**D208 Performance Assessment**

**MULTIPLE REGRESSION FOR MEDICAL DATA**

Fahim A. Akbar Student ID 001434895 Masters Data Analytics (January 1, 2021) Program Mentor: Lea Yoakem (877) 435-7948x6422 [fakbar3@wgu.edu](mailto:fakbar3@wgu.edu)

**Part I: Research Question**

**A. Purpose of the Multiple Regression Report**

**1. Question**

Multiple Linear Regression could be used to help answer the following: “How do patient observations correlate with initial days?”

**2. Goal**

One goal of this data analysis is to see if we can predict the readmission of a patient based on which independent variables in the dataset are a good indicator and fit. The independent variables identified will be used in the multiple linear regression to determine the probability of readmission. For this study we will use initial days as our target variable, and see how it compares to patient observations.

**Part II: Method Justification**

**B. Multiple Regression Methods**

**1. Multiple regression model assumptions**

According to Statistics Solutions (2021) these are the assumptions a multiple regression model needs to meet in order to be a good fit:

* Linear relationship – there must be a linear relationship between the independent and dependent variables. We can check if the relationship is linear by creating scatterplots.
* Errors should be normally distributed – errors between observed and predicted values should have a normal distribution. Histograms or Q-Q-Plots cab be used to check if errors are normally distributed. Normality can also be verified through the goodness of fit test. These are also referred to as the residuals of the regression.
* No multicollinearity – the independent variables should not be highly correlated with each other. A correlation matrix or a variance inflation factor can be used to check for high correlation. If multicollinearity is detected, centering the data by subtracting the mean score from each independent variables’ observations. If multicollinearity still exists the next best step would be to identify and remove the variables causing multicollinearity.
* Homoscedasticity –no clear pattern should be present in the distribution of data in a scatterplot of residuals versus predicted values.

**2. Benefits of using Python**

Python will be used to perform multiple regression of the medical data. Python has packages built for performing statistical analyses, including Pandas, Scipy, and Statsmodels. The syntax is easy to grasp, and Python also supports visualizations of variables and observations in them. This is especially helpful when presenting findings to people unfamiliar with coding. Python also has commands specific to multiple regression. The commands linear\_regression.fit, sm.OLS, and model.coef\_. will be used to run the regression analyses and create visulization.

**3. Why multiple regression is an appropriate technique to analyze the research question**

Multiple regression was chosen as the technique to analyze the research question because it allows us to analyze each patient observation variable. We can then see which variables have the strongest correlation with readmission. Most of the variables in the data set are numeric independent variables, and they have the potential to impact the number of initial days a patient is admitted for.

**Part III: Data Preparation**

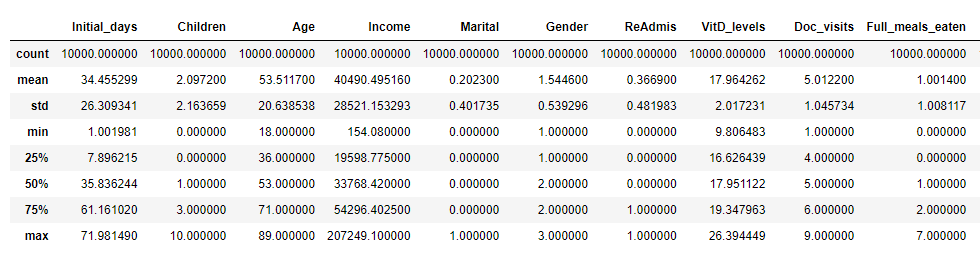
**C. Data Preparation Process for Multiple Regression analysis**

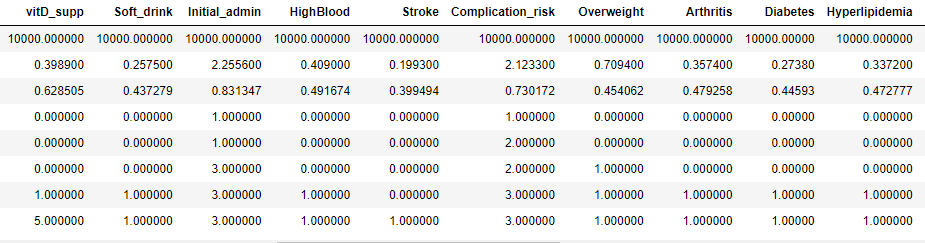
**1. Relevant data preprocessing goal and data manipulations**

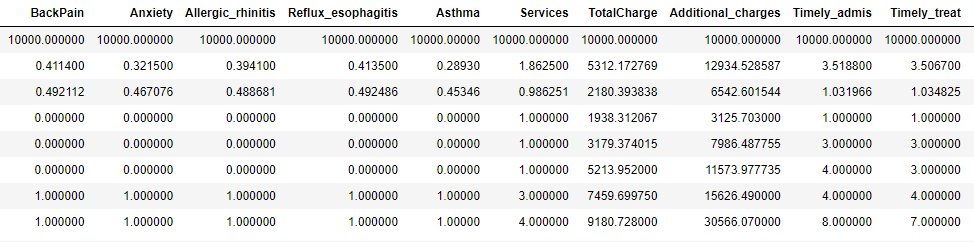
To prepare the data for multiple regression, we will first fix all of the null and missing values. This will be zone by either replacing them with zeros or populating based on average values for the variables. Categorical variables will be converted into numerical variables so that we can conduct linear regression. Demographic data will be removed since they do not pertain to our research question. If any duplicate data entries, rows, or columns are found, they will be removed.

