D208 Task 2: Logistic Regression Modeling

Part I: Research Question

A1. Research Question

The research question I would like to analyze is, “What variables are directly correlated to customer churn?”

A2. Data Analysis Goals

The objective of this data analysis is to use a predictive model, in this case, a logistic regression model, to gain more insight to determine which variables from the churn dataset correlate to customer churn. In this case, the variables from the dataset are the independent or explanatory variables and churn would be the dependent or target variable. Once the outcome has been determined, we should have an idea of what variables affect churn.

Part II: Logistic Regression

B1. Assumptions

Logistic regression is a statistical method used to predict the relationship between multiple independent (explanatory) variables and a single dependent (response) variable. Before conducting this method, we have to check for the following assumptions (Bobbitt 2020):

- Linearity: There is a linear relationship between the explanatory variables and the logit of response variable (inverse of a standard logistic function).
- No Multicollinearity: None of the explanatory variables are correlated with each other.
- Independence: The observations are independent of each other.
- The dependent variable is binary.
- There aren’t any extreme outliers in the dataset.
- Sample size of the dataset is large.

B2. Tool Benefits

I will be using Python to perform this data analysis. Python is a great tool to clean data and perform logistic regression because of the consistent syntax that makes it easy to learn and follow along, the flexibility to create and learn new things, and all of the libraries and packages that it has to offer. For example, I will be using the following libraries and packages for my analysis (R or Python 2023):

- pandas- to load datasets
- NumPy- to work with arrays
- Sci-kit Learn- for machine learning and to transform our data

- SciPy- for mathematical problems like checking for multicollinearity
- Matplotlib- for basic plotting generally consisting of bars, lines, pies, scatter plots, and graphs
- Seaborn- for a variety of visualization patterns

B3. Justification

Logistic regression is the appropriate technique to use to analyze the research question because the target variable, churn, is a binary categorical dependent variable. The multiple explanatory variables can be continuous and categorical though. However, if the target variable was continuous, then we would have to perform linear regression. Performing logistic regression will help to figure out if the explanatory variables have a positive or negative impact on the chosen target variable. This predictive model will give an indication of what independent variables directly affect customer churn as we add or remove them.

Part III: Data Preparation & Manipulation

C1. Data Cleaning Goals & Steps

The goal of data cleaning is to find any null or duplicated values in the dataset, correct any error or inconsistencies, and to get rid of any unnecessary variables that we will not be using for the regression analysis. I dropped the following columns since they are not important for the analysis question I have chosen: 'CaseOrder', 'Customer_id', 'Interaction', 'UID', 'City', 'State', 'County', 'Zip', 'Lat', 'Lng', 'Population', 'Area', 'TimeZone', 'Marital', 'Job', 'Age', 'PaperlessBilling', and 'PaymentMethod'. The reason I dropped these variables is because the customer's location, their personal life (age, job, and marital status), how they make payments, and their identification numbers does not impact if a customer leaves the company. I have the general steps I performed in order to clean and prepare the data for testing written below:

1. Import any necessary libraries and packages.
2. Load dataset into pandas data frame using read_csv command. The data frame is named "df".
3. Rename the survey columns to describe the variables better.
4. Print column names to check corrections made.
5. Calculate the total null values and total duplicate values in the dataset. If there are not any, the values will be shown as 0.
6. Check for the number of unique values in each column.
7. Print the columns with less than 100 unique values. This can help determine what variables I would like to drop from the analysis.
8. Drop columns that are unnecessary for the analysis.
9. Use the head() command to look at what data is left.

C2. Summary Statistics

The variables I will be using for the analysis include Churn (the categorical dependent variable) and the following explanatory variables are listed below:

- Categorical variables – Techie, Contract, Port_modem, Tablet, InternetService, Phone, Multiple, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, and StreamingMovies
- The proportion of each categorical variable is shown in the screenshot. For example, the churn variable is categorized into 2 groups, yes (people who left the company) and no, people who stayed with the company. Based on the proportion, 0.735 (73.5%) of customers stayed with the company and 0.265 (26.5%) of customers did not.

```
PROPORTION OF EACH CATEGORICAL VARIABLE
----------------------------------------
   Churn  Proportion
0    No       0.735
1   Yes       0.265
----------------------------------------
   Techie  Proportion
0     No      0.8321
1    Yes      0.1679
----------------------------------------
          Contract  Proportion
0  Month-to-month      0.5456
1        Two Year      0.2442
2        One year      0.2102
----------------------------------------
   Port_modem  Proportion
0          No      0.5166
1         Yes      0.4834
----------------------------------------
   Tablet  Proportion
0     No      0.7009
1    Yes      0.2991
----------------------------------------
   InternetService  Proportion
0     Fiber Optic      0.4408
1             DSL      0.3463
2            None      0.2129
----------------------------------------
   Phone  Proportion
0   Yes      0.9067
1    No      0.0933
----------------------------------------
   Multiple  Proportion
0       No      0.5392
1      Yes      0.4608
----------------------------------------
   OnlineSecurity  Proportion
0             No      0.6424
1            Yes      0.3576
----------------------------------------
   OnlineBackup  Proportion
0           No      0.5494
1          Yes      0.4506
----------------------------------------
   DeviceProtection  Proportion
0               No      0.5614
1              Yes      0.4386
----------------------------------------
   TechSupport  Proportion
0          No       0.625
1         Yes       0.375
----------------------------------------
   StreamingTV  Proportion
0          No      0.5071
1         Yes      0.4929
----------------------------------------
   StreamingMovies  Proportion
0              No       0.511
1             Yes       0.489
```

