D208: Predictive Modeling

Task I

Shawn Wheeler

000364855

Table of Contents

[A1: Research Question 3](#_Toc165408460)

[A2: Goals 3](#_Toc165408461)

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

[B2: Tool Benefits 4](#_Toc165408463)

[B3: Appropriate Technique 4](#_Toc165408464)

[C1: Data Cleaning 4](#_Toc165408465)

[C2: Summary Statistics 5](#_Toc165408466)

[C3: Visualizations 10](#_Toc165408467)

[C4: Data Transformation 24](#_Toc165408468)

[C5: Prepared Data 26](#_Toc165408469)

[D1: Initial Model 27](#_Toc165408470)

[D2: Justification of Model Reduction 27](#_Toc165408471)

[D3: Reduced Linear Regression Model 35](#_Toc165408472)

[E1: Model Comparison 35](#_Toc165408473)

[E2: Output and Calculations 36](#_Toc165408474)

[E3: Code 37](#_Toc165408475)

[F1: Results 41](#_Toc165408476)

[F1: Recommendations 42](#_Toc165408477)

[G: Panopto Demonstration 42](#_Toc165408478)

[H: Sources of Third-Party Code 42](#_Toc165408479)

[I: Sources 42](#_Toc165408480)

**Part I – Research Question**

## A1: Research Question

The research question for this project is “which variables affect Initial\_days?” Initial\_days is continuous and will be the target variable for analysis of the medical\_data data set. The independent variables that will be considered for their effect on Initial\_days are Age, Income, VitD\_levels, Doc\_visits, Full\_meals\_eaten, Soft\_drink, Initial\_admin, HighBlood, Stroke, Overweight, Arthritis, Diabetes, Asthma, and Services.

## A2: Goals

The goal of performing this analysis is to determine the extent of correlation between Initial\_days and the following independent variables: Age, Income, VitD\_levels, Doc\_visits, Full\_meals\_eaten, Soft\_drink, Initial\_admin, HighBlood, Stroke, Overweight, Arthritis, Diabetes, Asthma, and Services. Using the larger number of variables in the analysis should be beneficial in determining which variables are statistically significant to the length of days of an initial hospital stay for a patient in the data set and which are not. Overall, the goal would be to determine if there are any methods to consider that would either decrease or increase the length of the initial hospital stay of a patient.

**Part II – Method Justification**

## B1: Summary of Assumptions

In order to seek an answer for the research question, linear regression and multiple linear regression will be will be implemented. In order for the results of these methods to be useful, however, four assumptions are required to be met. These assumptions are: the dependent and each independent variable have a linear relationship, there is low correlation between the independent variables themselves, that all observations in the data set are independent and random from the population, and that residuals should be normally distributed while the mean equals zero instead of having constant variance (also known as heteroscedasticity) (Course materials, 2024).

The dependent and independent variables listed before must have a proportional relationship in order to pass the bar of the first assumption. For the second assumption, multicollinearity, which is the term for when there is a high correlation between independent variables and is tested for by calculating variance inflation factors, can not exist. Thirdly, observations are independent when the “residuals exhibit autocorrelation” (Zach, 2021). Fourthly, the residuals should not follow a pattern and they should have a mean of zero.

## B2: Tool Benefits

Python was chosen as the tool to attempt to successfully tackle the research question. This is due to its readability, computational speed, consistent syntax, and due to its fantastic ecosystem of packages and libraries allowing for exactly what is needed to perform our multiple linear regression analysis. (Western Governor’s University, n.d.). Below is a table of the libraries and packages that were used in the process of attempting to answer the research question via multiple linear regression along with their function.

*Table:*

|  |  |
| --- | --- |
| Libraries and Packages | Usage |
| pandas | Overall data manipulation / data wrangling |
| numpy | Mathematics operations |
| seaborn | Creating visualizations |
| matplotlib.pyplot | Creating visualizations |
| statsmodels.api | Visualization and ordinary least squares (OLS) |
| statsmodels.formula.api | Calculating OLS |
| statsmodels.stats.outliers\_influence | Calculating VIF (variance inflation factor) |
| sklearn.linear\_model | Linear regression |
| warnings | Ensure warnings do not appear during videos |

## B3: Appropriate Technique

With the intention of investigating the relationship between many independent variables and a singular target, dependent variable, multiple linear regression was selected as the appropriate technique for this project. Since I want to potentially determine specifically which variables have the greatest levels of association, this methodology will allow for in-depth examination of one-to-one relationships with the target variable as well as goodness of fit of the overall model. After refining the model, an equation will be expressed using data through this analysis technique along with other figures which should provide an answer as to which variables affect Initial\_days, if any.