**2. Summary statistics**

For the multiple regression to work, we need to identify the P-values and coefficients for the independent variables. These will help indicate which independent variables will impact the target variable, Initial\_days. The predictor variables in this study are the patient observations, such as Services, Children, and Full\_meals\_eaten. The tables below show standard deviations of each numerical variable as well as the dispersion in the interquartile ranges. Children, Full\_meals\_eaten, Income, and vitD\_supp are not normally distributed variables, while Age, VitD\_Levels, Doc\_visits, and Initial\_days are normally distributed.









All of the categorical variables were converted to numerical variables so that we can conduct the regression analysis and compare statistical data. The variable visualizations can be seen below in section 4. A summary of the statistics for these variables show that Age, Reflux\_esophagitis, BackPain, and Gender are normally distributed. In contrast, Marital, Soft\_drink, Initial\_admin, Stroke, Complication\_risk, and Diabetes are separated by large margins. Furthermore, we see that “No” responses for ReAdmis are more common than those with “Yes” responses. In regards to the bivariate analyses, Doc\_visits, Initial\_admin, Complication\_risk, Stroke and Diabetes are all normally distributed when compared against Initial\_days. In contrast, ReAdmi, Timely\_admis, and Timely\_visits are not normally distributed when compared against Initial\_days.

**3. Data Preparation Steps**

The first step in preparing the data is to make sure that there are no missing data entries in any of the columns. Next, we will ensure that none of the data in the columns is duplicated. We will make sure that none of the columns or rows are duplicated, to further prevent dealing with repeated entries. For the regression analysis, several columns in the dataset were deemed irrelevant and were subsequently dropped from the dataset (i.e latitude, longitude). Additionally, the predictor variables will need to be scaled, and the data will need to split into test and training sets. After the predictor variables have been scaled, the “yes/no” entries for the categorical variables will need to be converted to 1 and 0, respectively. Lastly, we’ll identify the target variable and move it to the first column of the dataset for easier visual cues.

**Code used for preparing data:**

#import packages and clean data before running multiple regression analysis. Rename the survey item variables accordingly.

import numpy as np

import pandas as pd

from sklearn import linear\_model

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

pd.set\_option('display.max\_columns', None)

import pylab

from pylab import rcParams

import statsmodels.api as sm

import statistics

from scipy import stats

import sklearn

from sklearn import preprocessing

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import train\_test\_split

from sklearn import metrics

from sklearn.metrics import classification\_report

from scipy.stats import chisquare

from scipy.stats import chi2\_contingency

df = pd.read\_csv (r'C:\Users\fahim\Documents\0\_WGUDocuments\d208\1medical\_clean.csv')

df.rename(columns={'Item1':'Timely\_admis','Item2':'Timely\_treat',

'Item3':'Timely\_visits','Item4':'Reliability',

'Item5':'Options','Item6':'Hrs\_treat',

'Item7':'Courteous','Item8':'Active\_listen'},inplace=True)

df.head()

df.info()

#check if there are any missing data entries - if there are none then the output should be False

df.isna().any()

#check if there is any duplicate data entries present in columns

df[df.duplicated()

# check if there are any duplicated rows in the data set - if there are none then the output should be False

df.duplicated().any()

# remove demographic data from the data set since these entries won't be necessary for the multiple regression analysis

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

# check to make sure that the columns for demographic data were dropped before proceeding

df.head()

# convert categorical yes/no values to numeric 1/0 values

df = df.replace(to\_replace = ['Yes','No'],value = [1,0])

df

# convert the categorical variable of genders to a numeric variable

df['Gender'] = df['Gender'].replace(['Male','Female','Nonbinary'],[1,2,3])

df

# convert the non-married Marital status values to "Married/Not Married", then convert "Married/Not Married" to "1/0"

#this will make the Marital variable easier to work with during regression analysis

df['Marital'] = df['Marital'].replace(['Divorced','Widowed','Separated','Never Married'],'Not Married')

df['Marital'] = df['Marital'].replace(['Married','Not Married'],[1,0])

# convert the Initial\_Admin, Complication\_risk, and Services variables into integers before proceeding

df['Initial\_admin'] = df['Initial\_admin'].replace(['Elective Admission','Observation Admission','Emergency Admission'],[1,2,3])

df['Complication\_risk'] = df['Complication\_risk'].replace(['Low','Medium','High'],[1,2,3])

df['Services'] = df['Services'].replace(['Blood Work','CT Scan','Intravenous','MRI'],[1,2,3,4])

df.info()

df.describe()

my\_list = df.columns.values.tolist()

print(my\_list)