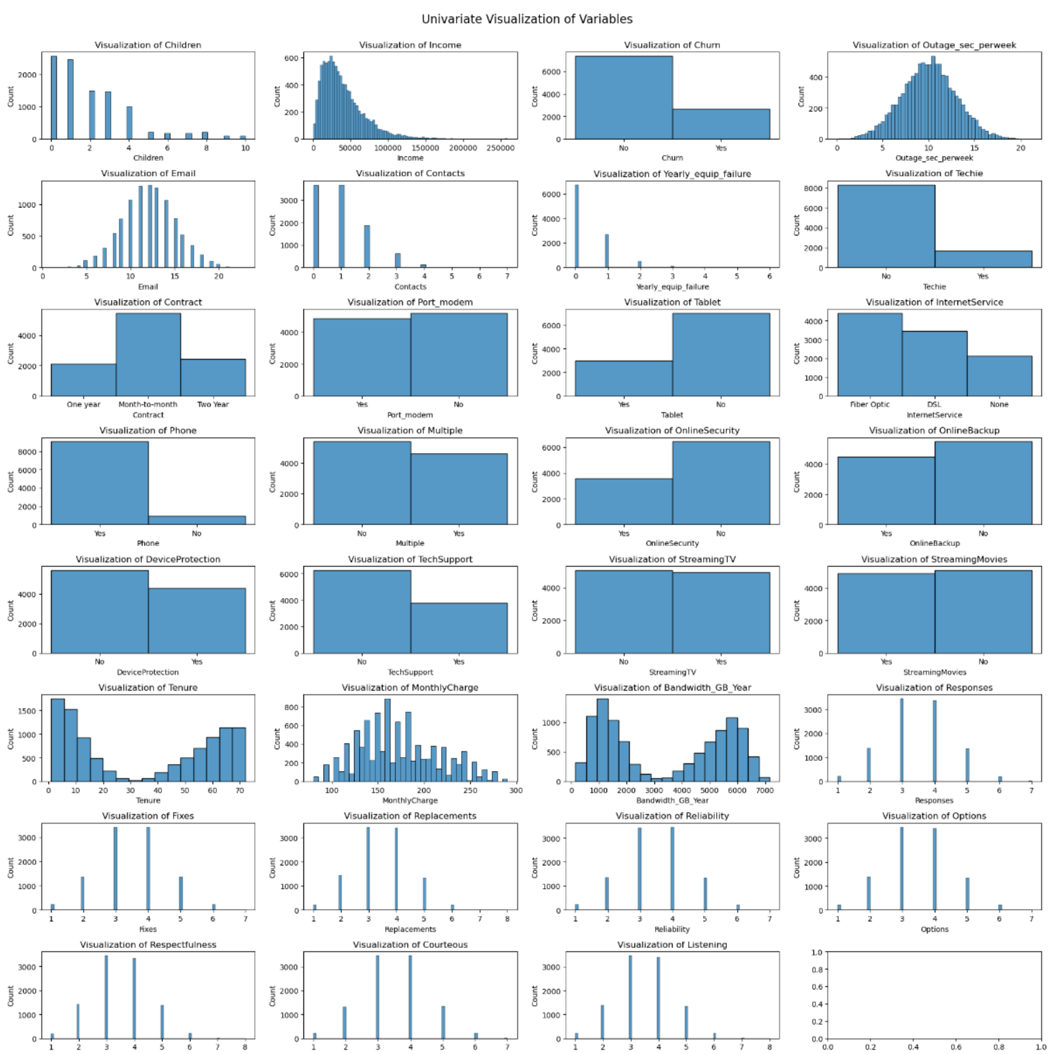
- Numerical variables – Children, Income, Outage_sec_perweek, Email, Contacts, Yearly_equip_failure, Tenure, MonthlyCharge, Bandwidth_GB_Year, Responses, Fixes, Replacements, Reliability, Options, Respectfulness, Courteous, and Listening
- The total count of each column, the mean (or average), the standard deviation (a measure of how the data is distributed compared to the mean), minimum and maximum number in that column, and the percentiles, 25%, 50%, and 75% (a number that represents the data point at a certain percentage of the dataset).

```
# Find the summary statistics for numerical variables
df.describe()
```

