# Predictive Capacity of Multiple Linear Regression for Patient Characteristics with Hospital Length of Stay

Joanne Senoren

Master of Science, Data Analytics

**Table of Contents**

## Part I: Research Question and Goals

### A1. Research Question

Hospital length of stay (LOS) is a primary measure used to determine the quality of care in hospitals worldwide. Other factors, such as readmission and mortality rate, are often used to determine areas of improvement across administration, patient care, facilities, and spread of resources (Lingsma et al., 2018). However, we should also consider specific patient-level factors to help guide a hospital's performance and treatment abilities.

This consideration leads us to this paper's research question: *What factors regarding a patient's hospital activities, medical history, and demographic background correlate with the length of their initial hospital stay?* Finding out which factors surrounding a patient's hospital stay can help direct the course of action across patient prioritization, improved care, and streamlined resources. Patients who must spend unnecessary additional days in the hospital are more at risk for medical complications. Additionally, prolonged hospital stays put more financial stress on the hospital and healthcare system. (Tipton et al., 2021)

### A2. Goals of Analysis

The overall goal of this analysis is to determine whether patient-level factors in the dataset influence a patient's hospital stay length. To reach this goal of this analysis, I will select which relevant variables in the dataset we can observe against 'Initial_days' (hospital stay) and develop a multiple linear regression model to specify which variables among those selected show a significant correlation with patient hospital stays. After generating a model, it will be scaled down to determine which combination of features can predict the length of hospital stay. Identifying these factors regarding patient demographics and medical information from the dataset can help increase efficiency in patient turnover, improve patient care, streamline hospital costs, and guide staffing decisions.

## Part II: Model and Programming Justifications

### B1. Four Assumptions of Multiple Linear Regression

The following are four assumptions of multiple linear regression: linearity, non-multicollinearity, homoscedasticity, and multivariate normality.

Linearity assumes a linear relationship between the response variable and the predictor variable. When selecting predictor variables, I established a loose correlation coefficient threshold for this analysis. Non-multicollinearity means that the model consists of predictor variables that are not highly correlated to each other to avoid redundancy that can affect coefficients, making predictions unreliable. Homoscedasticity demonstrates a constant variance among the residuals throughout the model, which helps us establish that the model is reliable. Multivariate normality is the assumption that the residuals in a linear regression model are normally distributed (Bobbitt, 2021). Whether the model meets these assumptions is discussed in the data analysis section of the paper.

### B2. Python Justification

My programming language of choice for this linear regression model is Python within the Jupyter Notebook environment. The first rationale for Python is that it is semantically easier to read than R. I also keep my code and annotations more organized with Python. By maintaining well-structured code, a reader can quickly understand what is happening in addition to the annotations that can help guide on what could happen next. The second justification for Python is my familiarity with several Python packages I benefitted from using for building this analysis. Packages such as NumPy, pandas, matplotlib, seaborn, and statsmodels were helpful in my analysis across data cleaning, preparation, wrangling, modeling, and feature selection. I used these packages repeatedly throughout the entire analytical process. Table 1 below demonstrates how each package contributes to the analysis goals, data cleaning, wrangling, and modeling.

Table 1

*Python Packages and Uses for Analysis*

| Package Name | Package Use in Analysis |
|---|---|
| NumPy | - used to work with arrays<br>- mathematical functions for calculations |
| pandas | - create DataFrames<br>- manipulate DataFrames<br>- generate statistical summaries |
| matplotlib | - establish and customize subplots for visualizations<br>- simple histograms and bar plots |
| seaborn | - customize boxplots, strip plot, heatmap, residual plot |
| statsmodels | - create a fitted regression model<br>- check VIF scores<br>- create a QQ plot |

### B3. Why Use Multiple Linear Regression Model for This Analysis

We can analyze a variety of continuous and categorical variables against the response variable 'Initial_days,' a continuous variable with the float data type. The float data type means that it is measured with any value within a range of any given values (Bruce et al., 2020). Multiple linear regression utilizes a specific calculation to determine if a combination of explanatory variables (continuous or categorical) can predict a continuous response variable. Since we are trying to find and select what categorical and continuous factors about a patient contribute to the continuous length of their hospital stay, multiple linear regression is an appropriate model to explore this type of analysis.

## Part III: Data Preparation

### C1. Data Cleaning Goals

The primary data cleaning goal for this dataset is to prepare the data for analysis by doing the following steps to eliminate data noise and gain more clarity about the dataset. Firstly, we should check for nulls and impute them if necessary (Figure 1). Second, duplicates should be addressed and removed as necessary (Figure 2). The third step would be to check the provided dictionary against the dataset to ensure the unique values in the variables are appropriate and make any necessary changes (Figure 3). Additional cleaning steps would include updating column names to match the dictionary to make the

dataset more precise (Figure 4). A more detailed look at the data also includes looking at outliers among continuous variables by generating visualizations and then individually considering each variable's outliers and whether to treat them based on realistic ranges (Figure 5). An example in Figure 6 shows how outliers for the variable 'Income' were examined.

These data preparation steps are essential tasks prior to predictive modeling, such as linear regression, because ensuring that the fitted model has accurate data is important. We gain an understanding of how reliable the model will be by addressing issues such as missing values, duplicates, and outliers. To see the annotated code used to clean the dataset in detail, please see the attached file named "D208_MultiLinearRegression.ipynb" and view the section with the header "Data Cleaning and Preparation."

