**A1. Research question:**

Continuing from the previous analysis, I selected medical data to perform predictive modeling for this assignment. The research question: what factor impacts the initial days’ length of stay of the patient? In this multiple linear regression model, the initial day is the dependent variable (continuous variable)

**A2. Objectives:**

The goal of this study is to identify the factors that have a strong effect on the length of stay of the initial day. From that, a predictive model can be built with proper feature selection. The ultimate result is to help focus on the right issue to resolve and possibly adjust these independent variables to reduce or strengthen their effects on the initial day’s length.

**B1. Assumption of the MLR:**

* The dependent variable and independent variables have a linear relationship.
* The correlation among independent variables is low.
* The observations of y are independently and randomly selected from the population.
* The residuals are distributed normally with a mean of zero.

**B2. Benefits of using Python for the analysis:**

Python has a wide range of libraries for data analysis and manipulation, such as NumPy, Pandas, and Scikit-learn, which make it easy to import, clean, and prepare data for multiple linear regression. Python's visualization libraries, such as Matplotlib and Seaborn, can be used to create visualizations of the data, which can help in understanding the relationship between the variables.

Python's statsmodels library provides a wide range of statistical functions, including multiple linear regression, which can be used to fit and evaluate the model. Python also offers multiple libraries to perform different types of linear regression, like OLS, Ridge, Lasso, and others, which can help to choose the best model for your problem.

**B3. Justification of applying multiple regression:**

As said above in the objective section, applying a multiple linear regression model for this study would help to predict the value of the initial day’s length (continuous variable) based on the value of multiple independent variables. This also shows the relationship between the dependent variable and the independent variables so that we can identify which one has the greatest impact on the dependent variable. Building the MLR equation will also help to identify patterns and trends in the data, from that to make predictions or forecasts based on those patterns.

**C1. Describe the data preparation goals and the data manipulation:**

For data preparation, the goal is to clean up any duplicated values or missing values by removing the duplication and filling in missing values by either mean/mode/median accordingly.

For data manipulation, the goal is to make sure there is no outlier in the chosen variable either by removing them or replacing them with a value. The extreme values of outliers can have a significant impact on the regression model. It can skew the results of the model and affect the model’s prediction performance. All these would lead to unreliable interpretation or poor fit. The next thing is to re-express chosen categorical variables into numerical ones to apply analysis.

For data selection, multicollinearity needs to be checked before running feature selection so that independent variables with high correlation can be removed.

**C2. Summary statistics**

In this assignment, the dependent variable is Initial\_days, and the independent variables are Children(int64), Age(int64), Income(float), Gender(object), VitD\_levels(float), Doc\_visits(int), Full\_meals\_eaten(int), Soft\_drink(object), Initial\_admin(object), HighBlood(object), Stroke(object), Complication\_risk(object), Overweight(object), Arthritis(object), Diabetes(object), Hyperlipidemia(object), BackPain(object), Anxiety(object), Allergic\_rhinitis(object), Reflux\_esophagitis(object), Asthma(object), Services(object), Initial\_days(float), TotalCharge(float), Additional\_charges(float), Item1(int), Item2(int), Item3(int), Item4(int), Item5(int), Item6(int), Item7(int), Item8(int). There are 34 variables with 16 categorical (object) variables and 18 continuous variables.

To determine which one of these independent variables will have a significant impact on the dependent variable, multiple linear regression needs to be performed. The focus will be on the p-value with alpha (significant level) set at 0.05 and R-square (coefficient of determination). If the p-value is greater than 0.05, the variable should be removed from the model. If the p-value is less than 0.05, the variable should be retained in the model. R-squared is used to explain how much the variance in the dependent variable can be explained by the change in independent variables. R-squared ranges from 0 to 1 (or 0 to 100%). The higher the r-squared, the better. An R-squared value of 0 means that the model explains none of the variability of the response data around its mean. On the other hand, an R-squared value of 1 means that the model explains all the variability of the response data around its mean. However, an almost absolute R-squared does not guarantee a good model, other assumptions checks will need to be performed to see. The assumptions check we perform in this assignment includes residual plot and multicollinearity.

**The size of each sample is 1000 entries, 34 independent variables as listed above, and 1 dependent variable Initial\_days. A statistic description of the variables listed as follows:**

**Table

Description automatically generated**

**Text

Description automatically generated with medium confidenceTable

Description automatically generated**

**Text

Description automatically generated with medium confidenceTable

Description automatically generated**

**Text

Description automatically generated with medium confidenceTable

Description automatically generated**

**When taking a look at the description provided, we can see that Age(53.5, 53), VitD\_levels(17.96, 17.95), Doc\_visits(5.01, 5.00), Full\_meals\_eaten(1.00, 1.00), Initial\_days(34.45, 35.83), and TotalCharge (5312, 5213) are relatively normally distributed with the mean and median similar to each other. While the mean is commonly used as a central tendency measure for continuous variables and is sensitive to outliers, the median is the middle value of a variable when the variable is arranged in order. It is not affected by outliers as much as the mean and it is commonly used as a central tendency measure for skewed data. Therefore, when the mean and the median are closely equal, there are less severe outliers. On the other hand, Children, Income, vitD\_supp, and Additional\_charges have outliers as the mean and median are far off each other**