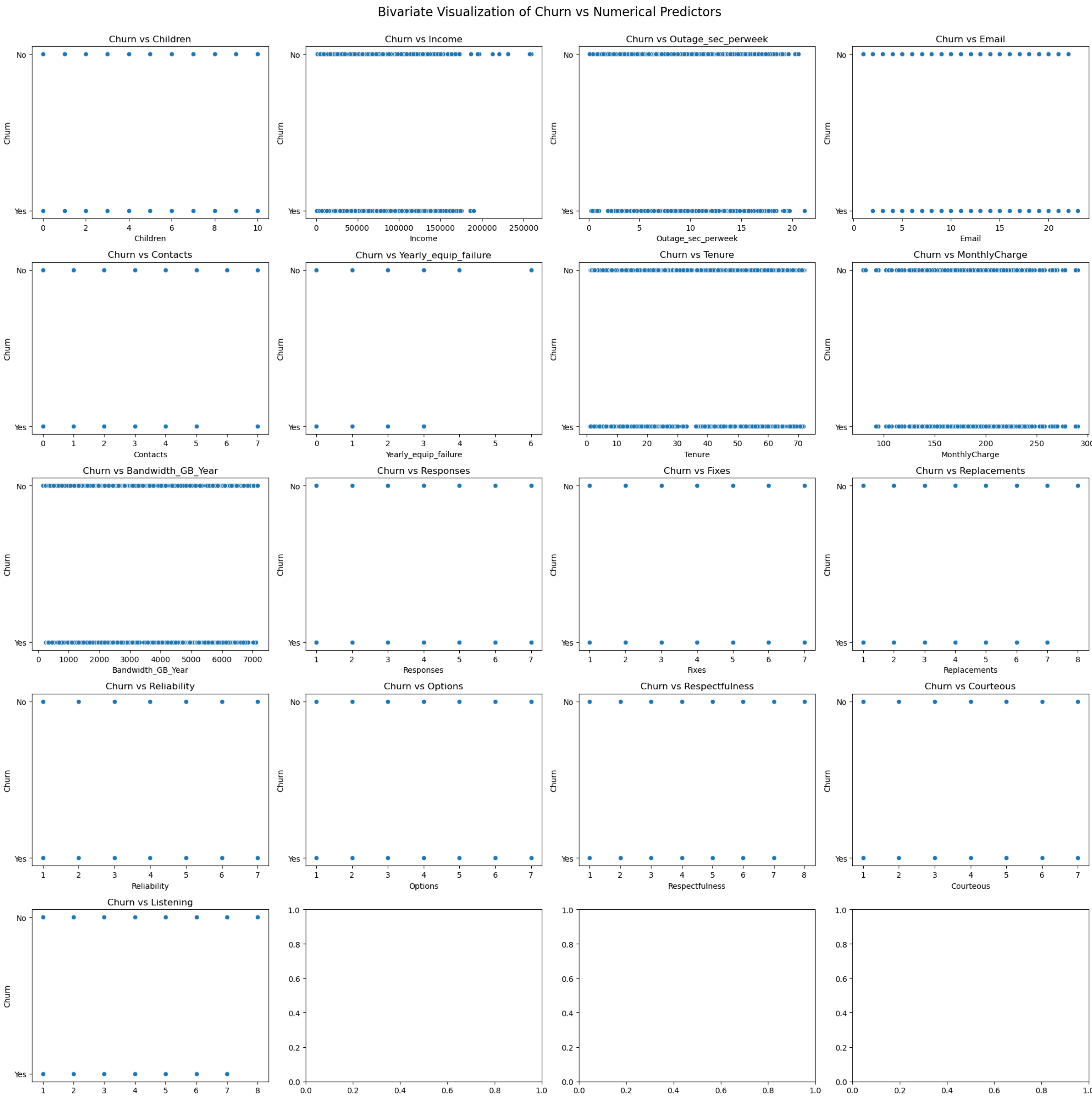
	Children	Income	Outage_sec_perweek	Email	Contacts	Yearly_equip_failure	Tenure	MonthlyCharge	Bandwidth_GB_Year	Responses	Fixes	Replacements	Reliability	Options	Respectfulness	Courteous	Listening
count	10000.0000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
mean	2.0877	39806.926771	10.001848	12.016000	0.994200	0.398000	34.526188	172.624816	3392.341550	3.490800	3.505100	3.487000	3.497500	3.492900	3.497300	3.509500	3.495600
std	2.1472	28199.916702	2.976019	3.025898	0.988466	0.635953	26.443063	42.943094	2185.294852	1.037797	1.034641	1.027977	1.025816	1.024819	1.033586	1.028502	1.028633
min	0.0000	348.670000	0.099747	1.000000	0.000000	0.000000	1.000259	79.978860	155.506715	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
25%	0.0000	19224.717500	8.018214	10.000000	0.000000	0.000000	7.917694	139.979239	1236.470827	3.000000	3.000000	3.000000	3.000000	3.000000	3.000000	3.000000	3.000000
50%	1.0000	33170.605000	10.018560	12.000000	1.000000	0.000000	35.430507	167.484700	3279.536903	3.000000	4.000000	3.000000	3.000000	3.000000	3.000000	4.000000	3.000000
75%	3.0000	53246.170000	11.969485	14.000000	2.000000	1.000000	61.479795	200.734725	5586.141370	4.000000	4.000000	4.000000	4.000000	4.000000	4.000000	4.000000	4.000000
max	10.0000	258900.700000	21.207230	23.000000	7.000000	6.000000	71.999280	290.160419	7158.981530	7.000000	7.000000	8.000000	7.000000	7.000000	8.000000	7.000000	8.000000