Figure 1

*Code for Checking for Nulls in Dataset*

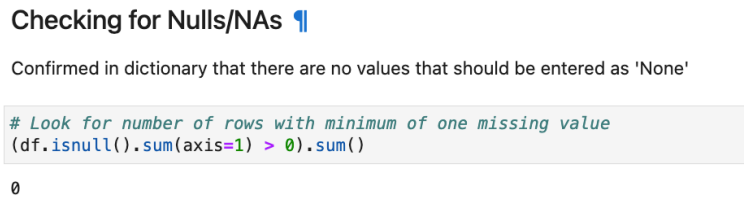



Figure 2

*Code for Checking Duplicates*

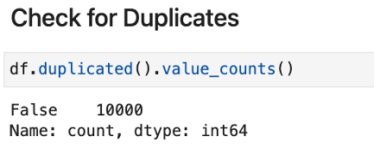



Figure 3

*Using .unique() to Check Input Values*

**Compare Unique Value Samples Against Dictionary**

```
# Check unique values against dictionary
for col in df:
    print(col, ':', df[col].unique())
```

Figure 4

*Code to Update Specific Column Names*

**Specify Survey Variable Names**

Change from 'Item...' to respective dictionary names

```
# Establish dictionary to map survey names
name_dict = {'Item1': 'TimelyAdmission', 'Item2':'TimelyTreatment', 'Item3':'TimelyVisits', 'Item4':'Reliability', 'Item5':'Options', 'Item6':'TreatmentHours',
             'Item7':'CourteousStaff', 'Item8':'ActiveListening'}

# Rename the dataframe and change the variable names so it persists
df.rename(columns=name_dict, inplace=True)

print(df.info())
```

Figure 5

*Code for Checking Outliers*

**Look at Outliers Among Continuous Variables**

Generate box plots to identify outliers quickly and then individually examine them to determine treatment.

```
cont_var = ['Population', 'Children', 'Age', 'Income', 'VitD_levels', 'Doc_visits', 'Full_meals_eaten',
            'vitD_supp', 'Initial_days', 'TotalCharge', 'Additional_charges']

# Set figure options
fig, axes = plt.subplots(4, 3, figsize=(20, 20))
fig.subplots_adjust(hspace=.5, wspace=.25)

# For loop over list of variables to generate box plots
for var, ax in zip(cont_var, axes.flat):
    ax.set_label(var)
    sns.boxplot(data=df, x=var, ax=ax)

axes[3,2].remove() # Removes unused chart
# Show figure
plt.show()
```

Figure 6

*Example Consideration for 'Income'*

```
### Income Outliers
This variable has the widest range of values with minimum of 154 and maximum of 207,249.
An explanation for the minimum income could be that patient has reported unemployment benefits.
On the maximum end, it's possible for someone to earn above 200,000.
The outliers are outside the upper limit so let's count how many there are.
```

```
# Gets Q1 and Q3 values
q1, q3 = np.percentile(df['Income'], [25, 75])

# Calculate interquartile range
iqr = q3 - q1

# Calculate upper limit
upper = q3 + (1.5 * iqr)

# Set variable for count
upper_count = 0

# Count outliers in Population
for x in df['Income']:
    if x > upper:
        upper_count += 1

print('Upper Outliers:', upper_count)

obs_perc = upper_count/len(df)
print('Percentage of dataset: {:.1%}'.format(obs_perc))
print(df['Income'].nlargest(n=upper_count))
```

```
Upper Outliers: 326
Percentage of dataset: 3.3%
8386    207249.10
841     204542.41
8598    203774.60
6406    197675.00
1778    197576.18
          ...
9791    106692.90
7391    106691.20
4027    106671.88
9596    106592.60
957     106521.92
Name: Income, Length: 326, dtype: float64
```

Outliers start at 106,521 to the maximum 207,249. These are realistic salaries and self-reported data. So we will keep Income the same.

## C2. Dependent and Independent Variables Summary Statistics

The following table (Table 2) shows the independent variables selected for this analysis. They are variables that describe a patient's demographic background, hospital interactions, and medical history. These will be modeled against the dependent variable, 'Initial_days,' which is the length of stay (in days) of the patient's first hospitalization.

Table 2

*Independent Variables Selected for MLR*

| Name | Variable Type | Data Type | Description |
|---|---|---|---|
| Age | Independent | Quantitative | patient's age upon admission |
| VitD_levels | Independent | Quantitative | patient's vitamin D levels (ng/ml) |
| TotalCharge | Independent | Quantitative | average per patient based on total charge and hospital stay |
| Doc_visits | Independent | Quantitative | number of times the doctor visited the patient during the hospital stay |
| Full_meals_eaten | Independent | Quantitative | number of full meals the patient ate during the hospital stay |
| Complication_risk | Independent | Categorical | level of patient complication risk |
| Initial_admin | Independent | Categorical | type of initial hospital admission |
| Services | Independent | Categorical | type of services the patient received while hospitalized |
| Soft_drink | Independent | Categorical | whether or not the patient drinks 3+ sodas a day |
| HighBlood | Independent | Categorical | Yes' or 'No' patient has high blood pressure |

| | | | |
|---|---|---|---|
| Overweight | Independent | Categorical | Yes' or 'No' patient is overweight |
| Arthritis | Independent | Categorical | Yes' or 'No' patient has arthritis |
| Diabetes | Independent | Categorical | Yes' or 'No' patient has diabetes |
| BackPain | Independent | Categorical | Yes' or 'No' patient has chronic back pain |
| Reflux_esophagitis | Independent | Categorical | Yes' or 'No' patient has reflux esophagitis |
| Asthma | Independent | Categorical | Yes' or 'No' patient has asthma |

Figure 7 below shows the summary statistics of the selected quantitative independent variables and the dependent variable.

Figure 7

*Summary Statistics for Dependent Variable and Continuous Independent Variables*

```
## Dependent Variable Summary Statistics
df[['Initial_days']].describe()
```

| | Initial_days |
|---|---|
| count | 10000.000000 |
| mean | 34.455299 |
| std | 26.309341 |
| min | 1.001981 |
| 25% | 7.896215 |
| 50% | 35.836244 |
| 75% | 61.161020 |
| max | 71.981490 |

```
df[['VitD_levels','TotalCharge','Age', 'Doc_visits', 'Full_meals_eaten']].describe()
```

| | VitD_levels | TotalCharge | Age | Doc_visits | Full_meals_eaten |
|---|---|---|---|---|---|
| count | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 |
| mean | 17.964262 | 5312.172769 | 53.511700 | 5.012200 | 1.001400 |
| std | 2.017231 | 2180.393838 | 20.638538 | 1.045734 | 1.008117 |
| min | 9.806483 | 1938.312067 | 18.000000 | 1.000000 | 0.000000 |
| 25% | 16.626439 | 3179.374015 | 36.000000 | 4.000000 | 0.000000 |
| 50% | 17.951122 | 5213.952000 | 53.000000 | 5.000000 | 1.000000 |
| 75% | 19.347963 | 7459.699750 | 71.000000 | 6.000000 | 2.000000 |
| max | 26.394449 | 9180.728000 | 89.000000 | 9.000000 | 7.000000 |

The figure above shows the summary statistics of the quantitative variables utilized in the model. The summary considers the 10,000 observations in the dataset as indicated in the 'count' row. The mean rows display the average value for each variable, the sum of all values divided by the number of rows. For example, the 'Initial_days' mean of 34.45 tells us that, on average, a patient in the dataset has had an initial hospitalization stay of about 34 days. The 'std' is the standard deviation or spread of the data around its mean. One standard deviation in 'Initial_days' is ~26 days, which tells us that a stay of 50 days is within one standard deviation. 'Min' and 'max' stand for minimum and maximum values, respectively. For 'Initial_days,' we see that the minimum value is 1 day of hospitalization, and the maximum value is about 72 days. The following univariate and bivariate sections provide additional summary statistics.

