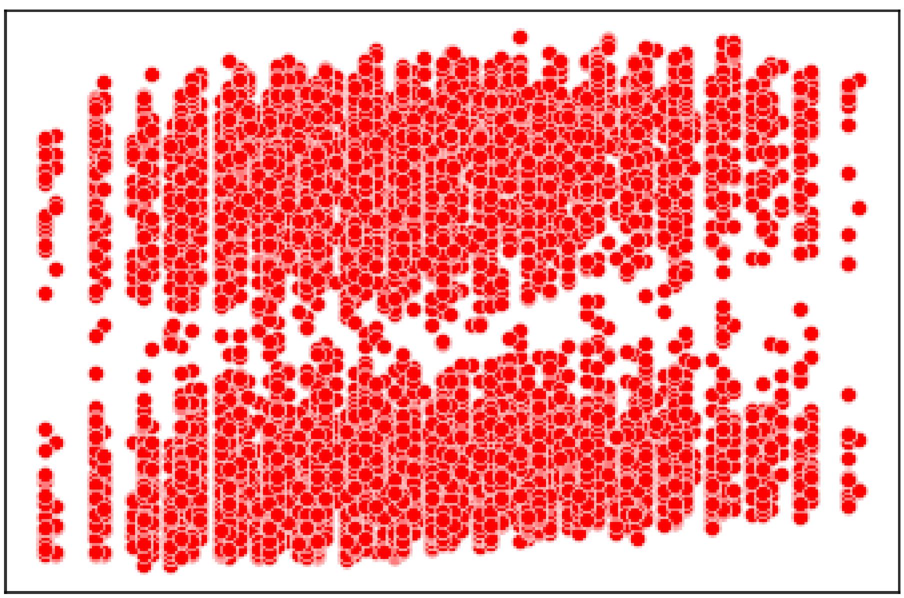
[20]:
```
sns.scatterplot(x=churn_df['Income'], y=churn_df['Bandwidth_GB_Year'],␣
 ↪color='red')
plt.show();
```



[21]:
```
sns.scatterplot(x=churn_df['Tenure'], y=churn_df['Bandwidth_GB_Year'],␣
 ↪color='red')
plt.show();
```

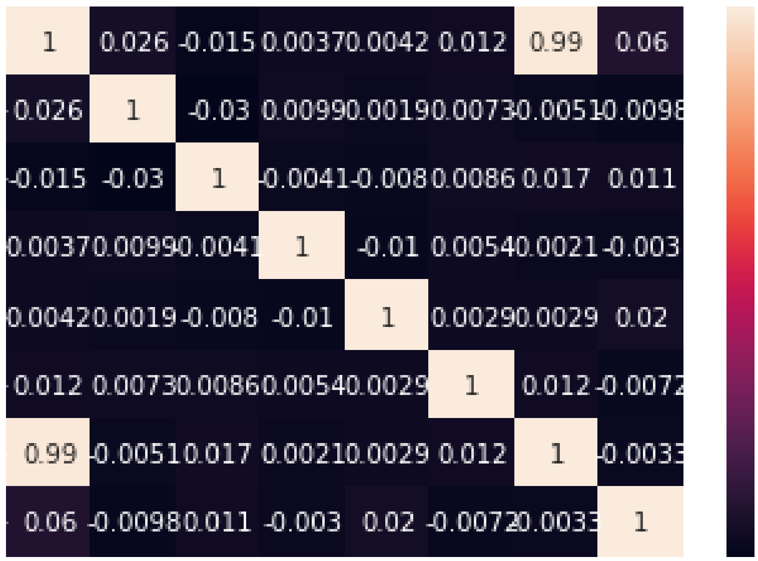


[22]:
```
sns.scatterplot(x=churn_df['MonthlyCharge'], y=churn_df['Bandwidth_GB_Year'],␣
 ↪color='red')
plt.show();
```



[23]:
```
# Create dataframe for heatmap bivariate analysis of correlation
churn_bivariate = churn_df[['Bandwidth_GB_Year', 'Children', 'Age', 'Income',
                            'Outage_sec_perweek', 'Yearly_equip_failure',
                            'Tenure', 'MonthlyCharge']]
```

[24]:
```
# Run Seaborn heatmap
sns.heatmap(churn_bivariate.corr(), annot=True)
plt.show()
```



1.3.2 Again, it appears that Tenure is the predictor for most of the varaince.

[25]:
```
"""Scree plots & PCA!!!"""
```

[25]: 'Scree plots & PCA!!!'