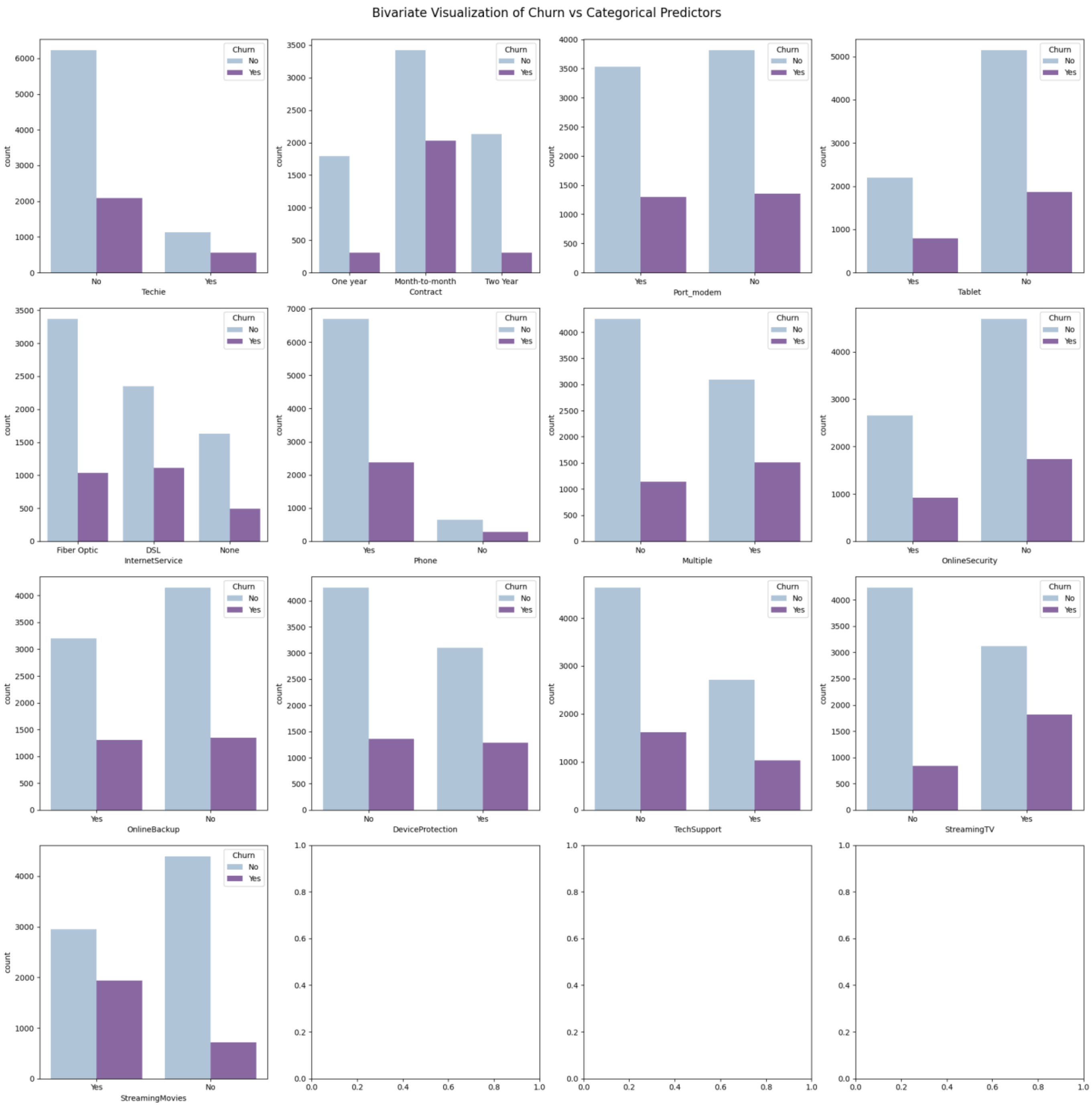
C3. <u>Univariate & Bivariate Visualizations</u>

Univariate Visualizations: I have provided a screenshot of the univariate visualizations of all 30 independent (predicting) variables and the 1 dependent (target) variable, Churn.