# move the chosen target variable "Initial\_days" to beginning of the columns

df=df[['Initial\_days','Children', 'Age', 'Income', 'Marital', 'Gender', 'ReAdmis', 'VitD\_levels', 'Doc\_visits', 'Full\_meals\_eaten', 'vitD\_supp', 'Soft\_drink', 'Initial\_admin', 'HighBlood', 'Stroke', 'Complication\_risk', 'Overweight', 'Arthritis', 'Diabetes', 'Hyperlipidemia', 'BackPain', 'Anxiety', 'Allergic\_rhinitis', 'Reflux\_esophagitis', 'Asthma', 'Services', 'TotalCharge', 'Additional\_charges', 'Timely\_admis', 'Timely\_treat', 'Timely\_visits', 'Reliability', 'Options', 'Hrs\_treat', 'Courteous', 'Active\_listen']]

# Confirm that the target variable was moved before exporting the prepared dataset my\_list = df.columns.values.tolist()

print(my\_list)

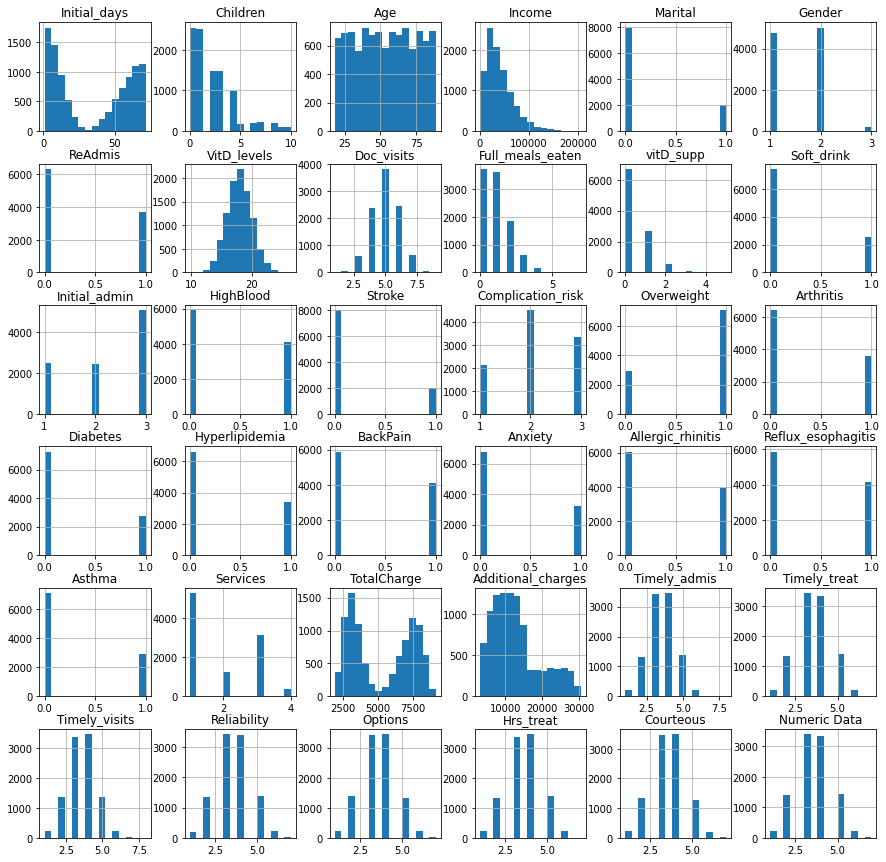
# describe the dataframe to identify distribution of variables

df.describe()

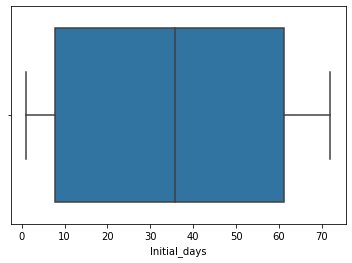
# now that all the modifications have been made, export the prepared datasetdf.to\_csv(r'C:\Users\fahim\Documents\0\_WGUDocuments\d208\1medical\_clean-PREPAREDTASK1.csv')

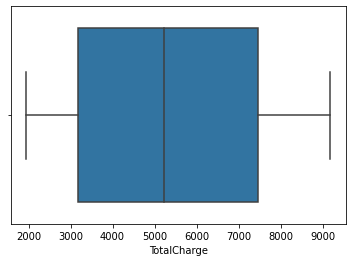
**4. Univariate and bivariate visualizations of the distributions of variables**

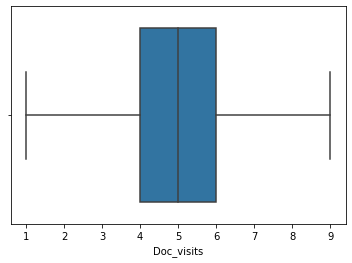
The cleaned data set was used to prepare visualizations of the variables. These will help determine which variables will be used in the linear regression model. The resulting histograms that were created are shown below.