Multiple linear regression requires a continuous dependent variable along with its multiple independent variables for analysis as opposed to another technique like logistic regression which can not be accomplished with a continuous dependent variable. A continuous variable is one which has a numerical value that could theoretically be infinite. Since our target variable is Initial\_days, a measure of the days a patient stayed in the hospital in their initial visit, which is by nature a continuous variable, it stands to reason to use multiple linear regression in an attempt to answer the initial research question.

**Part III – Data Preparation**

## C1: Data Cleaning

*See attached code:* d208complete.ipynb

The given data set has already been cleaned to an extent removing the need for many typical data cleaning tasks. Therefore, the goals of data cleaning for this project are to check for outliers, missing data, and duplicates, as well as examining variables for further needed tasks. To achieve these ends, I used duplicated() to check for duplicates, isnull() to determine if there were any missing data, and a series of boxplots to identify potential outliers. The data is being examined in this way to ensure that it is in the most usable form for answering the research question. Further below in C2, I will examine the individual variables to see if they require univariate cleaning or other alterations including re-expression of variables.

*Copy of code below:*

# Check for outliers

# Check for missing data

# Check for duplicates

# 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

Below I will be attaching screenshots of each independent variable and the dependent variables as well as an explanatory summary for what the visualizations represent. Variables not represented in this section are not going to be used in the answering of the original research question. Initial\_days is the dependent variables, while all others are independent for the purposes of this study.

It is important to understand the definitions of median, mean, interquartile ranges, count, standard deviation, minimum, and maximum. Median is the middle number in a set of numbers. In this case, it will be represented by the 50% interquartile from describe() as shown below. The interquartile ranges are represented as 25%, 50%, and 75% and each represents the range values above or below them. For instance, the 50% interquartile, which is also the median, means that 50% of the values are higher and 50% of the values are lower.

Mean represents the average of all of the values in a set of numbers. If you took all values, added them together, and then divided them by the number of values, you would arrive at the mean. The standard deviation, represented by std, is used with the mean to determine outliers. Finally, the min and max represent the lowest and highest values in the data set respectively, and count refers to the amount of values in the dataset. All of these should have 10,000.

The non-quantitative variables below are represented by using value\_counts() to describe the data, and I will include an additional brief explanation for each.

*Initial\_days:*

A screenshot of a computer

Description automatically generated

The dependent variable, Initial\_days, shows 10,000 values with an average of around 34.5 days for the initial hospital stay of a patient. The longest length was nearly 72 days.

*Age:*

A screenshot of a computer screen

Description automatically generated

Age reveals that the average age of patients are 53 out of the 10,000 values in the data set.

*Income:*

A screenshot of a computer

Description automatically generated

There was an average income of $40,490 for patients with 50% of patients falling below $33,768 per year and 25% above $54,296. This can be explained by the maximum value of $207,249 indicating a possibility of outliers using the median added to two standard deviations which is any value above around $90,000.

*VitD\_levels:*

A screenshot of a computer

Description automatically generated

Vitamin D levels are recorded on intake. This shows that the average levels for patients were 17.96 with a low of 9.8 and a high of 26.39.

*Doc\_visits:*

A screenshot of a computer code

Description automatically generated

Doctor visits for patients had an average of 5 per patient. However, looking at the values, there were also 25% that had less than 3.

*Full\_meals\_eaten:*

A screenshot of a computer code

Description automatically generated

Full meals eaten had an average of 1 per patient and a maximum of 7 meals. Only full meals are included in this data.

*Soft\_drink:*

A close up of numbers

Description automatically generated

Soft\_drink is a categorical variable that represents whether an individual consumes greater than three such beverages a day. Using value\_counts(), we can see that 7,425 do not consume more than three soft drinks per day, while the remaining 2,575 do so.