50% is our median, or middle value, about 36 days. The first and third quartiles are 25% and 75%, respectively. 50% of the data is inside the 25% and 75% markers. The other 50% are split in half outside the first and third quartiles. Since the first quartile has a value of 7.8, we can say that 25% of the data has a hospital stay of less than 7.8 days. The third quartile represents the higher values, so we can say that 25% of hospital stays are more than 61 days. The definitions apply to the rest of the numeric variables.

As the D208 PA step-by-step guide mentioned, the general statistics for categorical variables are generated in counts and percentages. Therefore, the figure below shows counts of each category per variable and their percentage within the dataset.

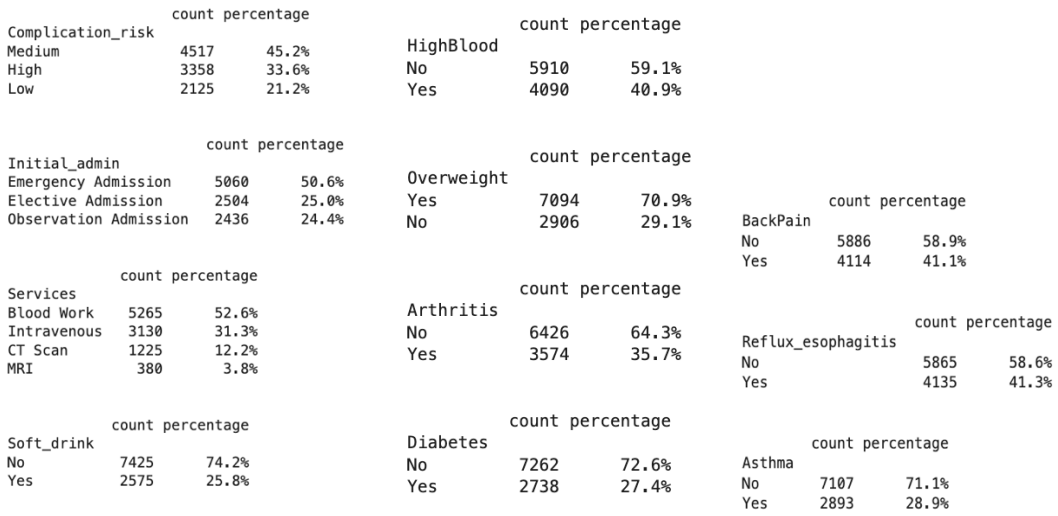
Figure 8

*Categorical Variables Summary Statistics*

```
category_vars = ['Complication_risk', 'Initial_admin', 'Services', 'Soft_drink', 'HighBlood', 'Overweight', 'Arthritis', 'Diabetes', 'BackPain',
    'Reflux_esophagitis', 'Asthma']

df[category_vars].describe()
```

| | Complication_risk | Initial_admin | Services | Soft_drink | HighBlood | Overweight | Arthritis | Diabetes | BackPain | Reflux_esophagitis | Asthma |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 | 10000 |
| **unique** | 3 | 3 | 4 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| **top** | Medium | Emergency Admission | Blood Work | No | No | Yes | No | No | No | No | No |
| **freq** | 4517 | 5060 | 5265 | 7425 | 5910 | 7094 | 6426 | 7262 | 5886 | 5865 | 7107 |

```
# Counts and Percentages
for col in df[category_vars]:
    counts = df[col].value_counts()

    percs = df[col].value_counts(normalize=True).mul(100).round(1).astype(str) + '%'

    print("\n")
    print(pd.concat([counts,percs], axis=1, keys=['count', 'percentage']))
```



| Complication_risk | count | percentage |
|---|---|---|
| Medium | 4517 | 45.2% |
| High | 3358 | 33.6% |
| Low | 2125 | 21.2% |

| Initial_admin | count | percentage |
|---|---|---|
| Emergency Admission | 5060 | 50.6% |
| Elective Admission | 2504 | 25.0% |
| Observation Admission | 2436 | 24.4% |

| Services | count | percentage |
|---|---|---|
| Blood Work | 5265 | 52.6% |
| Intravenous | 3130 | 31.3% |
| CT Scan | 1225 | 12.2% |
| MRI | 380 | 3.8% |

| Soft_drink | count | percentage |
|---|---|---|
| No | 7425 | 74.2% |
| Yes | 2575 | 25.8% |

| HighBlood | count | percentage |
|---|---|---|
| No | 5910 | 59.1% |
| Yes | 4090 | 40.9% |

| Overweight | count | percentage |
|---|---|---|
| Yes | 7094 | 70.9% |
| No | 2906 | 29.1% |

| Arthritis | count | percentage |
|---|---|---|
| No | 6426 | 64.3% |
| Yes | 3574 | 35.7% |

| Diabetes | count | percentage |
|---|---|---|
| No | 7262 | 72.6% |
| Yes | 2738 | 27.4% |

| BackPain | count | percentage |
|---|---|---|
| No | 5886 | 58.9% |
| Yes | 4114 | 41.1% |

| Reflux_esophagitis | count | percentage |
|---|---|---|
| No | 5865 | 58.6% |
| Yes | 4135 | 41.3% |