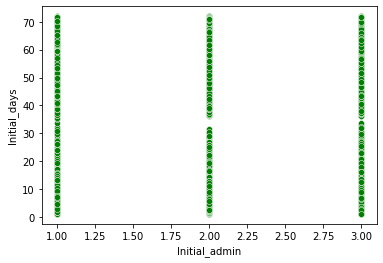
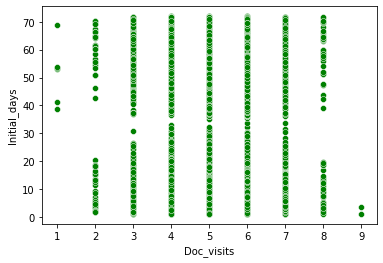
To address the outliers in the variables that were not evenly distributed, boxplots were created to remove the outliers as shown below:

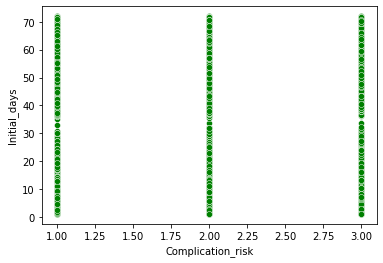




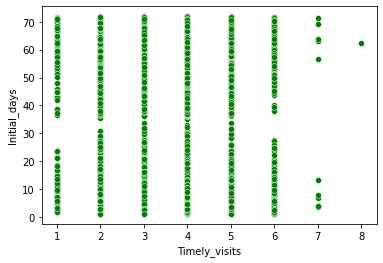
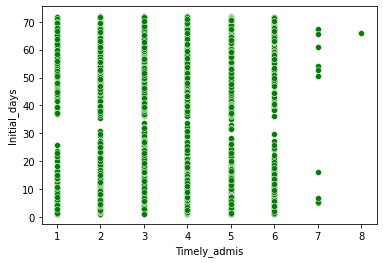


For visualizing bivariate analyses the target Initial\_days is compared with potential predictors in the scatterplots shown below:









**5. Provide a copy of the cleaned data set.**

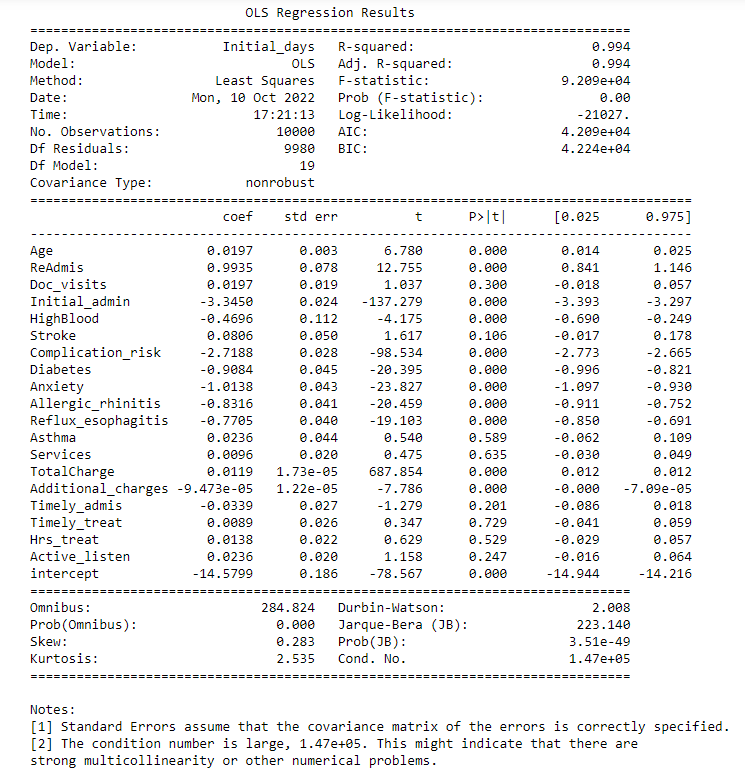
A copy of the cleaned data set, titled “1medical\_clean-PREPAREDTASK1.cvs” is provided in the task submission.