*Initial\_admin:*

A close up of a screen

Description automatically generated

Initial\_admin shows the reasons for admission or the initial hospital stay for a patient. This data shows that emergency admissions accounted for 5,060, elective admissions for 2,504, and lastly 2,436 admitted for observation adding to up to our 10,000 count.

*HighBlood:*

A close up of a number

Description automatically generated

HighBlood shows whether patients have a history of blood pressure or not. 5,910 of the 10,000 did not, whereas 4,090 did have high blood pressure.

*Stroke:*

A close-up of numbers

Description automatically generated

For patients with a history of stroke or not, the data shows that only 1,993 of the 10,000 do have a history of stroke while the remainder, 8,007, did not.

*Overweight:*

A close-up of a number

Description automatically generated

7,094 patients out of our 10,000 data set were shown to be overweight while 2,906 were not.

*Arthritis:*

A close up of numbers

Description automatically generated

There were 3,574 patients diagnosed with arthritis versus 6,426 that were not.

*Diabetes:*

A close up of numbers

Description automatically generated

The diabetes independent variable shows that 2,738 patients had a diagnosis of diabetes while 7,262 did not.

*Asthma:*

A close-up of numbers

Description automatically generated

Asthma was shown in 2,893 patients out of the 10,000 total leaving 7,107 without.

*Services:*

A white background with black text

Description automatically generated

Lastly, Services shows us that out of the 10,000 patients, 5,265 required blood work, 3,130 required intravenous service, 1,225 were given a CT Scan, and 280 needed an MRI.

## C3: Visualizations

Univariate Analyses

*Initial\_days:*

*A graph of a number of patients

Description automatically generated*

*Age:*

*A chart with a red line

Description automatically generated*

*Income:*

*A graph showing a distribution of income

Description automatically generated*

*VitD\_levels:*

*A chart of vitamin d levels

Description automatically generated*

*Doc\_visits:*

*A green graph with black lines

Description automatically generated*

*Full\_meals\_eaten:*

*A graph of a distribution of meals

Description automatically generated*

*Initial\_admin:*

*A blue line graph with black text

Description automatically generated*

*Initial\_admin:*

*A pie chart with numbers and text

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 numbers and text

Description automatically generated*

*Arthritis:*

*A graph with pink squares

Description automatically generated*

*Diabetes:*

*A purple and green pie chart

Description automatically generated*

*Asthma:*

*A graph of a distribution of asthma among patients

Description automatically generated*

*Services:*

*A bar graph with numbers and a number of numbers

Description automatically generated with medium confidence*

Bivariate Analyses

*Age and Initial\_days:*

*A graph of blue dots

Description automatically generated*

*Income and Initial\_days:*

*A graph of blue dots

Description automatically generated*

*VitD\_levels and Initial\_days:*

*A chart of blue dots

Description automatically generated*

*Doc\_visits and Initial\_days:*

*A graph of a number of blue dots

Description automatically generated*

*Full\_meals\_eaten and Initial\_days:*

*A graph of a number of food items

Description automatically generated with medium confidence*

*Soft\_drink and Initial\_days:*

*A chart of a soft drink

Description automatically generated*

*Initial\_admin and Initial\_days:*

*A diagram of a relationship between initial admission and initial admission

Description automatically generated*

*HighBlood and Initial\_days:*

*A diagram of a couple of blue squares

Description automatically generated*

*Stroke and Initial\_days:*

*A diagram of a relationship between stroke history and initial data

Description automatically generated*

*Overweight and Initial\_days:*

*A diagram of a relationship between overweight and initial days

Description automatically generated*

*Arthritis and Initial\_days:*

*A diagram of a person's relationship

Description automatically generated*

*Diabetes and Initial\_days:*

*A diagram of a couple of pink squares

Description automatically generated*

*Asthma and Initial\_days:*

*A diagram of a couple of squares

Description automatically generated*

*Services and Initial\_days:*

*A chart with green rectangular shapes

Description automatically generated with medium confidence*

## C4: Data Transformation

*See attached code:* d208complete.ipynb

Based on the information gained in the previous three sections, some data was transformed in order to allow the data to be used for analysis. Outliers were removed, values were re-expressed into numerical values as it is needed for our usage, dummy variables were created in order to combat multicollinearity, and all unneeded columns were dropped from our data set.