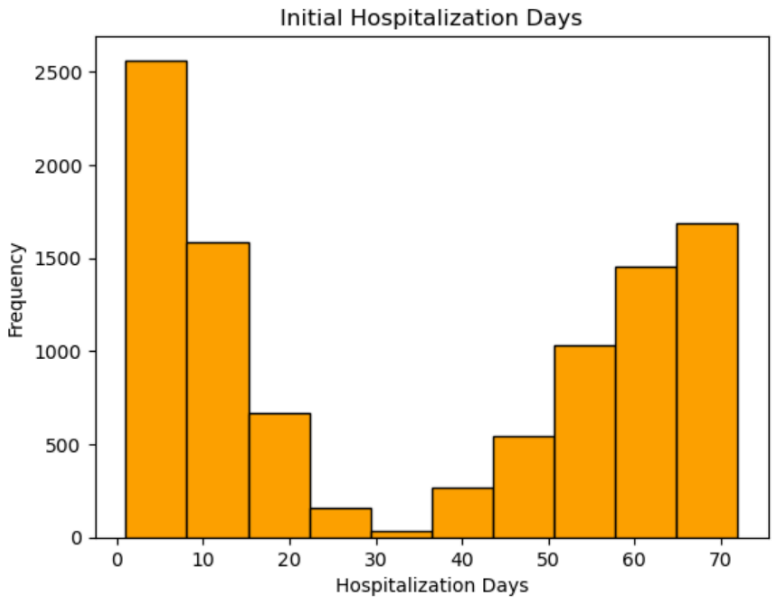
| Asthma | count | percentage |
|---|---|---|
| No | 7107 | 71.1% |
| Yes | 2893 | 28.9% |

The first chart in Figure 8 shows each variable's mode or most frequented category. The values in 'Soft_drink,' 'HighBlood,' 'Overweight,' 'Arthritis,' 'Diabetes,' 'BackPain,' 'Reflux_esophagitis,' and 'Asthma' are Boolean values consisting of Yes and No. It is interesting to note that the mode for these variables is mostly 'No' with an exception for Overweight, where 'Yes' is the mode. The third screenshot in Figure 8 goes over the percentages for each categorical variable in relation to the dataset. 'Initial_admin,' for example, tells us that 50.6% of hospitalized patients were admitted on an emergency admission status. The univariate section also reflects these percentages because of the limited statistics that categorical variables can provide.

### C3. Univariate and Bivariate Visualizations

Figure 9

*Univariate Statistic for Response Variable*




The variable 'Initial_days' and its univariate statistics are shown in Figure 9 above. The generated histogram shows that the dependent variable demonstrates a bimodal distribution with two peaks. This tells us that there are two distinct groups represented in the variable. Although we do not have more details about them, we can see that there are two sets of concentrations where many patients stay less than 10 days or more than 65 days. At this point, it's important to note that something else may be the cause of two subsets within one continuous variable. The histogram also tells us that the minimum and maximum values fall within two standard deviations and that no outliers exist.

Figure 10

*Univariate and Bivariate Statistics Continuous Explanatory Variables*

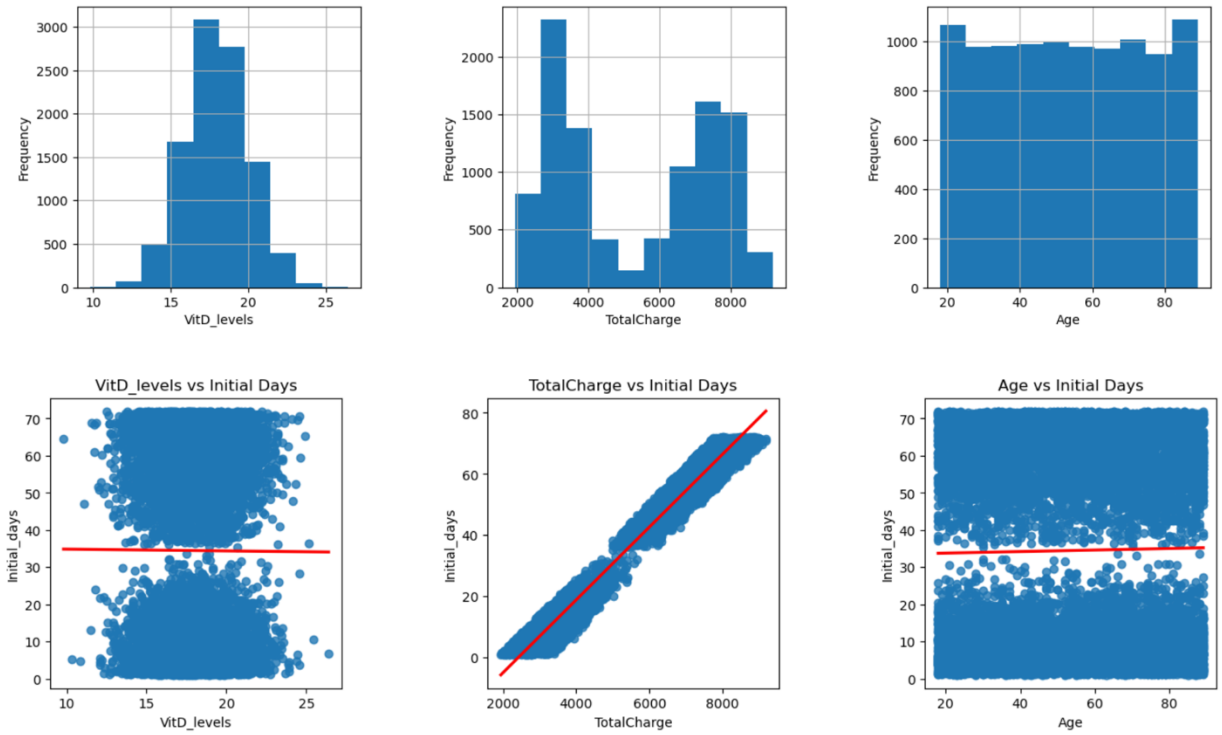
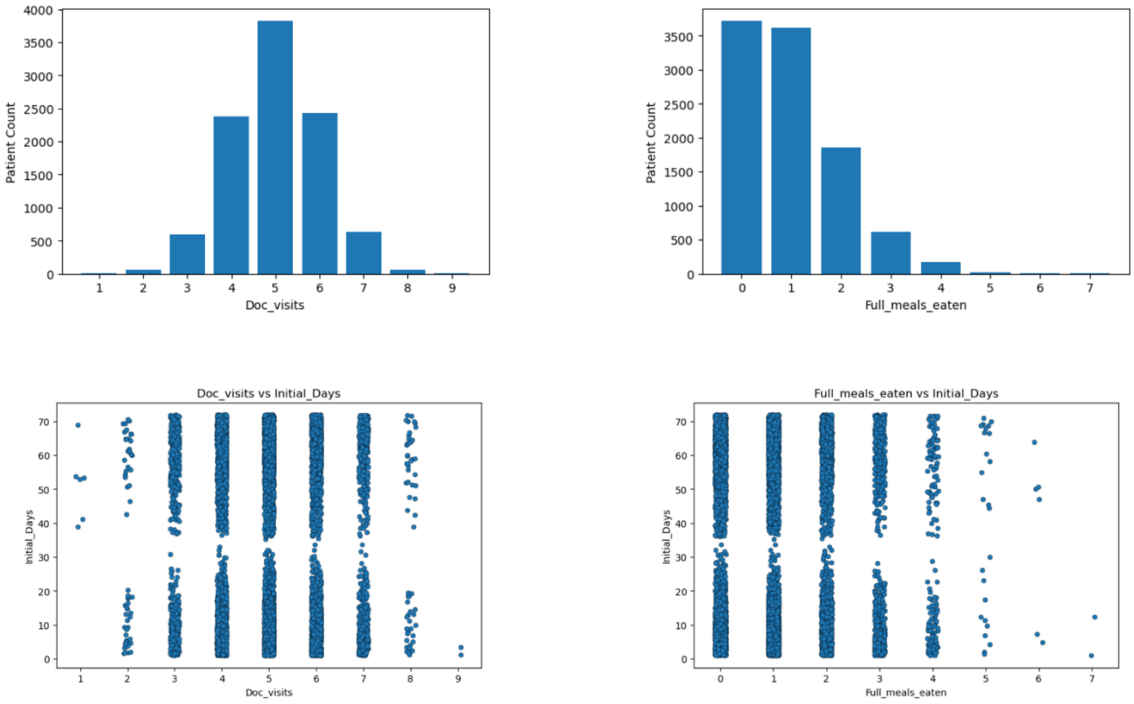