**For the other categorical variables that were re-expressed to continuous variables including Soft\_drink(object), Initial\_admin(object), HighBlood(object), Stroke(object), Complication\_risk(object), Overweight(object), Arthritis(object), Diabetes(object), Hyperlipidemia(object), BackPain(object), Anxiety(object), Allergic\_rhinitis(object), Reflux\_esophagitis(object), Asthma(object), Services(object), these are not affected by outliers and the proper method to check is mode measurement. This can be done by using histogram visualizations (section C4) to see which one occurs most often. From the graphs, we can see that except for Overweight, the rest of the categorical variables have more No than Yes as the answer.**

**C3. Data preparation steps:**

The preparation steps are as follows:

* Import all necessary packages and load data
* Check if there are any duplicated values and treat them accordingly. Using function duplicated(), the result shows no duplication.

Table

Description automatically generated

* Check if there are any missing values and treat them accordingly by filling in the missing values with median, mean, and mode. Using function isnull().sum(), the result shows no missing value. The data appeared to be clean.

Table

Description automatically generated

* Create another data frame called df\_new and drop all the non-relevant data for the study. The new dataset now contains 34 variables.

Graphical user interface, table

Description automatically generated

* From this new data frame, choose the qualified quantitative variables and run the seaborn boxplot to check outliers. These include Children, Age, Income, VitD levels, Doc\_visits, Full\_meals\_eaten, vitD\_supp, Initial\_days, TotalCharge, and Additional\_charges. The result shows Children, Income, VitD levels, Doc\_visits, Full\_meals\_eaten, vitD\_supp, and Additional\_charges. This time, I decided to use IQR capping method to treat these outliers as having outliers in the regression model will reduce the reliability of the model prediction.

IQR is the difference between the Q3 quantile and the Q1 quantile. The upper and lower bounds are then calculated by Q3 + (1.5\*IQR), Q1 – (1.5\*IQR). After that, the outliers outside of the upper and lower bounds are replaced with the upper and lower values.

Chart, box and whisker chart

Description automatically generated

Children (before and after treatment)

Chart, box and whisker chart

Description automatically generated Chart

Description automatically generated

Income (before and after treatment)

Chart, box and whisker chart

Description automatically generated Chart, bar chart, box and whisker chart

Description automatically generated

VitD\_levels (before and after treatment)

Chart, box and whisker chart

Description automatically generated Chart, box and whisker chart

Description automatically generated

Full\_meals\_eaten (before and after treatment)

Chart, box and whisker chart

Description automatically generated Chart, box and whisker chart

Description automatically generated

VitD\_supp (before and after treatment)

Chart, box and whisker chart

Description automatically generated Chart, box and whisker chart

Description automatically generated

Additional Charges (before and after treatment)

Chart, box and whisker chart

Description automatically generated Chart, box and whisker chart

Description automatically generated

* After treating outliers, all categorical variables are re-expressed to convert from Yes/No to 1/0. Columns initial\_admin and services are re-expressed by creating dummy variables instead. After converting, the original columns are all dropped, and add the new dummy variables into the data frame using the concatenate function. Note that when creating dummy variables, the first column is dropped to avoid multicollinearity later



A picture containing calendar

Description automatically generated

A picture containing diagram

Description automatically generated

**Codes:**

* import all necessary packages

import numpy as np

import pandas as pd

import plotnine as p9

import matplotlib.pyplot as plt

import seaborn as sns

import scipy.stats as stats

import statistics

import sklearn

import statsmodels.api as sm

import statsmodels.formula.api as smf

from sklearn import linear\_model

from pandas import DataFrame

from statsmodels.stats.outliers\_influence import variance\_inflation\_factor

* load data

df = pd.read\_csv(r'C:\Users\Administrator\OneDrive\MSDA\Predictive modeling 208\Medical\medical\_clean.csv')

* Check info

df.info()

* Check duplicated data

df.duplicated()

* Check missing values

df.isnull().sum()

* Drop non-relevant columns and create a new data frame for that

df\_new = df.drop(['CaseOrder','Customer\_id','Interaction','UID','City','State','County','Zip','Lat','Lng','Area','TimeZone','Population','Job','Marital','ReAdmis'], axis =1)

* Check the new data frame

df\_new.info()

* Replace all yes/no in categorical variables to 1/0

df\_new = df\_new.replace(to\_replace={'Yes':1, 'No':0})

* **statistical summary of variables**

**df\_new.describe()**

* Create a new data frame for qualified quantitative variables to check outliers

df\_quant = df\_new[['Children','Age','Income','VitD\_levels','Doc\_visits','Full\_meals\_eaten','vitD\_supp','Initial\_days','TotalCharge','Additional\_charges']]

* Boxplot to check outliers

plt.subplots(figsize = (15,8))

sns.boxplot(data = df\_quant)

* Replace children outliers with upper and lower values using the IQR capping method

Q3 = df\_new['Children'].quantile(0.75)