Bivariate Visualizations: I have provided a screenshot of the bivariate visualizations of all 30 independent (predicting) variables compared to the target variable, Churn. I chose to do scatterplots for the 17 numerical variables vs Churn and count plots for the 13 categorical variables vs Churn.





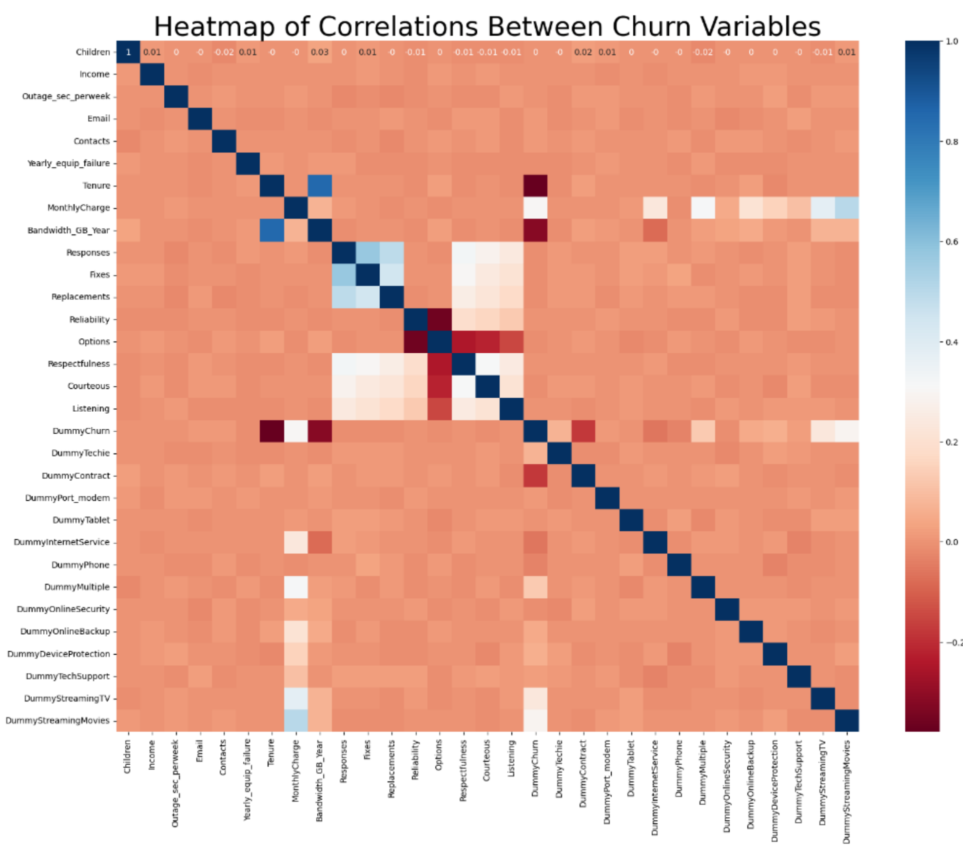
C4. <u>Data Transformation Goals & Steps</u>

Since my analysis does contain categorical variables, I have to re-express these variables as “dummy variables”. Dummy variables represent categorical variables as values of 0 or 1. For example, techie is categorized into 2 groups (yes or no). Therefore, 1 would represent “yes” and 0 would represent “no”.

For logistic regression, we would have to do one-hot encoding which is what I just mentioned where the categorical data is converted to binary numeric data. One-hot encoding follows the k-1 rule. We use k-1 to create a baseline to compare these dummy variables to all of the other dummy variables. Another example using this k-1 method would be internet service. In this analysis, internet service is categorized into 3 groups (fiber optic, DSL, or none), so 3-1=2 groups. As a result, 1 will represent "fiber optic" and 0 will represent everything else; in this case, "DSL" and "none". The same would be done for contract. There are 3 groups (month-to-month, one year, and two year). In this case, 1 will represent "two year" and 0 will represent everything else; "1 year" and "none". If we were to add an extra dummy to represent the third variable, this would cause an error . The data transformation steps can be found below:

1. Reformat the columns to have 3 decimal places.
2. Create dummy variables where "Yes" is represented by 1 & "No" is represented by 0.
3. Drop the original categorical columns so only dummy categorical columns are left.
4. Check the new data frame to make sure all the data transferred correctly.
5. Create a heatmap to check for multicollinearity.