Figure 10 shows two rows of visualizations. The first row is the univariate visualizations for the continuous variables, and the second row consists of their bivariate visualizations against 'Initial_days.' The 'VitD_levels' univariate graph demonstrates a normal distribution due to the approximate bell-shaped curve of the data. 'TotalCharge' shows a bimodal distribution with two peaks on either end of the histogram. 'Age' has a uniform distribution, which tells us that the patients have nearly equal subsets of age groups.

The red line in the second row within the scatterplots is a regression line that plots the relationship between the response variable and explanatory variables. 'TotalCharge' demonstrates a potential linear relationship with 'Initial_days.' We also know that the univariate graphs for 'TotalCharge' and 'Initial_days' are bimodal, indicating their positive correlation. The other two scatterplots show a nearly flat regression line or minimum relationship.

Figure 11

*Univariate and Bivariate Statistics Discrete Explanatory Variables*




The first row in Figure 11 shows the univariate bar graphs for two selected discrete variables. The data in 'Doc_visits' is normally distributed, and most patients visit their doctor approximately five times during their hospitalization. 'Full_meals_eaten' is right-skewed or positively skewed, where more patients eat none, partial, or one meal daily. The bivariate strip plots on the second row show some interesting points. 'Doc_visits' against 'Initial_days' shows that most people who stay as little as 1 day or as much as 70 days are visited by a doctor 5 times. The lower number of visits demonstrates less distribution of patients. It's interesting to note that those who are only visited by a doctor once have stayed in the hospital for more than 30 days. It is also notable in the 'Full_meals_eaten' vs. 'Initial_days' strip plot that many patients, regardless of their hospital stay length, have not eaten full meals. An assumption would be that short-term patients do not get to eat full meals before discharge, and long-term patients get the time to eat more than one full meal. However, the graph doesn't support this theory. This tells me there might be something specific about the patients in the dataset.

Figure 12

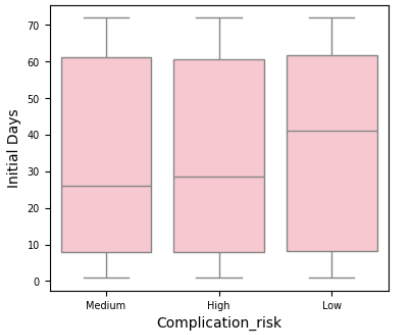
*Univariate Statistics Categorical Explanatory Variables*

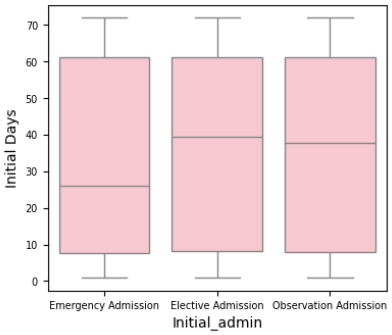
Figure 12 visualizes the summary statistics across the selected categorical variables in section C2. The percentages are showcased across the individual pie charts. We can quickly add additional details to the written summary statistics. For example, even though the summary statistic for 'HighBlood' told us that the mode is "No," we can easily see that the split between the patients who have and don't have high blood pressure is nearly fifty percent.

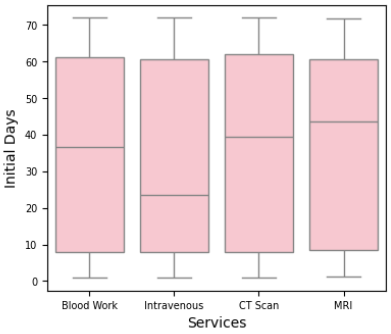
We can also better understand the non-Boolean categorical variables and note that across 'Complication_risk,' 'Initial_admin,' and 'Services,' the mode makes up approximately half of the patients in the data.