Q1 = df\_new['Children'].quantile(0.25)

median = df\_new['Children'].quantile(0.5)

IQR = Q3-Q1

upper = Q3 + (1.5\*IQR)

lower = Q1 -(1.5\*IQR)

df\_new['Children'] = np.where(df\_new['Children']>upper, upper, np.where(df\_new['Children']<lower, lower, df\_new['Children']))

* Replace income outliers with upper and lower values

Q3 = df\_new['Income'].quantile(0.75)

Q1 = df\_new['Income'].quantile(0.25)

median = df\_new['Income'].quantile(0.5)

IQR = Q3-Q1

upper = Q3 + (1.5\*IQR)

lower = Q1 -(1.5\*IQR)

df\_new['Income'] = np.where(df\_new['Income']>upper, upper, np.where(df\_new['Income']<lower, lower, df\_new['Income']))

* Replace vitD level with upper and lower values

Q3 = df\_new['VitD\_levels'].quantile(0.75)

Q1 = df\_new['VitD\_levels'].quantile(0.25)

median = df\_new['VitD\_levels'].quantile(0.5)

IQR = Q3-Q1

upper = Q3 + (1.5\*IQR)

lower = Q1 -(1.5\*IQR)

df\_new['VitD\_levels'] = np.where(df\_new['VitD\_levels']>upper, upper, np.where(df\_new['VitD\_levels']<lower, lower, df\_new['VitD\_levels']))

sns.boxplot(y='VitD\_levels',data = df\_new)

* Replace full meal eaten with upper and lower value

Q3 = df\_new['Full\_meals\_eaten'].quantile(0.75)

Q1 = df\_new['Full\_meals\_eaten'].quantile(0.25)

median = df\_new['Full\_meals\_eaten'].quantile(0.5)

IQR = Q3-Q1

upper = Q3 + (1.5\*IQR)

lower = Q1 -(1.5\*IQR)

df\_new['Full\_meals\_eaten'] = np.where(df\_new['Full\_meals\_eaten']>upper, upper, np.where(df\_new['Full\_meals\_eaten']<lower, lower, df\_new['Full\_meals\_eaten']))

sns.boxplot(y='Full\_meals\_eaten',data = df\_new)

* Replace vitD supp with upper and lower value

Q3 = df\_new['vitD\_supp'].quantile(0.75)

Q1 = df\_new['vitD\_supp'].quantile(0.25)

median = df\_new['vitD\_supp'].quantile(0.5)

IQR = Q3-Q1

upper = Q3 + (1.5\*IQR)

lower = Q1 -(1.5\*IQR)

df\_new['vitD\_supp'] = np.where(df\_new['vitD\_supp']>upper, upper, np.where(df\_new['vitD\_supp']<lower, lower, df\_new['vitD\_supp']))

sns.boxplot(y='vitD\_supp',data = df\_new)

* Replace additional charges with upper and lower value

Q3 = df\_new['Additional\_charges'].quantile(0.75)

Q1 = df\_new['Additional\_charges'].quantile(0.25)

median = df\_new['Additional\_charges'].quantile(0.5)

IQR = Q3-Q1

upper = Q3 + (1.5\*IQR)

lower = Q1 -(1.5\*IQR)

df\_new['Additional\_charges'] = np.where(df\_new['Additional\_charges']>upper, upper, np.where(df\_new['Additional\_charges']<lower, lower, df\_new['Additional\_charges']))

sns.boxplot(y='Additional\_charges',data = df\_new)

* Re-express categorical variables using dummy variables

df\_new['Gender\_sub'] = df\_new['Gender'].replace(('Male','Female','Nonbinary'),(0,1,2))

df\_new['Complication\_risk\_sub'] = df\_new['Complication\_risk'].replace(('Low','Medium','High'),(0,1,2))

Initial\_admin\_dummy = pd.get\_dummies(df\_new['Initial\_admin'], drop\_first=True)

Services\_dummy = pd.get\_dummies(df\_new['Services'], drop\_first=True)

* Attach the new dummy variables into the data frame and drop the original ones

df\_new = pd.concat([df\_new,Initial\_admin\_dummy,Services\_dummy], axis = 1)

df\_new = df\_new.drop(['Gender','Complication\_risk','Initial\_admin','Services'], axis=1)

**C4. Univariate and bivariate visualizations in the cleaned data set**

#histogram plot for continuous variables

df\_new.hist(figsize = (14,14))

Box and whisker chart

Description automatically generated

#Bivariate categorical and continous variable

y = 'Initial\_days'

x = ['Gender','Initial\_admin','Services','Complication\_risk']

sns.pairplot(df\_new, x\_vars = x, y\_vars = y, height =5)

Chart, bar chart

Description automatically generated

#Bivariate continuous and dependent variable

y = 'Initial\_days'

x = ['Children','Age','Income','VitD\_levels']

sns.pairplot(df\_new, x\_vars = x, y\_vars = y, height =5)

A picture containing graphical user interface

Description automatically generated

y = 'Initial\_days'

x = ['TotalCharge','Additional\_charges','Item1','Item2','Item3']

sns.pairplot(df\_new, x\_vars = x, y\_vars = y, height =6)