- The heatmap helps to easily identify if there are any correlations between any of the independent variables by simply looking for the dark blue color. Based on the correlation matrix and heatmap, Tenure and Bandwidth_GB_Year seem to be correlated. This means these two variables are at risk of being dropped.



6. Extract the cleaned & wrangled dataset.

C5. <u>Prepared Data Set</u>

I will provide an attached copy of the prepared data set, named "log_clean.csv".

Part IV: Model Comparison & Analysis

D1. <u>Initial Multiple Logistic Regression</u>

In my Jupyter Notebook, I ran a model of all 30 of the independent variables identified in part C5. I constructed this model by using the statsmodel library with the Logit() function and printed the initial model summary using the summary() function.

```
# Set the dependent variable
y = df.DummyChurn

# Set the multiple independent variables
X = df[['Children','Income','Outage_sec_perweek','Email','Contacts','Yearly_equip_failure','Tenure','MonthlyCharge',
        'Bandwidth_GB_Year','Responses','Fixes','Replacements','Reliability','Options','Respectfulness','Courteous',
        'Listening','DummyTechie','DummyContract','DummyPort_modem','DummyTablet','DummyInternetService','DummyPhone',
        'DummyMultiple','DummyOnlineSecurity','DummyOnlineBackup','DummyDeviceProtection','DummyTechSupport',
        'DummyStreamingTV','DummyStreamingMovies']].assign(const=1)

model = sm.Logit(y, X)
results = model.fit()
print(results.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.272221
         Iterations 8
                           Logit Regression Results
==============================================================================
Dep. Variable:             DummyChurn   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9969
Method:                           MLE   Df Model:                           30
Date:                Tue, 25 Jun 2024   Pseudo R-squ.:                  0.5292
Time:                        21:46:58   Log-Likelihood:                -2722.2
converged:                       True   LL-Null:                       -5782.2
Covariance Type:            nonrobust   LLR p-value:                     0.000
=========================================================================================
                            coef    std err          z      P>|z|      [0.025      0.975]
-----------------------------------------------------------------------------------------
Children                 -0.0362      0.018     -2.057      0.040      -0.071      -0.002
Income                  1.66e-07   1.22e-06      0.136      0.892   -2.22e-06    2.56e-06
Outage_sec_perweek        0.0007      0.012      0.064      0.949      -0.022       0.023
Email                    -0.0022      0.011     -0.195      0.845      -0.024       0.020
Contacts                  0.0335      0.035      0.970      0.332      -0.034       0.101
Yearly_equip_failure     -0.0302      0.054     -0.558      0.577      -0.136       0.076
Tenure                   -0.1937      0.020     -9.622      0.000      -0.233      -0.154
MonthlyCharge             0.0340      0.004      7.626      0.000       0.025       0.043
Bandwidth_GB_Year         0.0012      0.000      4.967      0.000       0.001       0.002
Responses                -0.0214      0.049     -0.440      0.660      -0.117       0.074
Fixes                     0.0239      0.046      0.519      0.604      -0.066       0.114
Replacements             -0.0169      0.042     -0.403      0.687      -0.099       0.065
Reliability              -0.0206      0.037     -0.553      0.580      -0.093       0.052
Options                  -0.0339      0.039     -0.872      0.383      -0.110       0.042
Respectfulness           -0.0337      0.040     -0.843      0.399      -0.112       0.045
Courteous                 0.0071      0.038      0.187      0.852      -0.067       0.082
Listening                -0.0070      0.036     -0.194      0.846      -0.077       0.063
DummyTechie               0.8053      0.089      9.037      0.000       0.631       0.980
DummyContract            -2.2781      0.103    -22.156      0.000      -2.480      -2.077
DummyPort_modem           0.1578      0.069      2.303      0.021       0.024       0.292
DummyTablet              -0.0814      0.074     -1.094      0.274      -0.227       0.064
DummyInternetService     -1.1619      0.170     -6.830      0.000      -1.495      -0.828
DummyPhone               -0.3322      0.117     -2.849      0.004      -0.561      -0.104
DummyMultiple             0.1210      0.152      0.794      0.427      -0.178       0.420
DummyOnlineSecurity      -0.2886      0.073     -3.928      0.000      -0.433      -0.145
DummyOnlineBackup        -0.2259      0.112     -2.013      0.044      -0.446      -0.006
DummyDeviceProtection    -0.2423      0.083     -2.912      0.004      -0.405      -0.079
DummyTechSupport         -0.1861      0.091     -2.056      0.040      -0.364      -0.009
DummyStreamingTV          0.5809      0.180      3.219      0.001       0.227       0.935
DummyStreamingMovies      0.7448      0.220      3.378      0.001       0.313       1.177
const                    -4.5577      0.490     -9.309      0.000      -5.517      -3.598
-----------------------------------------------------------------------------------------
```

D2. Justification of Model Reduction

To reduce the model, I first had to find the VIF of all of the independent variables to see which variables should be eliminated due to high multicollinearity. After I found the VIF, Bandwidth_GB_Year, MonthlyCharge, Responses, Fixes, Respectfulness, Email, Replacements, Listening, Courteous, and Outage_sec_perweek were removed one by one since they had VIF values greater than 10.