Outliers were detected in Income, VitD\_levels, and Full\_meals\_eaten. In each instance, the outliers were replaced with the median value. Furthermore, Income was truncated for easier readability. Next, all Boolean values represented as yes/no were replaced with 1 and 0 respectively. Dummy variables were then separated out and created for the Initial\_admin and Services variables with the columns for Observation Admission and MRIs being removed to allay multicollinearity. These dummy variables were then converted to numerical values, also. Finally, all columns unrelated to the analysis 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})

# 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)

# Full\_meals\_eaten outliers

df['Full\_meals\_eaten'] = np.where(df['Full\_meals\_eaten'] > 3, np.nan, df['Full\_meals\_eaten'])

df['Full\_meals\_eaten'].fillna(df['Full\_meals\_eaten'].median(), inplace=True)

# Reexpress booleans/categoricals into numeric values

df['Soft\_drink'] = df['Soft\_drink'].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'})

df['Asthma'] = df['Asthma'].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', 'ReAdmis', 'vitD\_supp', 'Initial\_admin', 'Complication\_risk', 'Hyperlipidemia', 'BackPain', 'Anxiety', 'Allergic\_rhinitis', 'Reflux\_esophagitis', 'Services', 'TotalCharge', 'Additional\_charges', 'Item1', 'Item2', 'Item3', 'Item4', 'Item5', 'Item6', 'Item7', 'Item8'], axis=1)

## C5: Prepared Data

*See attached .csv:* cleanmlrdata.csv

**Part IV – Model Comparison and Analysis**

## D1: Initial Model

A screenshot of a computer

Description automatically generated

## D2: Justification of Model Reduction

In order to reduce the model to enhance accuracy and provide better insight into answering the initial research question, backward stepwise elimination where variables with a p-value (represented as P>|t|) greater than our alpha level of 0.05 are removed from the model one at a time after which the model is reevaluated until no more variables fitting that criteria remain.

This method does not account for consideration of multicollinearity; therefore, I also calculated and considered the variance inflation factors (VIFs). Any VIF greater than 5 will be removed from this model.

*VIFs:*

A screenshot of a computer

Description automatically generated

*1st Reduction (VIF) (const and Services\_blood\_work):*

*A screenshot of a computer

Description automatically generated*

*2nd Reduction (VIF) (Services\_intravenous):*

A screenshot of a computer

Description automatically generated

*3rd Reduction (HighBlood):*

*A screenshot of a computer

Description automatically generated*

*4th Reduction (Stroke):*

*A screenshot of a computer

Description automatically generated*

*5th Reduction (Overweight):*

*A screenshot of a computer

Description automatically generated*

*6th Reduction (Full\_meals\_eaten):*

*A screenshot of a computer

Description automatically generated*

*7th Reduction (Diabetes):*

*A screenshot of a computer

Description automatically generated*

*8th Reduction (Asthma):*

*A screenshot of a document

Description automatically generated*

*9th Reduction (Soft\_drink):*

*A screenshot of a computer

Description automatically generated*

*10th Reduction (Initial\_admin\_emergency\_admission):*

*A screenshot of a computer

Description automatically generated*

*11th Reduction (Services\_ct\_scan):*

*A screenshot of a computer

Description automatically generated*

*12th Reduction (Income):*

*A screenshot of a computer

Description automatically generated*

*13th Reduction (Initial\_admin\_elective\_admission)*

*A screenshot of a computer

Description automatically generated*

## D3: Reduced Linear Regression Model

*Screenshot of reduced model:*

A screenshot of a computer

Description automatically generated

## E1: Model Comparison

For the purposes of comparing the initial model with the reduced model, a few factors will be considered from both, such as, adjusted R-squared, F-statistic, Prob, Residual Standard Error, AIC (Akaike’s Information Criteria), and BIC (Bayesian Information Criteria). Adjusted R-squared increases as the variables better explain the regression. Initially, the R-squared was 0.000 whereas the reduced model shows an adjusted R-squared of 0.626. A higher F-statistic also shows better association between a group of variables. The original model had an F-statistic of 1.071 while the reduced model has 4,192. The closer Prob is to 0, the more meaningful the regression is. Initially, that value was 0.377 whereas the reduced one has a Prob value of 0.