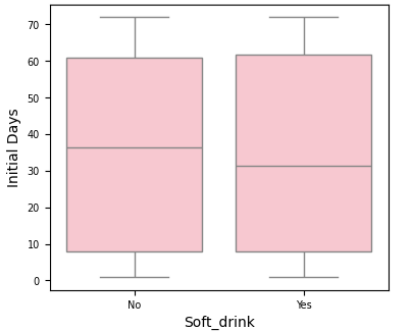
Figure 13

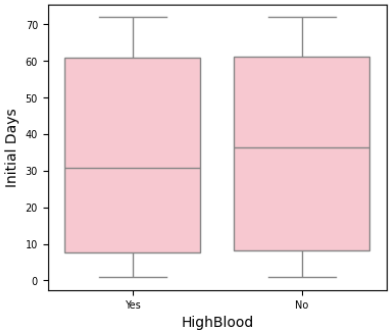
*Bivariate Statistics Categorical Explanatory Variables*

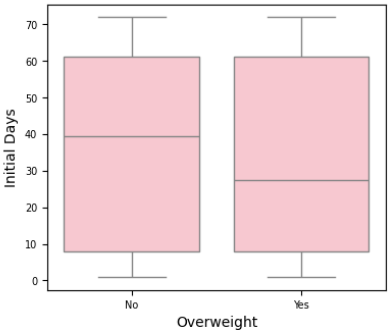

Initial Days
Medium
High
Low
Complication_risk

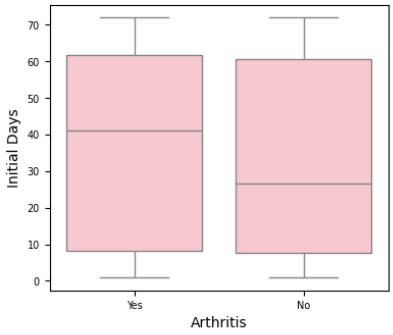

Initial Days
Emergency Admission
Elective Admission
Observation Admission
Initial_admin

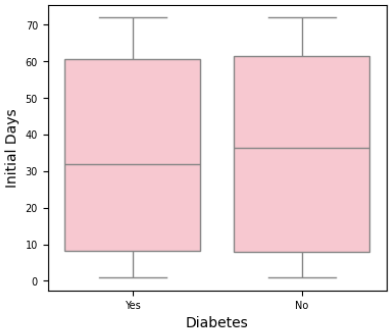

Initial Days
Blood Work
Intravenous
CT Scan
MRI
Services

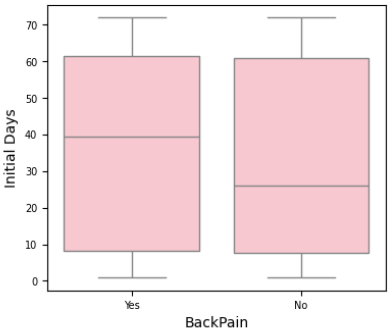

Initial Days
No
Yes
Soft_drink

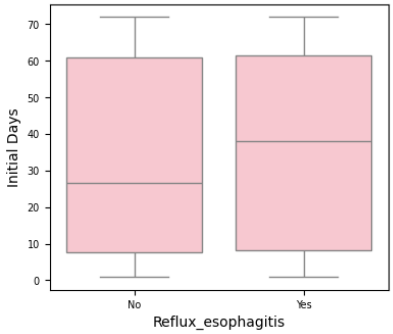

Initial Days
Yes
No
HighBlood


Initial Days
No
Yes
Overweight


Initial Days
Yes
No
Arthritis


Initial Days
Yes
No
Diabetes


Initial Days
Yes
No
BackPain


Initial Days
No
Yes
Reflux_esophagitis

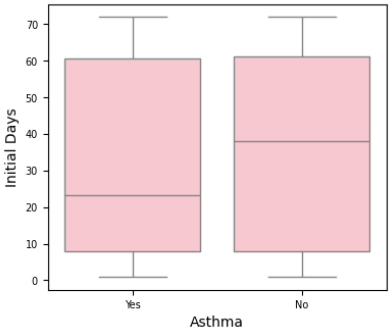

Initial Days
Yes
No
Asthma

Figure 13 above shows us the bivariate visualizations against 'Initial_days.' The boxplots add even more information to the data we are exploring by demonstrating the individual medians of each group within the categorical variable. We see some notable information here concerning hospital stay and the patient's medical history. Patients recorded as "No" across 'Soft_drink,' 'HighBlood,' 'Overweight,' and 'Asthma' have a median of over 30 days of hospital stay. Low 'Complication_risk' and elective 'Initial_admin' also show the highest median compared to the other categories in each variable. These groups are interesting because they display the opposite of another assumption: medical conditions, high complication risk, and emergency admission should demonstrate more extended hospital stays, given these factors could make treatment more complex and thus lengthy.

**C4. Data Transformation (Data Wrangling)**

Data wrangling across some variables must ensure an accurate multiple linear regression model (Middleton, 2022a). I transformed some of the selected categorical variables, which are included in Figure 14 and Figure 15.

I re-expressed Boolean input variables with 'Yes' and 'No' to 1 and 0, respectively. Qualitative variables are split into separate variables and then one-hot encoded to consist of numeric values, also known as creating dummy variables or indicator variables. I dropped one column for each one-hot encoded variable to eliminate redundancy in the model (Lingsma et al., 2018). See Figure 14 below for re-expression of categorical variables. Figure 15 shows the code for one-hot encoding to selected categorical variables. To see these data-wrangling actions in detail, refer to the "Data Wrangling" section in the Jupyter Notebook file.

Figure 14

*Re-expression of Categorical Variables Code and Output*

```
# New column names variable
new_col_list = []

# Function for looping through target columns to re-express 'Yes/No' to 1/0
for col in reexp_cols:
    # Make new name and duplicate original column
    new_col = col +'_numeric'
    df[new_col] = df[col]

    # Make dictionary to change boolean to numeric
    new_dict = {new_col: {'No': 0, 'Yes': 1}}

    # Change all values in duplicated columns to ordinal encoding
    df.replace(new_dict, inplace=True)

    # Add to new_col_list
    new_col_list.append(new_col)

# Check that numeric columns were created by filtering and printing them

numeric_cols = [var for var in df.columns if 'numeric' in var]
print(df[numeric_cols].info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 8 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Soft_drink_numeric         10000 non-null  int64
 1   HighBlood_numeric          10000 non-null  int64
 2   Overweight_numeric         10000 non-null  int64
 3   Arthritis_numeric          10000 non-null  int64
 4   Diabetes_numeric           10000 non-null  int64
 5   BackPain_numeric           10000 non-null  int64
 6   Reflux_esophagitis_numeric 10000 non-null  int64
 7   Asthma_numeric             10000 non-null  int64
dtypes: int64(8)
```

```
# Show unique values in new_col_list
for col in new_col_list:
    print(col, df[col].unique())
```

```
Soft_drink_numeric [0 1]
HighBlood_numeric [1 0]
Overweight_numeric [0 1]
Arthritis_numeric [1 0]
Diabetes_numeric [1 0]
BackPain_numeric [1 0]
Reflux_esophagitis_numeric [0 1]
Asthma_numeric [1 0]
```