**Part IV: Model Comparison and Analysis**

**D.** **Comparison between an initial and a reduced multiple regression model**

**1. Initial multiple regression model from all predictors identified in Part C2.**

We will run an initial regression on potential predictor variables, and then compare these against the target variable of Initial\_days. The OLS Regression Results are shown below:

****

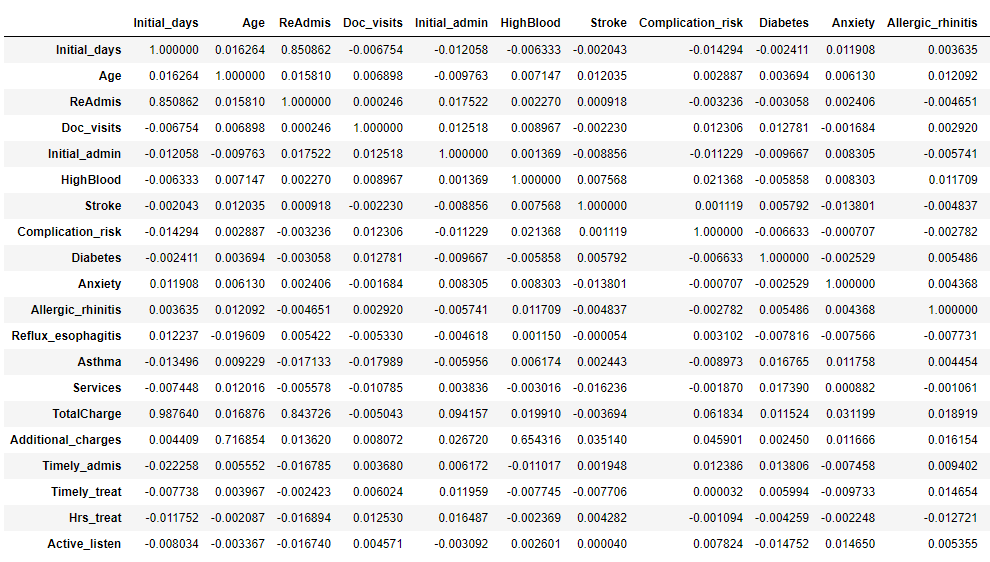
We removed minor observations/variables from the initial model. These included VitD\_levels, Income, Marital, Gender, Full\_meals\_eaten, vitD\_supp, and Soft\_drink. The initial model gave us the following equation:

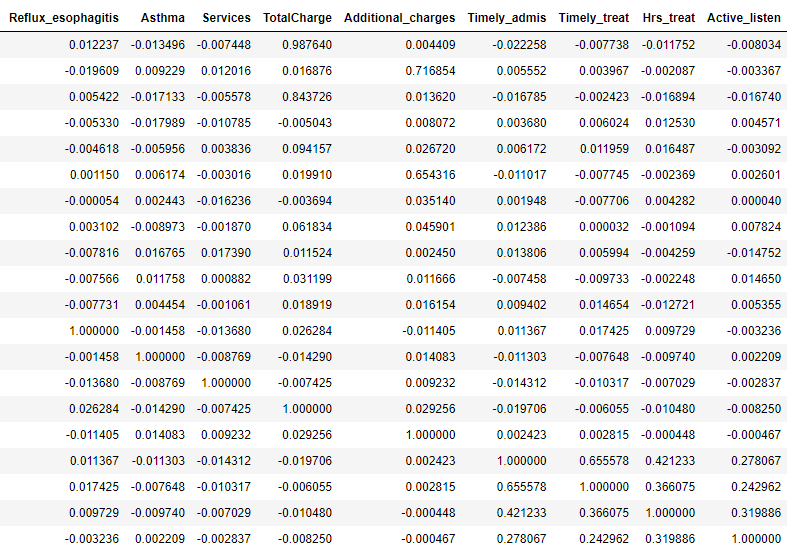
Ŷ= -14.5799 + 0.0197(Age) + 0.9935(ReAdmis) + 0.0197 (Doc\_visits) – 3.3450(Initial\_admin) – 0.4696(HighBlood) + 0.0806(Stroke) – 2.7188(Complication\_risk) – 0.9084(Diabetes) – 1.0138(Anxiety) – 0.8316(Allergic\_rhinitis) – 0.7705(Reflux\_esophagitis) + 0.0236(Asthma) + 0.0096(Services) + 0.0119(TotalCharge) – 9.473(Additional\_charges) – 0.0339(Timely\_admis) + 0.0089(Timely\_treat) + 0.0138(Hrs\_treat) + 0.0236(Active\_listen).

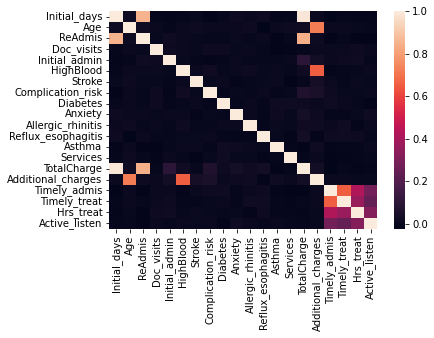
This equation is based on the standard multiple regression equation, according to Statistics Solutions (2010). This initial model has an R-squared value of 0.994, which means that 99.4% of the variation can be explained by this model. The large condition number suggests strong multicollinearity, and all the variables are not necessary for the reduced model. To identify which variables should be used in the reduced, we will use a heatmap and correlation matric to visualize where the multicollinearity occurs.