After that, I performed backward stepwise elimination. This is an important reduction method because it not only reduces the number of predictors, but also helps to resolve overfitting and remove the predictors that do not significantly affect the target variable. For this reduction method, I had to remove the least significant features based on their p-values one at a time. Therefore, if a variable had a p-value greater than 0.05, it was removed.

The following variables are what is left in the reduced linear regression model after the removal due to their p-values and insignificance to the analysis:
'Tenure', 'DummyTechie', 'DummyContract', 'DummyPort_modem', 'DummyInternetService', 'DummyPhone', 'DummyMultiple', 'DummyOnlineBackup', 'DummyDeviceProtection', 'DummyTechSupport', 'DummyStreamingTV', and 'DummyStreamingMovies'

D3. Reduced Logistic Regression Model

```python
# Run the model after the removal of "DummyOnlineSecurity" since it had the highest p-value (0.209)
y = df.DummyChurn
X = df[['Tenure','DummyTechie','DummyContract','DummyPort_modem','DummyInternetService',
        'DummyPhone','DummyMultiple','DummyOnlineBackup','DummyDeviceProtection',
        'DummyTechSupport','DummyStreamingTV','DummyStreamingMovies']].assign(const=1)

model = sm.Logit(y, X)
results = model.fit()
print(results.summary())

# This is the final reduced model because there are no p-values > 0.05
```

```
Optimization terminated successfully.
         Current function value: 0.282023
         Iterations 8
                           Logit Regression Results
==============================================================================
Dep. Variable:             DummyChurn   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9987
Method:                           MLE   Df Model:                           12
Date:                Tue, 25 Jun 2024   Pseudo R-squ.:                  0.5123
Time:                        22:23:48   Log-Likelihood:                -2820.2
converged:                       True   LL-Null:                       -5782.2
Covariance Type:            nonrobust   LLR p-value:                     0.000
=========================================================================================
                            coef    std err          z      P>|z|      [0.025      0.975]
-----------------------------------------------------------------------------------------
Tenure                   -0.0866      0.002    -41.915      0.000      -0.091      -0.083
DummyTechie               0.8126      0.088      9.191      0.000       0.639       0.986
DummyContract            -2.1365      0.097    -22.107      0.000      -2.326      -1.947
DummyPort_modem           0.1484      0.067      2.202      0.028       0.016       0.280
DummyInternetService     -0.6331      0.069     -9.157      0.000      -0.769      -0.498
DummyPhone               -0.3692      0.114     -3.237      0.001      -0.593      -0.146
DummyMultiple             1.3152      0.071     18.471      0.000       1.176       1.455
DummyOnlineBackup         0.6314      0.068      9.228      0.000       0.497       0.765
DummyDeviceProtection     0.2738      0.068      4.051      0.000       0.141       0.406
DummyTechSupport          0.2424      0.069      3.494      0.000       0.106       0.378
DummyStreamingTV          2.2921      0.079     28.992      0.000       2.137       2.447
DummyStreamingMovies      2.7526      0.082     33.443      0.000       2.591       2.914
const                    -2.0937      0.152    -13.804      0.000      -2.391      -1.796
=========================================================================================
```

E1. <u>Model Comparison</u>

Model Evaluation Metrics	Initial Model	Reduced Model
Log-Likelihood	-2722.2	-2820.2
LLR p-value	0.00	0.00
Number of Independent Variables	30	12

As stated in part D2, I found the variance inflation factors of all of the independent variables to determine which variables had the highest VIF that was causing multicollinearity. After removing one of the expected variables, it affected other variables' VIFs, so they had to be removed from the analysis too. Then I performed backward stepwise elimination to remove several more variables with p-values greater than 0.05. These reduction methods removed variables one-by-one to see how the removal of an insignificant variables would affect the other variables VIF and p-values.
All in all, these reduction methods reduced the number of independent variables from 30 to 12. Removing over half of these variables caused the Log-Likelihood value to decrease, but removing predictor values causes the decrease no matter if the variables are statistically significant or not. Therefore, it is only a fair comparison if both the models had the same number of predictor variables. Between the initial and reduced model, the LLR p-value remained at 0, implying that the regressions are meaningful. The closer this value is to 0, the better fitting the model is.