Figure 15

*One-Hot Encoding of Categorical Variables Code and Output*

```
# Establish selection for dummy variable creation
target_dummy_vars = ['Initial_admin', 'Complication_risk', 'Services']
# Create dummy variables and drop one level to remove redundancy
dummies = pd.get_dummies(df[target_dummy_vars], prefix=target_dummy_vars, drop_first=True, dtype=int)
dummies.head()
```

| | Initial_admin_Emergency Admission | Initial_admin_Observation Admission | Complication_risk_Low | Complication_risk_Medium | Services_CT Scan | Services_Intravenous | Services_MRI |
|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| **1** | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| **2** | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| **3** | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| **4** | 0 | 0 | 1 | 0 | 1 | 0 | 0 |

```
# Add dummies columns to df and make change permanent by reassigning df
df = pd.concat([df, dummies], axis=1)
```

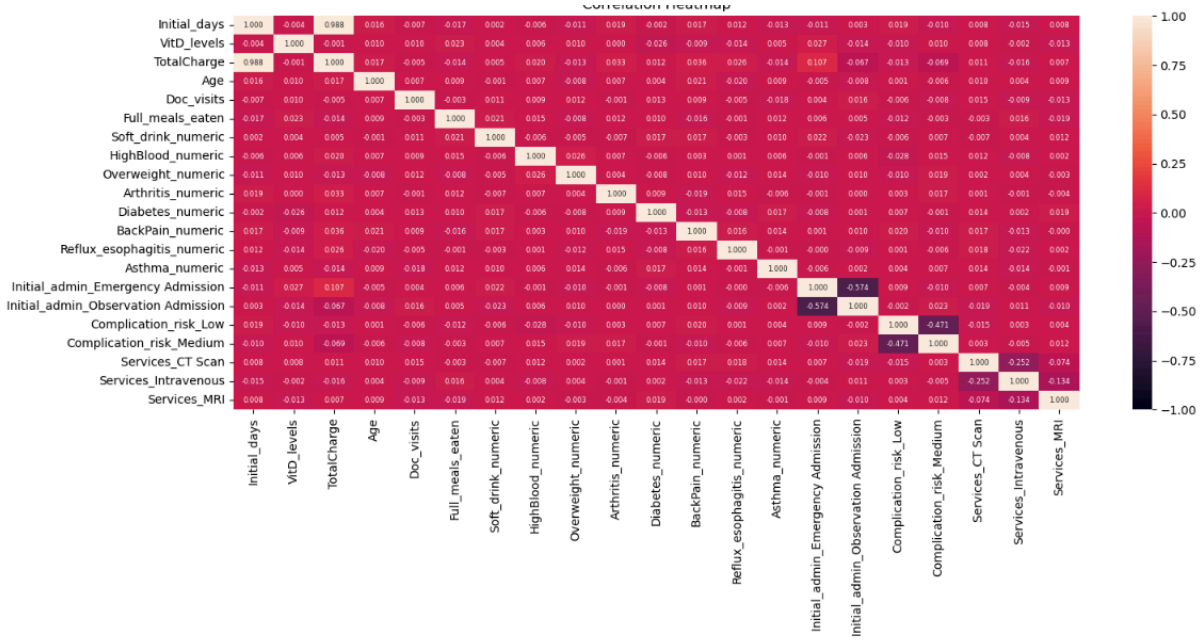
```
# Create array for generated dummy variables and check that they made it to the dataframe
dummy_vars = ['Initial_admin_Emergency Admission', 'Initial_admin_Observation Admission', 'Complication_risk_Low', 'Complication_risk_Medium',
              'Services_CT Scan', 'Services_Intravenous', 'Services_MRI']

print(df[dummy_vars].info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 7 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   Initial_admin_Emergency Admission   10000 non-null  int64
 1   Initial_admin_Observation Admission  10000 non-null  int64
 2   Complication_risk_Low               10000 non-null  int64
 3   Complication_risk_Medium            10000 non-null  int64
 4   Services_CT Scan                    10000 non-null  int64
 5   Services_Intravenous                10000 non-null  int64
 6   Services_MRI                        10000 non-null  int64
```

After cleaning and wrangling the selected data, I created a correlation matrix of dependent and independent variables to check the linearity assumption. Figure 16 below shows the correlation matrix that indicates that many of the patient-level variables do not show strong linear correlations with 'Initial_days' except for 'TotalCharge' where the correlation coefficient is .99. For the sake of the exercise, we will move forward with these knowing these variables have violated the assumption of linearity.

Figure 16

*Correlation Matrix*




## C5. CSV File of Prepared Data

Please refer to the generated CSV file titled 'Cleaned_Wrangled_MLR_Dataframe.'

