Matthew Blessing
Student ID: ***000507424***
D208 Predictive Modeling

Performance Assessment Task 2: Predictive Modeling - NBM2
Logistic Regression for Predictive Modeling

Part I: Research Question

A1.

Using the provided 'Churn' data set, we will utilize a logistic regression model to attempt to predict which customers will churn based on a set of independent variables, 'Age', 'Children', 'Gender', 'Income', 'MonthlyCharge' and 'Tenure'.

'Churn', as defined in the provided data dictionary, is whether the customer has discontinued service within the last month.

A2.

Understanding which factors lead to customer churn is beneficial for the business as they can use this information to be able to more accurately predict which customers are more likely to churn. With this information, the business can predict future revenue and make appropriate business decisions that impact customers, such as infrastructure upgrades, and better target marketing campaigns to attract, or retain, specific types of customers.

Part II: Method Justification

B1.

A logistic regression model generally assumes the following:

- Binary logistic regression requires our dependent variable to be binary
- Ordinal logistic regression requires our dependent variable to be ordinal
- Observations must be independent of one another
- Independent variables should have little or no multicollinearity , the independent variables should not be highly correlated with one another
- There are no extreme outliers in the data
- There is a linear relationship between the explanatory variables and the logit of the response variable
- The sample size of the data set is large enough to draw accurate conclusions from the model (*Assumptions of Logistic Regression*, n.d.)

B2.
For this analysis, we will be utilizing Python within Jupyter Notebooks. Python is an object-oriented programming language that is extremely popular for data science due to it being a powerful, easy to learn language that is extremely expandable with a large library of data science packages, such as NumPy, SciPy, Pandas, and Matplotlib. These libraries easily allow users to implement classification, regression, machine learning and more on chosen data sets (*Advantages of Learning Python for Data Science*, n.d.).

B3.
A logistic regression model should be useful in this analysis because our dependent variable is binary and we will be looking to predict whether a customer will churn using binary classification based on the input of several independent variables, such as age, number of children, income, etc.

Part III: Data Preparation

C1.
Before we can create a logistic regression model, there are some preparations required for the data.

To begin, we will want to ensure the data is clean by removing duplicate rows of data using the drop_duplicates() function. The data will also be checked for null values using the isnull().sum() function. Luckily, the data appears to be clean with no null values or duplicate rows.

We will want to change the values of our dependent variable, 'Churn' from 'Yes' and 'No' to 1 and 0. We can do this quickly using Pandas *.replace()* function.

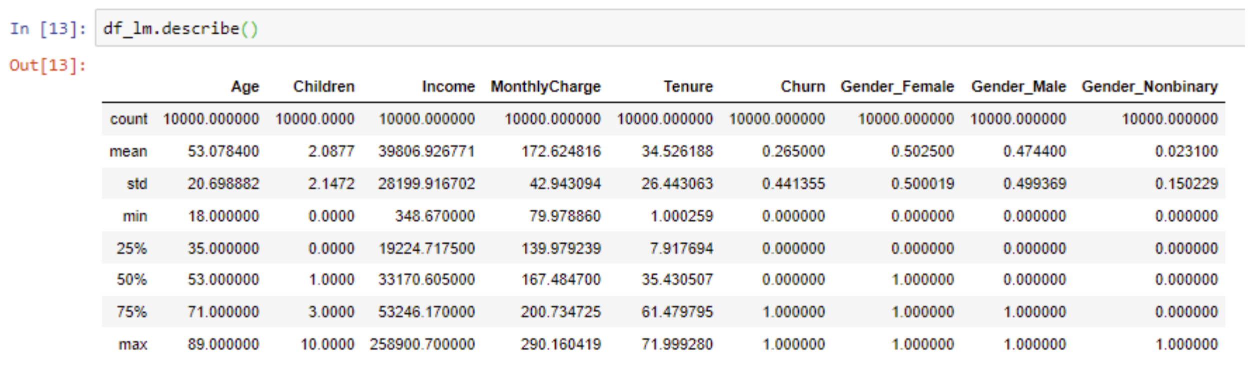
We will also want to utilize dummy variables since we will be utilizing categorical variables in our model. This can be done using Pandas get_dummies() function to convert the categorical variables to binary variables (1 = presence of the categorical variable, 0 = absence of the categorical variable).

We will also need to explore the variables we want to use for our model to ensure they fit the assumptions of the logistic regression model stated above. Once we have determined all of the variables we will want to use for the model, we will create a new dataframe containing only the variables we will be using for our model.

C2.
To answer the question of whether we can use age, children, income, gender identity, monthly charge, and tenure to predict if a customer will churn, we will need to first look at some summary statistics to understand the distributions of our independent and dependent variables.

We will want to look at histograms to view the distribution of variables, box plots to easily spot outliers and means, and bivariate scatter plots to see relationships between our independent and dependent variables. Lastly, we will want to compare all of our independent variables to look for multicollinearity between them, and if we discover multicollinearity, we will need to reframe our research question and select new variables.

Before we begin preparing our data for our model, we should explore our independent and dependent variables to explore the number of observations, measures of central tendency. We can get a good look at the measures of central tendency with our selected variables by using Panda's .describe() function.



```
In [13]: df_lm.describe()
Out[13]:
```

	Age	Children	Income	MonthlyCharge	Tenure	Churn	Gender_Female	Gender_Male	Gender_Nonbinary
count	10000.000000	10000.0000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
mean	53.078400	2.0877	39806.926771	172.624816	34.526188	0.265000	0.502500	0.474400	0.023100
std	20.698882	2.1472	28199.916702	42.943094	26.443063	0.441355	0.500019	0.499369	0.150229
min	18.000000	0.0000	348.670000	79.978860	1.000259	0.000000	0.000000	0.000000	0.000000
25%	35.000000	0.0000	19224.717500	139.979239	7.917694	0.000000	0.000000	0.000000	0.000000
50%	53.000000	1.0000	33170.605000	167.484700	35.430507	0.000000	1.000000	0.000000	0.000000
75%	71.000000	3.0000	53246.170000	200.734725	61.479795	1.000000	1.000000	1.000000	0.000000
max	89.000000	10.0000	258900.700000	290.160419	71.999280	1.000000	1.000000	1.000000	1.000000

As we can see, our selected variables for our model include seven variables - six independent variables - Age, Children, Income, MonthlyCharge, Tenure, Gender (Note that in the above screenshot, 'Gender' has been broken out into dummy variables for input into our model, the process for this is discussed below in section C3, the 'Gender' variable specifics are discussed below), and one dependent variable, 'Churn', that has also been broken out into dummy variables in the screenshot above, but will be discussed below. Some notes on our variables:

- All of our selected variables have 10,000 entries in this data set
- 'Age' is a continuous variable with a mean of 53.07 years
- 'Children' is a discrete variable with a mean of 2.087 children
- 'Income' is a continuous variable with a mean of $39,808

- ‘MonthlyCharge’ is a continuous variable with a mean of $172.6
- ‘Tenure’ is a discrete variable, with a mean of 34.5 months

Since ‘Gender’ is a nominal variable, we will have to look at the mode as a measure of central tendency. We can look at this using Panda’s .value_counts() function on the ‘Gender’ column with the following code:

```
df.Gender.value_counts()

Female 5025
Male 4744
Nonbinary 231
Name: Gender, dtype: int64
```

As we can see, there are 10,000 entries for ‘Gender’, with 5,025 Female, 4,744 Male, and 231 Nonbinary.

With ‘Churn’ being a binary variable, we will follow a similar process with ‘Gender’ and rely on the mode as the measure of central tendency using the following code:

```
df_lm.Churn.value_counts()

0    7350
1    2650
Name: Churn, dtype: int64
```

As we can see, there are 10,000 entries for ‘Churn’, with 7,350 No (0) and 2,650 Yes (1).

C3.
To prepare our data for the analysis, we begin by importing our data science libraries and tools we will use for the analysis, which include Pandas, NumPy, Matplotlib, StatsModel and Seaborn:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import statsmodels.api as sm
sns.set_style('darkgrid')

#sets the jupyter notebook window to take up 90% width of the
browser window
from IPython.core.display import display, HTML display(HTML(""))
```

We then read in the data into a dataframe via Pandas with the following code:

```
df = pd.read_csv('churn_clean.csv')
```

We then drop any duplicate rows:

```
#drops any duplicate rows
df.drop_duplicates()
```

We also check for any null values:

```
#gives us the count of any null values in the data
df.isnull().sum()
```

We then replace the 'Yes' and 'No' values in the 'Churn' column with 1 and 0, where a 1 = 'Yes', and a 0 = 'No':

```
#replaces the 'Yes' and 'No' values in the 'Churn' column with 1
and 0, 1=Yes, 0=No
df.Churn.replace(('Yes', 'No'), (1, 0), inplace=True)
```

We then create a new dataframe containing only our independent and dependent variables:

```
#creates a new dataframe containing only the columns 'Age',
'Children', 'Gender', 'Income', 'MonthlyCharge', 'Tenure', and
'Churn' by copying those columns and values from the existing
dataframe
df_lm = df[['Age','Children', 'Gender', 'Income',
'MonthlyCharge', 'Tenure', 'Churn']].copy()
```

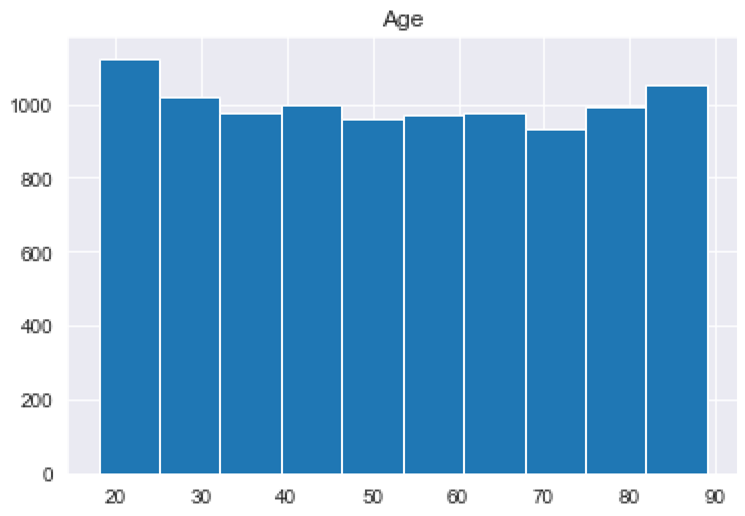
We will need to create dummy variables for our categorical variable (Gender), in order to more accurately use these variables in our model:

```
#creates dummy variables to convert the categorical variables to
binary values in order to more accurately use them in our model
df_lm = pd.get_dummies(data=df_lm, drop_first=False) df_lm.head()
```

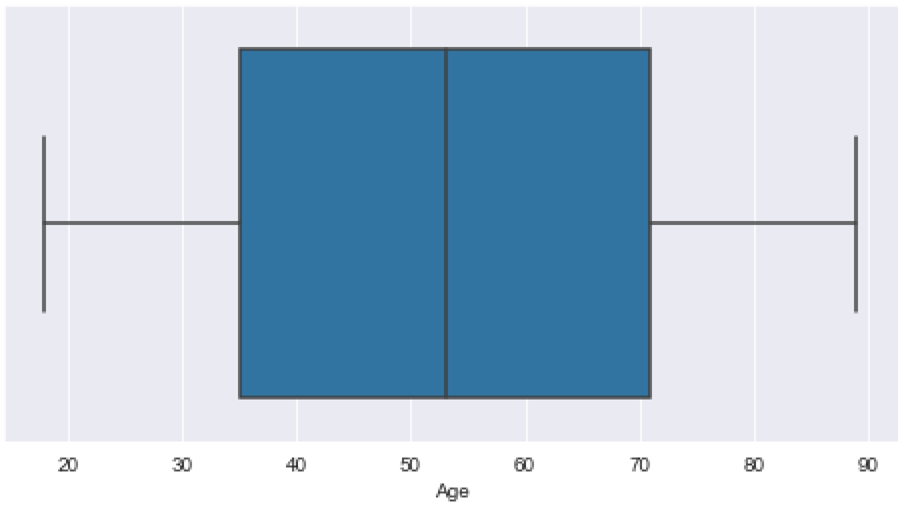
We also export the newly created dataframe to a .csv:

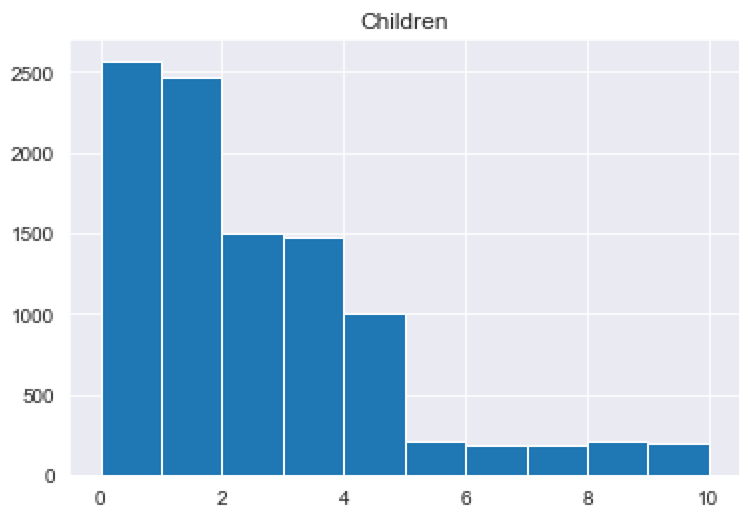
```
#exports the new dataframe to a .csv
df_lm.to_csv('D208_Data_set.csv')
```

C4.
To explore our data, we will look at univariate and bivariate plots utilizing histograms, box plots, and scatterplots to be able to more easily understand variable distribution, interquartile ranges, and how our independent variables relate to our dependent variable.

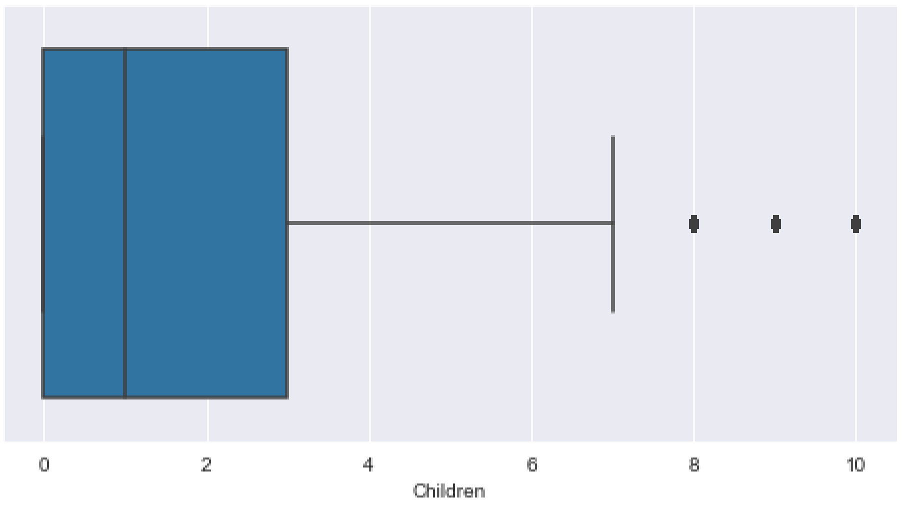


‘Age’ has a relatively uniform distribution

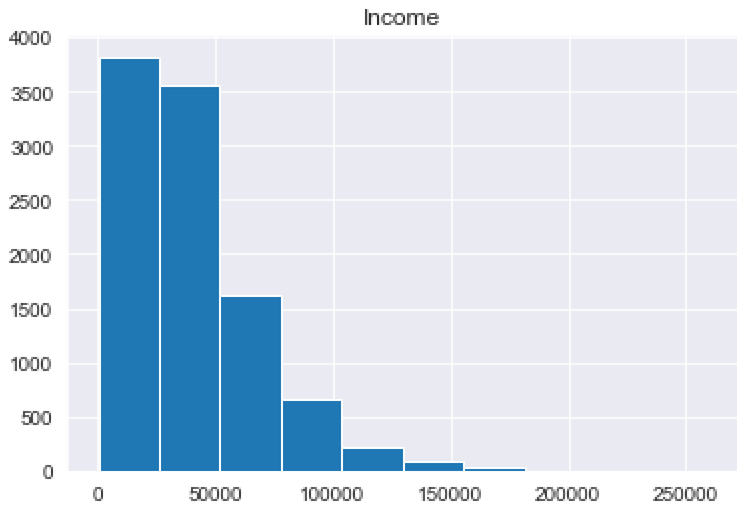




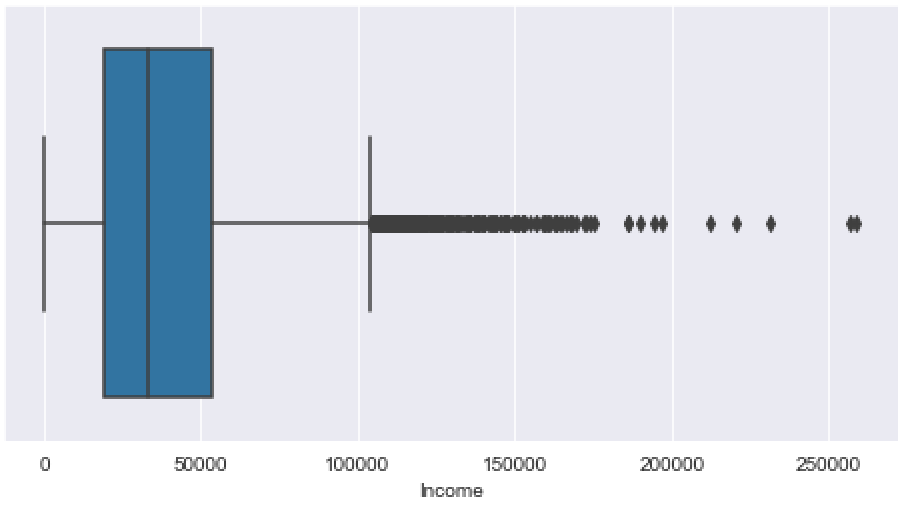
‘Children’ has a right-skewed distribution, which makes sense and is to be expected.



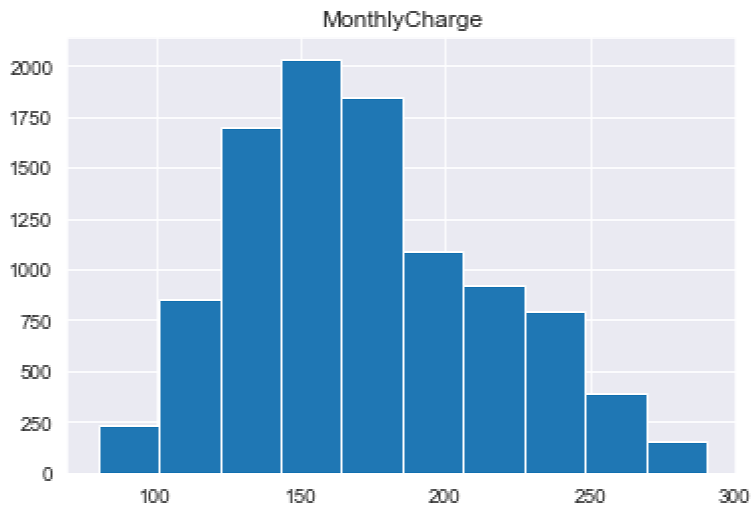
Some outliers here, but that is to be expected.



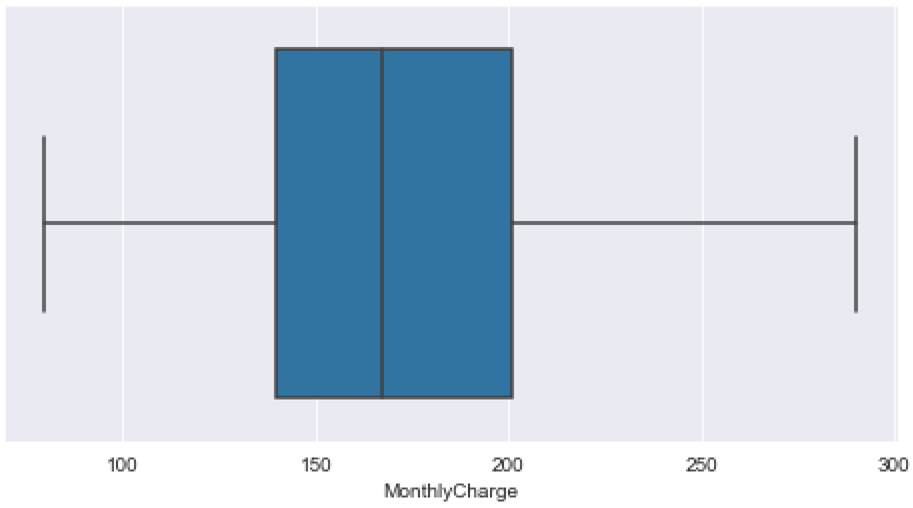
‘Income’ is also right-skewed in its distribution, but this is also to be expected.

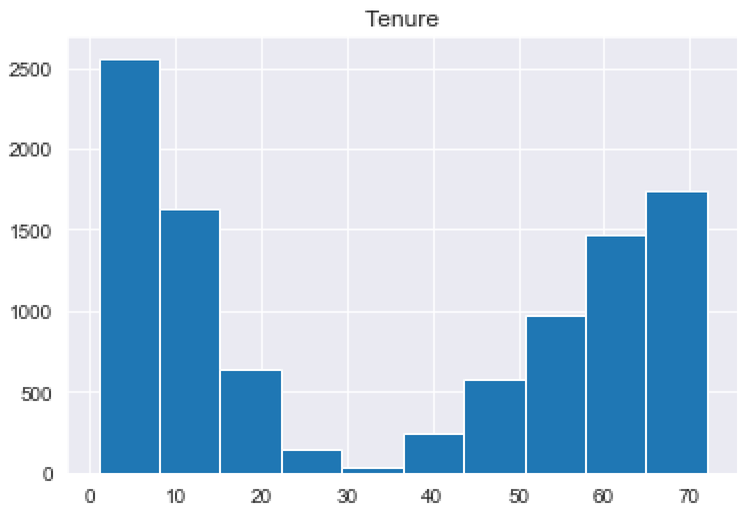


‘Income’ also has several outliers, but this is to be expected.

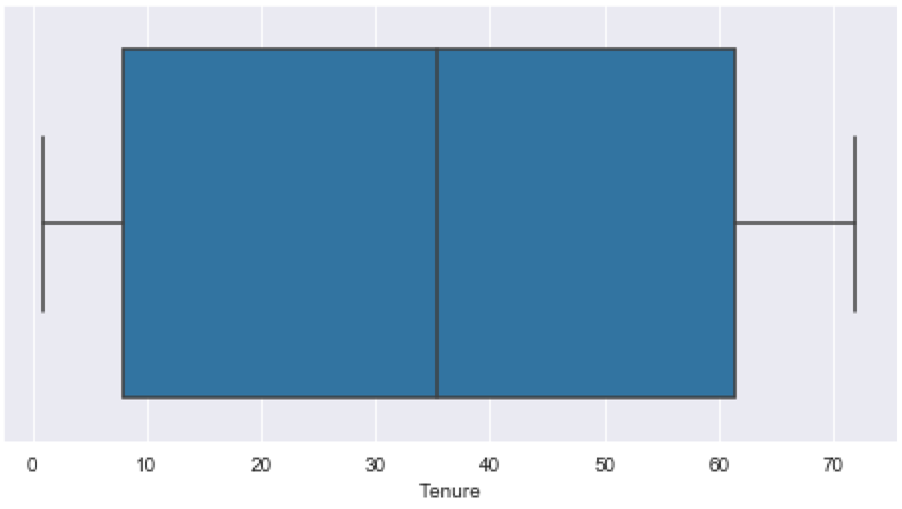


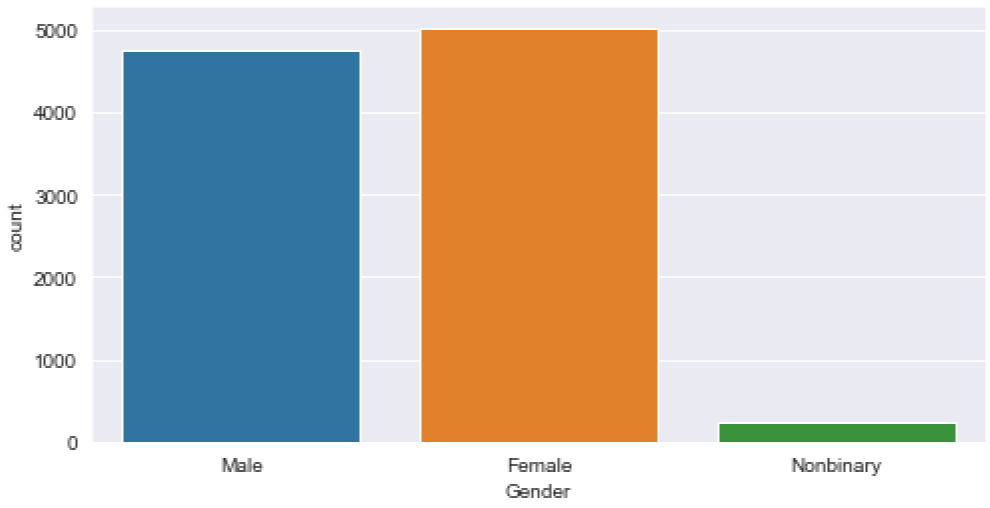
‘MonthlyCharge’ has a nearly normal distribution.



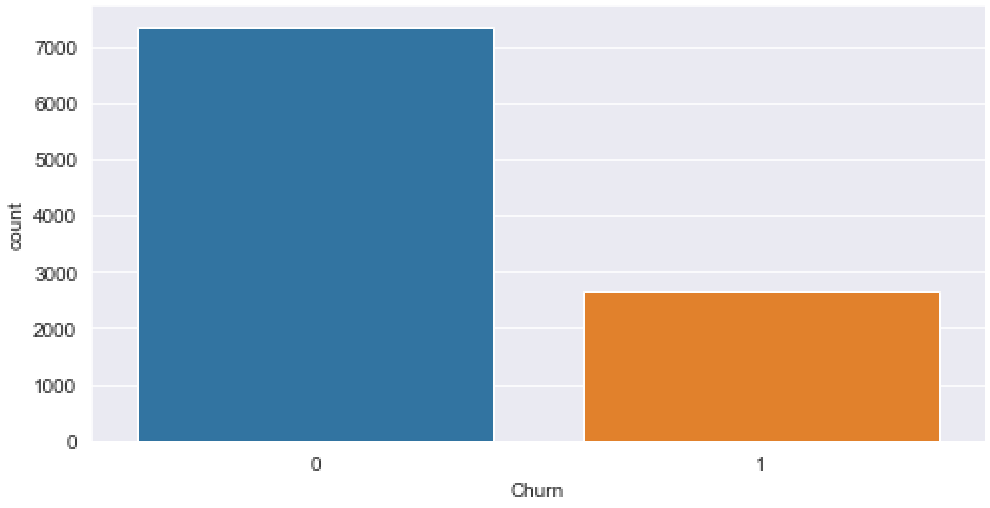


‘Tenure’ has a bimodal distribution.





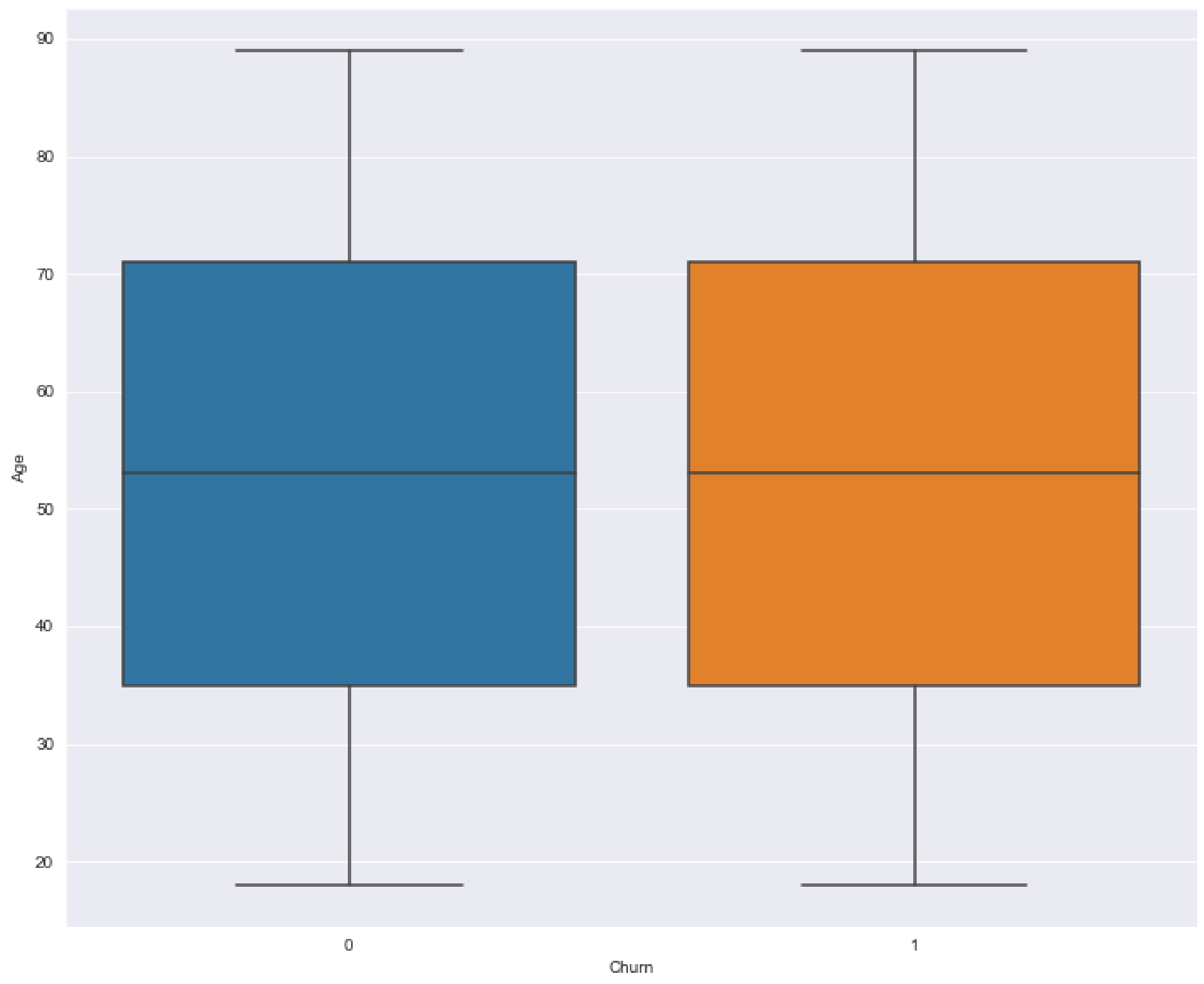
A countplot of 'Gender' reveals that Female customers slightly outnumber Male, followed by Nonbinary.



A countplot of 'Churn' reveals a significantly higher number of 'No' churn

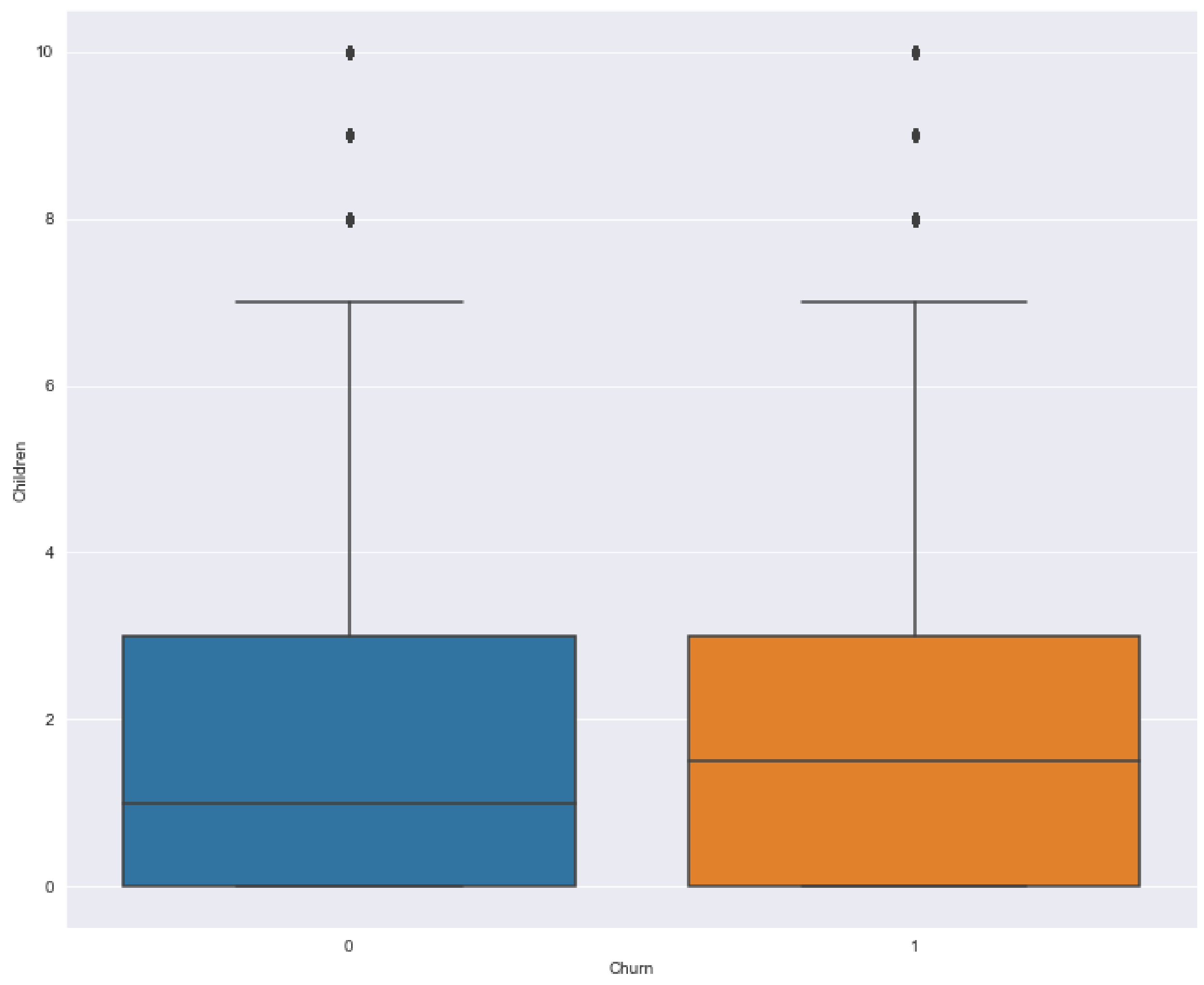
Now that we have seen our univariate distributions, let's look at how our independent variables compare to our dependent variable.

Churn vs. Age



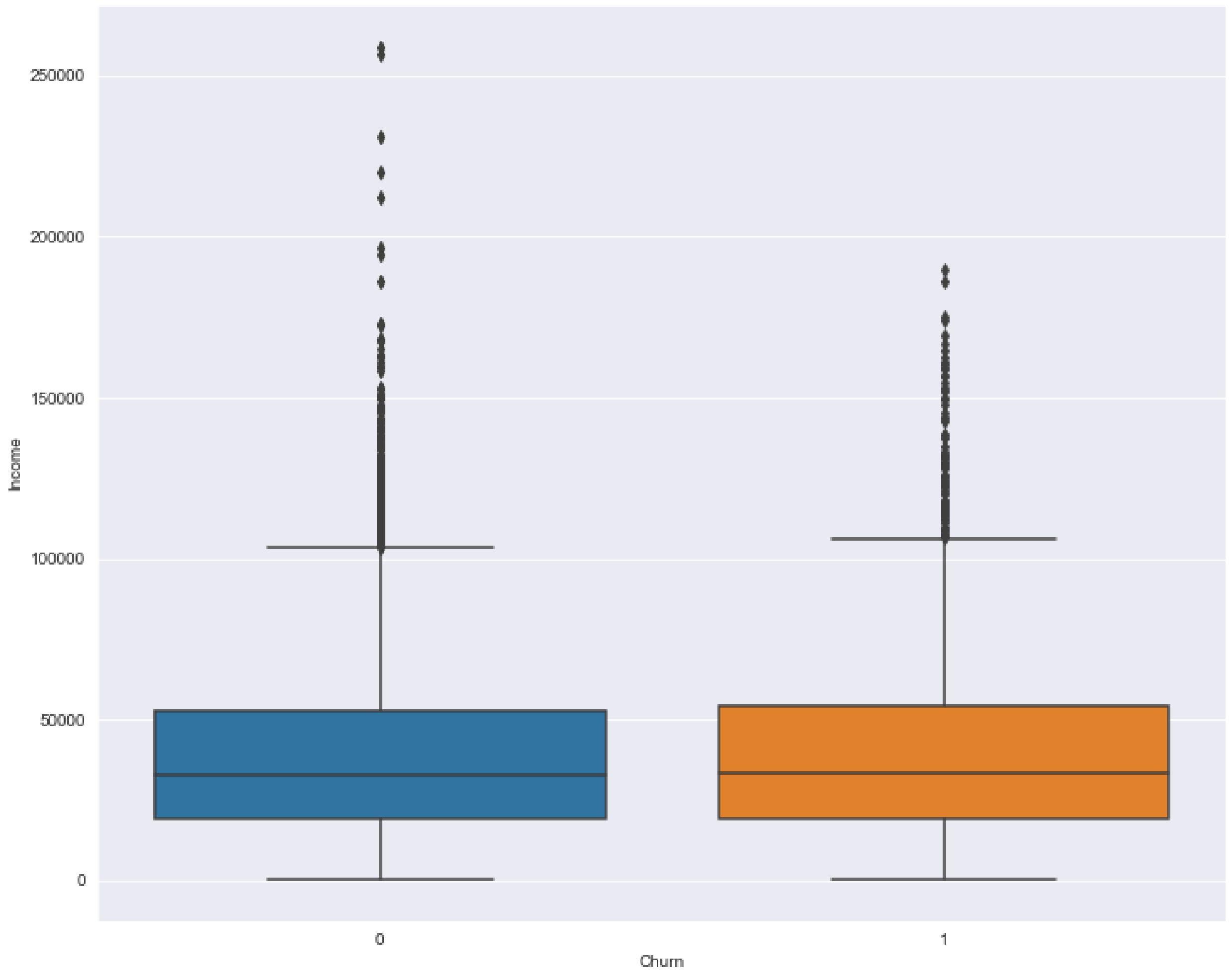
We see a relatively even distribution between customers who churn and those who do not within the same age range.

Churn vs. Children



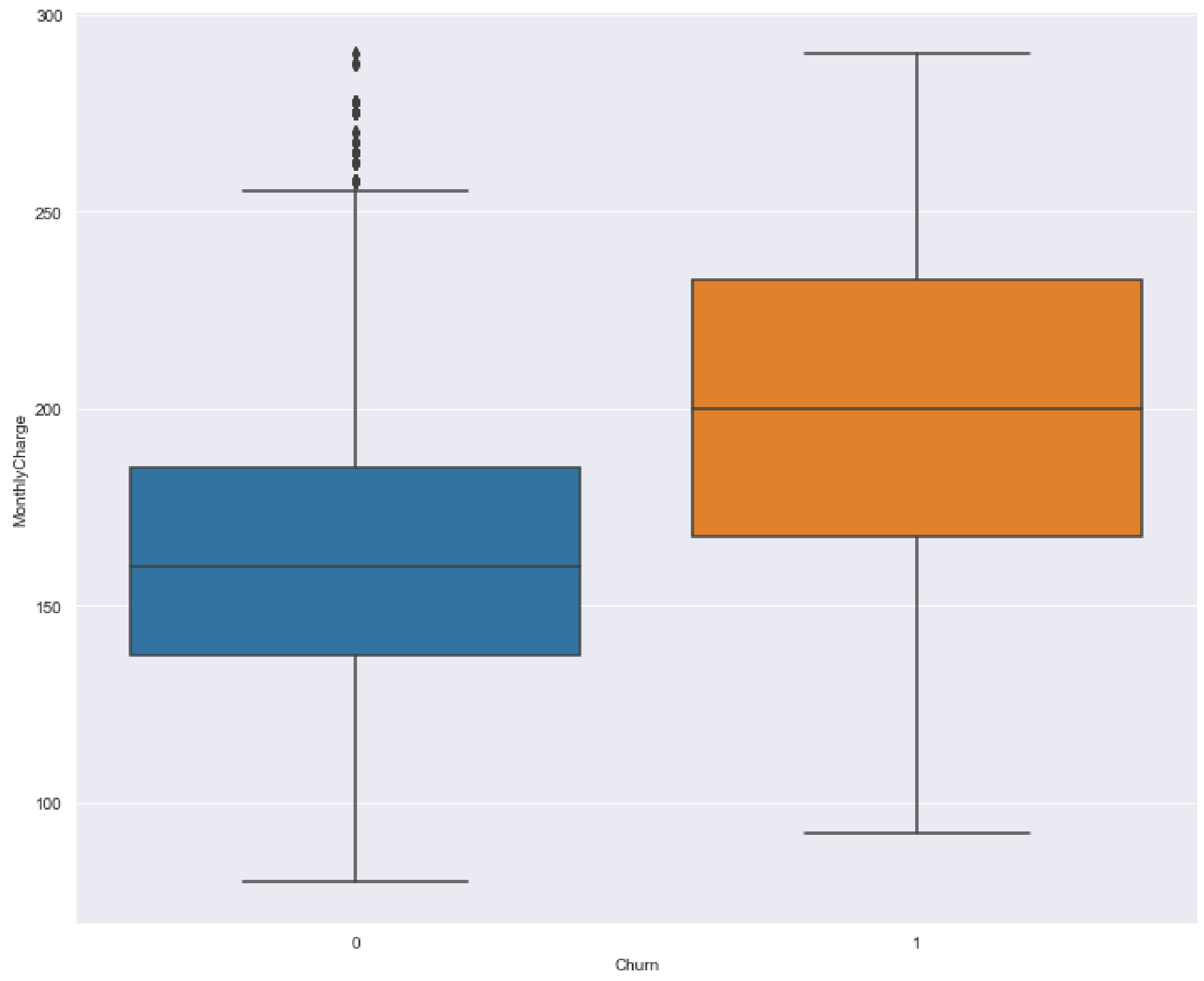
We can see customers who churn have slightly higher mean in terms of children.

Churn vs. Income



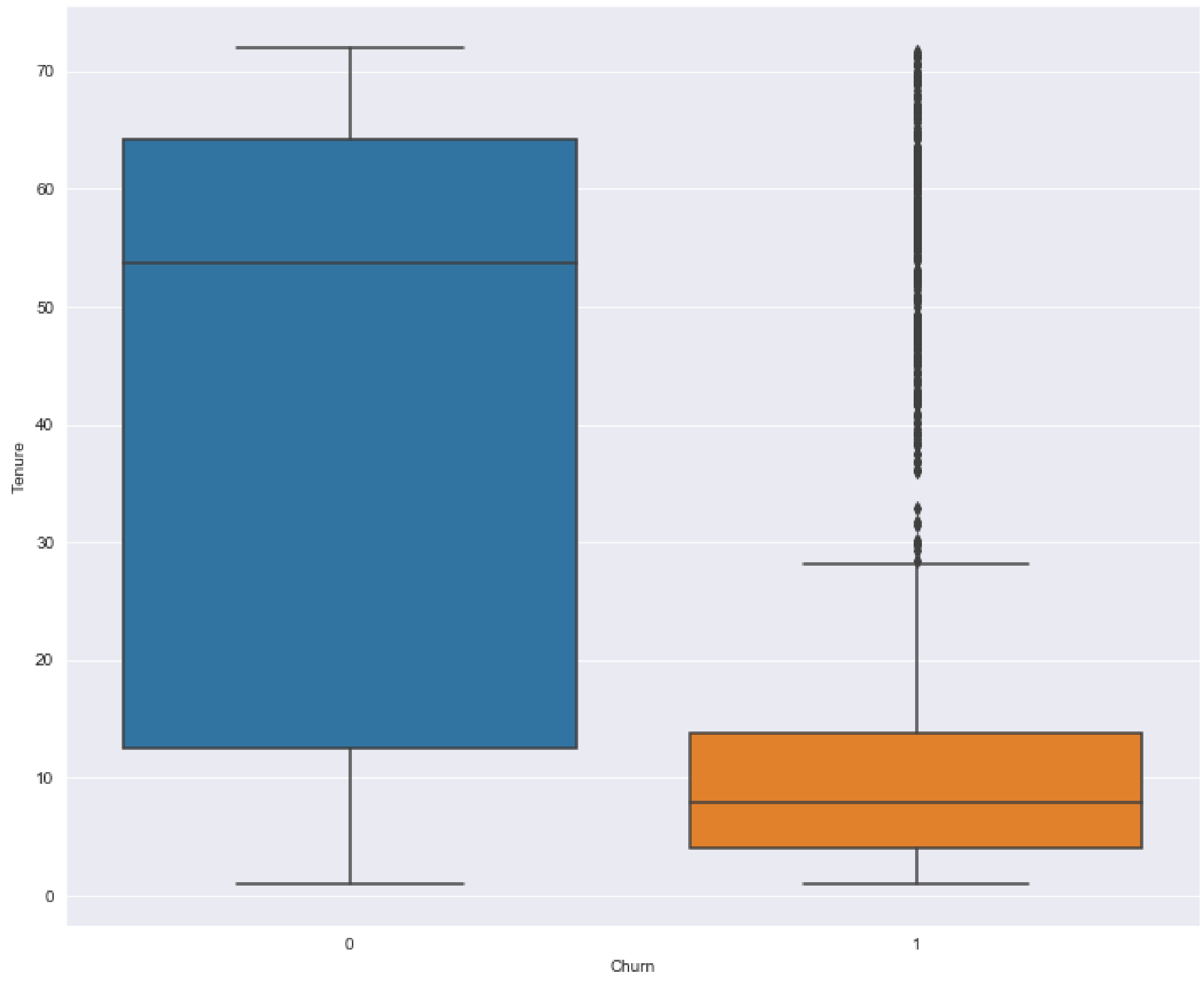
We can see here that the customers with large outlier incomes churn less.

Churn vs. MonthlyCharge



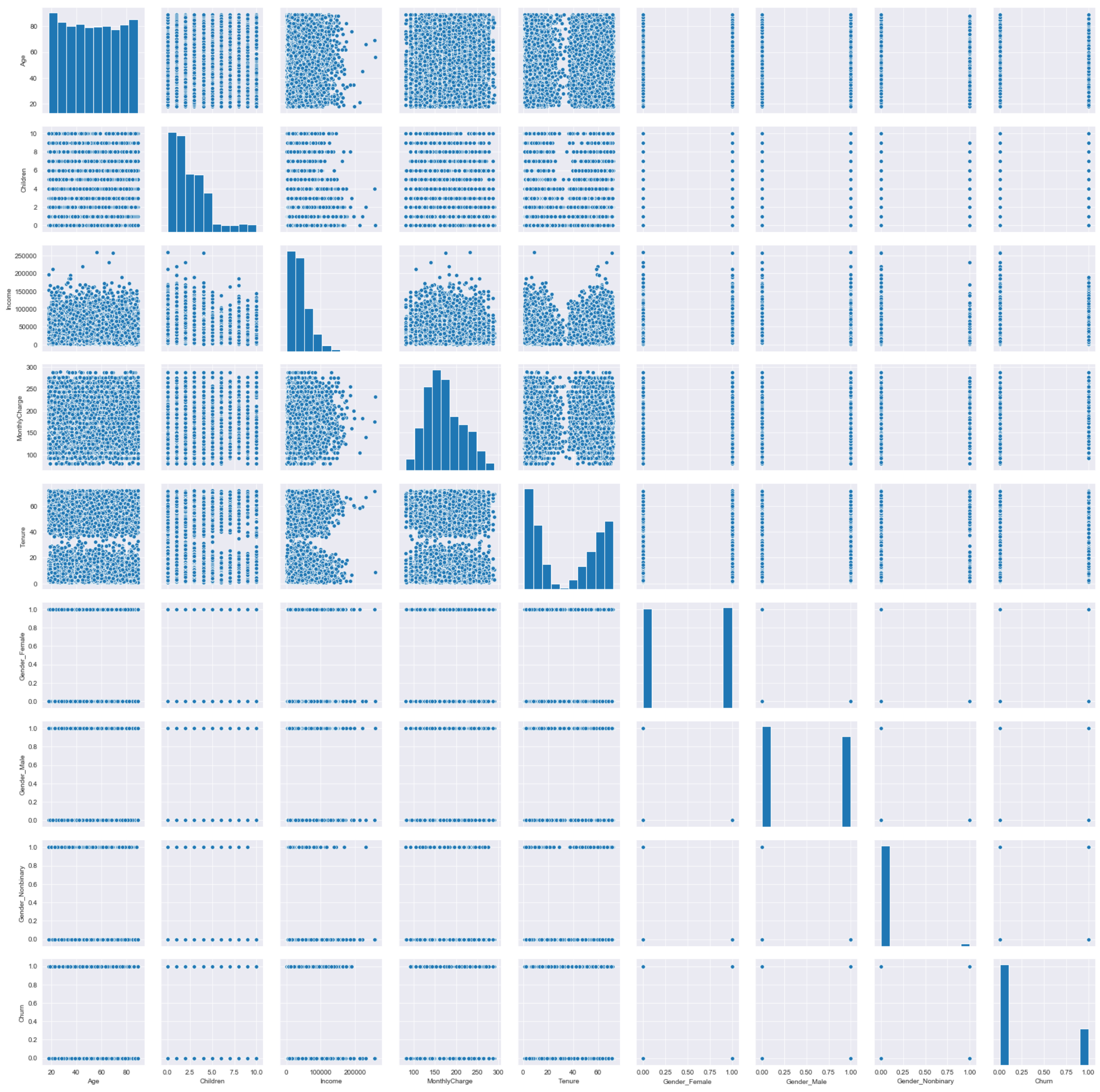
We can see that customers with a larger average MonthlyCharge churn more.

Churn vs. Tenure



It seems that customers with a longer average tenure churn less.

Lastly, we will utilize a pairplot to visualize relationships between all of our variables, which will help us identify any multicollinearity issues within our independent variables:



It appears that there are no strong linear relationships between our independent variables, so it would seem that we do not have any multicollinearity issues within our selected variables.

C5.
The last thing we will do to prepare our data is to export our cleaned and prepared dataset by using the following code:

```
#exports the new dataframe to a .csv
df_lm.to_csv('D208_Task2_Data_set.csv')
```

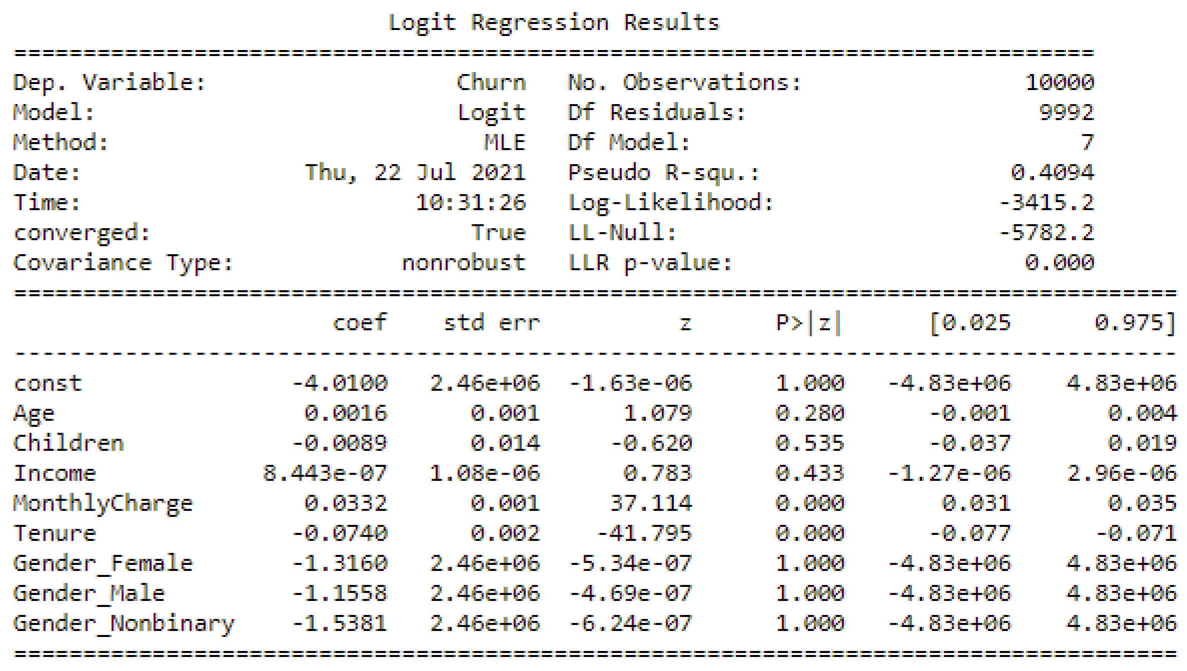
Please see the attached ‘D208_Task2_Data_set.csv’.

Part IV: Model Comparison and Analysis

D1.
We construct our initial logistic regression model by defining our independent / predictor variables and our dependent / target variable:

```
#defines the variables for input into our model
X = df_lm[['Age', 'Children', 'Income', 'MonthlyCharge',
'Tenure', 'Gender_Female', 'Gender_Male', 'Gender_Nonbinary']]
# X = independent variable(s) / predictor variable
y = df_lm['Churn']
# y = dependent variable / response / target variable
Xc = sm.add_constant(X)
# adds the constant coefficient / intercept to our model
```

Our model returns the following output:



```
                           Logit Regression Results
==============================================================================
Dep. Variable:                  Churn   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9992
Method:                           MLE   Df Model:                            7
Date:                Thu, 22 Jul 2021   Pseudo R-squ.:                  0.4094
Time:                        10:31:26   Log-Likelihood:                -3415.2
converged:                       True   LL-Null:                       -5782.2
Covariance Type:            nonrobust   LLR p-value:                     0.000
=====================================================================================
                      coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------
const              -4.0100   2.46e+06  -1.63e-06      1.000   -4.83e+06    4.83e+06
Age                 0.0016      0.001      1.079      0.280      -0.001       0.004
Children           -0.0089      0.014     -0.620      0.535      -0.037       0.019
Income           8.443e-07   1.08e-06      0.783      0.433   -1.27e-06    2.96e-06
MonthlyCharge       0.0332      0.001     37.114      0.000       0.031       0.035
Tenure             -0.0740      0.002    -41.795      0.000      -0.077      -0.071
Gender_Female      -1.3160   2.46e+06  -5.34e-07      1.000   -4.83e+06    4.83e+06
Gender_Male        -1.1558   2.46e+06  -4.69e-07      1.000   -4.83e+06    4.83e+06
Gender_Nonbinary   -1.5381   2.46e+06  -6.24e-07      1.000   -4.83e+06    4.83e+06
=====================================================================================
```

D2.
As we can see from our initial model summary results, we have a pseudo R-squared value of .41, meaning 41% of the variance can be explained using this model, which suggests the model is not a great fit. We can also see that all of our ‘Gender’ variables are identified by the model as being statistically insignificant, with a *p* value of 1. Standard error for ‘Income’ and our ‘Gender’ variables are also extreme.

Before we go further, let us reduce this model by removing any of our statistically insignificant variables, leaving only ‘MonthlyCharge’ and ‘Tenure’.

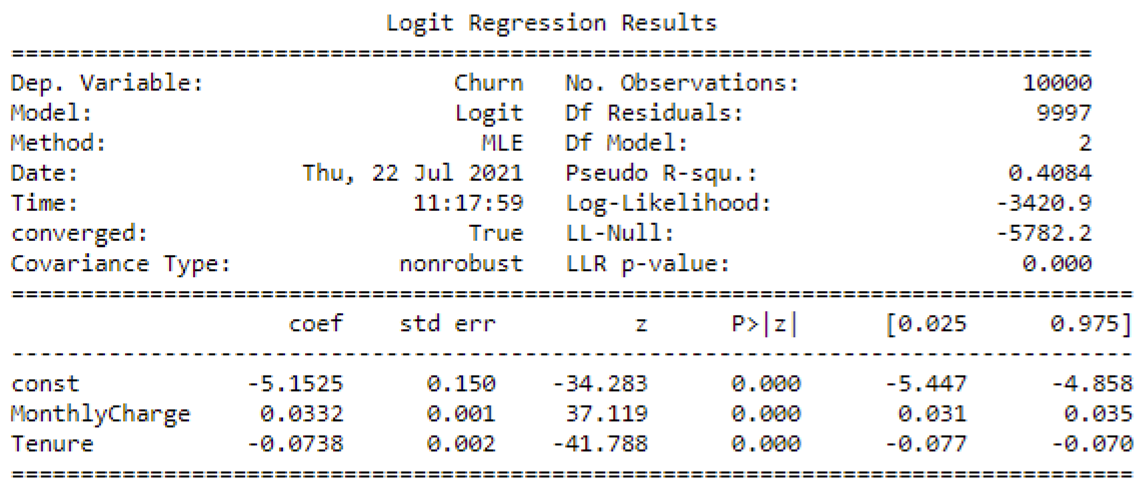
D3.
We prepare our reduced model by creating a new model without the ‘Children’, ‘Income’, and ‘Gender’ variables:

```
#defines the variables for input into our reduced model
Xr = df_lm[['MonthlyCharge', 'Tenure']] # X = independent
variable(s) / predictor variable
yr = df_lm['Churn'] # y = dependent variable / response / target
variable
Xcr = sm.add_constant(Xr)

log_reg_red = sm.Logit(yr, Xcr).fit()

model_red.summary()
```

Our reduced model returns the following output:



```
                           Logit Regression Results
==============================================================================
Dep. Variable:                  Churn   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9997
Method:                           MLE   Df Model:                            2
Date:                Thu, 22 Jul 2021   Pseudo R-squ.:                  0.4084
Time:                        11:17:59   Log-Likelihood:                -3420.9
converged:                       True   LL-Null:                       -5782.2
Covariance Type:            nonrobust   LLR p-value:                     0.000
=================================================================================
                    coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------
const            -5.1525      0.150    -34.283      0.000      -5.447      -4.858
MonthlyCharge     0.0332      0.001     37.119      0.000       0.031       0.035
Tenure           -0.0738      0.002    -41.788      0.000      -0.077      -0.070
=================================================================================
```

E1.
In our reduced model, the Pseudo R-squared value remains nearly the same at .41, suggesting 41% of the variance in the data can be explained using this model. However, it does appear that our remaining independent variables are at least statistically significant, with a 0.0 *p* value. Standard errors are much better as well at 0.001 and 0.002 for ‘MonthlyCharge’ and ‘Tenure’, respectively. This should provide us with a decent model to input into our classifier.

E2.
Let's train our reduced model using SciKit Learn's train/test split and use the scoring metrics (Precision, Recall, Support, F1 score) to determine how well our classification model performs.

We begin by utilizing SciKit Learn's train/test split to divide the dataset into subsets that will fit and train the model and make predictions compared to the expected values (*Train-Test Split for Evaluating Machine Learning Algorithms*, n.d.).

```
#creates a train/test split for our model
X_train,X_test,y_train,y_test =
train_test_split(Xr,yr,test_size=0.25,random_state=0)
```

We then train the model on the training data set:

```
logistic_regression= LogisticRegression()
logistic_regression.fit(X_train,y_train)
y_pred=logistic_regression.predict(X_test)
```

We can then define and output the results of our confusion matrix:

```
confusion_matrix = pd.crosstab(y_test, y_pred,
rownames=['Actual'], colnames=['Predicted'])
Confusion_matrix
```

Predicted	0	1
Actual		
0	1685	155
1	261	399

As we can see with our confusion matrix:

- Number of samples is 2,500
- Number of samples labeled '0' in the original data is 1,840
- Number of samples labeled '1' in the original data is 660
- Number of samples labeled '0' in the predicted data is 1,946
- Number of samples labeled '1' in the predicted data is 554
- Number of correct classifications is 2,084
- Number of misclassifications is 416

We can then print our classification report that will give us our scores on our classifier:

```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.87      0.92      0.89      1840
           1       0.72      0.60      0.66       660

    accuracy                           0.83      2500
   macro avg       0.79      0.76      0.77      2500
weighted avg       0.83      0.83      0.83      2500
```

Overall, our accuracy of the classification model appears to be 0.83, or 84%. Which is to say, our reduced model, using 'MonthlyCharge' and 'Tenure' as independent/predictor variables, was able to predict customers that churned with 84% accuracy.

E3.

Code:

```
#defines the variables for input into our model
X = df_lm[['Age', 'Children', 'Income', 'MonthlyCharge',
'Tenure', 'Gender_Female', 'Gender_Male', 'Gender_Nonbinary']]
# X = independent variable(s) / predictor variable
y = df_lm['Churn']
# y = dependent variable / response / target variable
Xc = sm.add_constant(X)
# adds the constant coefficient / intercept to our model


#defines the variables for input into our reduced model
Xr = df_lm[['MonthlyCharge', 'Tenure']] # X = independent
variable(s) / predictor variable
yr = df_lm['Churn'] # y = dependent variable / response / target
variable
Xcr = sm.add_constant(Xr)

log_reg_red = sm.Logit(yr, Xcr).fit()

model_red.summary()
```

Part V: Data Summary and Implications

F1.
Our reduced model's regression equation:

$$\hat{Y} = b_0 + b_1X_1 + b_2X_2 + \ldots + b_pX_p$$

Churn = -5.152 + 0.033 MonthlyCharge + -0.073 Tenure

Our reduced logistic regression model is far from perfect. Our Pseudo R-squared value suggests that our reduced model only accounts for ~41% of the variance in the data. We can, however, make some assumptions based on the coefficients of our variables:

```
                  Coeff
MonthlyCharge     0.033
Tenure           -0.073
```

In a logistic regression model, the coefficients of each independent variable show a positive or negative correlation with the dependent variable, showing the log-odds of the predictor variable being true or false. The amount by which the odds of the independent variable contributing to the dependent variable being true or false is indicated by the coefficient.
With this information we can make the following assumptions:

- For every one unit increase in 'MonthlyCharge', assuming all other variables remain the same, a customer is more likely to churn.
- For every one unit increase in 'Tenure', assuming all other variables remain the same, a customer is less likely to churn.

These coefficients indicate that 'MonthlyCharge' contributes to an increase, albeit a small one, in the likelihood that a customer will churn, which makes perfect sense. Customers that pay more will likely look to leave the service for another provider offering similar services at lower costs.

Conversely, 'Tenure' appears to have a negative correlation to 'Churn', suggesting that the longer a customer stays with the service, the less likely they are to leave the service. It could be that customers feel a sense of brand loyalty to the provider and do not wish to leave.

With an accuracy of only 84%, our classification model is not particularly useful for being able to definitively offer accurate predictions. We can, however, make somewhat accurate predictions about customer churn based on 'MonthlyCharge' and 'Tenure', especially when combined with the knowledge gained from our reduced model when we look at our coefficients.

While we can understand the relationship of ‘MonthlyCharge’ and ‘Tenure’ and how it relates to customer churn, the best approach would be to iterate on this model and take more variables into account. Similarly to our linear regression model ,we have perhaps limited our ability to build an accurate model by being too specific in our research question. However, we have discovered that there is a correlation between our reduced model’s independent variables and our dependent variable.

F2.
With the information we have gathered with our reduced logistic regression model, the best course of action we can recommend is expand our model and reframe our research question. While our research question initially set out to predict customer churn based on a number of independent variables, we ended up eliminating most of those due to our model informing us that the majority of the variables had statistically insignificant *p* values. It may be beneficial to not limit our independent variable selection in our research question, and instead be more general in our approach and see if *any* independent variables within our dataset can be used to predict churn.

It should also be noted that this dataset only contains 10,000 entries, which is not a huge sample size. Our model could be constricted by the limited number of data and, as such, it would be recommended to obtain a larger dataset to be able to more accurately make predictions.

Part VI: Demonstration
See attached Panopto video.

SOURCES

Assumptions of Logistic Regression. (n.d.). Statistics Solutions. Retrieved July 22, 2021, from https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/assumptions-of-logistic-regression/

Advantages of Learning Python for Data Science. (n.d.). BSD MAG. Retrieved July 22, 2021, from https://bsdmag.org/advantages-of-learning-python-for-data-science/

Logistic Regression using Statsmodels. (n.d.). GeeksforGeeks. Retrieved July 22, 2021, from https://www.geeksforgeeks.org/logistic-regression-using-statsmodels/

Example of Logistic Regression in Python. (n.d.). Data to Fish. Retrieved July 22, 2021, from https://datatofish.com/logistic-regression-python/

Train-Test Split for Evaluating Machine Learning Algorithms. (n.d.). Machine Learning Mastery. Retrieved July 22, 2021, from https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/

Frost, J. (2017, May 5). *Standard error of the regression*. Statistics By Jim. https://statisticsbyjim.com/glossary/standard-error-regression/

How to Interpret Logistic Regression Coefficients. (n.d.). Displayr. Retrieved July 22, 2021, from https://www.displayr.com/how-to-interpret-logistic-regression-coefficients/