Chart, bar chart, histogram

Description automatically generated

**C5. Please see attached CSV file for a cleaned data set**

**D1. Initial multiple regression model with all independent variables**

Table

Description automatically generated

The initial model is as follows:

Y = a + bx1 + bx2 + …. Or

Initial\_days = -21.3017

-0.0010Children+0.0059Age -5.531e-07Income+ 0.0061VitD\_levels-0.0082Doc\_visits-0.0025Full\_meals\_eaten-0.0246vitD\_supp+ 0.0240Soft\_drink-1.0850HighBlood+ 0.0126 Stroke+ 0.0474 Overweight-0.8341Arthritis-0.9218Diabetes-1.1142Hyperlipidemia -1.0657BackPain -1.0320Anxiety-0.7807Allergic\_rhinitis-0.7368Reflux\_esophagitis+ 0.0115 Asthma+ 0.0122TotalCharge-2.789e-05Additional\_charges-0.0067Item1 -0.0032Item2+ 0.0210Item3+0.0059Item4+ 0.0125Item5+ 0.0196Item6 -0.0301Item7+ 0.0036Item8-0.0234Gender\_sub -2.7726Complication\_risk\_sub -6.2221Emergency Admission +0.0613Observation Admission+ 0.0237CT Scan -0.0074Intravenous+ 0.0733MRI

**D2. Justify the selection procedure and a model evaluation to reduce the initial model**

I decided to use the backward elimination method and evaluated independent variables by their p-values. As said previously, after running the regression model, I checked the R-squared value which showed here as 0.998 or translating to 99.8%, this is exceedingly high, and may need to check other factors impacting this. Naturally, the more variables we have, the higher the r-squared becomes. From the initial model, it included 36 variables that possibly influence initial days. The interpretation for this model was simply that it intercepted the Y-axis at -21.3017 and for every plus (+) sign it meant every time that variable happens initial\_days was going up to the degree of the coefficient of that IV indicating a positive relationship, for every minus (-) sign in the equation meant every time that variable happens initial\_days was going down to the degree of the coefficient of that IV indicating negative relationship. Before we move on to the multicollinearity check and linearity check, we need to reduce the initial model. To do that, I started off by removing the one with the largest p-value one by one. After every removal of the largest p-value variable, the OLS was run again, and the process repeats until there was no more variable to remove (Singh, 2021). In total, there were 23 OLS run to get rid of all the high p-value variables. The removal was done in this order: Children, Item2, Full\_meal\_eatens, Intravenous, Item8, Stroke, Asthma, item4, item1, item5, doc\_visits, CT scan, Soft\_drink, vitD\_levels, gender\_sub, MRI, income, item6, vitd\_supp, observation admission, item3, overweight, item7

**D3. Reduced model:**

After implementing the backward elimination method and running OLS 23 times, the equation had 13 significant independent variables left, including both continuous and categorical variables. Please note, although shown in the graph as all continuous, many of them were re-expressed from categorical to continuous variables

Table

Description automatically generated

Table

Description automatically generated with low confidence

The new equation is

Initial\_days = -21.1169 +0.0058Age-1.0876HighBlood-0.8347Arthritis-0.9213Diabetes-1.1143Hyperlipidemia-1.0645BackPain-1.0332Anxiety-0.7837Allergic\_rhinitis-0.7377Reflux\_esophagitis+0.0122TotalCharge-2.757e-05Additional\_charges-2.7739Complication\_risk\_sub-6.2505Emergency Admission

This is not yet the completed model as I need to check multicollinearity. To do that, I used the function corr to create a heatmap visualization and then VIF. The process is similar to backward elimination where I would remove one by one the variable with VIF larger than 10 (not all at once)

Graphical user interface, application

Description automatically generated

Chart, waterfall chart

Description automatically generated

Setting vmin and vmax as -1 to 1, it was shown that there was not much correlation between IVs. The one with the highest dark green colors were Additional\_charges, Age, and Highblook with correlations of 0.72 and 0.66. This correlation is not higher than 0.8 but I performed the VIF anyway.

Graphical user interface, text, application, email

Description automatically generated

From the table, those 3 indeed showed an exceedingly high VIF > 10. I removed Additional\_charges as it had the highest VIF. After it was done, everything was back to less than 10. There is no need to remove any further variables.

Graphical user interface, text, application, email

Description automatically generated

The final regression equation:

Initial\_days = -21.1169 +0.0058Age-1.0876HighBlood-0.8347Arthritis-0.9213Diabetes-1.1143Hyperlipidemia-1.0645BackPain-1.0332Anxiety-0.7837Allergic\_rhinitis-0.7377Reflux\_esophagitis+0.0122TotalCharge -2.7739Complication\_risk\_sub-6.2505Emergency Admission

**E1. Comparing initial and reduced models**

* Variable selection techniques:

For the initial model, variables were selected based on EDA techniques by plotting scatterplots to check the relationship between independent variables and dependent variables. Admittedly, from all the plots demonstrated above, I did not find significant relationships between them. The one that I found that had a strong relationship with is total\_charges and initial\_days. Moreover, by scanning through histograms, I noticed many of them did not have a normal distribution, which was one of the assumptions for the regression model. The initial model was therefore a selection of variables that I consider most relevant and most likely to have an impact on the dependent variables, including both categorical and continuous ones.