**2. Justification for statistically based variable selection procedure, and a model evaluation metric to reduce the initial model**

Here, we reduce the initial model in a way that better aligns with our proposed research question. To address the multicollinearity in the initial model, a correlation matrix and heatmap will be generated. The correlation matrix is used to identify which best variables are best for a reduced regression model. A heatmap will help to further visualize this data as well, and better determine which variables are causing the multicollinearity. The correlation matrix and heatmap are shown below:







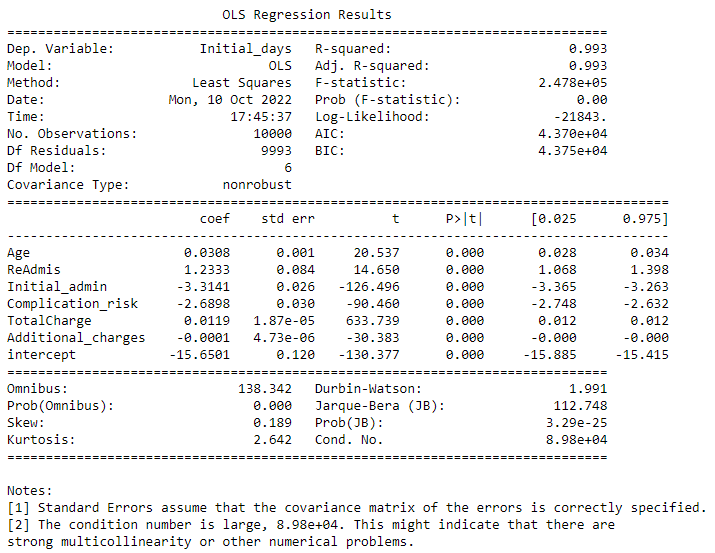
The correlation matrix and heatmap helped to identify which variables were not strong as predictors. Based on the heatmap ReAdmis and TotalCharge are strong predictors. To further narrow down the list to other potential variables, we’ll remove diagnosis and survey variables. Age and Complication\_risk are left because tend to lead to longer admission days. The new heatmap is shown below:



Even after removing the diagnosis and survey variables, ReAdmis and TotalCharge were still very strong predictors for most of the variance. A linear relationship was found between the number of initial days a patient was admitted and their potential readmission and high total charges. We will continue with the analysis and run a multiple linear regression model on these reduced variables, including both categorical and continuous variables.

**3. Reduced multiple regression model including both categorical and continuous variables**

We will run a reduced multiple regression model using the identified variables above. The reduced OLS Regression Results as identified in the correlation matrix and the heat map are shown below.



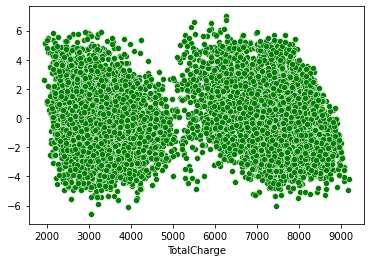
After conducting the regression, the reduced model still accounts for 99.3% of the variance. The final multiple linear regression model equation was produced:

Ŷ= -15.6501 + 0.0308(Age) + 1.2333(ReAdmis) – 3.3141(Initial\_admin) – 2.6898(Complication\_risk + 0.0119(TotalCharge) – 0.0001(Additional\_charges).

**E. Analysis of the data set using the reduced multiple regression model:**

**1. Data analysis process of comparing the initial and reduced multiple regression models,**

Variable selection was determined based on the results of the correlation matrix and the heatmap used to map out the variables. We seleteced the variables that had the highest correlation to our target variable, initial days. The model evaluation metric is shown above with the model equation and analysis, including the R-squared values. The residual plot for the model is shown below:



**2. Output and calculations of the analysis performed, including the model’s residual error.**

The output of the calculations as well as the model’s residual error is noted above in the tables and visualizations.

**3. Code used to support implementation of multiple regression models**

The full code for the project is provided at the end section **Part VI: Demonstration.** Furthermore, a pdf print of the Jupyter notebook used for running the python scripts is attached in task submission. The full code used is also provided as a txt document in the task submission as well.

**Part V: Data Summary and Implications**

**F. Summary of findings and assumptions**

**1. Results of data analysis**

After conducting the data analysis, this is the final multiple linear regression equation for the reduced model we generated:

Ŷ=-15.6501 + 0.0308(Age) + 1.2333(ReAdmis) – 3.3141(Initial\_admin) – 2.6898(Complication\_risk + 0.0119(TotalCharge) – 0.0001(Additional\_charges).

The coefficients of the statistically significant variables from the reduced model show a high correlation with the Initial\_days. The coefficients indicated that for every 1 unit of:

* Age – Initial\_days will increase by 0.0308 units
* ReAdmis – Initial\_days will increase by 1.2333 units
* Initial\_admin – Initial\_days will decrease by 3.3141 units
* Complication\_risk – Initial\_days will decrease by 2.6898 units
* TotalCharge – Initial\_days will increase by 0.0119 units
* Additional\_charges – Initial\_days will decrease by 0.0001 units

The p-values for variables listed above all are statistically significant at 0.000.

**Statistical and practical significance of the model:**

The multiple regression model was both statistically and and practically significant. We were able to determine which variables had the greatest impact on the initial days a patient is admitted. We were also able to see the how when these variables increase or decrease, the initial days will also increase or decrease. It identifies there is a strong correlation between a patient being readmitted and high total charges with the Initial days. The data analysis showed a strong correlation between patient readmissions and high total charges with the initial days. In practice, the hospital can now monitor patients based on these predictor variables and see if a patient is likely to be readmitted. The hospital can now predict that if a patient has been readmitted and has high total charges, then their initial days will likely be high. The analysis can also be used to predict that if a patient is going to be readmitted. If they have high total charges and had a high number of initial days, then they have a higher chance of being readmitted.

**Data analysis limitations:**

The data analysis was limited to six specific variables with a p-value of 0.000. While these variables were statistically significant, other variables did have a low p-value but were not used in the analysis. Furthermore, the independent variables and coefficients where chosen based on the target variable of initial days. If we were to select for a different target variable, then the independent variables and coefficients would have changed based on their relationship to the target. Thus, are additional potential models that could be created or used that we did not account for. Furthermore, it’s important to emphasis that these results are purely predictive. While there is strong confidence in the results, it’s always important to account for anomalies. Furthermore, the analysis is limited by the data provided, as there could be additional variables that were not accounted for during the surveys. These could range from genetic history, recent exposure to certain viruses, additional health complications, and so on. The data provided is what the hospital deemed most important, so we are limited to work within that given framework.

**2. Recommended course of action**

Based on the results of the analysis, the hospital could use the number of initial days a patient stays to reduce their chance of readmission and total charges. They could start by making sure the patient receives the most accurate diagnosis and procedures as soon as possible so that they don’t have to spend too much time at the hospital getting treated. Furthermore, the hospital could follow up on patients with a high number of initial days and create new plans and procedures for them to follow to reduce their chance of being readmitted. To address the limitations of the analysis, we could run additional multiple regressions based on other target variables. Furthermore, the hospital could continue to grow the sample size of the already determined variables via additional patient surveys. They can also include new variables to test for, such as additional health complications or exposure to viruses.

**Part VI: Demonstration**

**Link to the Panopto Video recording:**

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=97cef0a1-67cc-42bb-b337-af2b002a6213>

**Sources for third party code:**

**Code for importing packages, preparing data, and performing multiple regression:**

**GitHub:**

<https://github.com/JangirSumit/data_science/blob/master/22nd%20June%20Assignments/case%20study/questions.py>

<https://github.com/soroosh-rz/Data-Science/blob/master/linear%20regression/linear_regression_exercise.py>

**Kaggle:**

<https://www.kaggle.com/code/vipulgandhi/linear-regression/notebook>

**Code used to create the reduced multiple regression model**

<https://www.kaggle.com/code/divan0/multiple-linear-regression>

<https://www.statsmodels.org/dev/generated/statsmodels.regression.linear_model.OLS.html>

**References**

Statistics Solutions. (2021, August 11). Assumptions of multiple linear regression. Retrieved September 20, 2022, from <https://www.statisticssolutions.com/freeresources/directory-of-statistical-analyses/assumptions-of-multiple-linear-regression/>

Statistics Solutions. (2010, December 21). Multiple Regression. Retrieved September 11, 2022 from <https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/multiple-regression/#:~:text=The%20multiple%20regression%20equation%20explained,when%20the%20predictor%20variable%20changes>

**Full Code used for this Project**

#import packages and clean data before running multiple regression analysis. Rename the survey item variables accordingly.

import numpy as np

import pandas as pd

from sklearn import linear\_model

import matplotlib.pyplot as plt

import seaborn as sns

%matplotlib inline

pd.set\_option('display.max\_columns', None)

import pylab

from pylab import rcParams

import statsmodels.api as sm

import statistics

from scipy import stats

import sklearn

from sklearn import preprocessing

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import train\_test\_split

from sklearn import metrics

from sklearn.metrics import classification\_report

from scipy.stats import chisquare

from scipy.stats import chi2\_contingency

df = pd.read\_csv (r'C:\Users\fahim\Documents\0\_WGUDocuments\d208\1medical\_clean.csv')

df.rename(columns={'Item1':'Timely\_admis','Item2':'Timely\_treat',

'Item3':'Timely\_visits','Item4':'Reliability',

'Item5':'Options','Item6':'Hrs\_treat',

'Item7':'Courteous','Item8':'Active\_listen'},inplace=True)

df.head()

df.info()

#check if there are any missing data entries - if there are none then the output should be False

df.isna().any()

#check if there is any duplicate data entries present in columns

df[df.duplicated()

# check if there are any duplicated rows in the data set - if there are none then the output should be False

df.duplicated().any()

# remove demographic data from the data set since these entries won't be necessary for the multiple regression analysis

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

# check to make sure that the columns for demographic data were dropped before proceeding

df.head()

# convert categorical yes/no values to numeric 1/0 values

df = df.replace(to\_replace = ['Yes','No'],value = [1,0])

df

# convert the categorical variable of genders to a numeric variable

df['Gender'] = df['Gender'].replace(['Male','Female','Nonbinary'],[1,2,3])

df

# convert the non-married Marital status values to "Married/Not Married", then convert "Married/Not Married" to "1/0"

#this will make the Marital variable easier to work with during regression analysis

df['Marital'] = df['Marital'].replace(['Divorced','Widowed','Separated','Never Married'],'Not Married')

df['Marital'] = df['Marital'].replace(['Married','Not Married'],[1,0])

# convert the Initial\_Admin, Complication\_risk, and Services variables into integers before proceeding

df['Initial\_admin'] = df['Initial\_admin'].replace(['Elective Admission','Observation Admission','Emergency Admission'],[1,2,3])

df['Complication\_risk'] = df['Complication\_risk'].replace(['Low','Medium','High'],[1,2,3])

df['Services'] = df['Services'].replace(['Blood Work','CT Scan','Intravenous','MRI'],[1,2,3,4])

df.info()

df.describe()

my\_list = df.columns.values.tolist()

print(my\_list)

# move the chosen target variable "Initial\_days" to beginning of the columns

df=df[['Initial\_days','Children', 'Age', 'Income', 'Marital', 'Gender', 'ReAdmis', 'VitD\_levels', 'Doc\_visits', 'Full\_meals\_eaten', 'vitD\_supp', 'Soft\_drink', 'Initial\_admin', 'HighBlood', 'Stroke', 'Complication\_risk', 'Overweight', 'Arthritis', 'Diabetes', 'Hyperlipidemia', 'BackPain', 'Anxiety', 'Allergic\_rhinitis', 'Reflux\_esophagitis', 'Asthma', 'Services', 'TotalCharge', 'Additional\_charges', 'Timely\_admis', 'Timely\_treat', 'Timely\_visits', 'Reliability', 'Options', 'Hrs\_treat', 'Courteous', 'Active\_listen']]

# Confirm that the target variable was moved before exporting the prepared dataset

my\_list = df.columns.values.tolist()

print(my\_list)

# describe the dataframe to identify distribution of variables

df.describe()

# now that all the modifications have been made, export the prepared dataset

df.to\_csv(r'C:\Users\fahim\Documents\0\_WGUDocuments\d208\1medical\_clean-PREPAREDTASK1.csv')

# identify the columns for numerical data

NumericalData = df.select\_dtypes(include = "number").columns

print (NumericalData)

# retrieve the OLS Regression Results

print(lm\_initialdays.summary())

# to address the strong multicollinearity, create heatmap and correlation matrix

medical\_heatmap = df[['Initial\_days','Age', 'ReAdmis', 'Doc\_visits', 'Initial\_admin', 'HighBlood', 'Stroke', 'Complication\_risk', 'Diabetes','Anxiety', 'Allergic\_rhinitis', 'Reflux\_esophagitis', 'Asthma', 'Services', 'TotalCharge', 'Additional\_charges', 'Timely\_admis', 'Timely\_treat', 'Hrs\_treat','Active\_listen']]

#Initial model heatmap

sns.heatmap(medical\_heatmap.corr(), annot=False)

plt.show

medical\_heatmap.corr()

# to narrow the results, remove the diagnosis and survey variables and create a reduced initial model heatmap

medical\_heatmap = df[['Initial\_days','Age', 'ReAdmis', 'Initial\_admin','Complication\_risk', 'TotalCharge','Additional\_charges']]

sns.heatmap(medical\_heatmap.corr(), annot=True)

plt.show

# create the reduced multiple regression model

df['intercept'] = 1

lm\_initialdays\_reduced = sm.OLS(df['Initial\_days'],df[['Age', 'ReAdmis','Initial\_admin', 'Complication\_risk','TotalCharge', 'Additional\_charges','intercept']]).fit()

print(lm\_initialdays\_reduced.summary())

# load cleaned data to use for the residual plot; we will name the dataframe "regression\_df" to make it easier to keep track of

regression\_df = pd.read\_csv(r'C:\Users\fahim\Documents\0\_WGUDocuments\d208\1medical\_clean-PREPAREDTASK1.csv')

# create the residual plot

regression\_df['intercept'] = 1

residuals = regression\_df['Initial\_days'] - lm\_initialdays\_reduced.predict(regression\_df[['Age','ReAdmis','Initial\_admin','Complication\_risk', 'TotalCharge','Additional\_charges','intercept']])

sns.scatterplot(x=regression\_df['TotalCharge'],y=residuals,color='green')

plt.show();