Western Governors University

D208 – Predictive Modeling – Task 2 – Logistic Regression

By: Krista Moik

Table of Contents

Part I: Research Question

A1. Research Question

Using the medical_clean CSV, my research question is: **Which variables have the greatest impact on whether a patient is readmitted within one month of discharge?** This is important to know as hospitals are fined for excessive readmission.

A2. Goals

Knowing which variables affect whether a patient is readmitted or not is important for stakeholders. This knowledge could allow hospitals to predict hospital bed availability, staffing needs, and could also allow for techniques that reduce the number of readmissions thus also reducing costs and fines. My goal is to develop a logistic regression model that will allow me to find which variables in the dataset have the greatest impact on whether a patient is readmitted. In this analysis, the ReAdmis column is my dependent variable and the other variables in the dataset are my independent variables that I will be testing.

Part II: Method Justification

B1. Summary of Assumptions

Four assumptions of the logistic regression include the assumption that predicted values are nominal values, such as yes and no responses, or another type of label or category such as education level, marital status, and gender. The next assumption of the logistic model is that observations are independent of each other. The logistic regression model also assumes there is no multicollinearity among the independent variables. Finally, the logistic regression model assumes there are no extreme outliers in the dataset (Statology, 2020).

B2. Tool Benefits

Python and R are both capable of supporting the various phases of this analysis. While R is a programming language that has many of the statistical abilities needed for this analysis already built in, I will be using Python as it was recommended to not change programming languages for this project. That being said, Python has many libraries that can be imported that are just as statistically capable as R. Additionally, Python is also beneficial to use in terms of its quickness, readability, mathematical abilities, and visualizations (Western Governors University Information Technology, 2024). As always, I will import pandas and numpy for their database and basic mathematical abilities. Additionally, I will also be importing libraries from scipy.stats, statsmodels, and sklearn to complete the statistical portions of this analysis. I will also be importing matplotlib and seaborn to complete the visualization portions of this analysis. While I am familiar with most of these packages, the statistical models were suggested in WGU Course Materials (Western Governors University, n.d.).

B3. Appropriate Technique

Logistic Regression is an appropriate technique to develop a model to determine the relationship between a categorical dependent variable with multiple independent explanatory variables, which include those that are qualitative or quantitative in nature. My dependent variable is ReAdmis, which is a binary categorical variable with yes and no responses. My independent variables include those that are both quantitative and qualitative in nature. Thus, my research question and chosen variables align with the capabilities and requirements of logistic regression. Using logistic regression will allow me to estimate the probability of whether any of the independent variables have a relationship with the dependent variable that could be used to predict whether a patient will be readmitted.

Part III: Data Preparation
C1. Data Cleaning Goals

First, even though the data states it is cleaned, I will check for duplicate and null values using the print(df.duplicated().value_counts()) and df.isnull().sum() functions. I then plan to check for outliers in my chosen variables using the describe() function, histograms (using matplotlib plt.hist() function), and boxplots (using seaborn - sns.boxplot()) and will perform imputation as needed using fillna(). Once the data is cleaned, I will check the dataset again using boxplots to see if there are any changes. Finally, I will review my variables for categorical values and use ordinal encoding or one hot encoding as needed using functions like creating a dictionary, replace(), drop(), and onehot_encoder() as appropriate. To confirm the data has been cleaned in alignment with my research question, I will visualize my data again to confirm the effects. See the attached document in pdf and ipynb format titled KMoikD208Code2 to see the actual code used per column. By verifying the data is cleaned and ready for further analysis, I am preventing errors and unclean data from affecting my analysis.

C2. Summary Statistics

The dependent variable I am using is ReAdmis. This is a binary categorical variable using yes and no responses. By using the describe() and value_count() functions, I obtained the below descriptive statistical information regarding this variable. Over 63% of patients were NOT readmitted within one month.

```
#statistically describe dependent variable - ReAdmis
df.ReAdmis.describe()
```

```
count     10000
unique        2
top          No
freq       6331
Name: ReAdmis, dtype: object
```

```
df.ReAdmis.value_counts()
```

```
ReAdmis
No     6331
Yes    3669
Name: count, dtype: int64
```

For the independent variables, I will be using the columns Children, Age, Income, Doc_visits, Overweight, Diabetes, Anxiety, HighBlood, Stroke, Gender, Marital, Complication_risk, Initial_days, and Initial_admin. I chose these independent variables as I felt that they would be more likely to have an impact on whether a patient was readmitted rather than some of the other variables present in the dataset such as those that counted how many meals a patient ate while in the hospital or whether or not the patient received a Vitamin D supplement.

The statistics for my chosen independent variables are below:

1. Children, a quantitative and discrete variable, counts the number of children each patient has. Per the below statistics, the minimum number is 0 children and the most children any patient has is 10. The average number of children patients have is approximately 2.

```
#Statistically describe independent variable - Children
df.Children.describe()
```

```
count    10000.000000
mean         2.097200
std          2.163659
min          0.000000
25%          0.000000
50%          1.000000
75%          3.000000
max         10.000000
Name: Children, dtype: float64
```

```
df.Children.value_counts()
```

```
Children
0     2548
1     2509
3     1489
2     1475
4      995
7      213
8      209
6      191
5      169
9      108
10      94
Name: count, dtype: int64
```

2. Age, a quantitative and continuous variable, is the age of the patient in years. As you can see by the minimum age of 18, data is not available for patients under the age of 18. The highest age is 89.0 years old. The average age of patients is about 53.5 years old.

```
#Statistically describe independent variable - Age
df.Age.describe()
```

```
count    10000.000000
mean        53.511700
std         20.638538
min         18.000000
25%         36.000000
50%         53.000000
75%         71.000000
max         89.000000
Name: Age, dtype: float64
```

3. Income, a quantitative and continuous variable, is the reported income of each patient or the primary insurance holder. The minimum is $154.08, which seems low, but would not be unreasonable for an 18-year-old patient who just started working. The highest is over $200,000 a year, which is also not unreasonable for an older professional who has progressed their career over the years. The average salary amongst patients is approximately $40,490.

```
#Statistically describe independent variable - Income
df.Income.describe()
```

```
count     10000.000000
mean      40490.495160
std       28521.153293
min         154.080000
25%       19598.775000
50%       33768.420000
75%       54296.402500
max      207249.100000
Name: Income, dtype: float64
```

4. Marital, a qualitative and nominal variable, provides the marital status of the patient or primary insurance holder. Somewhat surprisingly, widowed is the most common response out of 5 response options of never married, separated, married, divorced, and widowed. Looking deeper into this data using value_counts() we can see that both widowed and married are close in number to each other. The other three options are also relatively similar to each other.

```
#Statistically describe independent variable - Marital
df.Marital.describe()
```

```
count       10000
unique          5
top       Widowed
freq         2045
Name: Marital, dtype: object
```

```
df.Marital.value_counts()
```

```
Marital
Widowed          2045
Married          2023
Separated        1987
Never Married    1984
Divorced         1961
Name: count, dtype: int64
```

5. Gender, a qualitative and nominal variable, is the self-identification of the patient as male, female, or nonbinary. From the statistics below, we can see that there were more female patients than male patients or nonbinary patients. Looking closer at the data using value_counts(), we can see that the number of male patients is only slightly smaller than female, but that the number of patients reporting as nonbinary is much smaller than those who reported as male or female.

```
#Statistically describe independent variable - Gender
df.Gender.describe()
```

```
count      10000
unique         3
top       Female
freq        5018
Name: Gender, dtype: object
```

```
df.Gender.value_counts()
```

```
Gender
Female       5018
Male         4768
Nonbinary     214
Name: count, dtype: int64
```

6. Doc_visits, a quantitative and discrete variable, is a count of the number of times the primary physician visited the patient during the initial hospitalization. The average number of doctor visits during initial admission was a little over 5 days. No physician visited more than 9 times, and every patient had at least 1 visit by the doctor.

```
#Statistically describe independent variable - Doc_visits
df.Doc_visits.describe()
```

```
count    10000.000000
mean         5.012200
std          1.045734
min          1.000000
25%          4.000000
50%          5.000000
75%          6.000000
max          9.000000
Name: Doc_visits, dtype: float64
```

```
df.Doc_visits.value_counts()
```

```
Doc_visits
5    3823
6    2436
4    2385
7     634
3     595
8      61
2      58
1       6
9       2
Name: count, dtype: int64
```

7. Initial_admin, a qualitative and nominal variable, indicates whether a patient was initially admitted as an emergency, elective, or observation admission. A little over half of all patients were admitted as an Emergency. Looking closer at the data using value.counts() we can see that patients admitted as elective and observation are almost equal.

```
#Statistically describe independent variable - Initial_admin
df.Initial_admin.describe()
```

```
count                   10000
unique                      3
top       Emergency Admission
freq                     5060
Name: Initial_admin, dtype: object
```

```
df.Initial_admin.value_counts()
```

```
Initial_admin
Emergency Admission      5060
Elective Admission       2504
Observation Admission    2436
Name: count, dtype: int64
```

8. HighBlood, a qualitative and nominal variable, indicates as yes or no whether the patient has high blood pressure. The below statistics show that a little over half of all patients did not have high blood pressure.

```
#Statistically describe independent variable - HighBlood
df.HighBlood.describe()
```

```
count     10000
unique        2
top          No
freq       5910
Name: HighBlood, dtype: object
```

```
df.HighBlood.value_counts()
```

```
HighBlood
No     5910
Yes    4090
Name: count, dtype: int64
```

9. Stroke, a qualitative and nominal variable, indicates as yes or no whether the patient has had a stroke. The below statistics shows that the majority of patients did not have a stroke.

```
#Statistically describe independent variable - Stroke
df.Stroke.describe()
```

```
count     10000
unique        2
top          No
freq       8007
Name: Stroke, dtype: object
```

```
df.Stroke.value_counts()
```

```
Stroke
No     8007
Yes    1993
Name: count, dtype: int64
```

10. Complication_risk, a qualitative and ordinal variable, uses the classifiers high, medium, and low to indicate the level of complication risk for the patient. The below statistics show that almost half of all patients were considered medium for complication risks. Using value_counts() we can see that patients determined at high risk were the next highest, and that patients determined to be at low risk accounted for the least number of patients.

```
#Statistically describe independent variable - Complication_risk
df.Complication_risk.describe()
```

```
count      10000
unique         3
top       Medium
freq        4517
Name: Complication_risk, dtype: object
```

```
df.Complication_risk.value_counts()
```

```
Complication_risk
Medium    4517
High      3358
Low       2125
Name: count, dtype: int64
```

11. Overweight, a qualitative and nominal variable, indicates as either yes or no whether the patient is overweight. The below statistics show that more than half of all patients are overweight.

```
#Statistically describe independent variable - Overweight
df.Overweight.describe()
```

```
count     10000
unique        2
top         Yes
freq       7094
Name: Overweight, dtype: object
```

```
df.Overweight.value_counts()
```

```
Overweight
Yes    7094
No     2906
Name: count, dtype: int64
```

12. Diabetes, a qualitative and nominal variable, indicates as either yes or no whether the patient has diabetes. The below statistics show that more than half of patients do not have diabetes.

```
#Statistically describe independent variable - Diabetes
df.Diabetes.describe()
```

```
count     10000
unique        2
top          No
freq       7262
Name: Diabetes, dtype: object
```

```
df.Diabetes.value_counts()
```

```
Diabetes
No     7262
Yes    2738
Name: count, dtype: int64
```

13. Anxiety, a qualitative and nominal variable, indicates as either yes or no whether the patient has anxiety. The below statistics show that more than half of patients did not have anxiety.

```
#Statistically describe independent variable - Anxiety
df.Anxiety.describe()
```

```
count     10000
unique        2
top          No
freq       6785
Name: Anxiety, dtype: object
```

```
df.Anxiety.value_counts()
```

```
Anxiety
No     6785
Yes    3215
Name: count, dtype: int64
```

14. Initial_days, a continuous variable, is a count of how many days a patient was initially admitted to the hospital. By using the describe() function, I obtained the below descriptive statistical information regarding this variable. The minimum days spent in the hospital is slightly over 1 day and the maximum is almost 72 days. The one day minimum is affected by the dataset only including stays that were at least 1 day. The average number of days a patient was initially admitted for was almost 34.5 days.

```
#Statistically describe independent variable - Initial_days
df.Initial_days.describe()
```

```
count    10000.000000
mean        34.455299
std         26.309341
min          1.001981
25%          7.896215
50%         35.836244
75%         61.161020
max         71.981490
Name: Initial_days, dtype: float64
```

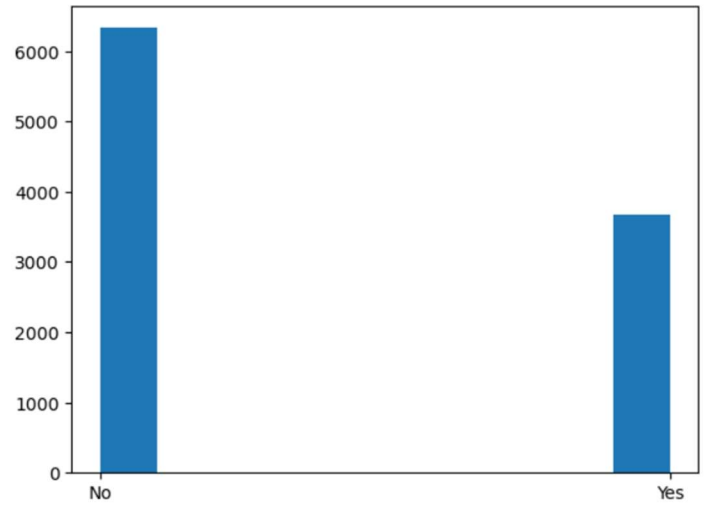
```
df.Initial_days.value_counts()
```

```
Initial_days
63.544320    2
67.421390    2
70.325420    2
63.334690    1
67.036510    1
            ..
5.977596     1
5.799041     1
6.415853     1
7.328631     1
70.850590    1
Name: count, Length: 9997, dtype: int64
```

C3. Visualizations

Please see the below univariate visualization of the dependent variable ReAdmis using the plt.hist() function. The histogram is in line with what the descriptive statistics told us regarding the amount of yes and no responses.

```
plt.hist(df['ReAdmis'])
plt.show()
```

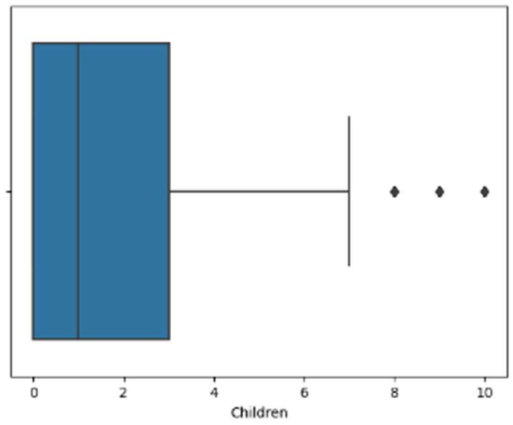


Univariate visualizations of my independent variables:

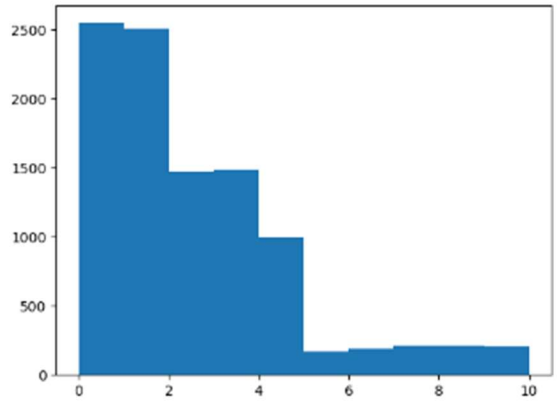
1. The Children variable is shown below as a boxplot and histogram. The boxplot clearly shows outliers, and the histogram indicates the data is positively skewed.

```
sns.boxplot(df, x='Children')
```

```
<Axes: xlabel='Children'>
```



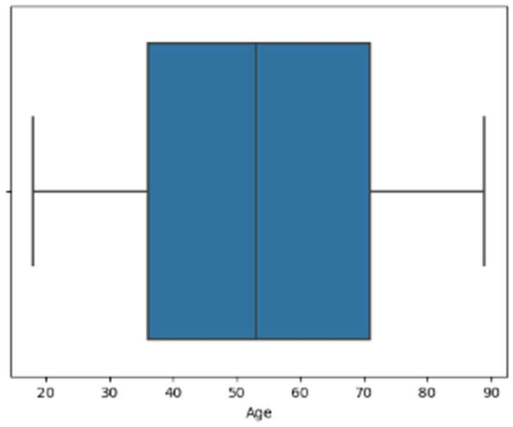

```
plt.hist(df['Children'])
plt.show()
```



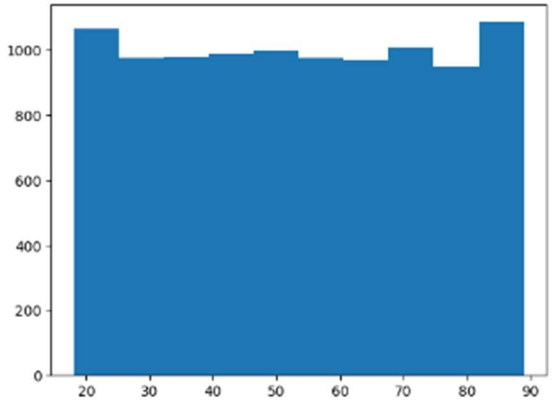
2. The Age variable is shown below as a boxplot and histogram. Per the boxplot, there do not appear to be any outliers. The histogram shows a uniform distribution.

```
sns.boxplot(df, x='Age')
```

```
<Axes: xlabel='Age'>
```



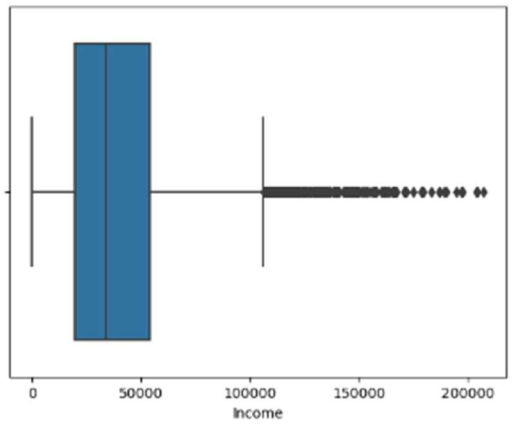

```
plt.hist(df['Age'])
plt.show()
```



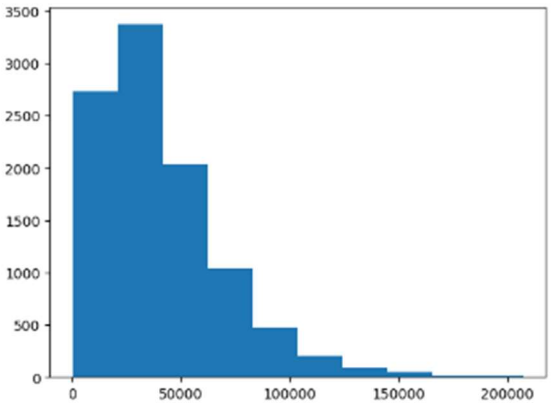
3. The Income variable is shown below as a boxplot and histogram. The boxplot shows there are many outliers. The histogram shows a positive skew.

```
sns.boxplot(df, x='Income')
```

```
<Axes: xlabel='Income'>
```

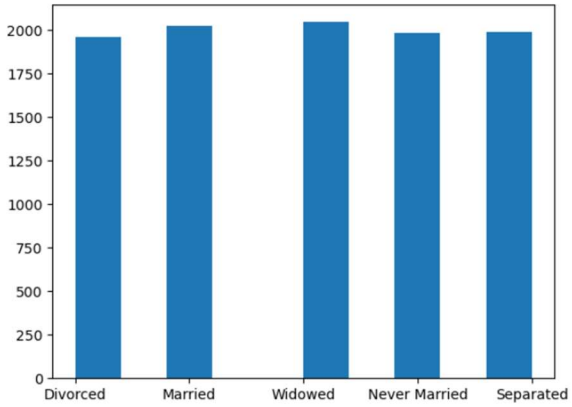



```
plt.hist(df['Income'])
plt.show()
```



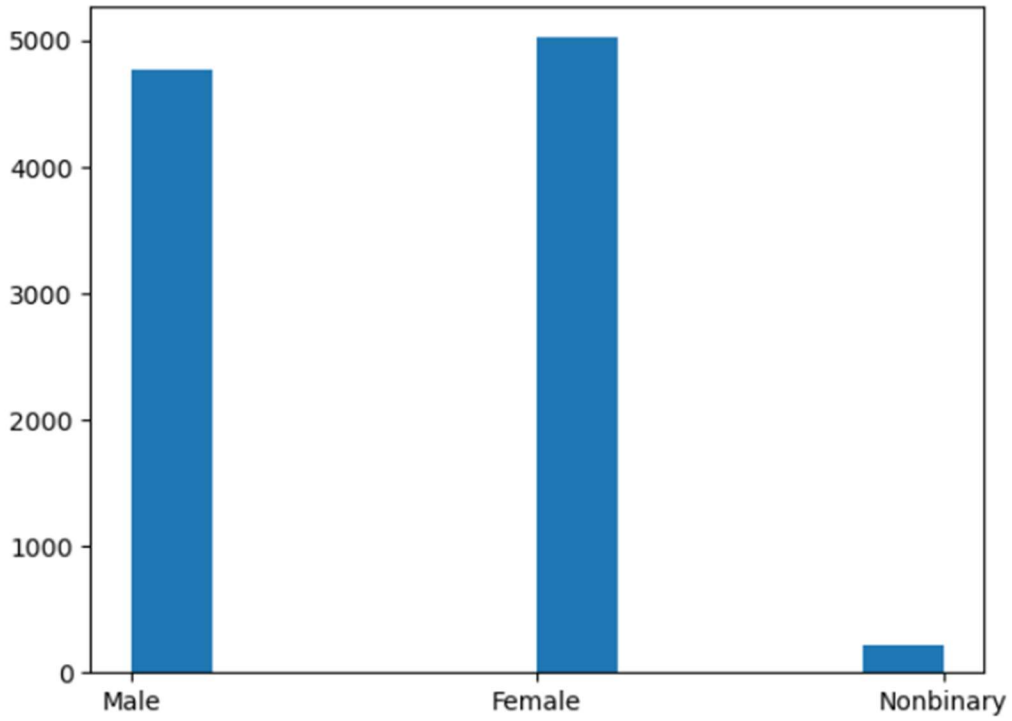
4. The Marital variable is shown below as a histogram and seems to be a somewhat uniform distribution and in line with what the descriptive statistics told us.

```
#unable to create boxplot for categorical data
plt.hist(df['Marital'])
plt.show()
```



5. The Gender variable is shown below as a histogram. The values are in line with what the describe() and value.counts() functions showed us.

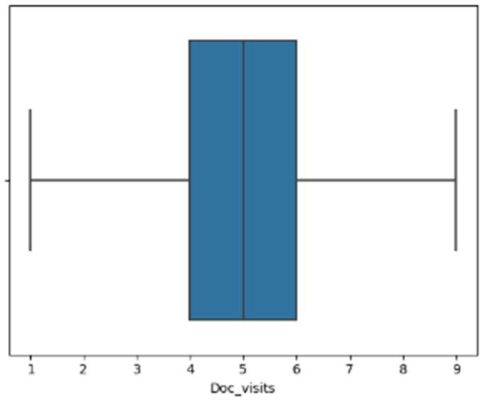
```
#unable to create boxplot for categorical data
plt.hist(df['Gender'])
plt.show()
```



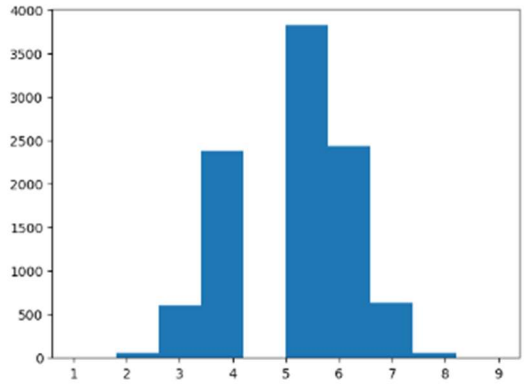
6. The Doc_visits variable is shown below as a boxplot and histogram. The boxplot does not seem to show any outliers. This histogram shows a bimodal distribution.

```
sns.boxplot(df, x='Doc_visits')
```

```
<Axes: xlabel='Doc_visits'>
```

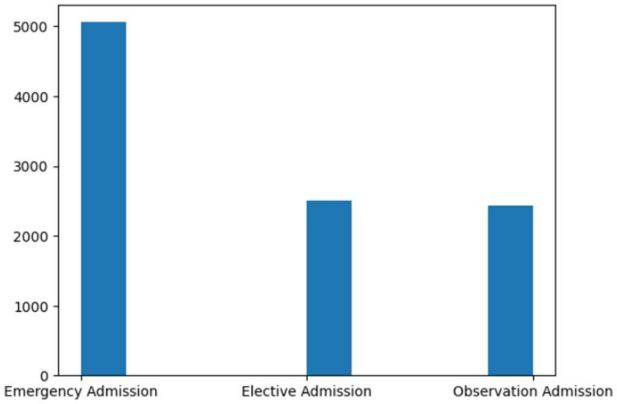



```
plt.hist(df['Doc_visits'])
plt.show()
```



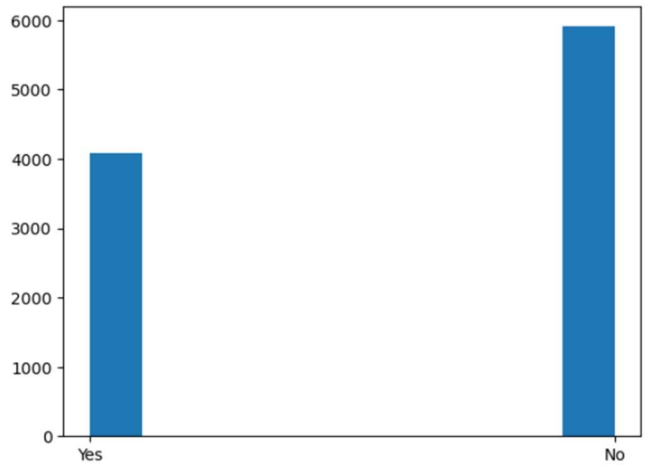
7. The Initial_admin variable is shown below as a histogram. This visualization is in line with what the describe() and value.counts() functions showed us.

```
#unable to create boxplot for categorical data
plt.hist(df['Initial_admin'])
plt.show()
```



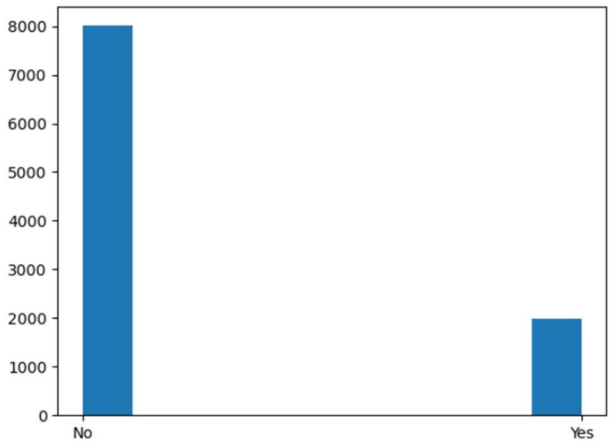
8. The HighBlood variable is shown below as a histogram. This visualization is in line with what the describe() function showed us.

```
#unable to create boxplot for categorical data
plt.hist(df['HighBlood'])
plt.show()
```



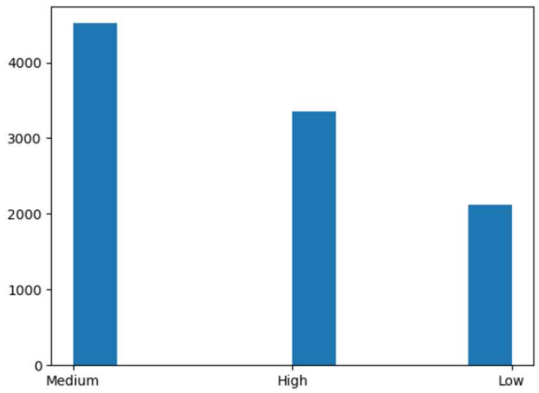
9. The Stroke variable is shown below as a histogram. This visualization is in line with what the describe() function showed us.

```
#unable to create boxplot for categorical data
plt.hist(df['Stroke'])
plt.show()
```



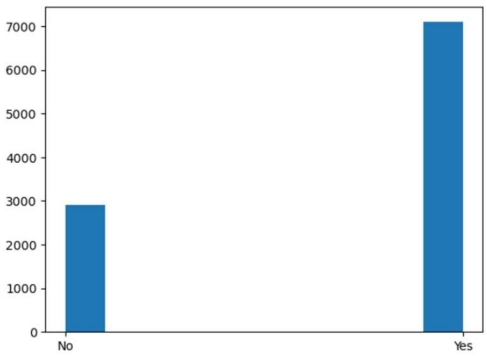
10. The Complication_risk variable is shown below as a histogram. This visualization is in line with what the describe() and value.counts() functions showed us.

```
#unable to create boxplot for categorical data
plt.hist(df['Complication_risk'])
plt.show()
```



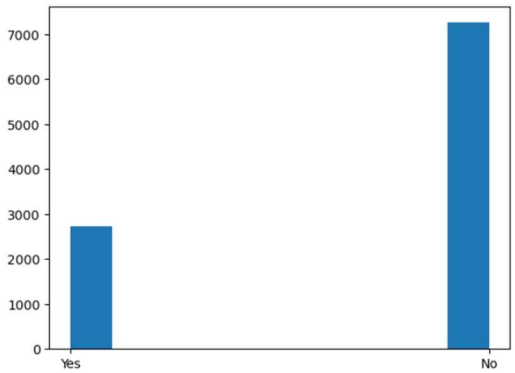
11. The Overweight variable is shown below as a histogram. This visualization is in line with what the describe() function showed us.

```
#unable to create boxplot for categorical data
plt.hist(df['Overweight'])
plt.show()
```



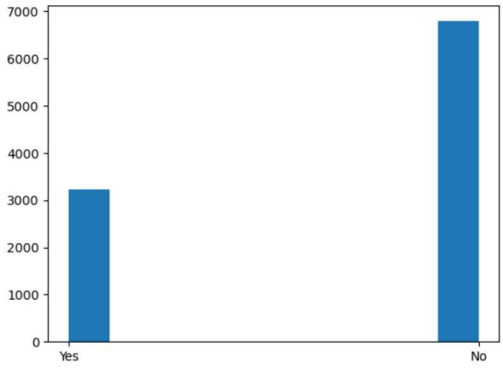
12. The Diabetes variable is shown below as a histogram. This visualization is in line with what the describe() function showed us.

```
#unable to create boxplot for categorical data
plt.hist(df['Diabetes'])
plt.show()
```



13. The Anxiety variable is shown below as a histogram. This visualization is in line with what the describe() function showed us.

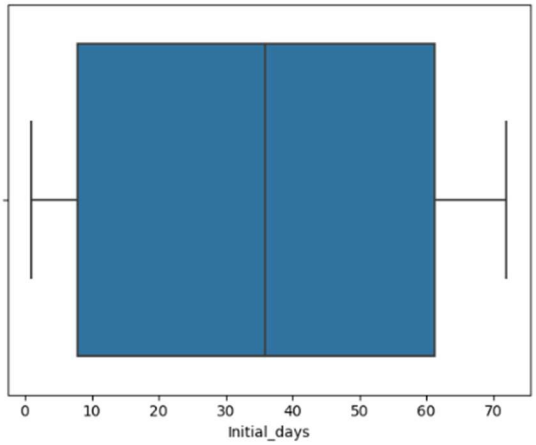
```
#unable to create boxplot for categorical data
plt.hist(df['Anxiety'])
plt.show()
```



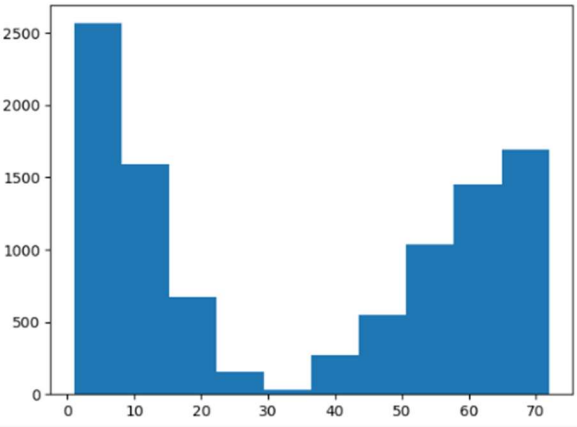
14. The Initial_days variable is shown using the sns.boxplot() and plt.hist() functions. These visualizations show a bimodal skew without any apparent outliers.

```
sns.boxplot(df, x='Initial_days')
```

```
<Axes: xlabel='Initial_days'>
```



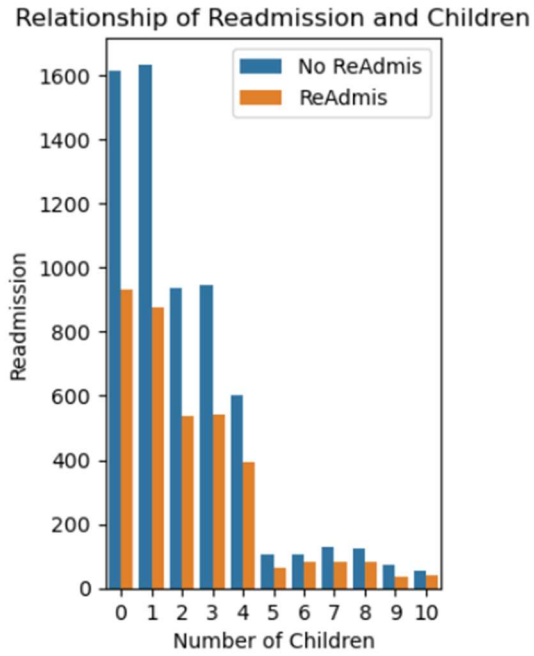

```
plt.hist(df['Initial_days'])
plt.show()
```



Bivariate visualizations of my dependent and independent variables using the countplot() function are below:

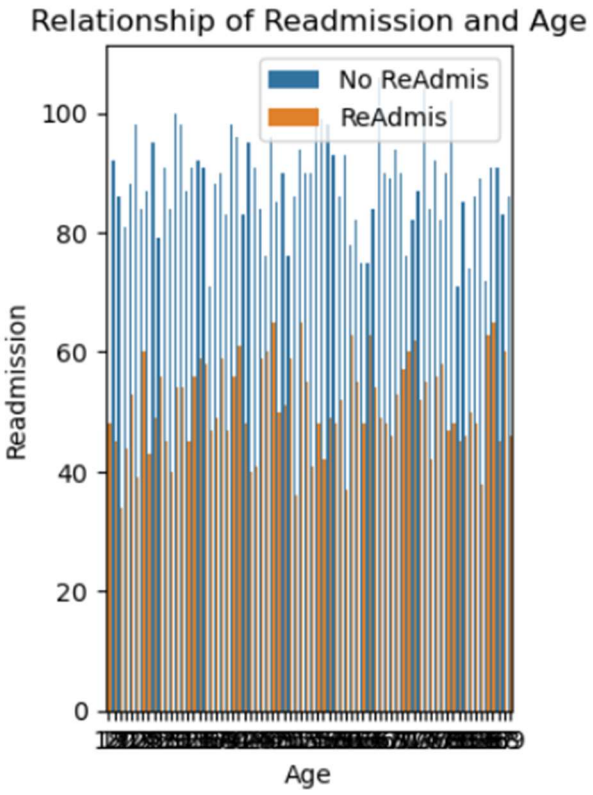
1. The ReAdmis and Children countplot is below. It shows that generally, regardless of the number of children, patients had a higher rate of not being readmitted than they were readmitted.

```
#Bivariate exploration of ReAdmis and Children using Countplot
plt.subplot(1, 2, 2)
plt.title("Relationship of Readmission and Children")
sns.countplot(data = df, x="Children", hue="ReAdmis")
plt.legend(["No ReAdmis", "ReAdmis"])
plt.xlabel("Number of Children")
plt.ylabel("Readmission");
```



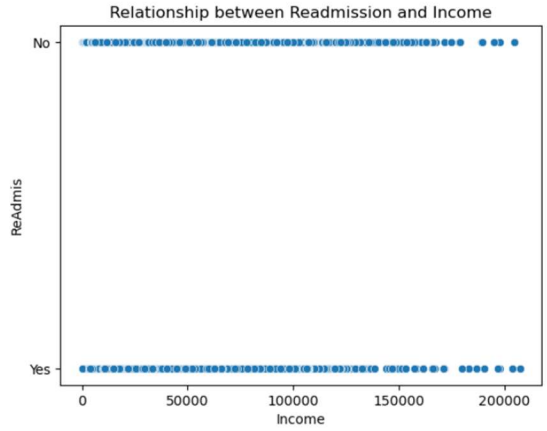
2. The ReAdmis and Age countplot is below. Due to the number of ages (from 18 to 89 years), it is hard to read, however it is clear that rates of no readmission exceed rates of readmission across the age groups.

```
#Bivariate exploration of ReAdmis and Age using Countplot
plt.subplot(1, 2, 2)
plt.title("Relationship of Readmission and Age")
sns.countplot(data = df, x="Age", hue="ReAdmis")
plt.legend(["No ReAdmis", "ReAdmis"])
plt.xlabel("Age")
plt.ylabel("Readmission");
```



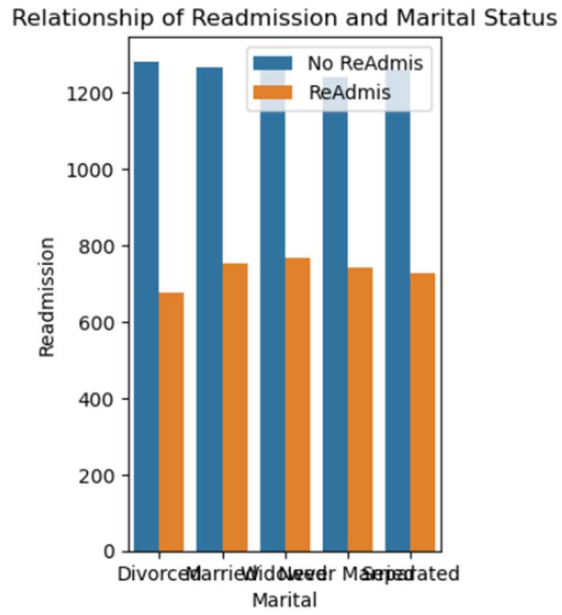
3. The ReAdmis and Income scatterplot is below. Simply looking at this visually, the rates of no readmission and readmission appear to be distributed across all incomes.

```
#Bivariate visualization of Readmis and Income using scatterplots
sns.scatterplot(x='Income', y='ReAdmis', data=df).set(title="Relationship between Readmission and Income")
plt.show()
```



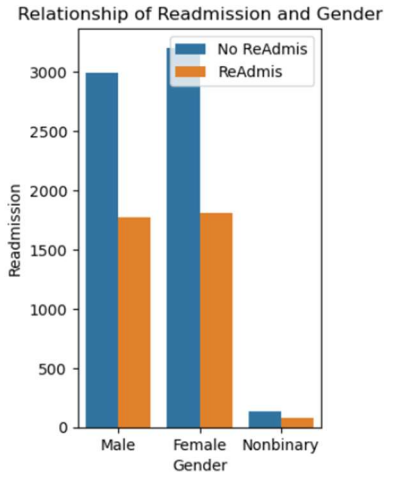
4. The ReAdmis and Marital Status countplot is below. We continue to see the no readmission exceeds rates of admission across all marital status.

```
#Bivariate exploration of ReAdmis and Marital Status using Countplot
plt.subplot(1, 2, 2)
plt.title("Relationship of Readmission and Marital Status")
sns.countplot(data = df, x="Marital", hue="ReAdmis")
plt.legend(["No ReAdmis", "ReAdmis"])
plt.xlabel("Marital")
plt.ylabel("Readmission");
```



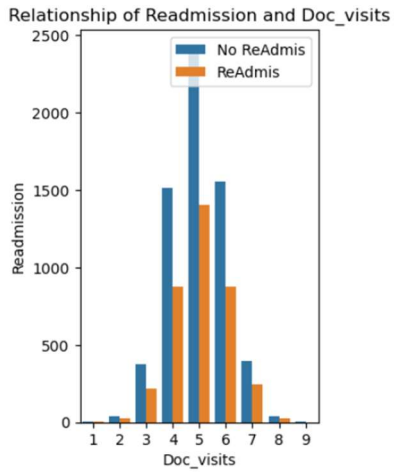
5. The ReAdmis and Gender countplot is below. Patients across all gender categories had higher rates of not being readmitted than they were readmitted.

```
#Bivariate exploration of ReAdmis and Gender using Countplot
plt.subplot(1, 2, 2)
plt.title("Relationship of Readmission and Gender")
sns.countplot(data = df, x="Gender", hue="ReAdmis")
plt.legend(["No ReAdmis", "ReAdmis"])
plt.xlabel("Gender")
plt.ylabel("Readmission");
```



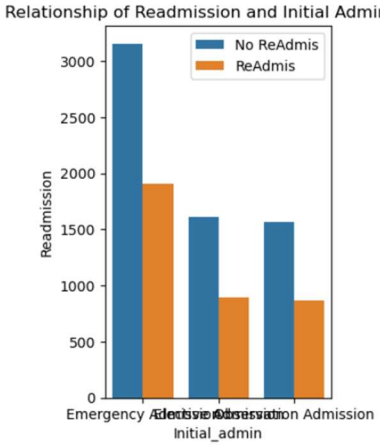
6. The ReAdmis and Doc_visits countplot is below. The rates of no readmission are larger than readmission across the number of doctor visits.

```
#Bivariate exploration of ReAdmis and Doc_visits using Countplot
plt.subplot(1, 2, 2)
plt.title("Relationship of Readmission and Doc_visits")
sns.countplot(data = df, x="Doc_visits", hue="ReAdmis")
plt.legend(["No ReAdmis", "ReAdmis"])
plt.xlabel("Doc_visits")
plt.ylabel("Readmission");
```



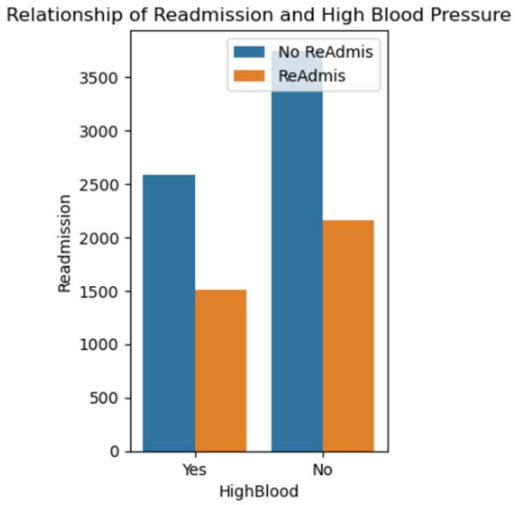
7. The ReAdmis and Initial_admin countplot is below. The rates of no readmission exceed the rates of admission across all initial admin types.

```
#Bivariate exploration of ReAdmis and Initial_admin using Countplot
plt.subplot(1, 2, 2)
plt.title("Relationship of Readmission and Initial Admin Type")
sns.countplot(data = df, x="Initial_admin", hue="ReAdmis")
plt.legend(["No ReAdmis", "ReAdmis"])
plt.xlabel("Initial_admin")
plt.ylabel("Readmission");
```



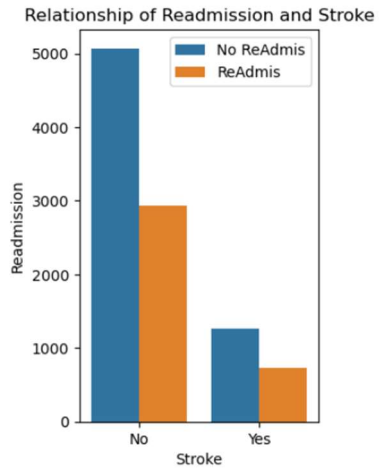
8. The ReAdmis and HighBlood countplot is below. Regardless of whether the patient had high blood pressure or not, they were most often not readmitted.

```
#Bivariate exploration of ReAdmis and HighBlood using Countplot
plt.subplot(1, 2, 2)
plt.title("Relationship of Readmission and High Blood Pressure")
sns.countplot(data = df, x="HighBlood", hue="ReAdmis")
plt.legend(["No ReAdmis", "ReAdmis"])
plt.xlabel("HighBlood")
plt.ylabel("Readmission");
```



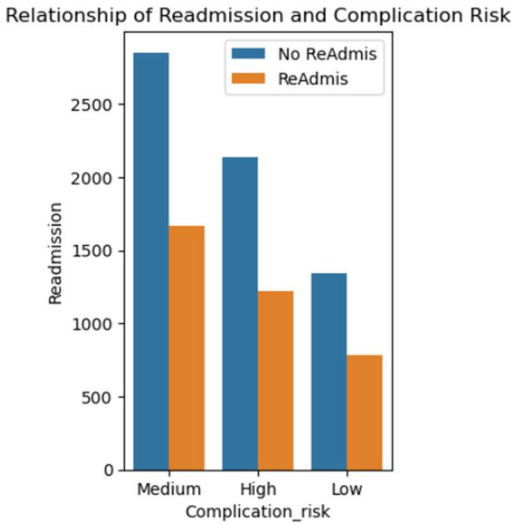
9. The ReAdmis and Stroke countplot is below. Regardless of whether the patient had a stroke or not, they were most often not readmitted.

```
#Bivariate exploration of ReAdmis and Stroke using Countplot
plt.subplot(1, 2, 2)
plt.title("Relationship of Readmission and Stroke")
sns.countplot(data = df, x="Stroke", hue="ReAdmis")
plt.legend(["No ReAdmis", "ReAdmis"])
plt.xlabel("Stroke")
plt.ylabel("Readmission");
```



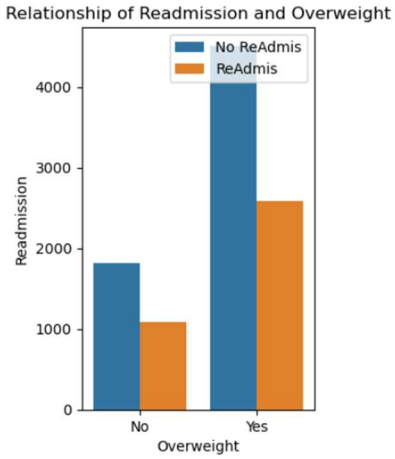
10. The ReAdmis and Complication_risk countplot is below. Regardless of complication risk rating, patients were most often not readmitted.

```
#Bivariate exploration of ReAdmis and Complication_risk using Countplot
plt.subplot(1, 2, 2)
plt.title("Relationship of Readmission and Complication Risk")
sns.countplot(data = df, x="Complication_risk", hue="ReAdmis")
plt.legend(["No ReAdmis", "ReAdmis"])
plt.xlabel("Complication_risk")
plt.ylabel("Readmission");
```



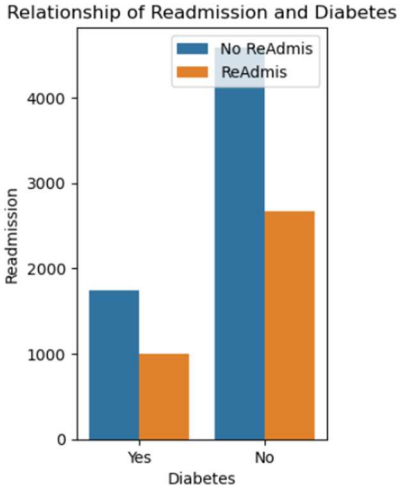
11. The ReAdmis and Overweight countplot is below. Regardless of whether the patient was overweight or not, they were most often not readmitted.

```
#Bivariate exploration of ReAdmis and Overweight using Countplot
plt.subplot(1, 2, 2)
plt.title("Relationship of Readmission and Overweight")
sns.countplot(data = df, x="Overweight", hue="ReAdmis")
plt.legend(["No ReAdmis", "ReAdmis"])
plt.xlabel("Overweight")
plt.ylabel("Readmission");
```



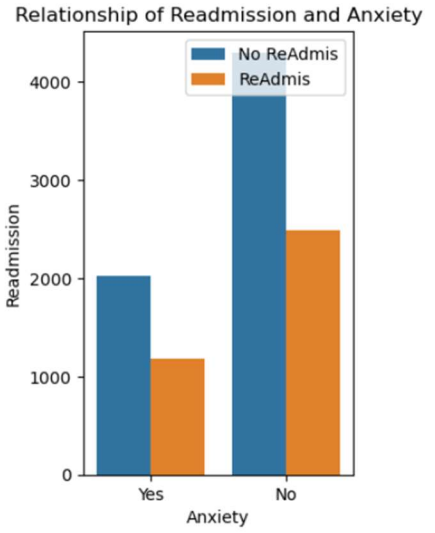
12. The ReAdmis and Diabetes countplot is below. Regardless of whether the patient was diabetic or not, they were most often not readmitted.

```
#Bivariate exploration of ReAdmis and Diabetes using Countplot
plt.subplot(1, 2, 2)
plt.title("Relationship of Readmission and Diabetes")
sns.countplot(data = df, x="Diabetes", hue="ReAdmis")
plt.legend(["No ReAdmis", "ReAdmis"])
plt.xlabel("Diabetes")
plt.ylabel("Readmission");
```



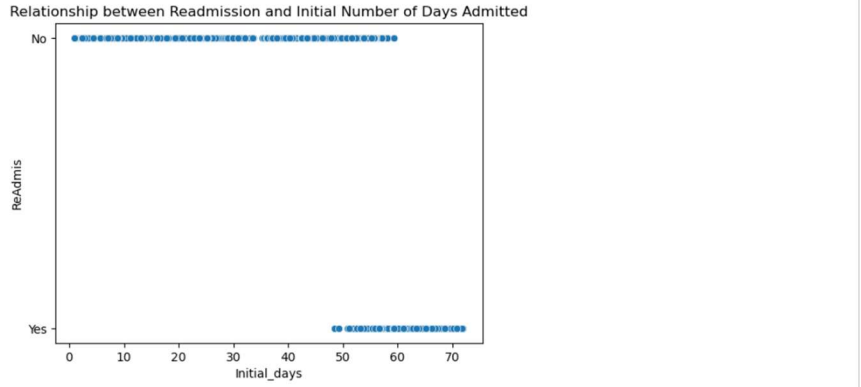
13. The ReAdmis and Anxiety countplot is below. Regardless of whether the patient had anxiety or not, they were most often not readmitted.

```
#Bivariate exploration of ReAdmis and Anxiety using Countplot
plt.subplot(1, 2, 2)
plt.title("Relationship of Readmission and Anxiety")
sns.countplot(data = df, x="Anxiety", hue="ReAdmis")
plt.legend(["No ReAdmis", "ReAdmis"])
plt.xlabel("Anxiety")
plt.ylabel("Readmission");
```



14. The ReAdmis and Initial_days scatterplot is below. This shows that starting around 50 initial days, patients were more likely to be readmitted than not.

```
#Bivariate visualization of ReAdmis and Initial_days using scatterplots
sns.scatterplot(x='Initial_days', y='ReAdmis', data=df).set(title="Relationship between Readmission and Initial Number of Days Ad
plt.show()
```



C4. Data Transformation

Just as in the first task, my data transformation goals include re-expressing my qualitative variables to numeric values so I can use them in my logistic regression model. For the variables that have yes/no responses, I will use ordinal encoding to set the yes responses as 1s and the no responses as 0s. For the categorical values that have 3 or more response types, I will use one hot encoding as explained by Practical Business Python (Moffitt, 2017). Additionally, I will use the df.drop() function to drop all columns I am not using in my analysis. This will provide me with a dataset of my

dependent and independent variables which are all now numeric values so I can create my regression model.

Here is an example of the ordinal encoding code I used to change Yes and No responses to 1s and 0s:

Re-Expression of Categorical Values

```
#re-expressing ReAdmis
df['ReAdmis_numeric']=df['ReAdmis']
```

```
#set up dictionary
dict_readmis={'ReAdmis_numeric' : {'No':0, 'Yes':1}}
```

```
#replace variable's values
df.replace(dict_readmis, inplace=True)
```

```
# drop original column
df=df.drop(columns=['ReAdmis'])
```

Below is a screenshot using onehot encoding to re-express categorical variables with more than two response types:

```
#onehot encoding
onehot_encoder = OneHotEncoder(sparse=False)
```

```
onehot_encoded=onehot_encoder.fit_transform(df[['Gender', 'Marital', 'Complication_risk', 'Initial_admin']])
```

```
C:\Users\Kmoik WGU\anaconda3\Lib\site-packages\sklearn\preprocessing\_encoders.py:972: FutureWarning: `sparse` was renamed to `
sparse_output` in version 1.2 and will be removed in 1.4. `sparse_output` is ignored unless you leave `sparse` to its default v
alue.
  warnings.warn(
```

```
df_encoded=pd.DataFrame(onehot_encoded, columns=onehot_encoder.get_feature_names_out(['Gender', 'Marital', 'Complication_risk', '
```

```
#drop first columns to reduce multicollinearity
df_encoded.drop(['Gender_Female', 'Marital_Divorced', 'Complication_risk_High', 'Initial_admin_Elective Admission'], axis=1, inpl
```

```
#merging encoded columns to df
df=pd.concat([df, df_encoded], axis=1)
```

After the encoding was completed, I then used the drop() function to drop the columns I was not including in my analysis as well as the first column of the re-expressed variables to reduce instances of multicollinearity.

```
#drop unused columns
df.drop(['Initial_admin', 'Complication_risk', 'Gender', 'Marital', 'CaseOrder', 'Customer_id', 'Interaction', 'UID', 'City', 'St
```

Please see the attached pdf and ipynb files titled KMoikD208Code2 for the full code used.

C5. Prepared Data Set

Please see the attached CSV file titled: KMoikclean_medical.csv showing my prepared dataset. The columns in the clean data set are below:

```
#confirm data set has been updated appropriately
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 21 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   Children                            10000 non-null  int64
 1   Age                                 10000 non-null  int64
 2   Income                              10000 non-null  float64
 3   Doc_visits                          10000 non-null  int64
 4   Initial_days                        10000 non-null  float64
 5   ReAdmis_numeric                     10000 non-null  int64
 6   Overweight_numeric                  10000 non-null  int64
 7   Diabetes_numeric                    10000 non-null  int64
 8   Anxiety_numeric                     10000 non-null  int64
 9   HighBlood_numeric                   10000 non-null  int64
 10  Stroke_numeric                      10000 non-null  int64
 11  Gender_Male                         10000 non-null  float64
 12  Gender_Nonbinary                    10000 non-null  float64
 13  Marital_Married                     10000 non-null  float64
 14  Marital_Never Married               10000 non-null  float64
 15  Marital_Separated                   10000 non-null  float64
 16  Marital_Widowed                     10000 non-null  float64
 17  Complication_risk_Low               10000 non-null  float64
 18  Complication_risk_Medium            10000 non-null  float64
 19  Initial_admin_Emergency Admission   10000 non-null  float64
 20  Initial_admin_Observation Admission 10000 non-null  float64
dtypes: float64(12), int64(9)
memory usage: 1.6 MB
```

No duplicate or missing values were in the dataset. The only outliers in the data were those that were previously explored and retained in D206 and in the first task of D208. I chose to still retain these outliers as I found them to be reasonable and justifiable to keep in the dataset.

After re-expressing my variables and dropping unused and unneeded columns, my resulting data set has 21 columns which includes my 1 dependent variable and 20 independent variables. The columns that changed were the re-expressed columns. I added _numeric to the end of the columns where the Yes and No responses were re-encoded as 1s and 0s. My dependent variable was renamed as ReAdmis_numeric. For the columns with more than 2 variables that needed to be re-expressed, a new column was made for every response, with the first column being dropped, and 1s and 0s added. So, the marital column with 5 possible responses became: Marital_Married, Marital_Never Married, Marital_Separated, and Marital_Widowed. Marital_Divorced was dropped to reduce multicollinearity.

Part IV: Model Comparison and Analysis

D1. Initial Model

I used the Maximum Likelihood Estimation (MLE) method and code to obtain my initial logistic regression model (Western Governors University, n.d.). My initial model included 21 variables,

where 1 variable was my dependent variable, and 20 were my independent variables. The initial model is below:

Logistic Regression Model

```
y = df.ReAdmis_numeric
X = df[["Children", "Age", "Income", "Doc_visits", "Initial_days", "Overweight_numeric", "Anxiety_numeric", "Diabetes_numeric", '
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.039465
         Iterations 13
                           Logit Regression Results
==============================================================================
Dep. Variable:        ReAdmis_numeric   No. Observations:                10000
Model:                          Logit   Df Residuals:                    9979
Method:                           MLE   Df Model:                          20
Date:                Tue, 06 Feb 2024   Pseudo R-squ.:                 0.9400
Time:                        09:31:45   Log-Likelihood:                -394.65
converged:                       True   LL-Null:                       -6572.9
Covariance Type:            nonrobust   LLR p-value:                     0.000
=====================================================================================================
                                        coef    std err          z      P>|z|      [0.025      0.975]
-----------------------------------------------------------------------------------------------------
Children                              0.0871      0.042      2.091      0.036       0.005       0.169
Age                                   0.0030      0.005      0.656      0.512      -0.006       0.012
Income                             1.691e-06   3.3e-06      0.512      0.608   -4.78e-06    8.16e-06
Doc_visits                           -0.0309      0.086     -0.359      0.720      -0.200       0.138
Initial_days                          1.2080      0.061     19.834      0.000       1.089       1.327
Overweight_numeric                   -0.2789      0.205     -1.359      0.174      -0.681       0.123
Anxiety_numeric                      -0.8776      0.202     -4.350      0.000      -1.273      -0.482
Diabetes_numeric                      0.3974      0.209      1.902      0.057      -0.012       0.807
HighBlood_numeric                     0.7188      0.196      3.675      0.000       0.335       1.102
Stroke_numeric                        1.5031      0.244      6.155      0.000       1.025       1.982
Gender_Male                           0.0366      0.189      0.194      0.846      -0.334       0.407
Gender_Nonbinary                      0.4912      0.619      0.794      0.427      -0.722       1.704
Marital_Married                       0.1155      0.294      0.392      0.695      -0.461       0.692
Marital_Never Married                 0.2276      0.302      0.753      0.451      -0.365       0.820
Marital_Separated                    -0.1211      0.307     -0.394      0.693      -0.723       0.481
Marital_Widowed                       0.2682      0.296      0.908      0.364      -0.311       0.848
Complication_risk_Low                -1.3881      0.259     -5.365      0.000      -1.895      -0.881
Complication_risk_Medium             -0.2959      0.213     -1.388      0.165      -0.714       0.122
Initial_admin_Emergency Admission     1.9412      0.241      8.069      0.000       1.470       2.413
Initial_admin_Observation Admission   0.5478      0.257      2.130      0.033       0.044       1.052
const                               -66.8721      3.401    -19.663      0.000     -73.538     -60.207
=====================================================================================================

Possibly complete quasi-separation: A fraction 0.78 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
```

This initial model gives us a logical regression equation of:

ReAdmis = -66.8721 + 0.0871*Children + 0.003*Age + 1.691e-06*Income – 0.0309*Doc_visits + 1.2.80*Initial_days – 0.2789*Overweight_numeric – 0.8776*Anxiety_numeric + 0.3974*Diabetes_numeric + 0.7188*HighBlood_numeric + 1.5031*Stroke_numeric + 0.0366*Gender_Male + 0.4912*Gender_Nonbinary + 0.1155*Marital_Married+0.2276*Marital_Never Married – 0.1211*Marital_Separated + 0.2685*Marital_Widowed – 1.3881*Complication_risk_Low – 0.2959*Complication_risk_Medium + 1.9412*Initial_admin_Emergency Admission + 0.5478*Initial_admin_Observation_admission

This model has a Pseudo R-squared value of 0.94 which indicates that this is a pretty good model. Additionally, the LLR p-value is 0.000, which is smaller than our alpha value of 0.05. This indicates the model as a whole is useful (Western Governors University, n.d.).

D2. Justification of Model Reduction

While the initial model seems to be relevant, it does show that several of the variables have p-values greater than the alpha value of 0.05, meaning they are not statistically significant. Additionally, the notes indicate that there is a possible complete quasi-separation. This means that my model may perfectly or almost perfectly predict the response (University of California, Los Angeles Advanced Research Computing, n.d.). As such, we will use Variance Inflation Factors (VIF) and backward stepwise elimination to remove the variables that have high levels of multicollinearity and are not statistically relevant to our data and evaluate the model from there. The VIF value represents multicollinearity, and we want to remove any variables that have a value of 10 or greater. The backward stepwise elimination process will then be performed until all variables remaining have a p-value smaller than our alpha value, indicating they are statistically significant. The result of these processes will be variables that have low multicollinearity and are statistically significant to our dependent variable.

D3. Reduced Logistic Regression Model

First, I calculated the VIF using code obtained from WGU Course Materials (Western Governors University, n.d.):

```
#calculate VIF for all independent variables
vif_df["VIF"]=[variance_inflation_factor(X.values, i) for i in range(len(X.columns))]
```

```
print(vif_df)
```

```
                              feature        VIF
0                            Children   1.908811
1                                 Age   6.689004
2                              Income   2.891617
3                          Doc_visits  13.063681
4                        Initial_days   2.635161
5                  Overweight_numeric   3.287509
6                     Anxiety_numeric   1.461876
7                    Diabetes_numeric   1.369756
8                   HighBlood_numeric   1.677805
9                      Stroke_numeric   1.243323
10                        Gender_Male   1.906911
11                  Gender_Nonbinary   1.041777
12                   Marital_Married   1.927804
13               Marital_Never Married   1.895149
14                   Marital_Separated   1.905218
15                   Marital_Widowed   1.932970
16               Complication_risk_Low   1.595106
17            Complication_risk_Medium   2.262637
18   Initial_admin_Emergency Admission   2.865296
19  Initial_admin_Observation Admission   1.909743
```

Once I calculated the VIF values for each independent variable, I proceeded to remove Doc_visits from my model as the VIF was greater than 10 at 13.06. After removing Doc_visits, I checked the updated VIF model:

```
#calculating VIF for independent variables
vif_df["VIF"]=[variance_inflation_factor(X.values, i) for i in range(len(X.columns))]
```

```
print(vif_df)
```

```
                              feature       VIF
0                            Children  1.880873
1                                 Age  5.758430
2                              Income  2.759933
3                        Initial_days  2.569008
4                  Overweight_numeric  3.127351
5                     Anxiety_numeric  1.451134
6                    Diabetes_numeric  1.357735
7                   HighBlood_numeric  1.657749
8                      Stroke_numeric  1.238293
9                          Gender_Male  1.872013
10                  Gender_Nonbinary  1.040078
11                   Marital_Married  1.848077
12               Marital_Never Married  1.810949
13                   Marital_Separated  1.822964
14                   Marital_Widowed  1.849150
15               Complication_risk_Low  1.567946
16            Complication_risk_Medium  2.204364
17   Initial_admin_Emergency Admission  2.700515
18  Initial_admin_Observation Admission  1.831224
```

Removing Doc_visits resulted in all the remaining variables to have a VIF of 5.6 or less. Instead of removing Age, which has the next highest VIF of 5.75, I will keep it for now as, thinking logically, Age could be an important variable in my model and is below 10 for the VIF.

Now that my reduction based on VIF is complete, I will create a new MLE model with the remaining variables:

Logit Regression Results

Dep. Variable:	ReAdmis_numeric	No. Observations:	10000
Model:	Logit	Df Residuals:	9980
Method:	MLE	Df Model:	19
Date:	Tue, 06 Feb 2024	Pseudo R-squ.:	0.9399
Time:	09:48:53	Log-Likelihood:	-394.71
converged:	True	LL-Null:	-6572.9
Covariance Type:	nonrobust	LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
Children	0.0868	0.042	2.084	0.037	0.005	0.168
Age	0.0030	0.005	0.666	0.506	-0.006	0.012
Income	1.773e-06	3.29e-06	0.538	0.590	-4.68e-06	8.23e-06
Initial_days	1.2071	0.061	19.849	0.000	1.088	1.326
Overweight_numeric	-0.2793	0.205	-1.361	0.173	-0.681	0.123
Anxiety_numeric	-0.8767	0.202	-4.345	0.000	-1.272	-0.481
Diabetes_numeric	0.3933	0.209	1.884	0.060	-0.016	0.802
HighBlood_numeric	0.7146	0.195	3.662	0.000	0.332	1.097
Stroke_numeric	1.5044	0.244	6.161	0.000	1.026	1.983
Gender_Male	0.0358	0.189	0.189	0.850	-0.334	0.406
Gender_Nonbinary	0.4790	0.617	0.776	0.438	-0.731	1.689
Marital_Married	0.1158	0.294	0.393	0.694	-0.461	0.693
Marital_Never Married	0.2289	0.302	0.757	0.449	-0.364	0.821
Marital_Separated	-0.1175	0.307	-0.383	0.702	-0.719	0.484
Marital_Widowed	0.2714	0.296	0.918	0.359	-0.308	0.851
Complication_risk_Low	-1.3884	0.259	-5.363	0.000	-1.896	-0.881
Complication_risk_Medium	-0.2930	0.213	-1.375	0.169	-0.711	0.125
Initial_admin_Emergency Admission	1.9367	0.240	8.064	0.000	1.466	2.407
Initial_admin_Observation Admission	0.5411	0.256	2.110	0.035	0.038	1.044
const	-66.9769	3.390	-19.756	0.000	-73.622	-60.332

Possibly complete quasi-separation: A fraction 0.78 of observations can be perfectly predicted. This might indicate that there is complete quasi-separation. In this case some parameters will not be identified.

The Pseudo R-Squared value is still good, but slightly lower, at 0.9399, and the LLR p-value remains the same at 0.000. There are quite a few variables with p-values that are greater than the 0.05 alpha value, indicating those variables are not statistically significant. Also, the model still notes there is possible complete quasi-separation. Per WGU Course Materials, I will now use backward stepwise elimination process to create updated MLE models by systematically removing the variables with the highest p-values until no variables with p-values greater than the alpha value of 0.05 exist in the model (Western Governors University, n.d.). I will start by removing the variable with the highest p-value, which is Gender_Male and has a p-value of 0.85:

```
#Our Pseudo R value and LLR p value are unchanged.  We do have several variables with p values greater than the alpha value 0.05.
y = df.ReAdmis_numeric
X = df[["Children", "Age", "Income", "Initial_days", "Overweight_numeric", "Anxiety_numeric", "Diabetes_numeric", "HighBlood_nume
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.039473
         Iterations 13
                           Logit Regression Results
==============================================================================
Dep. Variable:        ReAdmis_numeric   No. Observations:                10000
Model:                          Logit   Df Residuals:                    9981
Method:                           MLE   Df Model:                          18
Date:                Tue, 06 Feb 2024   Pseudo R-squ.:                 0.9399
Time:                        09:50:13   Log-Likelihood:                -394.73
converged:                       True   LL-Null:                       -6572.9
Covariance Type:            nonrobust   LLR p-value:                     0.000
=================================================================================================
                                    coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------------------
Children                          0.0870      0.042      2.091      0.037       0.005       0.169
Age                               0.0029      0.004      0.649      0.516      -0.006       0.012
Income                         1.762e-06   3.29e-06      0.535      0.593   -4.69e-06    8.22e-06
Initial_days                      1.2078      0.061     19.879      0.000       1.089       1.327
Overweight_numeric               -0.2802      0.205     -1.366      0.172      -0.682       0.122
Anxiety_numeric                  -0.8787      0.202     -4.361      0.000      -1.274      -0.484
Diabetes_numeric                  0.3926      0.209      1.881      0.060      -0.016       0.802
HighBlood_numeric                 0.7133      0.195      3.658      0.000       0.331       1.095
Stroke_numeric                    1.5080      0.243      6.193      0.000       1.031       1.985
Gender_Nonbinary                  0.4623      0.611      0.756      0.449      -0.736       1.660
Marital_Married                   0.1187      0.294      0.404      0.686      -0.457       0.695
Marital_Never Married             0.2276      0.302      0.753      0.451      -0.365       0.820
Marital_Separated                -0.1154      0.307     -0.376      0.707      -0.716       0.486
Marital_Widowed                   0.2750      0.295      0.933      0.351      -0.303       0.853
Complication_risk_Low            -1.3911      0.258     -5.383      0.000      -1.898      -0.885
Complication_risk_Medium         -0.2954      0.213     -1.388      0.165      -0.712       0.122
Initial_admin_Emergency Admission 1.9356      0.240      8.062      0.000       1.465       2.406
Initial_admin_Observation Admission 0.5416    0.256      2.112      0.035       0.039       1.044
const                           -66.9885      3.391    -19.755      0.000     -73.635     -60.342
=================================================================================================

Possibly complete quasi-separation: A fraction 0.78 of observations can be
```

The Pseudo R-Squared value remains the same and I still have large p-values. I will continue with my backward stepwise elimination process. The next variable I will eliminate is Marital_Separated which has a p-value of 0.707:

```
#I will now remove the next highest p-value from the model - Marital_Separated at 0.707
y = df.ReAdmis_numeric
X = df[["Children", "Age", "Income", "Initial_days", "Overweight_numeric", "Anxiety_numeric", "Diabetes_numeric", "HighBlood_nume
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.039480
         Iterations 13
                           Logit Regression Results
==============================================================================
Dep. Variable:        ReAdmis_numeric   No. Observations:                10000
Model:                          Logit   Df Residuals:                    9982
Method:                           MLE   Df Model:                          17
Date:                Tue, 06 Feb 2024   Pseudo R-squ.:                 0.9399
Time:                        09:51:26   Log-Likelihood:                -394.80
converged:                       True   LL-Null:                       -6572.9
Covariance Type:            nonrobust   LLR p-value:                     0.000
=================================================================================================
                                    coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------------------
Children                          0.0862      0.042      2.073      0.038       0.005       0.168
Age                               0.0028      0.004      0.626      0.531      -0.006       0.012
Income                         1.739e-06   3.29e-06      0.529      0.597   -4.71e-06    8.19e-06
Initial_days                      1.2081      0.061     19.878      0.000       1.089       1.327
Overweight_numeric               -0.2827      0.205     -1.379      0.168      -0.685       0.119
Anxiety_numeric                  -0.8767      0.201     -4.354      0.000      -1.271      -0.482
Diabetes_numeric                  0.3891      0.208      1.867      0.062      -0.019       0.798
HighBlood_numeric                 0.7151      0.195      3.669      0.000       0.333       1.097
Stroke_numeric                    1.5061      0.243      6.188      0.000       1.029       1.983
Gender_Nonbinary                  0.4792      0.608      0.788      0.431      -0.712       1.671
Marital_Married                   0.1780      0.248      0.717      0.473      -0.308       0.664
Marital_Never Married             0.2858      0.260      1.101      0.271      -0.223       0.795
Marital_Widowed                   0.3338      0.250      1.335      0.182      -0.156       0.824
Complication_risk_Low            -1.3880      0.258     -5.373      0.000      -1.894      -0.882
Complication_risk_Medium         -0.2967      0.213     -1.395      0.163      -0.714       0.120
Initial_admin_Emergency Admission 1.9339      0.240      8.057      0.000       1.463       2.404
Initial_admin_Observation Admission 0.5353    0.256      2.092      0.036       0.034       1.037
const                           -67.0552      3.389    -19.788      0.000     -73.697     -60.413
=================================================================================================

Possibly complete quasi-separation: A fraction 0.78 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
```

The Pseudo R-Squared value and LLR p-values remain the same, as does the note regarding possible complete quasi-separation. I will next remove the variable Income which has a p-value of 0.597:

```
#I will now remove the next highest p-value from the model - Income at 0.597
y = df.ReAdmis_numeric
X = df[["Children", "Age", "Initial_days", "Overweight_numeric", "Anxiety_numeric", "Diabetes_numeric", "HighBlood_numeric", "Str
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.039494
         Iterations 13
                           Logit Regression Results
==============================================================================
Dep. Variable:        ReAdmis_numeric   No. Observations:                10000
Model:                          Logit   Df Residuals:                    9983
Method:                           MLE   Df Model:                          16
Date:                Tue, 06 Feb 2024   Pseudo R-squ.:                 0.9399
Time:                        09:51:58   Log-Likelihood:                -394.94
converged:                       True   LL-Null:                       -6572.9
Covariance Type:            nonrobust   LLR p-value:                     0.000
=========================================================================================================
                                            coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------------------------------
Children                                  0.0864      0.042      2.079      0.038       0.005       0.168
Age                                       0.0026      0.004      0.589      0.556      -0.006       0.011
Initial_days                              1.2075      0.061     19.886      0.000       1.089       1.327
Overweight_numeric                       -0.2832      0.205     -1.383      0.167      -0.685       0.118
Anxiety_numeric                          -0.8753      0.201     -4.346      0.000      -1.270      -0.481
Diabetes_numeric                          0.3837      0.208      1.843      0.065      -0.024       0.792
HighBlood_numeric                         0.7137      0.195      3.662      0.000       0.332       1.096
Stroke_numeric                            1.5005      0.243      6.173      0.000       1.024       1.977
Gender_Nonbinary                          0.4762      0.607      0.785      0.433      -0.713       1.666
Marital_Married                           0.1760      0.248      0.710      0.478      -0.310       0.662
Marital_Never Married                     0.2813      0.259      1.084      0.278      -0.227       0.790
Marital_Widowed                           0.3326      0.250      1.330      0.184      -0.158       0.823
Complication_risk_Low                    -1.3828      0.258     -5.359      0.000      -1.889      -0.877
Complication_risk_Medium                 -0.2947      0.213     -1.385      0.166      -0.712       0.122
Initial_admin_Emergency Admission         1.9262      0.239      8.043      0.000       1.457       2.396
Initial_admin_Observation Admission       0.5295      0.256      2.072      0.038       0.029       1.030
const                                   -66.9349      3.376    -19.828      0.000     -73.551     -60.318
=========================================================================================================

Possibly complete quasi-separation: A fraction 0.78 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
```

The next variable with the highest p-value is Age with a p-value of 0.556. I will now remove that variable from the MLE model:

```
#I will now remove the next highest p-value from the model - Age at 0.556
y = df.ReAdmis_numeric
X = df[["Children", "Initial_days", "Overweight_numeric", "Anxiety_numeric", "Diabetes_numeric", "HighBlood_numeric", "Stroke_num
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.039511
         Iterations 13
                           Logit Regression Results
==============================================================================
Dep. Variable:        ReAdmis_numeric   No. Observations:                10000
Model:                          Logit   Df Residuals:                    9984
Method:                           MLE   Df Model:                          15
Date:                Tue, 06 Feb 2024   Pseudo R-squ.:                 0.9399
Time:                        09:52:22   Log-Likelihood:                -395.11
converged:                       True   LL-Null:                       -6572.9
Covariance Type:            nonrobust   LLR p-value:                     0.000
=================================================================================================
                                    coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------------------
Children                          0.0865      0.042      2.081      0.037       0.005       0.168
Initial_days                      1.2085      0.061     19.890      0.000       1.089       1.328
Overweight_numeric               -0.2815      0.205     -1.375      0.169      -0.683       0.120
Anxiety_numeric                  -0.8671      0.201     -4.320      0.000      -1.260      -0.474
Diabetes_numeric                  0.3760      0.208      1.811      0.070      -0.031       0.783
HighBlood_numeric                 0.7143      0.195      3.667      0.000       0.333       1.096
Stroke_numeric                    1.4990      0.243      6.170      0.000       1.023       1.975
Gender_Nonbinary                  0.4678      0.609      0.768      0.443      -0.727       1.662
Marital_Married                   0.1750      0.248      0.706      0.480      -0.311       0.660
Marital_Never Married             0.2635      0.257      1.024      0.306      -0.241       0.768
Marital_Widowed                   0.3223      0.250      1.291      0.197      -0.167       0.812
Complication_risk_Low            -1.3820      0.258     -5.355      0.000      -1.888      -0.876
Complication_risk_Medium         -0.3004      0.212     -1.414      0.157      -0.717       0.116
Initial_admin_Emergency Admission 1.9220      0.239      8.037      0.000       1.453       2.391
Initial_admin_Observation Admission 0.5300    0.255      2.077      0.038       0.030       1.030
const                           -66.8372      3.371    -19.827      0.000     -73.444     -60.230
=================================================================================================

Possibly complete quasi-separation: A fraction 0.78 of observations can be
perfectly predicted. This might indicate that there is complete
```

The Pseudo R-Squared value and LLR p-values remain the same. I still have several variables with p-values greater than the alpha value 0.05. I will now remove Marital_Married as it has the highest p-value left of 0.48:

```
#I will now remove the next highest p-value from the model - Marital_Married at 0.48
y = df.ReAdmis_numeric
X = df[["Children", "Initial_days", "Overweight_numeric", "Anxiety_numeric", "Diabetes_numeric", "HighBlood_numeric", "Stroke_num
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.039536
         Iterations 13
                           Logit Regression Results
==============================================================================
Dep. Variable:        ReAdmis_numeric   No. Observations:                10000
Model:                          Logit   Df Residuals:                    9985
Method:                           MLE   Df Model:                          14
Date:                Tue, 06 Feb 2024   Pseudo R-squ.:                0.9398
Time:                        09:52:59   Log-Likelihood:               -395.36
converged:                       True   LL-Null:                      -6572.9
Covariance Type:            nonrobust   LLR p-value:                     0.000
=======================================================================================================
                                          coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------------------------
Children                                0.0867      0.042      2.085      0.037       0.005       0.168
Initial_days                            1.2086      0.061     19.887      0.000       1.089       1.328
Overweight_numeric                     -0.2742      0.204     -1.343      0.179      -0.674       0.126
Anxiety_numeric                        -0.8680      0.201     -4.324      0.000      -1.261      -0.475
Diabetes_numeric                        0.3655      0.207      1.766      0.077      -0.040       0.771
HighBlood_numeric                       0.7217      0.194      3.711      0.000       0.341       1.103
Stroke_numeric                          1.5006      0.243      6.176      0.000       1.024       1.977
Gender_Nonbinary                        0.4704      0.610      0.771      0.441      -0.726       1.666
Marital_Never Married                   0.1971      0.240      0.823      0.411      -0.272       0.667
Marital_Widowed                         0.2558      0.231      1.107      0.268      -0.197       0.709
Complication_risk_Low                  -1.3781      0.258     -5.344      0.000      -1.884      -0.873
Complication_risk_Medium               -0.2970      0.212     -1.400      0.161      -0.713       0.119
Initial_admin_Emergency Admission       1.9310      0.239      8.083      0.000       1.463       2.399
Initial_admin_Observation Admission     0.5359      0.255      2.101      0.036       0.036       1.036
const                                 -66.7881      3.369    -19.822      0.000     -73.392     -60.184
=======================================================================================================

Possibly complete quasi-separation: A fraction 0.78 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
```

The Pseudo R-Squared value has lowered fractionally to 0.9398. The LLR p-value remains the same, and there are still variables with p-values greater than 0.05. The next variable I will remove is Gender_Nonbinary that has a p-value of 0.441:

```
#I will now remove the next highest p-value from the model - Gender_Nonbinary at 0.441
y = df.ReAdmis_numeric
X = df[["Children", "Initial_days", "Overweight_numeric", "Anxiety_numeric", "Diabetes_numeric", "HighBlood_numeric", "Stroke_num
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.039566
         Iterations 13
                           Logit Regression Results
==============================================================================
Dep. Variable:        ReAdmis_numeric   No. Observations:                10000
Model:                          Logit   Df Residuals:                    9986
Method:                           MLE   Df Model:                          13
Date:                Tue, 06 Feb 2024   Pseudo R-squ.:                0.9398
Time:                        09:53:29   Log-Likelihood:               -395.66
converged:                       True   LL-Null:                      -6572.9
Covariance Type:            nonrobust   LLR p-value:                     0.000
=====================================================================================================
                                        coef    std err          z      P>|z|      [0.025      0.975]
-----------------------------------------------------------------------------------------------------
Children                              0.0886      0.042      2.132      0.033       0.007       0.170
Initial_days                          1.2078      0.061     19.896      0.000       1.089       1.327
Overweight_numeric                   -0.2657      0.204     -1.305      0.192      -0.665       0.133
Anxiety_numeric                      -0.8645      0.200     -4.313      0.000      -1.257      -0.472
Diabetes_numeric                      0.3734      0.207      1.807      0.071      -0.032       0.778
HighBlood_numeric                     0.7224      0.194      3.714      0.000       0.341       1.104
Stroke_numeric                        1.4928      0.242      6.157      0.000       1.018       1.968
Marital_Never Married                 0.1969      0.240      0.822      0.411      -0.273       0.667
Marital_Widowed                       0.2547      0.231      1.102      0.271      -0.198       0.708
Complication_risk_Low                -1.3764      0.258     -5.338      0.000      -1.882      -0.871
Complication_risk_Medium             -0.2898      0.212     -1.368      0.171      -0.705       0.125
Initial_admin_Emergency Admission     1.9186      0.238      8.056      0.000       1.452       2.385
Initial_admin_Observation Admission   0.5261      0.254      2.068      0.039       0.028       1.025
const                               -66.7423      3.366    -19.829      0.000     -73.339     -60.145
=====================================================================================================

Possibly complete quasi-separation: A fraction 0.78 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
```

The next variable I will remove from the model is Marital_Never Married, which has a p-value of 0.411:

```
#I will now remove the next highest p-value from the model - Marital_Never Married at 0.411
y = df.ReAdmis_numeric
X = df[["Children", "Initial_days", "Overweight_numeric", "Anxiety_numeric", "Diabetes_numeric", "HighBlood_numeric", "Stroke_num
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.039600
         Iterations 13
                           Logit Regression Results
==============================================================================
Dep. Variable:        ReAdmis_numeric   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9987
Method:                           MLE   Df Model:                           12
Date:                Tue, 06 Feb 2024   Pseudo R-squ.:                 0.9398
Time:                        09:54:08   Log-Likelihood:                -396.00
converged:                       True   LL-Null:                       -6572.9
Covariance Type:            nonrobust   LLR p-value:                     0.000
=================================================================================================
                                    coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------------------
Children                          0.0876      0.041      2.111      0.035       0.006       0.169
Initial_days                      1.2072      0.061     19.896      0.000       1.088       1.326
Overweight_numeric               -0.2676      0.204     -1.314      0.189      -0.667       0.132
Anxiety_numeric                  -0.8645      0.200     -4.317      0.000      -1.257      -0.472
Diabetes_numeric                  0.3679      0.206      1.784      0.074      -0.036       0.772
HighBlood_numeric                 0.7190      0.194      3.701      0.000       0.338       1.100
Stroke_numeric                    1.4927      0.242      6.156      0.000       1.017       1.968
Marital_Widowed                   0.2050      0.223      0.919      0.358      -0.232       0.642
Complication_risk_Low            -1.3818      0.257     -5.370      0.000      -1.886      -0.878
Complication_risk_Medium         -0.2894      0.212     -1.367      0.172      -0.704       0.126
Initial_admin_Emergency Admission 1.9185      0.238      8.067      0.000       1.452       2.385
Initial_admin_Observation Admission 0.5175    0.254      2.038      0.042       0.020       1.015
const                           -66.6485      3.361    -19.830      0.000     -73.236     -60.061
=================================================================================================

Possibly complete quasi-separation: A fraction 0.78 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
```

The next variable I will remove from the model is Marital_Widowed, which has a p-value that increased to 0.358:

```
#I will now remove the next highest p-value from the model - Marital_Widowed at 0.358
y = df.ReAdmis_numeric
X = df[["Children", "Initial_days", "Overweight_numeric", "Anxiety_numeric", "Diabetes_numeric", "HighBlood_numeric", "Stroke_num
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.039643
         Iterations 13
                           Logit Regression Results
==============================================================================
Dep. Variable:        ReAdmis_numeric   No. Observations:                10000
Model:                          Logit   Df Residuals:                    9988
Method:                           MLE   Df Model:                          11
Date:                Tue, 06 Feb 2024   Pseudo R-squ.:                0.9397
Time:                        12:47:15   Log-Likelihood:               -396.43
converged:                       True   LL-Null:                      -6572.9
Covariance Type:            nonrobust   LLR p-value:                     0.000
=================================================================================================
                                    coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------------------
Children                          0.0870      0.041      2.099      0.036       0.006       0.168
Initial_days                      1.2061      0.061     19.880      0.000       1.087       1.325
Overweight_numeric               -0.2674      0.204     -1.314      0.189      -0.666       0.132
Anxiety_numeric                  -0.8654      0.200     -4.326      0.000      -1.257      -0.473
Diabetes_numeric                  0.3659      0.206      1.776      0.076      -0.038       0.770
HighBlood_numeric                 0.7125      0.194      3.674      0.000       0.332       1.093
Stroke_numeric                    1.4774      0.242      6.110      0.000       1.004       1.951
Complication_risk_Low            -1.3791      0.257     -5.371      0.000      -1.882      -0.876
Complication_risk_Medium         -0.2788      0.211     -1.320      0.187      -0.693       0.135
Initial_admin_Emergency Admission 1.9256      0.238      8.107      0.000       1.460       2.391
Initial_admin_Observation Admission 0.5169    0.254      2.038      0.042       0.020       1.014
const                           -66.5461      3.358    -19.818      0.000     -73.127     -59.965
=================================================================================================

Possibly complete quasi-separation: A fraction 0.78 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
```

The Pseudo R-Squared value has lowered fractionally to 0.9397. The LLR p-value remains the same, and there are still variables with p-values greater than 0.05. The next variable I will remove is Overweight_numeric that has a p-value of 0.189:

```
#I will now remove the next highest p-value from the model - Overweight_numeric at 0.189
y = df.ReAdmis_numeric
X = df[["Children", "Initial_days", "Anxiety_numeric", "Diabetes_numeric", "HighBlood_numeric", "Stroke_numeric", "Complication_r
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.039729
         Iterations 13
                           Logit Regression Results
==============================================================================
Dep. Variable:        ReAdmis_numeric   No. Observations:                10000
Model:                          Logit   Df Residuals:                    9989
Method:                           MLE   Df Model:                          10
Date:                Tue, 06 Feb 2024   Pseudo R-squ.:                0.9396
Time:                        09:55:39   Log-Likelihood:               -397.29
converged:                       True   LL-Null:                      -6572.9
Covariance Type:            nonrobust   LLR p-value:                     0.000
=================================================================================================================
                                    coef    std err          z      P>|z|      [0.025      0.975]
-----------------------------------------------------------------------------------------------------------------
Children                          0.0888      0.041      2.142      0.032       0.008       0.170
Initial_days                      1.2013      0.060     19.926      0.000       1.083       1.319
Anxiety_numeric                  -0.8639      0.200     -4.318      0.000      -1.256      -0.472
Diabetes_numeric                  0.3668      0.206      1.783      0.075      -0.036       0.770
HighBlood_numeric                 0.6943      0.193      3.594      0.000       0.316       1.073
Stroke_numeric                    1.4734      0.241      6.107      0.000       1.001       1.946
Complication_risk_Low            -1.3669      0.256     -5.338      0.000      -1.869      -0.865
Complication_risk_Medium         -0.2656      0.211     -1.260      0.208      -0.679       0.148
Initial_admin_Emergency Admission 1.9119      0.237      8.066      0.000       1.447       2.376
Initial_admin_Observation Admission 0.5015    0.253      1.981      0.048       0.005       0.998
const                           -66.4697      3.348    -19.855      0.000     -73.031     -59.908
=================================================================================================================

Possibly complete quasi-separation: A fraction 0.78 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
```

The Pseudo R-Squared value has lowered fractionally again to 0.9396. The LLR p-value remains the same, and there are still variables with p-values greater than 0.05. The next variable I will remove is Complication_risk_Medium that has a p-value that increased to 0.208:

```
#I will now remove the next highest p-value from the model - Complication_risk_Medium at 0.208
y = df.ReAdmis_numeric
X = df[["Children", "Initial_days", "Anxiety_numeric", "Diabetes_numeric", "HighBlood_numeric", "Stroke_numeric", "Complication_r
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.039809
         Iterations 13
                           Logit Regression Results
==============================================================================
Dep. Variable:        ReAdmis_numeric   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9990
Method:                           MLE   Df Model:                            9
Date:                Tue, 06 Feb 2024   Pseudo R-squ.:                 0.9394
Time:                        09:56:22   Log-Likelihood:                -398.09
converged:                       True   LL-Null:                       -6572.9
Covariance Type:            nonrobust   LLR p-value:                     0.000
=====================================================================================================
                                        coef    std err          z      P>|z|      [0.025      0.975]
-----------------------------------------------------------------------------------------------------
Children                              0.0925      0.041      2.232      0.026       0.011       0.174
Initial_days                          1.1999      0.060     19.924      0.000       1.082       1.318
Anxiety_numeric                      -0.8803      0.200     -4.409      0.000      -1.272      -0.489
Diabetes_numeric                      0.3748      0.205      1.825      0.068      -0.028       0.777
HighBlood_numeric                     0.6920      0.193      3.587      0.000       0.314       1.070
Stroke_numeric                        1.4733      0.241      6.120      0.000       1.001       1.945
Complication_risk_Low                -1.2155      0.225     -5.408      0.000      -1.656      -0.775
Initial_admin_Emergency Admission     1.9115      0.237      8.080      0.000       1.448       2.375
Initial_admin_Observation Admission   0.4948      0.253      1.956      0.050      -0.001       0.991
const                               -66.5483      3.348    -19.875      0.000     -73.111     -59.986
=====================================================================================================

Possibly complete quasi-separation: A fraction 0.78 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
```

The Pseudo R-Squared value has lowered fractionally again to 0.9394. The LLR p-value remains the same, and there is still one variable with a p-value greater than 0.05. The next variable I will remove is Diabetes_numeric that has a p-value that increased to 0.068:

```
#I will now remove the next highest p-value from the model - Diabetes_numeric at 0.068
y = df.ReAdmis_numeric
X = df[["Children", "Initial_days", "Anxiety_numeric", "HighBlood_numeric", "Stroke_numeric", "Complication_risk_Low", "Initial_a
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.039977
         Iterations 13
                           Logit Regression Results
==============================================================================
Dep. Variable:        ReAdmis_numeric   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9991
Method:                           MLE   Df Model:                            8
Date:                Tue, 06 Feb 2024   Pseudo R-squ.:                 0.9392
Time:                        09:57:08   Log-Likelihood:                -399.77
converged:                       True   LL-Null:                       -6572.9
Covariance Type:            nonrobust   LLR p-value:                     0.000
=================================================================================================
                                    coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------------------
Children                          0.0915      0.041      2.218      0.027       0.011       0.172
Initial_days                      1.1973      0.060     19.913      0.000       1.079       1.315
Anxiety_numeric                  -0.8851      0.199     -4.441      0.000      -1.276      -0.494
HighBlood_numeric                 0.6406      0.190      3.376      0.001       0.269       1.013
Stroke_numeric                    1.4434      0.239      6.030      0.000       0.974       1.913
Complication_risk_Low            -1.2117      0.225     -5.393      0.000      -1.652      -0.771
Initial_admin_Emergency Admission 1.9022      0.235      8.090      0.000       1.441       2.363
Initial_admin_Observation Admission 0.4846    0.252      1.924      0.054      -0.009       0.978
const                           -66.2655      3.335    -19.869      0.000     -72.802     -59.729
=================================================================================================

Possibly complete quasi-separation: A fraction 0.78 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
```

The Pseudo R-Squared value has lowered fractionally again to 0.9392. The LLR p-value remains the same, and there is still one variable with a p-value greater than 0.05. The p-value of Initial_admin_Observation Admission has increased to 0.054, just over the 0.005 alpha. I will remove that variable next:

```
#I will now remove the next highest p-value from the model - Initial_admin_Observation Admission at 0.054
y = df.ReAdmis_numeric
X = df[["Children", "Initial_days", "Anxiety_numeric", "HighBlood_numeric", "Stroke_numeric", "Complication_risk_Low", "Initial_a
logit_model=sm.Logit(y,X)
result=logit_model.fit()
print(result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.040163
         Iterations 13
                           Logit Regression Results
==============================================================================
Dep. Variable:        ReAdmis_numeric   No. Observations:                10000
Model:                          Logit   Df Residuals:                    9992
Method:                           MLE   Df Model:                           7
Date:                Tue, 06 Feb 2024   Pseudo R-squ.:                 0.9389
Time:                        09:57:42   Log-Likelihood:                -401.63
converged:                       True   LL-Null:                       -6572.9
Covariance Type:            nonrobust   LLR p-value:                     0.000
=================================================================================================
                                    coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------------------
Children                          0.0918      0.041      2.231      0.026       0.011       0.172
Initial_days                      1.1915      0.060     19.962      0.000       1.075       1.308
Anxiety_numeric                  -0.8900      0.199     -4.477      0.000      -1.280      -0.500
HighBlood_numeric                 0.6346      0.189      3.352      0.001       0.264       1.006
Stroke_numeric                    1.4540      0.238      6.120      0.000       0.988       1.920
Complication_risk_Low            -1.2153      0.224     -5.429      0.000      -1.654      -0.777
Initial_admin_Emergency Admission  1.6643      0.197      8.446      0.000       1.278       2.050
const                           -65.7128      3.297    -19.930      0.000     -72.175     -59.250
=================================================================================================

Possibly complete quasi-separation: A fraction 0.77 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
```

My final logistic regression model using MLE (above) shows that I have 7 variables left that are statistically significant with p-values less than 0.05: Children, Initial_days, Anxiety_numeric, HighBlood_numeric, Stroke_numeric, Complication_risk_Low, and Initial_admin_Emergency Admission. The Pseudo R-squared value is 0.9389, which is slightly lower than the initial model, however, the LLR p-value is still at 0.000 confirming the model is good. Further analysis of the initial and final models is included in the next section.

E1. Model Comparison

My initial multiple logistic regression model had many variables in it, not all of which were statistically important to the model itself. One variable was removed due to a VIF greater than 10, indicating a high probability of multicollinearity. An additional 12 variables were removed from the model using the backward stepwise elimination method. These variables were removed as their p-value was greater than the 0.05 alpha value. Variables with p-values greater than the alpha value generally are not considered statistically significant, and therefore should be removed from the model. This process was completed one variable at a time starting with the variable with the highest p-value and then reviewing the updated model to see how the removal of that variable affected the other variables left including their p-value. Ultimately, it was determined that 7 variables had p-value less than the alpha value and thus were found to be statistically significant.

My initial model had a very good Pseudo R-Squared value of 0.94. My final model had a lower but still good Pseudo R-Squared value of 0.9389. The LLR p-value of both models was 0.00, which is less

than the 0.05 alpha and indicates the model is useful. The final model only includes variables that are statistically significant to the model, whereas the initial model includes variables that are not statistically significant. With the other factors being close and both models being determined useful, the final model is most likely the best model due to only including statistically significant variables.

Another way to analyze how these models compare is finding the AIC, Akaike's Information Criteria, where the smaller the AIC value is, the better the model fits (Western Governors University, n.d.). The AIC of the initial model and final model are calculated below, where the initial model is calculated first, then the final (Statology, 2021):

```
#calculating AIC for initial model
y = df.ReAdmis_numeric
X = df[["Children", "Age", "Income", "Doc_visits", "Initial_days", "Overweight_numeric", "Anxiety_numeric", "Diabetes_numeric",
X=sm.add_constant(X)
model=sm.OLS(y,X). fit()
print(model.aic)
```

```
898.351157515357
```

```
#calculating AIC for reduced model
y = df.ReAdmis_numeric
X = df[["Children", "Initial_days", "Anxiety_numeric", "HighBlood_numeric", "Stroke_numeric", "Complication_risk_Low", "Initial_
X=sm.add_constant(X)
model=sm.OLS(y,X). fit()
print(model.aic)
```

```
879.5525237007641
```

As the final model has a lower AIC value of 879.55 (versus 898.35 for the initial model), it is confirmed via AIC that the final model is a better fit than the initial model.

E2. Output and Calculations

Using code from Geeks to Geeks to obtain the accuracy calculation and confusion matrix of the initial and final models, I was able to determine that my initial model is 0.91, or 91%, accurate, and my final reduced model is 0.98, or 98%, accurate (Geeks for Geeks, n.d.). See the below initial and final confusion matrices:

Initial Accuracy Calculation and Confusion Matrix:

```
#confusion matrix and accuracy of initial model
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=0)
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
y_pred = logreg.predict(X_test)
print('Accuracy of logistic regression: {:.2f}'.format(logreg.score(X_test, y_test)))
final_matrix = confusion_matrix(y_test, y_pred)
print(final_matrix)
```

```
Accuracy of logistic regression: 0.91
[[1419  185]
 [  49  847]]
```

Final Accuracy Calculation and Confusion Matrix:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=0)
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
y_pred = logreg.predict(X_test)
print('Accuracy of logistic regression: {:.2f}'.format(logreg.score(X_test, y_test)))
final_matrix = confusion_matrix(y_test, y_pred)
print(final_matrix)
```

```
Accuracy of logistic regression: 0.98
[[1580   24]
 [  26  870]]
```

Just as the AIC confirmed that the final model was a better fit than the initial model, the accuracy calculation confirms this by showing that the final model is more accurate than the initial model. The full code is also attached in the pdf and ipynb documents titled KMoikD208Code2.

E3. Code

Please see the attached document titled KMoikD208Code2 in both pdf and ipynb format.

Part V: Data Summary and Implications

F1. Results

The initial logistic equation for my model was:

ReAdmis = -66.8721 + 0.0871*Children + 0.003*Age + 1.691e-06*Income – 0.0309*Doc_visits + 1.2.80*Initial_days – 0.2789*Overweight_numeric – 0.8776*Anxiety_numeric + 0.3974*Diabetes_numeric + 0.7188*HighBlood_numeric + 1.5031*Stroke_numeric + 0.0366*Gender_Male + 0.4912*Gender_Nonbinary + 0.1155*Marital_Married+0.2276*Marital_Never Married – 0.1211*Marital_Separated + 0.2685*Marital_Widowed – 1.3881*Complication_risk_Low – 0.2959*Complication_risk_Medium + 1.9412*Initial_admin_Emergency Admission + 0.5478*Initial_admin_Observation_admission

The logistic equation for my final model is:

ReAdmis = -65.718 + 0.0918*Children + 1.1915*Initial_days – 0.89*Anxiety_numeric + 0.6346*HighBlood_numeric + 1.4540*Stroke_numeric – 1.2153*Complication_risk_Low + 1.6643*Initial_admin_Emergency Admission

The coefficients in my final model mean that assuming everything else remains constant:

An increase of 1 child increases the log-odds of readmission by 0.0918
An increase of 1 initial day increases the log-odds of readmission by 1.1915
An increase of 1 for anxiety decreases the log-odds of readmission by 0.89
An increase of 1 for high blood pressure increases the log-odds of readmission by 0.6346
An increase of 1 for stroke increases the log-odds of readmission by 1.454

An increase of 1 for complication_risk_Low decreases the log-odds of readmission by 1.2153
An increase of 1 for initial admin Emergency Admission increases the log-odds of readmission by 1.6643

The final model appears to be both statistically and practically significant. The LLR p-value of 0.000, which is less than the alpha value of 0.05, indicates the model is useful. Since all variables included in the final model have p-values less than 0.05, this indicates they are statistically significant. Additionally, the AIC of the final model of 0.98 indicates that this final model is very accurate and thus both statistically and practically significant to use to predict rates of readmission.

Some of the limitations of my analysis include the type of data provided, or not provided as the case may be. Additional data points would be helpful to know such as certain diagnoses, comorbidities, and the like that would most likely affect readmission rates. Additionally, this dataset and thus my model would benefit from more data in general, including for the data that is already present. For instance, it may be important to have the data for patients under the age of 18, and for those admitted for less than 1 day.

F2. Recommendations

While my results and model were overall good, I would recommend obtaining more and different data to use in the model. Additionally, all my models, including the final reduced model, indicated there was possible quasi-complete separation. It may be beneficial into digging deeper into that. Obtaining a larger dataset could help as well as trying other methods, such as using recursive feature elimination instead of backward stepwise elimination to see how the models compare and if the quasi-complete separation is still noted as a possibility.

Part VI: Demonstration

G. Panopto Demonstration

The link to my Panopto video recording is:
https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=63936846-b3f3-4127-9bc4-b11000005ab3

H. Sources of Third-Party Code

For B2: Western Governors University. (n.d.). *D208 Predictive Modeling Episode 5* [Video]. https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=15852089-76a2-4828-8a42-ad3100e8460c

For D1: Western Governors University. (n.d.). *D208 Predictive Modeling Episode 4* [Video]. https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=5328e088-48c2-42d7-a5e3-ad27015d138f

For C4: Moffitt, C. (2017). *Guide to Encoding Categorical Values in Python*. Practical Business Python. https://pbpython.com/categorical-encoding.html

For D3: Western Governors University. (n.d.). *D208 Predictive Modeling Episode 5* [Video]. https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=15852089-76a2-4828-8a42-ad3100e8460c

For E1: Statology. (2021). *How to Calculate AIC of Regression Models in Python*. Statology. https://www.statology.org/aic-in-python/

For E2: Geeks for Geeks. (n.d.). *How To Do Train Test Split Using Sklearn In Python*. Geeks for Geeks. https://www.geeksforgeeks.org/how-to-do-train-test-split-using-sklearn-in-python/

I. Sources

For B1: Statology. (2020). *The 6 Assumptions of Logistic Regression*. Statology. https://www.statology.org/assumptions-of-logistic-regression/

For B2: Western Governors University Information Technology. (2024). *R or Python*. Western Governors University. https://www.wgu.edu/online-it-degrees/programming-languages/r-or-python.html

For B2: Western Governors University. (n.d.). *D208 Predictive Modeling Episode 5* [Video]. https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=15852089-76a2-4828-8a42-ad3100e8460c

For D1: Western Governors University. (n.d.). *D208 Predictive Modeling Episode 4* [Video]. https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=5328e088-48c2-42d7-a5e3-ad27015d138f

For D1: Western Governors University. (n.d.) *Getting Started with D208 Part II* [Video]. https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=39bbe2db-de7d-4bf5-913b-af5c0003da9d

For D2: University of California, Los Angeles Advanced Research Computing. (n.d.). *FAQ What is Complete or Quasi-Complete Separation in Logistic Regression and What Are Some Strategies to Deal with the Issue?.* University of California, Los Angeles. https://stats.oarc.ucla.edu/other/mult-pkg/faq/general/faqwhat-is-complete-or-quasi-complete-separation-in-logistic-regression-and-what-are-some-strategies-to-deal-with-the-issue/

For E1: Western Governors University. (n.d.) *Getting Started with D208 Part II* [Video]. https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=39bbe2db-de7d-4bf5-913b-af5c0003da9d

J. Professional Communication