Residual standard error for the initial model was 26.31 while the reduced model shows a 26.5 which is a minimal increase but still not a better fit. The model with a lower AIC is a better fit. Initially, the model showed a 9.379e+04 whereas the reduced model shows a 9.392e+04 which is lower and therefore a better fit albeit very slightly. BIC also follows the rule where the lower value is superior. In this case, the initial BIC value was 9.392e+04 with the reduced model showing 9.395e+04 which is not indicative of a greater model. Overall, neither model seems to be excellent, but the reduced one mostly wins out in comparison to the initial.

## E2: Output and Calculations

*Residual plot for reduced model:*

*A blue dotted diagram with numbers and a line

Description automatically generated with medium confidence*

*Q-Q plot for reduced model:*

*A graph with a line and a red line

Description automatically generated*

*Residual Standard Error for the reduced model using mse\_resid:*



## E3: Code

*See attached code:* d208complete.ipynb

# Create model

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

# Multiple Linear Regression (Course Materials, 2024)

# Independent variables dataframe w/ added constant

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

# Dependent variable

y = df.Initial\_days

# OLS regression model for MLR

mod = smf.ols((y,X), data=df)

res = mod.fit()

print(res.summary())

# Grabbing original RSE for E2

rse = np.sqrt(res.mse\_resid)

print(rse)

# (Zach, 2020)

# Higher than 5 or so should be removed

vif = pd.DataFrame()

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

vif['variable'] = X.columns

vif

# Use p-values and VIFs and remove one at a time

# Remove const

# Remove Services\_blood\_work

# Remove Services\_intravenous

# Remove all except arthritis

# 1st reduction

# Remove const and Services\_blood\_work

X = df[['Age', 'Income', 'VitD\_levels', 'Doc\_visits', 'Full\_meals\_eaten', 'Soft\_drink', 'Initial\_admin\_emergency\_admission', 'Initial\_admin\_elective\_admission', 'HighBlood', 'Stroke', 'Overweight', 'Arthritis', 'Diabetes', 'Asthma', 'Services\_intravenous', 'Services\_ct\_scan']]

y = df.Initial\_days

mod = smf.ols((y,X), data=df)

res = mod.fit()

print(res.summary())

# 2nd reduction Services\_intravenous

X = df[['Age', 'Income', 'VitD\_levels', 'Doc\_visits', 'Full\_meals\_eaten', 'Soft\_drink', 'Initial\_admin\_emergency\_admission', 'Initial\_admin\_elective\_admission', 'HighBlood', 'Stroke', 'Overweight', 'Arthritis', 'Diabetes', 'Asthma', 'Services\_ct\_scan']]

y = df.Initial\_days

mod = smf.ols((y,X), data=df)

res = mod.fit()

print(res.summary())

# END removed for VIF, now removed for p-values

# HighBlood removed

X = df[['Age', 'Income', 'VitD\_levels', 'Doc\_visits', 'Full\_meals\_eaten', 'Soft\_drink', 'Initial\_admin\_emergency\_admission', 'Initial\_admin\_elective\_admission', 'Stroke', 'Overweight', 'Arthritis', 'Diabetes', 'Asthma', 'Services\_ct\_scan']]

y = df.Initial\_days

mod = smf.ols((y,X), data=df)

res = mod.fit()

print(res.summary())

# 4th Reduction Stroke removed

X = df[['Age', 'Income', 'VitD\_levels', 'Doc\_visits', 'Full\_meals\_eaten', 'Soft\_drink', 'Initial\_admin\_emergency\_admission', 'Initial\_admin\_elective\_admission', 'Overweight', 'Arthritis', 'Diabetes', 'Asthma', 'Services\_ct\_scan']]

y = df.Initial\_days

mod = smf.ols((y,X), data=df)

res = mod.fit()

print(res.summary())

# 5th Reduction Overweight removed

X = df[['Age', 'Income', 'VitD\_levels', 'Doc\_visits', 'Full\_meals\_eaten', 'Soft\_drink', 'Initial\_admin\_emergency\_admission', 'Initial\_admin\_elective\_admission', 'Arthritis', 'Diabetes', 'Asthma', 'Services\_ct\_scan']]

y = df.Initial\_days

mod = smf.ols((y,X), data=df)

res = mod.fit()

print(res.summary())

# 6th Reduction Full\_meals\_eaten removed

X = df[['Age', 'Income', 'VitD\_levels', 'Doc\_visits', 'Soft\_drink', 'Initial\_admin\_emergency\_admission', 'Initial\_admin\_elective\_admission', 'Arthritis', 'Diabetes', 'Asthma', 'Services\_ct\_scan']]

y = df.Initial\_days

mod = smf.ols((y,X), data=df)

res = mod.fit()

print(res.summary())

# 7th Reduction Diabetes removed

X = df[['Age', 'Income', 'VitD\_levels', 'Doc\_visits', 'Soft\_drink', 'Initial\_admin\_emergency\_admission', 'Initial\_admin\_elective\_admission', 'Arthritis', 'Asthma', 'Services\_ct\_scan']]

y = df.Initial\_days

mod = smf.ols((y,X), data=df)

res = mod.fit()

print(res.summary())

# 8th Reduction Asthma removed

X = df[['Age', 'Income', 'VitD\_levels', 'Doc\_visits', 'Soft\_drink', 'Initial\_admin\_emergency\_admission', 'Initial\_admin\_elective\_admission', 'Arthritis', 'Services\_ct\_scan']]

y = df.Initial\_days

mod = smf.ols((y,X), data=df)

res = mod.fit()

print(res.summary())

# 9th Reduction Soft\_drink removed

X = df[['Age', 'Income', 'VitD\_levels', 'Doc\_visits', 'Initial\_admin\_emergency\_admission', 'Initial\_admin\_elective\_admission', 'Arthritis', 'Services\_ct\_scan']]

y = df.Initial\_days

mod = smf.ols((y,X), data=df)

res = mod.fit()

print(res.summary())

# 10th Reduction Initial\_admin\_emergency\_admission removed

X = df[['Age', 'Income', 'VitD\_levels', 'Doc\_visits', 'Initial\_admin\_elective\_admission', 'Arthritis', 'Services\_ct\_scan']]

y = df.Initial\_days

mod = smf.ols((y,X), data=df)

res = mod.fit()

print(res.summary())

# 11th Reduction Services\_ct\_scan removed

X = df[['Age', 'Income', 'VitD\_levels', 'Doc\_visits', 'Initial\_admin\_elective\_admission', 'Arthritis']]

y = df.Initial\_days

mod = smf.ols((y,X), data=df)

res = mod.fit()

print(res.summary())

# 12th Reduction Income removed

X = df[['Age', 'VitD\_levels', 'Doc\_visits', 'Initial\_admin\_elective\_admission', 'Arthritis']]

y = df.Initial\_days

mod = smf.ols((y,X), data=df)

res = mod.fit()

print(res.summary())

# 13th Reduction Initial\_admin\_elective\_admission removed

X = df[['Age', 'VitD\_levels', 'Doc\_visits', 'Arthritis']]

y = df.Initial\_days

mod = smf.ols((y,X), data=df)

res = mod.fit()

print(res.summary())

X = df[['Age', 'VitD\_levels', 'Doc\_visits', 'Arthritis']]

y = df.Initial\_days

mod = smf.ols((y,X), data=df)

res = mod.fit()

print(res.summary())

# Residual plot

# Residual standard error

# Checking RSE for new model

# (Anonymous, n.d.)

rse = np.sqrt(res.mse\_resid)

print(rse)

# Create a residual plot (aka density plot) AND a Q-Q plot (normal probability plot)

# Singular Residual Plot (Course Materials, n.d.)

# Define Predictor variables

X = df[['Age', 'VitD\_levels', 'Doc\_visits', 'Arthritis']]

y = df['Initial\_days']

# Fit model

mod = sm.OLS(y, sm.add\_constant(X)).fit()

# Create theoretical values

y\_theor = mod.predict(sm.add\_constant(X))

# Subtract theoretical values from actual values

residuals = y - y\_theor

# Create the residual plot

plt.scatter(y\_theor, residuals, c="cyan")

plt.title("Residual Plot for Reduced Model")

plt.ylabel("Residuals")

plt.xlabel("Theoretical Values")

plt.axhline(y=0, color='blue', linestyle='--')

plt.show()

# Q-Q Plot

sm.qqplot(residuals, line='45', fit=True)

plt.xlabel("Theoretical Values")

plt.ylabel("Residuals")

plt.show()

**Part V – Data Summary and Implications**

## F1: Results

*Regression equation:*

Initial\_days = 43188.325 + 1.4734\*Arthritis + 1.4024\*VitD\_levels + 1.1318\*Doc\_visits + 0.055\*Age

*Interpretation of coefficients of reduced model:*

I will include a table of the coefficients below and their interpretations based on the regression equation created. For each of these, the description refers to only when that specific variable is being manipulated and all other variables are held, as is.

|  |  |
| --- | --- |
| **Coefficient** | **Interpretation (all else held at 0)** |
| 43188.325 | y-intercept (where the line crosses the x-axis) |
| 1.4734 | Arthritis is a binary variable with values of only 1 or 0. If the value is 1, then it increases Initial\_days by 1.4734 |
| 1.4024 | VitD\_levels is a continuous variable and can therefore have the largest effect. An increase of the value by 1 leads to an increase of 1.4024 for Initial\_days |
| 1.1318 | Doc\_visits has values from 1 – 9 in our data set. An increase of 1 leads to an increase of 1.1318 for Initial\_days |
| 0.055 | Age is a numerical value between 18-89 in this data set. An increase of 1 in this equation leads to a 0.055 increase in Initial\_days |

*Statistical and practical significance of the reduced model:*

Although the reduced model did not have a better BIC value than the original, it did surpass the initial model in all other statistical considerations. According to Prob, with a value (0.00) of less than 0.05, the reduced model is significant. Furthermore, as described previously in E1, the AIC, F-statistic, and adjusted R-squared values also indicate that the reduced model is a better fit than the original mathematically.

In terms of practical significance, the reduced model shows that there is some association between the variables remaining. Those variables that remained are Arthritis, VitD\_levels, Doc\_visits, and Age with regards to Initial\_days as shown in the regression equation. Practically, it states that there is an association and that there are differing magnitudes of those associations. Based on that, it would be more sensible to examine variables with a higher coefficient to determine their practicality and how to use that information to make business decisions.

*The limitations of the data analysis:*

One of the biggest limitations that I see with this data analysis is that the data set is small and there are a quite limited number of variables. More data would allow for more useful models to be created. Another limitation is that the backward stepwise elimination method may not be the best method for reducing the model which would be a knowledge issue. Finally, not all variables were tested leading to a poorer model overall than if all had been considered. This limited information makes it quite difficult to determine whether causation exists in any form or not.

## F2: Recommendations

There has been shown to be a statistically significant association in the reduced model. Based on the reduced model, arthritis, vitamin D levels, doctor visits, and age are worth consideration for further analysis. Arthritis, having the highest coefficient, would be prudent to investigate further; however, what use that information may have is questionable at best. It may be useful in assessing the likelihood of a lengthier stay for a patient initially, but other use cases remain elusive. Continuing, initial vitamin D levels, although having a high coefficient, suffers similar issues of usability. Investigating further, however, in this case may lead to determining a treatment for low, or less ideal, levels of vitamin D in order to reduce the initial stay of a patient. The number of doctor visits should be explored once more data can be accumulated to see if the visits are themselves beneficial or merely a function of time. If more visits early in a hospital stay lead to less days in the initial stay, that could be financially prudent to investigate. Therefore, based on the results and applicable values of the reduced model and equation, I would recommend more in depth data collection and subsequent investigation into these variables to see if the association is of tangible value or not.

**Part VI - Demonstration**

## G: Panopto Demonstration

*See Panopto Link:*

*https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=2d0d555d-66ad-436e-9b53-b16f010a04b3*

## H: Sources of Third-Party Code

Anonymous. (n.d.). *Find RSME and Standard Deviation of a StatsModels Multiple Regression*. (n.d.). Stack Overflow. https://stackoverflow.com/questions/68532863/find-rsme-and-standard-deviation-of-a-statsmodels-multiple-regression

Course Materials

Zach. (2020, July 20). *How to Calculate VIF in Python*. Statology. https://www.statology.org/how-to-calculate-vif-in-python/

## I: Sources

Course Materials

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. (2021, November 16). *The Five Assumptions of Multiple Linear Regression*. Statology. https://www.statology.org/multiple-linear-regression-assumptions/