For the reduced model, it was done using the backward elimination (or kitchen sink) method, I ran the OSL linear regression based on the initial model and repeatedly removed variables with the largest p-value one after another in each run until I could not remove any more variables.

* The model evaluation metric:

As mentioned above, when first running the initial regression model, it was shown that the R-squared was 0.998 and F-stat was 0.00. This was exceedingly considering there were many variables with a p-value higher than 0.05 on the table. The backward elimination was then applied to remove insignificant variables. After 23 times running the OSL, the model has 13 variables left that fit the requirement of having p-values smaller than 0.05 and the total model F stat was still at 0.00. The R-squared was still at 0.998. A multicollinearity check was then done to evaluate if there were any IVs has a strong correlation with each other. The heatmap and VIF function were then used to evaluate the remaining 13 IVs. There was none that had a correlation higher than 0.8 but Age, Highblood, and Additional\_charges had a high of 0.72 and 0.66. The VIF then showed that 3 of them had exceedingly high VIF (as table above). Similar to backward elimination, I removed the variable with the highest VIF one by one, in this case, additional charges with a VIF score of 72. After removing this variable, the remaining variables showed a good fit, there is no further removal needed.

* Residual plot:

Chart, scatter chart

Description automatically generated

Chart, histogram

Description automatically generated

**E2. The output of calculation and residual error:**

Initial model: Initial\_days = -21.3017

-0.0010Children+0.0059Age -5.531e-07Income+ 0.0061VitD\_levels-0.0082Doc\_visits-0.0025Full\_meals\_eaten-0.0246vitD\_supp+ 0.0240Soft\_drink-1.0850HighBlood+ 0.0126 Stroke+ 0.0474 Overweight-0.8341Arthritis-0.9218Diabetes-1.1142Hyperlipidemia -1.0657BackPain -1.0320Anxiety-0.7807Allergic\_rhinitis-0.7368Reflux\_esophagitis+ 0.0115 Asthma+ 0.0122TotalCharge-2.789e-05Additional\_charges-0.0067Item1 -0.0032Item2+ 0.0210Item3+0.0059Item4+ 0.0125Item5+ 0.0196Item6 -0.0301Item7+ 0.0036Item8-0.0234Gender\_sub -2.7726Complication\_risk\_sub -6.2221Emergency Admission +0.0613Observation Admission+ 0.0237CT Scan -0.0074Intravenous+ 0.0733MRI

Reduced model before multicollinearity check:

Initial\_days = -21.1169 +0.0058Age-1.0876HighBlood-0.8347Arthritis-0.9213Diabetes-1.1143Hyperlipidemia-1.0645BackPain-1.0332Anxiety-0.7837Allergic\_rhinitis-0.7377Reflux\_esophagitis+0.0122TotalCharge-2.757e-05Additional\_charges-2.7739Complication\_risk\_sub-6.2505Emergency Admission

Final regression model

Initial\_days = -21.1169 +0.0058Age-1.0876HighBlood-0.8347Arthritis-0.9213Diabetes-1.1143Hyperlipidemia-1.0645BackPain-1.0332Anxiety-0.7837Allergic\_rhinitis-0.7377Reflux\_esophagitis+0.0122TotalCharge -2.7739Complication\_risk\_sub-6.2505Emergency Admission

Residual:

Graphical user interface, table

Description automatically generated

**E3. Codes:**

* Initial model:

y = df\_new['Initial\_days']

x = df\_new[['Children','Age','Income','VitD\_levels','Doc\_visits','Full\_meals\_eaten','vitD\_supp','Soft\_drink','HighBlood','Stroke','Overweight','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','Asthma','TotalCharge','Additional\_charges','Item1','Item2','Item3','Item4','Item5','Item6','Item7','Item8','Gender\_sub','Complication\_risk\_sub','Emergency Admission','Observation Admission','CT Scan','Intravenous','MRI']]

regr = linear\_model.LinearRegression()

regr.fit(x,y)

print('Intercep: \n',regr.intercept\_)

print('Coefficients: \n',regr.coef\_)

x=sm.add\_constant(x)

model = sm.OLS(y,x).fit()

predictions = model.predict(x)

print\_model = model.summary()

print(print\_model)

* remove children

y = df\_new['Initial\_days']

x = df\_new[['Age','Income','VitD\_levels','Doc\_visits','Full\_meals\_eaten','vitD\_supp','Soft\_drink','HighBlood','Stroke','Overweight','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','Asthma','TotalCharge','Additional\_charges','Item1','Item2','Item3','Item4','Item5','Item6','Item7','Item8','Gender\_sub','Complication\_risk\_sub','Emergency Admission','Observation Admission','CT Scan','Intravenous','MRI']]

regr = linear\_model.LinearRegression()

regr.fit(x,y)

x=sm.add\_constant(x)

model = sm.OLS(y,x).fit()

print(model.summary())