E2. <u>Output & Calculations</u>

Confusion Matrix & Accuracy Calculation on reduced model:

```
# Split the datset
y = df.DummyChurn
X = df[['Tenure','DummyTechie','DummyContract','DummyPort_modem','DummyInternetService',
        'DummyPhone','DummyMultiple','DummyOnlineBackup','DummyDeviceProtection',
        'DummyTechSupport','DummyStreamingTV','DummyStreamingMovies']]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)

# Create the confusion matrix of the reduced model
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
y_pred = logreg.predict(X_test)
print('Accuracy of logistic regression classifier on test set: {:.2f}%'.format(logreg.score(X_test, y_test)*100))
cm = confusion_matrix(y_test, y_pred)
print(cm)
```

```
Accuracy of logistic regression classifier on test set: 86.03%
[[2012  189]
 [ 230  569]]
```

E3. <u>Copy of Code</u>

Provided as the file “D208 Task 2.ipynb”.

Part V: Data Summary & Implications

F1. <u>Data Analysis Results</u>

The regression equation for the reduced model:
y = -2.0937 - 0.0866(Tenure) + 0.8126(Techie) - 2.1365(Contract) + 0.1484(Port_modem) -0.0136(InternetService) - 0.3692(Phone) + 1.3152(Multiple) + 0.6314(OnlineBackup) + 0.2738(DeviceProtection) + 0.2424(TechSupport) + 2.2921(StreamingTV) + 2.7526(StreamingMovies)

An interpretation of the coefficients of the reduced model:

- y represents customer churn.
- Keeping all things constant, for one unit of increase in tenure, the log odds of customer churn decrease by 8.66%.
- Keeping all things constant, customers who considered themselves techies increase the log odds of customer churn by 81.26%.
- Keeping all things constant, customers with a two-year contract decrease the log odds of customer churn by 213.65%.
- Keeping all things constant, customers with port-modem increase the log odds of customer churn by 14.84%.
- Keeping all things constant, customers with fiber optic internet service decrease the log odds of customer churn by 1.36%.
- Keeping all things constant, customers with phone service decrease the log odds of customer churn by 36.92%.
- Keeping all things constant, customers with multiple services increase the log odds of customer churn by 131.52%.
- Keeping all things constant, customers with online backup increase the log odds of customer churn by 63.14%.
- Keeping all things constant, customers with device protection decrease the log odds of customer churn by 27.38%.
- Keeping all things constant, customers who need technical support increase the log odds of customer churn by 24.24%.
- Keeping all things constant, customers who have streaming TV increase the log odds of customer churn by 229.21%.
- Keeping all things constant, customers who have streaming movies increase the log odds of customer churn by 275.26%.

Statistical and practical significance of the reduced model:
The reduced model and results are statistically significant because the LLR p-value is 0 and the accuracy of the model being 86%. Since the LLR p-value is 0, this means that the reduced model is a good fitting model. This also means that there is much confidence that the results are not based on luck. Although some variables I feel would have been important for the analysis were dropped, I agree with all of the variables left in the analysis. However, I do not think the reduced model is practically significant. Based on the interpretation of the coefficients, some of the results don't make sense. For example, customers who have multiple services and have the streaming TV and movies option have a significant increase in log odds of customer churn. However, it seems like customers

with these variables are loyal customers so why would they be more likely to leave the company? It seems that any customer who pays for an extra service or add-on, excluding phone services and device protection, is more likely to leave the telecommunications company. Therefore, in my opinion, the reduced model is not practically significant.

Disadvantages of the data analysis:
During the data reduction process, I feel as if some of the variables that were dropped from the analysis due to their p-values or variance inflation factors, would have been important to use for the analysis in the real world. For example, I thought monthly charge would be important for the analysis. Another disadvantage is the data set size. Logistic regression favors large datasets and if there aren't enough variables, this could lead to overfitting. As mentioned above, the reduced regression model seems to be a good fitting model according to the model evaluation metrics. But realistically, it is not practical. I feel as if the model's predicted probabilities don't necessarily follow the actual probabilities.

F2. Recommendations

Regarding the practical significance of the model, I do believe it is an impractical model based on some of the results from the model summary and some of the explanatory variables that were removed due to high p-values or VIFs. However, as mentioned above, it is considered statistically significant. This model is a decent starting point since it is 86% accurate and most of the data in the summary results makes sense regarding the variables included in the analysis. I do recommend gathering more customer records next time for a logistic regression model since this type of model favors larger data sets. Some recommendations based on the model results would be to try to persuade customers to get the fiber optic internet service. It seems that customers who choose that option rather than no internet service or DSL are more likely to stay with the company. The same goes for customers who opt in for device protection and a two-year contract.

Part VI: Demonstration

G. Panopto Video

H. Third-Party Code Sources

N/A

I. Sources

Bobbitt, Z. (2020, October 13). The 6 assumptions of logistic regression (with examples). Statology. https://www.statology.org/assumptions-of-logistic-regression/

R or python. Western Governors University. (2023, July 7). https://www.wgu.edu/online-it-degrees/programming-languages/r-or-python.html

J. Professional Communication

Demonstrate professional communication in the content and presentation of your submission.