1.3.3 There is clearly a direct linear relationship between customer tenure with the telecom company & the amount of data (in GBs) that is being used. Let's run a simple linear regression model on those two variables.

[26]:
```
churn_df['intercept'] = 1
lm_bandwidth = sm.OLS(churn_df['Bandwidth_GB_Year'], churn_df[['Children',␣
↪'Tenure', 'intercept']]).fit()
print(lm_bandwidth.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:      Bandwidth_GB_Year   R-squared:                       0.984
Model:                            OLS   Adj. R-squared:                  0.984
Method:                 Least Squares   F-statistic:                 3.074e+05
Date:                Sun, 11 Jul 2021   Prob (F-statistic):               0.00
Time:                        14:02:24   Log-Likelihood:                -70408.
No. Observations:               10000   AIC:                         1.408e+05
Df Residuals:                    9997   BIC:                         1.408e+05
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Children      31.1771      1.288     24.215      0.000      28.653      33.701
Tenure        81.9516      0.105    783.869      0.000      81.747      82.156
intercept    497.7782      5.291     94.079      0.000     487.407     508.150
==============================================================================
Omnibus:                      380.523   Durbin-Watson:                   1.978
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              295.655
Skew:                           0.335   Prob(JB):                     6.30e-65
Kurtosis:                       2.489   Cond. No.                         84.0
==============================================================================

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

1.3.4 Well, there it is. Removing all the other predictor variables except "Tenure" & our model still explains 98% of the variance.

1.3.5 Reduced Multiple Linear Regression Model

With two indpendent variables: y = 497.78 + 31.18 * Children + 81.94 * Tenure

[27]:
```
# Extract Clean dataset
churn_df.to_csv('churn_prepared.csv')
```

1.3.6 Part IV: Model Comparison and Analysis

D. Compare an initial and a reduced multiple regression model by doing the following:

1. Construct an initial multiple regression model from all predictors that were identified in Part C2.

Note: Clearly state regression equation, for example:
"four indpendent vars: y = -0.878 + 0.01 * Age + 0.31 * Female + 0.22 * Education + 0.09 * Income"

2. Justify a statistically based variable selection procedure and a model evaluation metric to reduce the initial model in a way that aligns with the research question.

3. Provide a reduced multiple regression model that includes both categorical and continuous variables.

Note: The output should include a screenshot of each model.

1.3.7 D1. Initial Model

Construct an initial multiple regression model from all predictors that were identified in Part C2.

1.3.8 D2. Justification of Model Reduction

Justify a statistically based variable selection procedure and a model evaluation metric to reduce the initial model in a way that aligns with the research question.
Note: Heatmap of missing values vs observed

1.3.9 D3. Reduced Multiple Regression Model

Provide a reduced multiple regression model that includes both categorical and continuous variables.

1.3.10 Part IV: E

E. Analyze the data set using your reduced multiple regression model by doing the following:

1. Explain your data analysis process by comparing the initial and reduced multiple regression models, including the following elements:
   the logic of the variable selection technique
   the model evaluation metric
   a residual plot
2. Provide the output and any calculations of the analysis you performed, including the model's residual error.

Note: The output should include the predictions from the refined model you used to perform the analysis.

3. Provide the code used to support the implementation of the multiple regression models.

1.3.11 E1. Model Comparison

Explain your data analysis process by comparing the initial and reduced multiple regression models, including the following elements:

```
<li>
the logic of the variable selection technique
</li>
<li>
the model evaluation metric
</li>
<li>
a residual plot
</li>
```

Note: Verbatim from fasttrack description of analysis of Titanic dataset, "Since male is the dummy variable, being male reduces the log odds by 2.75 while a unit increase in age reduces log odds by 0.037."

1.3.12 E2. Output & Calculations

Provide the output and any calculations of the analysis you performed, including the model's residual error.

Note: The output should include the predictions from the refined model you used to perform the analysis.

1.3.13 E3. Code

Provide the code used to support the implementation of the multiple regression models.

1.3.14 Part V: Data Summary and Implications

F. Summarize your findings and assumptions by doing the following:

1. Discuss the results of your data analysis, including the following elements:
   a regression equation for the reduced model
   an interpretation of coefficients of the statistically significant variables of the model
   the statistical and practical significance of the model
   the limitations of the data analysis
2. Recommend a course of action based on your results.

1.3.15 F1. Results

Discuss the results of your data analysis, including the following elements:

```
<li>
a regression equation for the reduced model
</li>
<li>
an interpretation of coefficients of the statistically significant variables of the model
```

```
</li>
<li>
the statistical and practical significance of the model
</li>
<li>
the limitations of the data analysis
</li>
```

1.3.16 F2. Recommendations

Recommend a course of action based on your results.

1.3.17 Part VI: Demonstration

G. Provide a Panopto video recording that includes all of the following elements:

a demonstration of the functionality of the code used for the analysis
an identification of the version of the programming environment
a comparison of the two multiple regression models you used in your analysis
an interpretation of the coefficients.

1.3.18 G. Video

link

1.3.19 H. Sources for Third-Party Code

Kaggle. (2018, May 01). Bivariate plotting with pandas. Kaggle. https://www.kaggle.com/residentmario/bivariate-plotting-with-pandas#

Sree. (2020, October 26). Predict Customer Churn in Python. Towards Data Science. https://towardsdatascience.com/predict-customer-churn-in-python-e8cd6d3aaa7

Wikipedia. (2021, May 31). Bivariate Analysis. https://en.wikipedia.org/wiki/Bivariate_analysis#:~:text=Biv

1.3.20 I. Sources

Ahmad, A. K., Jafar, A & Aljoumaa, K. (2019, March 20). Customer churn prediction in telecom using machine learning in big data platform. Journal of Big Data. https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0191-6

Altexsoft. (2019, March 27). Customer Churn Prediction Using Machine Learning: Main Approaches and Models. Altexsoft. https://www.altexsoft.com/blog/business/customer-churn-prediction-for-subscription-businesses-using-machine-learning-main-approaches-and-models/

Bruce, P., Bruce A. & Gedeck P. (2020). Practical Statistics for Data Scientists. O'Reilly.

CBTNuggets. (2018, September 20). Why Data Scientists Love Python. https://www.cbtnuggets.com/blog/technology/data/why-data-scientists-love-python

Freedman, D. Pisani, R. & Purves, R. (2018). Statistics. W. W. Norton & Company, Inc.

Frohbose, F. (2020, November 24). Machine Learning Case Study: Telco Customer Churn Prediction. Towards Data Science. https://towardsdatascience.com/machine-learning-case-study-telco-customer-churn-prediction-bc4be03c9e1d

Griffiths, D. (2009). A Brain-Friendly Guide: Head First Statistics. O'Reilly.

Grus, J. (2015). Data Science from Scratch. O'Reilly.
Massaron, L. & Boschetti, A. (2016). Regression Analysis with Python. Packt Publishing.
McKinney, W. (2018). Python for Data Analysis. O'Reilly.
Rossant, C. (2018). IPython Interactive Computing & Visualization Cookbook, 2nd Edition. Packt Publishing.
Rossant, C. (2015). Learning IPython Interactive Computing & Visualization, 2nd Edition. Packt Publishing.
VanderPlas, J. (2017). Python Data Science Handbook. O'Reilly.

[ ]:
```
!wget -nc https://raw.githubusercontent.com/brpy/colab-pdf/master/colab_pdf.py
from colab_pdf import colab_pdf
colab_pdf('D208_Performance_Assessment_NBM2_Task_1.ipynb')
```

```
--2021-07-11 14:02:35--  https://raw.githubusercontent.com/brpy/colab-
pdf/master/colab_pdf.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)...
185.199.108.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com
(raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1864 (1.8K) [text/plain]
Saving to: colab_pdf.py

colab_pdf.py        100%[===================>]   1.82K  --.-KB/s    in 0s

2021-07-11 14:02:35 (29.3 MB/s) - colab_pdf.py saved [1864/1864]

Mounted at /content/drive/

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.


WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

Extracting templates from packages: 100%
```

[ ]: