D208: Predictive Modeling

Task II

Shawn Wheeler

000364855

Table of Contents

[A1: Research Question 3](#_Toc165408460)

[A2: Goals 3](#_Toc165408461)

[B1: Summary of Assumptions 3](#_Toc165408462)

[B2: Tool Benefits 3](#_Toc165408463)

[B3: Appropriate Technique 4](#_Toc165408464)

[C1: Data Cleaning 4](#_Toc165408465)

[C2: Summary Statistics 5](#_Toc165408466)

[C3: Visualizations 8](#_Toc165408467)

[C4: Data Transformation 20](#_Toc165408468)

[C5: Prepared Data 21](#_Toc165408469)

[D1: Initial Model 21](#_Toc165408470)

[D2: Justification of Model Reduction 22](#_Toc165408471)

[D3: Reduced Logistic Regression Model 27](#_Toc165408472)

[E1: Model Comparison 27](#_Toc165408473)

[E2: Output and Calculations 28](#_Toc165408474)

[E3: Code 28](#_Toc165408475)

[F1: Results 31](#_Toc165408476)

[F1: Recommendations 32](#_Toc165408477)

[G: Panopto Demonstration 33](#_Toc165408478)

[H: Sources of Third-Party Code 33](#_Toc165408479)

[I: Sources 33](#_Toc165408480)

**Part I – Research Question**

## A1: Research Question

What factors contribute to a patient being readmitted to the hospital? In other words, which independent variables contribute to our dependent, categorical variable ReAdmis?

## A2: Goals

The goal of this project is to determine if there are any variables that would directly affect ReAdmis and how much of an effect they have on ReAdmis. The independent, explanatory variables being included for analysis are Age, Income, VitD\_levels, Doc\_visits, Initial\_admin, HighBlood, Stroke, Overweight, Arthritis, Diabetes, Services, and Initial\_days.

**Part II – Method Justification**

## B1: Summary of Assumptions

There are four assumptions that this project will be concerned with regarding logistic regression models.

1) The dependent variable is binary, categorical only.

2) Each row of data is an independent observation.

3) There is not a great deal of multicollinearity between the independent variables.

4) No extreme outliers (Zach, 2020).

## B2: Tool Benefits

The tool used for this project is Python. With its great ecosystem of libraries and packages, it aligns perfectly with our goals for this project. Additional Python benefits include computational speed and consistent syntax (Western Governor’s University, n.d.). Multiple libraries and packages were utilized in this project as described in the table below.

|  |  |
| --- | --- |
| **Libraries and Packages** | **Usage** |
| pandas | Overall data manipulation |
| numpy | Mathematics operations |
| seaborn | Creating various visualizations |
| matplotlib.pyplot | Creating various visualizations |
| statsmodels.api | Logistic regression calculations |
| statsmodels.stats.outliers\_influence | Calculating variance inflation factors (VIFs) |
| Sklearn.metrics | Confusion matrix creation, calculating accuracy |
| warnings | Eliminate needless warnings on Panopto video |

## B3: Appropriate Technique

Logistic regression was chosen as the technique to attempt to answer the research question because it allows for prediction of our categorical, binary, target variable using a number of explanatory variables. This fits the intended data and should produce a result revealing greater understanding where other techniques such as multiple linear regression would not be appropriate. This is due to the fact that multiple linear regression requires a continuous dependent variable while logistic requires a binary, categorical dependent variable. ReAdmis will only contain values of 1 or 0 to represent “Yes” or “No” which means that, by definition, it would only be usable as the target variable utilizing logistic regression.

**Part III – Data Preparation**

## C1: Data Cleaning

*See attached code:* d208task2complete.ipynb

The initial data set has been previously cleaned but will still be examined for further cleaning. The goals of data cleaning here are checking for duplicates, missing data, outliers, and to see if variables require re-expression to fit the needs of logistic regression in our next step towards preparing the data.

*Copy of code below:*

# Check for duplicates, missing data, outliers

# Checking for duplicates

df.duplicated().value\_counts()

# Checking for missing data

df.isnull().sum()

# Checking for outliers in quantitative variables

quant\_columns = ['Lat', 'Lng', 'Population', 'Children', 'Age', 'Income', 'VitD\_levels', 'Doc\_visits', 'Full\_meals\_eaten', 'VitD\_supp', 'Initial\_days', 'TotalCharge', 'Additional\_charges']

for column in df:

if column in quant\_columns:

plt.figure()

plt.gca().set\_title(column)

df.boxplot([column])

## C2: Summary Statistics

In order to understand the data summaries for the variables to be used, a few definitions are necessary to understand what the screenshots below are conveying. Mean is the average of all values in a number set. Median is the middle number in a number set. 25%, 50%, and 75% refer to interquartile ranges. That means that the value for 75% has 25% of all other values above it and 75% below. 50% is the same as the median. Std means standard deviation and is used for calculating outliers. Count is the total number of values for that variable which will be 10,000 in all of these instances. Min and max refer to the smallest value and the largest value respectively. The categorical variables will be explained uniquely.

*ReAdmis:*

A close-up of numbers

Description automatically generated

Out of the 10,000 patients, 3,669 (36.7%) were readmitted while 6,331 (63.3%) were not.

*Age:*

*A screenshot of a computer screen

Description automatically generated*

The average age of patients is 53. The oldest in the data set was 89.

*Income:*

*A screenshot of a computer

Description automatically generated*

Average income of patients is $40,490. The highest was $207,249 per year while the lowest was reported at $154 per year.

*VitD\_levels:*

*A screenshot of a computer

Description automatically generated*

The average vitamin D level on intake for patients is 17.96 with mostly a variance of about 4 points. There was a maximum vitamin D level of 26.39 reported.

*Doc\_visits:*

*A screenshot of a computer code

Description automatically generated*

Doctors visited patients on average of 5 times.

*Initial\_admin:*

*A screenshot of a computer

Description automatically generated*

The initial reason for admission for patients in the data set was 50.6% emergency admissions, 25.04% elective admissions, and 24.36% observation admissions.

*HighBlood:*

*A close-up of a number

Description automatically generated*

40.9% of patients were reported as having high blood pressure while 59.1% were not.

*Stroke:*

*A close-up of numbers

Description automatically generated*

19.93% of patients had a history of stroke while 80.07% did not.

*Overweight:*

*A close-up of a number

Description automatically generated*

70.94% of patients were reported as overweight while 29.06% were not.

*Arthritis:*

*A close up of numbers

Description automatically generated*

35.74% of patients had arthritis while 64.26% did not.

*Diabetes:*

*A close up of numbers

Description automatically generated*

27.38% of patients were diabetic while 72.62% were not.

*Services:*

*A white background with black text

Description automatically generated*

For the 10,000 patients, 52.65% received blood work, 31.30% had an intravenous service, 12.25% received a CT scan, and 3.8% were given an MRI.

*Initial\_days:*

A screenshot of a computer

Description automatically generated

The length of days of the average initial stay in the hospital for a patient in the data set was shown to be around 34.5 days. The shortest stays were one day and the longest was nearly 72 days.

## C3: Visualizations

Univariate Analyses

*ReAdmis:*

*A pink rectangular object with black text

Description automatically generated*

*Age:*

*A graph of a patient age

Description automatically generated*

*Income:*

*A graph of a distribution of income

Description automatically generated*

*VitD\_levels:*

*A graph of vitamin d levels

Description automatically generated*

*Doc\_visits:*

*A green bar graph with black lines

Description automatically generated*

*Initial\_admin:*

*A pie chart with numbers and a few percentages

Description automatically generated*

*HighBlood:*

*A pie chart with numbers and a green circle

Description automatically generated*

*Stroke:*

*A graph of a patient

Description automatically generated*

*Overweight:*

*A pie chart with text on it

Description automatically generated*

*Arthritis:*

*A pink rectangular graph with black lines

Description automatically generated with medium confidence*

*Diabetes:*

*A purple and green pie chart

Description automatically generated*

*Services:*

*A bar graph with blue bars

Description automatically generated*

*Initial\_days:*

*A graph of a number of patients

Description automatically generated*

Bivariate Analyses

*ReAdmis and Age:*

*A diagram of a relationship between age and readmissing

Description automatically generated*

*ReAdmis and Income:*

*A graph of a diagram

Description automatically generated with medium confidence*

*ReAdmis and VitD\_levels:*

*A diagram of a chart

Description automatically generated with medium confidence*

*ReAdmis and Doc\_visits:*

*A diagram of a patient

Description automatically generated with medium confidence*

*ReAdmis and Initial\_admin:*

*A graph of a patient admission

Description automatically generated with medium confidence*

*ReAdmis and HighBlood:*

*A graph of a patient

Description automatically generated with medium confidence*

*ReAdmis and Stroke:*

*A graph with green and pink squares

Description automatically generated*

*ReAdmis and Overweight:*

*A graph with green and blue bars

Description automatically generated*

*ReAdmis and Arthritis:*

*A graph of red and blue squares

Description automatically generated*

*ReAdmis and Services:*

*A graph of different colored bars

Description automatically generated*

*ReAdmis and Initial\_days:*

*A diagram of a patient's relationship

Description automatically generated*

## C4: Data Transformation

*See attached code:* d208task2complete.ipynb

To conform the data set to be better utilized by logistic regression, many aspects of the data was transformed. Outliers were replaced with median values in Income and VitD\_levels. Many variables were re-expressed as “1” and “0” instead of “Yes” and “No.” Dummy variables were then created for Initial\_admin and Services using get\_dummies from pandas. Finally, all unused variables for this project were dropped from the data set.

*Copy of code below:*

# Remove outliers

# Change booleans/categoricals into numbers

# Create dummy variables and drop regular columns

# Remove outliers (> {Mean + 2\*std})

df['Initial\_days'] = df.Initial\_days.round(1)

# Setting Income outliers to NAs

df['Income'] = np.where(df['Income'] > 97532.79, np.nan, df['Income'])

# Setting Income NaN outliers to Median value

df['Income'].fillna(df['Income'].median(), inplace=True)

# Setting values to 2 decimal places to better represent income

df['Income'] = df.Income.round(2)

# VitD\_levels outliers

df['VitD\_levels'] = np.where(df['VitD\_levels'] > 21.99, np.nan, df['VitD\_levels'])

df['VitD\_levels'].fillna(df['VitD\_levels'].median(), inplace=True)

# Reexpress booleans/categoricals into numeric values

df['ReAdmis'] = df['ReAdmis'].map({'Yes': '1', 'No': '0'})

df['HighBlood'] = df['HighBlood'].map({'Yes': '1', 'No': '0'})

df['Stroke'] = df['Stroke'].map({'Yes': '1', 'No': '0'})

df['Overweight'] = df['Overweight'].map({'Yes': '1', 'No': '0'})

df['Arthritis'] = df['Arthritis'].map({'Yes': '1', 'No': '0'})

df['Diabetes'] = df['Diabetes'].map({'Yes': '1', 'No': '0'})

# Create dummy variables and adding to dataframe

# Initial\_admin dummies

initial\_admin\_dummies = pd.get\_dummies(df.Initial\_admin, columns=['Elective Admission', 'Emergency Admission'])

df.insert(25, 'Initial\_admin\_elective\_admission', initial\_admin\_dummies['Elective Admission'])

df.insert(25, 'Initial\_admin\_emergency\_admission', initial\_admin\_dummies['Emergency Admission'])

# Services dummies

services\_dummies = pd.get\_dummies(df.Services, columns=['Blood Work', 'Intravenous', 'CT Scan'])

df.insert(41, 'Services\_blood\_work', services\_dummies['Blood Work'])

df.insert(41, 'Services\_intravenous', services\_dummies['Intravenous'])

df.insert(41, 'Services\_ct\_scan', services\_dummies['CT Scan'])

# Map dummies to numerical values

df['Initial\_admin\_elective\_admission'] = df['Initial\_admin\_elective\_admission'].astype(int)

df['Initial\_admin\_emergency\_admission'] = df['Initial\_admin\_emergency\_admission'].astype(int)

df['Services\_blood\_work'] = df['Services\_blood\_work'].astype(int)

df['Services\_intravenous'] = df['Services\_intravenous'].astype(int)

df['Services\_ct\_scan'] = df['Services\_ct\_scan'].astype(int)

# Drop unneeded columns

df = df.drop(['CaseOrder', 'Customer\_id', 'Interaction', 'UID', 'City', 'State', 'County', 'Zip', 'Lat', 'Lng', 'Population', 'Area', 'TimeZone', 'Job', 'Children', 'Marital', 'Gender', 'Full\_meals\_eaten', 'vitD\_supp', 'Soft\_drink', 'Initial\_admin', 'Complication\_risk', 'Hyperlipidemia', 'BackPain', 'Anxiety', 'Allergic\_rhinitis', 'Reflux\_esophagitis', 'Asthma', 'Services', 'TotalCharge', 'Additional\_charges', 'Item1', 'Item2', 'Item3', 'Item4', 'Item5', 'Item6', 'Item7', 'Item8'], axis=1)

## C5: Prepared Data

*See attached .csv:* cleanlogisticdata.csv

**Part IV – Model Comparison and Analysis**

## D1: Initial Model

*Screenshot of initial model:*

*A screenshot of a computer

Description automatically generated*

## D2: Justification of Model Reduction

In an attempt to create a superior model, I reduced the model down based on the assumptions of logistic regression mentioned earlier and via iterative alpha value comparison. Firstly, the data needs to be checked for multicollinearity by calculating variance inflation values to see if any were greater than 10 and would therefore require removal. None existed. Next, the variable with the highest p-value was removed, the model recalculated, and repeated until no p-values exceeded 0.05 as determined by our initial alpha value.

*VIFs:*

*A screenshot of a computer

Description automatically generated*

*1st Reduction – Age:*

*A screenshot of a computer

Description automatically generated*

*2nd Reduction – Doc\_visits:*

*A screenshot of a computer

Description automatically generated*

*3rd Reduction – Income:*

*A screenshot of a computer

Description automatically generated*

*4th Reduction – Overweight:*

*A screenshot of a computer

Description automatically generated*

*5th Reduction – VitD\_levels:*

*A screenshot of a computer

Description automatically generated*

*6th Reduction – Diabetes:*

*A screenshot of a computer

Description automatically generated*

*7th Reduction – Services\_ct\_scan*

*A screenshot of a computer

Description automatically generated*

## D3: Reduced Logistic Regression Model

*Screenshot of reduced model:*

A screenshot of a computer

Description automatically generated

## E1: Model Comparison

Pseudo R-squared and AIC (Akaike’s Information Criteria) will be used primarily for model comparison of the initial logistic regression model and the newly reduced one. The reduced model has a pseudo R-squared value of .9405 while the initial model was shown to have 0.9412. For pseudo R-squared, a higher value indicates a better model fit. AIC was also calculated and compared. The initial model’s AIC is 804.66 while the reduced is 800.6. For AIC, a lower value implies that the model is a better overall fit. Therefore, the initial model was shown to be slightly superior via the pseudo R-squared comparison while the reduced was the better fit if comparing using AIC. Out of curiosity, accuracy of the initial model was also juxtaposed with the reduced model’s as shown in E2. The initial model showed an accuracy of 98.31% while the reduced had 98.29% which is inferior.

## E2: Output and Calculations

*Confusion matrix for reduced model:*

*A close up of numbers

Description automatically generated*

*Accuracy calculation for reduced model:*

**

## E3: Code

*See attached code:* d208task2complete.ipynb

*Copy of code below:*

# Create model

df = pd.read\_csv("C:/Users/Owner/cleanlogisticdata.csv")

# Define independent variables to be used for initial model

X = df[['Age', 'Income', 'VitD\_levels', 'Doc\_visits', 'Initial\_admin\_emergency\_admission', 'Initial\_admin\_elective\_admission', 'HighBlood', 'Stroke', 'Overweight', 'Arthritis', 'Diabetes', 'Services\_blood\_work', 'Services\_intravenous', 'Services\_ct\_scan', 'Initial\_days']].assign(const=1)

# (*Logistic Regression Using Statsmodels*, 2020)

# INITIAL MODEL

y = df.ReAdmis.astype(int)

log\_reg = sm.Logit(y, X).fit()

print(log\_reg.summary())

# AIC for initial model

aic\_initial = log\_reg.aic

print("AIC: ", aic\_initial)

# Predictions

yhat = log\_reg.predict(X)

prediction = list(map(round, yhat))

# confusion matrix

cm = confusion\_matrix(y, prediction)

print("Confusion Matrix : \n", cm)

# Accuracy

acc = accuracy\_score(y, prediction)

print("Accuracy :", acc)

# VIFs

vif = pd.DataFrame()

vif['VIF'] = [variance\_inflation\_factor(X.values, i) for i in range(X.shape[1])]

vif['variable'] = X.columns

vif

D2: Justification of Model Reduction

# Reduction 1: Age

X = df[['Income', 'VitD\_levels', 'Doc\_visits', 'Initial\_admin\_emergency\_admission', 'Initial\_admin\_elective\_admission', 'HighBlood', 'Stroke', 'Overweight', 'Arthritis', 'Diabetes', 'Services\_blood\_work', 'Services\_intravenous', 'Services\_ct\_scan', 'Initial\_days']].assign(const=1)

y = df.ReAdmis

log\_reg = sm.Logit(y, X).fit()

print(log\_reg.summary())

# Reduction 2: Doc\_visits

X = df[['Income', 'VitD\_levels', 'Initial\_admin\_emergency\_admission', 'Initial\_admin\_elective\_admission', 'HighBlood', 'Stroke', 'Overweight', 'Arthritis', 'Diabetes', 'Services\_blood\_work', 'Services\_intravenous', 'Services\_ct\_scan', 'Initial\_days']].assign(const=1)

y = df.ReAdmis

log\_reg = sm.Logit(y, X).fit()

print(log\_reg.summary())

# Reduction 3: Income

X = df[['VitD\_levels', 'Initial\_admin\_emergency\_admission', 'Initial\_admin\_elective\_admission', 'HighBlood', 'Stroke', 'Overweight', 'Arthritis', 'Diabetes', 'Services\_blood\_work', 'Services\_intravenous', 'Services\_ct\_scan', 'Initial\_days']].assign(const=1)

y = df.ReAdmis

log\_reg = sm.Logit(y, X).fit()

print(log\_reg.summary())

# Reduction 4: Overweight

X = df[['VitD\_levels', 'Initial\_admin\_emergency\_admission', 'Initial\_admin\_elective\_admission', 'HighBlood', 'Stroke', 'Arthritis', 'Diabetes', 'Services\_blood\_work', 'Services\_intravenous', 'Services\_ct\_scan', 'Initial\_days']].assign(const=1)

y = df.ReAdmis

log\_reg = sm.Logit(y, X).fit()

print(log\_reg.summary())

# Reduction 5: VitD\_levels

X = df[['Initial\_admin\_emergency\_admission', 'Initial\_admin\_elective\_admission', 'HighBlood', 'Stroke', 'Arthritis', 'Diabetes', 'Services\_blood\_work', 'Services\_intravenous', 'Services\_ct\_scan', 'Initial\_days']].assign(const=1)

y = df.ReAdmis

log\_reg = sm.Logit(y, X).fit()

print(log\_reg.summary())

# Reduction 6: Diabetes

X = df[['Initial\_admin\_emergency\_admission', 'Initial\_admin\_elective\_admission', 'HighBlood', 'Stroke', 'Arthritis', 'Services\_blood\_work', 'Services\_intravenous', 'Services\_ct\_scan', 'Initial\_days']].assign(const=1)

y = df.ReAdmis

log\_reg = sm.Logit(y, X).fit()

print(log\_reg.summary())

# Reduction 7: Services\_ct\_scan

X = df[['Initial\_admin\_emergency\_admission', 'Initial\_admin\_elective\_admission', 'HighBlood', 'Stroke', 'Arthritis', 'Services\_blood\_work', 'Services\_intravenous', 'Initial\_days']].assign(const=1)

y = df.ReAdmis

log\_reg = sm.Logit(y, X).fit()

print(log\_reg.summary())

D3: Reduced Logistic Regression Model

# Reduced model

X = df[['Initial\_admin\_emergency\_admission', 'Initial\_admin\_elective\_admission', 'HighBlood', 'Stroke', 'Arthritis', 'Services\_blood\_work', 'Services\_intravenous', 'Initial\_days']].assign(const=1)

y = df.ReAdmis

log\_reg\_reduced = sm.Logit(y, X).fit()

print(log\_reg\_reduced.summary())

E1: Model Comparison

# AIC for reduced model

aic\_reduced = log\_reg\_reduced.aic

print("AIC: ", aic\_reduced)

E2: Logistic Regression

# Confusion matrix and accuracy

# Predictions

yhat = log\_reg\_reduced.predict(X)

prediction = list(map(round, yhat))

# Confusion matrix

cm = confusion\_matrix(y, prediction)

print("Confusion Matrix : \n", cm)

# Accuracy

acc = accuracy\_score(y, prediction)

print("Accuracy :", acc\*100, "%")

**Part V – Data Summary and Implications**

## F1: Results

*Regression equation:*

ReAdmis = -65.9722 + 1.4179\*Initial\_admin\_emergency\_admission + 1.3742\*Stroke + 1.2272\*Initial\_days + 0.716\*HighBlood – 0.6907\*Initial\_admin\_elective\_admission – 1.0426\*Arthritis – 1.6486\*Services\_blood\_work – 1.6381\*Services\_intravenous

ReAdmis, the dependent variable, refers to the log odds of readmission. As this is a logistic regression equation, the left side of the equation could alternately be expressed as ln(p/(1-p)) where p = probability that a patient is readmitted (ReAdmis). As explained below, this is a natural logarithm (represented by ln) of the odds of readmission ( p / (1-p) ).

*Interpretation of coefficients of reduced model:*

In order to understand the table of interpretation of coefficients below, note that it refers to only that single, specific variable mentioned being manipulated at a time and all other variables being held or set at 0. Each instance of increase or decrease of coefficients affects the log odds of ReAdmis, the dependent variable. Log odds are quite literally the natural logarithm of the odds of readmission. Odds of readmission refers to the chances of readmission divided by the chances of not being readmitted. (GeeksforGeeks, 2021). In the case of all independent variables being set to 0, the log odds of readmission (ReAdmis) would be -65.9722, the y-intercept, for instance.

The log odds can be converted to percentage change in odds, as well. Using the formula (odds – 1)\*100, where odds is e^(coefficient), a 1 value increase in Initial\_days would be ((e^(1.2272)-1)\*100) or a 241% increase in the odds of being readmitted (Course Material, n.d.). For a negative correlated variable, if a patient received an intravenous service (Services\_intravenous = 1), then the formula shows an 80.6% decrease in the odds of being readmitted (ReAdmis).

Although log odds is the term being used, it can also, and perhaps more accurately, be stated as natural log odds.

|  |  |
| --- | --- |
| **Coefficient** | Interpretation and Effect |
| -65.9722 | y-intercept |
| 1.4719 | Value of Initial\_admin\_emergency\_admissions can only be 1 or 0. So, 1 increases the log odds of readmission by 1.4719 while 0 does nothing. |
| 1.3742 | Value of Stroke can only be 1 or 0. A value of 1 increases log odds of ReAdmis by 1.3742 while 0 does nothing. |
| 1.2272 | Initial\_days is positive and continuous. A 1 point increase leads to an increase of 1.2272 for log odds of ReAdmis. |
| 0.716 | Value of HighBlood can only be 1 or 0. If value is 1, then log odds of ReAdmis are increased by 0.716 while a 0 does nothing. |
| -0.6907 | Initial\_admin\_elective\_admission can only be 1 or 0. It value is 1, then it decreases the log odds for ReAdmis by 0.6907. A value of 0 does nothing. |
| -1.0426 | Arthritis can only be 1 or 0. If it has a value of 1, then ReAdmis log odds are decreased by 1.0426. A value of 0 does nothing. |
| -1.6486 | Services\_blood\_work can only have values of 1 or 0. If it has a value of 1, then ReAdmis log odds are decreased by 1.6486. If the value is 0, then it does nothing. |
| -1.6381 | Services\_intravenous can only have values of 1 or 0. If it has a value of 1, then log odds of ReAdmis are decreased by 1.6381 while a 0 does nothing. |

*Statistical and practical significance of the reduced model:*

Statistically, there are some ways of determining significance of the reduced model. No p-value exceeds our original alpha value (0.05). The LLR p-value is also shown as 0.000 and further indicates statistical significance as it is below 0.05.

As there is statistical significance of the reduced model, there might be some practical significance of the data, in addition. The reduced model shows that the greatest potential factor when estimating chance of readmission is the length of the initial hospital stay as the variable Initial\_days is the only variable that is continuous as opposed to binary. Initial\_admin\_emergency\_admission, Stroke, and HighBlood all increased the likelihood of readmission which is also useful information as future predictors in a real-world setting. Initial\_admin\_elective\_admission, Arthritis, Services\_blood\_work, and Services\_intravenous all had an inverse relationship with readmission to varying degrees which can be further analyzed.

*The limitations of the data analysis:*

There were limitations considered over the course of this project. A larger data set would have allowed for the creation of a better model. More variables could also be included in the analysis in order to attempt to create a better fitting model. Furthermore, additional variables could have been collected with the rest of the data. It is possible that there was a better method of reduction than the backwards step elimination utilized here. As always, just because there is a correlation, it does not mean that there is a direct causation or even a worthwhile corollary.

## F2: Recommendations

The initial research question asked which factors contribute to patients being readmitted to the hospital. According to the reduced model, there is now a list of statistically significant factors and a statistically significant model worthy of further investigation. I would recommend much greater analysis into the causes of Initial\_days as, according to the reduced model, it was shown to be the largest manipulable variable that contributed to likelihood of readmission. Other binary factors shown to be correlated are worth looking into whether a checklist should perhaps be created for the purposes of attempting to approximate patient flow over time. This would be beneficial for human resource allocation and bed availability, for instance, as a way to increase efficiency and potentially minimize costs.

**Part VI - Demonstration**

## G: Panopto Demonstration

*See Panopto Link: https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=dcad246d-ad1e-4770-a8e5-b17100e22140*

## H: Sources of Third-Party Code

*Logistic Regression using Statsmodels*. (2020, July 17). GeeksforGeeks. https://www.geeksforgeeks.org/logistic-regression-using-statsmodels/

## I: Sources

Course Materials

GeeksforGeeks. (2021, July 18). *Role of Log Odds in Logistic Regression*. (2021, July 18). GeeksforGeeks. https://www.geeksforgeeks.org/role-of-log-odds-in-logistic-regression

Western Governors University. (n.d.). *R or Python*. Western Governors University. https://www.wgu.edu/online-it-degrees/programming-languages/r-or-python.html

Zach. (2020, October 13). *The 6 Assumptions of Logistic Regression (With Examples)*. Statology. https://www.statology.org/assumptions-of-logistic-regression/