* Removal Item2

y = df\_new['Initial\_days']

x = df\_new[['Age','Income','VitD\_levels','Doc\_visits','Full\_meals\_eaten','vitD\_supp','Soft\_drink','HighBlood','Stroke','Overweight','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','Asthma','TotalCharge','Additional\_charges','Item1','Item3','Item4','Item5','Item6','Item7','Item8','Gender\_sub','Complication\_risk\_sub','Emergency Admission','Observation Admission','CT Scan','Intravenous','MRI']]

x=sm.add\_constant(x)

model = sm.OLS(y,x).fit()

print(model.summary())

* Remove full meal eaten

y = df\_new['Initial\_days']

x = df\_new[['Age','Income','VitD\_levels','Doc\_visits','vitD\_supp','Soft\_drink','HighBlood','Stroke','Overweight','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','Asthma','TotalCharge','Additional\_charges','Item1','Item3','Item4','Item5','Item6','Item7','Item8','Gender\_sub','Complication\_risk\_sub','Emergency Admission','Observation Admission','CT Scan','Intravenous','MRI']]

x=sm.add\_constant(x)

model = sm.OLS(y,x).fit()

print(model.summary())

* remove Intravenous

y = df\_new['Initial\_days']

x = df\_new[['Age','Income','VitD\_levels','Doc\_visits','vitD\_supp','Soft\_drink','HighBlood','Stroke','Overweight','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','Asthma','TotalCharge','Additional\_charges','Item1','Item3','Item4','Item5','Item6','Item7','Item8','Gender\_sub','Complication\_risk\_sub','Emergency Admission','Observation Admission','CT Scan','MRI']]

x=sm.add\_constant(x)

model = sm.OLS(y,x).fit()

print(model.summary())

* remove item8

y = df\_new['Initial\_days']

x = df\_new[['Age','Income','VitD\_levels','Doc\_visits','vitD\_supp','Soft\_drink','HighBlood','Stroke','Overweight','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','Asthma','TotalCharge','Additional\_charges','Item1','Item3','Item4','Item5','Item6','Item7','Gender\_sub','Complication\_risk\_sub','Emergency Admission','Observation Admission','CT Scan','MRI']]

x=sm.add\_constant(x)

model = sm.OLS(y,x).fit()

print(model.summary())

* remove stroke

y = df\_new['Initial\_days']

x = df\_new[['Age','Income','VitD\_levels','Doc\_visits','vitD\_supp','Soft\_drink','HighBlood','Overweight','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','Asthma','TotalCharge','Additional\_charges','Item1','Item3','Item4','Item5','Item6','Item7','Gender\_sub','Complication\_risk\_sub','Emergency Admission','Observation Admission','CT Scan','MRI']]

x=sm.add\_constant(x)

model = sm.OLS(y,x).fit()

print(model.summary())

* remove asthma

y = df\_new['Initial\_days']

x = df\_new[['Age','Income','VitD\_levels','Doc\_visits','vitD\_supp','Soft\_drink','HighBlood','Overweight','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','TotalCharge','Additional\_charges','Item1','Item3','Item4','Item5','Item6','Item7','Gender\_sub','Complication\_risk\_sub','Emergency Admission','Observation Admission','CT Scan','MRI']]

x=sm.add\_constant(x)

model = sm.OLS(y,x).fit()

print(model.summary())

* remove item4

y = df\_new['Initial\_days']

x = df\_new[['Age','Income','VitD\_levels','Doc\_visits','vitD\_supp','Soft\_drink','HighBlood','Overweight','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','TotalCharge','Additional\_charges','Item1','Item3','Item5','Item6','Item7','Gender\_sub','Complication\_risk\_sub','Emergency Admission','Observation Admission','CT Scan','MRI']]

x=sm.add\_constant(x)

model = sm.OLS(y,x).fit()

print(model.summary())

* remove item1

y = df\_new['Initial\_days']

x = df\_new[['Age','Income','VitD\_levels','Doc\_visits','vitD\_supp','Soft\_drink','HighBlood','Overweight','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','TotalCharge','Additional\_charges','Item3','Item5','Item6','Item7','Gender\_sub','Complication\_risk\_sub','Emergency Admission','Observation Admission','CT Scan','MRI']]

x=sm.add\_constant(x)

model = sm.OLS(y,x).fit()

print(model.summary())

* remove item5

y = df\_new['Initial\_days']

x = df\_new[['Age','Income','VitD\_levels','Doc\_visits','vitD\_supp','Soft\_drink','HighBlood','Overweight','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','TotalCharge','Additional\_charges','Item3','Item6','Item7','Gender\_sub','Complication\_risk\_sub','Emergency Admission','Observation Admission','CT Scan','MRI']]

x=sm.add\_constant(x)

model = sm.OLS(y,x).fit()

print(model.summary())

* remove doc\_visit

y = df\_new['Initial\_days']

x = df\_new[['Age','Income','VitD\_levels','vitD\_supp','Soft\_drink','HighBlood','Overweight','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','TotalCharge','Additional\_charges','Item3','Item6','Item7','Gender\_sub','Complication\_risk\_sub','Emergency Admission','Observation Admission','CT Scan','MRI']]

x=sm.add\_constant(x)

model = sm.OLS(y,x).fit()

print(model.summary())

* remove CT scan

y = df\_new['Initial\_days']

x = df\_new[['Age','Income','VitD\_levels','vitD\_supp','Soft\_drink','HighBlood','Overweight','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','TotalCharge','Additional\_charges','Item3','Item6','Item7','Gender\_sub','Complication\_risk\_sub','Emergency Admission','Observation Admission','MRI']]

x=sm.add\_constant(x)

model = sm.OLS(y,x).fit()

print(model.summary())

* remove soft\_drink

y = df\_new['Initial\_days']

x = df\_new[['Age','Income','VitD\_levels','vitD\_supp','HighBlood','Overweight','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','TotalCharge','Additional\_charges','Item3','Item6','Item7','Gender\_sub','Complication\_risk\_sub','Emergency Admission','Observation Admission','MRI']]

x=sm.add\_constant(x)

model = sm.OLS(y,x).fit()

print(model.summary())

* remove vitD\_levels

y = df\_new['Initial\_days']

x = df\_new[['Age','Income','vitD\_supp','HighBlood','Overweight','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','TotalCharge','Additional\_charges','Item3','Item6','Item7','Gender\_sub','Complication\_risk\_sub','Emergency Admission','Observation Admission','MRI']]

x=sm.add\_constant(x)

model = sm.OLS(y,x).fit()

print(model.summary())

* remove gender\_sub

y = df\_new['Initial\_days']

x = df\_new[['Age','Income','vitD\_supp','HighBlood','Overweight','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','TotalCharge','Additional\_charges','Item3','Item6','Item7','Complication\_risk\_sub','Emergency Admission','Observation Admission','MRI']]

x=sm.add\_constant(x)

model = sm.OLS(y,x).fit()

print(model.summary())

* remove MRI

y = df\_new['Initial\_days']

x = df\_new[['Age','Income','vitD\_supp','HighBlood','Overweight','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','TotalCharge','Additional\_charges','Item3','Item6','Item7','Complication\_risk\_sub','Emergency Admission','Observation Admission']]

x=sm.add\_constant(x)

model = sm.OLS(y,x).fit()

print(model.summary())

* remove income

y = df\_new['Initial\_days']

x = df\_new[['Age','vitD\_supp','HighBlood','Overweight','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','TotalCharge','Additional\_charges','Item3','Item6','Item7','Complication\_risk\_sub','Emergency Admission','Observation Admission']]

x=sm.add\_constant(x)

model = sm.OLS(y,x).fit()

print(model.summary())

* remove item6

y = df\_new['Initial\_days']

x = df\_new[['Age','vitD\_supp','HighBlood','Overweight','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','TotalCharge','Additional\_charges','Item3','Item7','Complication\_risk\_sub','Emergency Admission','Observation Admission']]

x=sm.add\_constant(x)

model = sm.OLS(y,x).fit()

print(model.summary())

* remove vitD\_supp

y = df\_new['Initial\_days']

x = df\_new[['Age','HighBlood','Overweight','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','TotalCharge','Additional\_charges','Item3','Item7','Complication\_risk\_sub','Emergency Admission','Observation Admission']]

x=sm.add\_constant(x)

model = sm.OLS(y,x).fit()

print(model.summary())

* remove Observation Admission

y = df\_new['Initial\_days']

x = df\_new[['Age','HighBlood','Overweight','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','TotalCharge','Additional\_charges','Item3','Item7','Complication\_risk\_sub','Emergency Admission']]

x=sm.add\_constant(x)

model = sm.OLS(y,x).fit()

print(model.summary())

* remove item3

y = df\_new['Initial\_days']

x = df\_new[['Age','HighBlood','Overweight','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','TotalCharge','Additional\_charges','Item7','Complication\_risk\_sub','Emergency Admission']]

x=sm.add\_constant(x)

model = sm.OLS(y,x).fit()

print(model.summary())

* remove Overweight

y = df\_new['Initial\_days']

x = df\_new[['Age','HighBlood','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','TotalCharge','Additional\_charges','Item7','Complication\_risk\_sub','Emergency Admission']]

x=sm.add\_constant(x)

model = sm.OLS(y,x).fit()

print(model.summary())

* remove item7

y = df\_new['Initial\_days']

x = df\_new[['Age','HighBlood','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','TotalCharge','Additional\_charges','Complication\_risk\_sub','Emergency Admission']]

x=sm.add\_constant(x)

model = sm.OLS(y,x).fit()

print(model.summary())

regr = linear\_model.LinearRegression()

model = regr.fit(x, y)

print('Intercept:', model.intercept\_)

print('Coefficients:', model.coef\_)

* Check data type after

df\_new[['Age','HighBlood','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','TotalCharge','Additional\_charges','Complication\_risk\_sub','Emergency Admission']].info()

* create heatmap and correlation check

df\_heatmap = df\_new[['Age','HighBlood','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','TotalCharge','Additional\_charges','Complication\_risk\_sub','Emergency Admission']]

plt.subplots(figsize=(30,30))

c = df\_heatmap.corr()

sns.heatmap(c,cmap = 'BrBG',annot=True, vmin = -1, vmax =1)

c

* VIF check

x = df\_new[['Age','HighBlood','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','TotalCharge','Additional\_charges','Complication\_risk\_sub','Emergency Admission']]

vif\_data = pd.DataFrame()

vif\_data["feature"] = x.columns

# calculating VIF for each feature

vif\_data["VIF"] = [variance\_inflation\_factor(x.values, i)

for i in range(len(x.columns))]

print(vif\_data)

* remove Additional charge

x = df\_new[['Age','HighBlood','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','TotalCharge','Complication\_risk\_sub','Emergency Admission']]

vif\_data = pd.DataFrame()

vif\_data["feature"] = x.columns

# calculating VIF for each feature

vif\_data["VIF"] = [variance\_inflation\_factor(x.values, i)

for i in range(len(x.columns))]

print(vif\_data)

* scatterplot residuals

y= df\_new['Initial\_days']

x = df\_new[['Age','HighBlood','Arthritis','Diabetes','Hyperlipidemia','BackPain','Anxiety','Allergic\_rhinitis','Reflux\_esophagitis','TotalCharge','Complication\_risk\_sub','Emergency Admission']]

model = sm.OLS(y,x).fit()

residuals = model.resid

plt.scatter(model.predict(), residuals)

plt.xlabel("Predicted Values")

plt.ylabel("Residuals")

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

plt.show()

* check normal distribution

sns.displot(model.resid)

plt.show()

* print residuals

print(residuals)

MSE = sum((residuals)\*\*2)/len(residuals)

print(MSE)

**F1. Discuss the results of the data analysis:**

* The regression equation: As mentioned above, the final reduced regression model is Initial\_days = -21.1169 +0.0058Age-1.0876HighBlood-0.8347Arthritis-0.9213Diabetes-1.1143Hyperlipidemia-1.0645BackPain-1.0332Anxiety-0.7837Allergic\_rhinitis-0.7377Reflux\_esophagitis+0.0122TotalCharge -2.7739Complication\_risk\_sub-6.2505Emergency Admission

This is concluded after backward elimination and multicollinearity check. This means the intercept on the y-axis is at -21.1169 (when all other variables are 0). When age and total charges go up, the length of initial days goes up by 0.0058 units and 0.0122 units respectively. For the rest of the variables, when they go up, the length of initial days will go down respectively to the coefficient of that variable units. -1.0876HighBlood, -0.8347Arthritis, -0.9213Diabetes, -1.1143Hyperlipidemia, -1.0645BackPain, -1.0332Anxiety, -0.7837Allergic\_rhinitis, -0.7377Reflux, -2.7739Complication\_risk\_sub, -6.2505Emergency Admission

* The final table also showed the r-squared at 0.998 and F-stat at 0.00, which was still extremely high, meaning 99.8% of the variant in the dependent variable can be explained by the change in those independent variables and the confidence level is almost 100%
* Statistically, those OSL numbers look good. The residual number as calculated above showed a mean sum error of 23.8066. The residual plot showed a non-random pattern as it skewed more to both sides. The histogram plot, however, showed a normal distribution with a mean of almost 0. Practically, since the scatterplot did now show a good linear relationship and the residual plots did not show a random pattern, I am not confident this is the best model to run.
* Limitation of the data analysis: As mentioned above, the thing I’m concerned about is the linearity check and the residual plots that may affect the accuracy of the model. Moreover, there are possibly other variables that were missing from the dataset that may affect the prediction of the model.

**F2. Recommendation**

Based on the result, a more in-depth analysis with different approaches may need to be applied to validate this model. A different set of data may need to be collected to verify if there are any other variables affecting the initial days’ length of stay. On the model itself, we can see that when age and total charge go up, the initial stay will go up. While age is a natural factor we can’t control, the complication factors that come with age are something we might want to do further research on. For the total charge, it is logical thing that when someone stays longer, the total charge will go up. Regarding all other variables, it’s interesting to see that when those diseases go up, the initial days go down. Again, this is something that needs to be further analyzed by collecting more information on whether these people are transferred to somewhere else right after.

**G. Please see the Panopto video in a separate attached.**

**H. Third-party web reference**

In this study, I used the dataset medical\_clean provided by the WGU assessment link. It was downloaded and then saved in my personal One Drive

*Medical Data and Dictionary Files*. (2022). [Dataset]. Western Governors University. https://"C:\Users\Administrator\OneDrive\MSDA\Predictive modeling 208\Medical\medical\_clean.csv"

**I. References**

*Medical Data and Dictionary Files*. (2022). [Dataset]. Western Governors University. https://"C:\Users\Administrator\OneDrive\MSDA\Predictive modeling 208\Medical\medical\_clean.csv"

Singh, H. (2021, April 7). *Backward Feature Elimination and its Implementation*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/04/backward-feature-elimination-and-its-implementation/