## Part IV: Model Comparison and Analysis

### D1. Initial Model

Figure 17

*Model A: Initial Multiple Linear Regression Model Code & Summary*

```
# Establish the explanatory columns
expl_cols = df[['VitD_levels','TotalCharge','Age', 'Doc_visits', 'Full_meals_eaten', 'Complication_risk_Low',
                'Complication_risk_Medium', 'Initial_admin_Emergency Admission', 'Initial_admin_Observation Admission',
                'Services_CT Scan', 'Services_Intravenous', 'Services_MRI', 'Soft_drink_numeric', 'HighBlood_numeric',
                'Overweight_numeric', 'Arthritis_numeric', 'Diabetes_numeric', 'BackPain_numeric', 'Reflux_esophagitis_numeric',
                'Asthma_numeric']]
```

```
y = df[['Initial_days']]
X = expl_cols.assign(const=1)

model_A = sm.OLS(y, X).fit()

print(model_A.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:           Initial_days   R-squared:                       0.999
Model:                            OLS   Adj. R-squared:                  0.999
Method:                 Least Squares   F-statistic:                 5.271e+05
Date:                Mon, 12 Aug 2024   Prob (F-statistic):               0.00
Time:                        15:57:12   Log-Likelihood:                -12070.
No. Observations:               10000   AIC:                         2.418e+04
Df Residuals:                    9979   BIC:                         2.433e+04
Df Model:                          20
Covariance Type:            nonrobust
==============================================================================================
                                        coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------------
VitD_levels                           0.0024      0.004      0.599      0.549      -0.005       0.010
TotalCharge                           0.0122   3.76e-06   3242.730      0.000       0.012       0.012
Age                                  -0.0004      0.000     -1.066      0.287      -0.001       0.000
Doc_visits                            0.0135      0.008      1.746      0.081      -0.002       0.029
Full_meals_eaten                     -0.0100      0.008     -1.238      0.216      -0.026       0.006
Complication_risk_Low                 5.0380      0.023    223.842      0.000       4.994       5.082
Complication_risk_Medium              5.0346      0.019    271.520      0.000       4.998       5.071
Initial_admin_Emergency Admission    -6.2691      0.020   -315.380      0.000      -6.308      -6.230
Initial_admin_Observation Admission   0.0213      0.023      0.922      0.357      -0.024       0.067
Services_CT Scan                      0.0028      0.026      0.108      0.914      -0.048       0.053
Services_Intravenous                  0.0013      0.018      0.072      0.943      -0.035       0.037
Services_MRI                          0.0292      0.043      0.678      0.498      -0.055       0.114
Soft_drink_numeric                   -0.0246      0.019     -1.328      0.184      -0.061       0.012
HighBlood_numeric                    -1.3766      0.016    -83.457      0.000      -1.409      -1.344
Overweight_numeric                    0.0156      0.018      0.874      0.382      -0.019       0.051
Arthritis_numeric                    -0.8867      0.017    -52.397      0.000      -0.920      -0.854
Diabetes_numeric                     -0.9326      0.018    -51.270      0.000      -0.968      -0.897
BackPain_numeric                     -1.0486      0.016    -63.570      0.000      -1.081      -1.016
Reflux_esophagitis_numeric           -0.7150      0.016    -43.416      0.000      -0.747      -0.683
Asthma_numeric                       -0.0050      0.018     -0.279      0.781      -0.040       0.030
const                               -28.7021      0.091   -315.271      0.000     -28.881     -28.524
==============================================================================
Omnibus:                      481.369   Durbin-Watson:                   1.975
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              383.476
Skew:                          -0.396   Prob(JB):                     5.36e-84
Kurtosis:                       2.459   Cond. No.                     6.48e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.48e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

The model (Figure 17) using the Ordinary Least Squares method (OLS) shows an adjusted R-value of .999, which indicates that 99% of the variance of 'Initial_days' is explained by the combination of the variables. Two things stand out upon viewing the model summary. First, we see a multicollinearity error telling us we must check which variables are causing it. Multicollinearity among the explanatory variables affects the quality of the model. Second, we see that $R^2$ and adjusted $R^2$ are the same values. We set an alpha of 0.05 and see several of the variables' p-values are above the significance threshold.

Before reducing the model with backward stepwise elimination, we could check for multicollinearity using each variable's variance inflation factor (VIF). Removing these variables recursively by removing one at a time, starting with the highest VIF, will help us meet the non-multicollinearity assumption of a linear regression model. Addressing multicollinearity is essential because correlated predictor variables can cause unreliable coefficients and make models unstable (Larose & Larose, 2019). Figure 18 below shows the code and output for the variable inflation factors across all our explanatory variables. We set a VIF threshold of 10 to address severe multicollinearity, and anything over 5 and under 10 presents moderate collinearity. 'VitD_levels' and 'Doc_visits' have the highest VIFs. To see the recursion outputs, please refer to the "Addressing Multicollinearity" section in the Jupyter notebook titled "D208_MultiLinearRegression.ipynb." After recursively removing 'VitD_levels' and 'Doc_visits,' we have the VIFs in Figure 19.

Figure 18

*First VIF Code and Output*

```
from statsmodels.stats.outliers_influence import variance_inflation_factor

vif_check = expl_cols

X = vif_check

# Create dataframe to load variable name and VIF scores
vif_df = pd.DataFrame()

# Create column called feature to list all variables
vif_df["feature"] = X.columns
# Creates column called 'VIF' to list all corresponding VIF scores
vif_df["VIF"] = [variance_inflation_factor(X.values, i)
for i in range(len(X.columns))]


print(vif_df)
```

```
                              feature        VIF
0                         VitD_levels  31.116574
1                         TotalCharge   6.769467
2                                 Age   7.376655
3                          Doc_visits  19.882159
4                    Full_meals_eaten   1.983054
5                Complication_risk_Low   1.619661
6             Complication_risk_Medium   2.327733
7     Initial_admin_Emergency Admission   3.016369
8   Initial_admin_Observation Admission   1.952083
9                     Services_CT Scan   1.233109
10                Services_Intravenous   1.585808
11                        Services_MRI   1.072191
12                  Soft_drink_numeric   1.347784
13                   HighBlood_numeric   1.691125
14                  Overweight_numeric   3.392016
15                   Arthritis_numeric   1.556067
16                    Diabetes_numeric   1.375266
17                    BackPain_numeric   1.700139
18          Reflux_esophagitis_numeric   1.695831
19                       Asthma_numeric   1.405584
```

Figure 19

*VIFs After Removing 'VitD_levels' and 'Doc_visits'*

```
max_val = vif_df['VIF'].max()
max_name = vif_df['feature'][vif_df['VIF'] == max_val].values[0]
max_index = vif_df['feature'][vif_df['VIF'] == max_val].index[0]
print(max_val, max_name, max_index)


vif_df.drop(max_index, inplace=True)
print(vif_df)
```

```
19.882158972165122 Doc_visits 3
                                feature       VIF
1                           TotalCharge  6.769467
2                                   Age  7.376655
4                         Full_meals_eaten  1.983054
5                    Complication_risk_Low  1.619661
6                 Complication_risk_Medium  2.327733
7     Initial_admin_Emergency Admission  3.016369
8   Initial_admin_Observation Admission  1.952083
9                         Services_CT Scan  1.233109
10                   Services_Intravenous  1.585808
11                           Services_MRI  1.072191
12                    Soft_drink_numeric  1.347784
13                     HighBlood_numeric  1.691125
14                    Overweight_numeric  3.392016
15                     Arthritis_numeric  1.556067
16                     Diabetes_numeric  1.375266
17                     BackPain_numeric  1.700139
18           Reflux_esophagitis_numeric  1.695831
19                         Asthma_numeric  1.405584
```

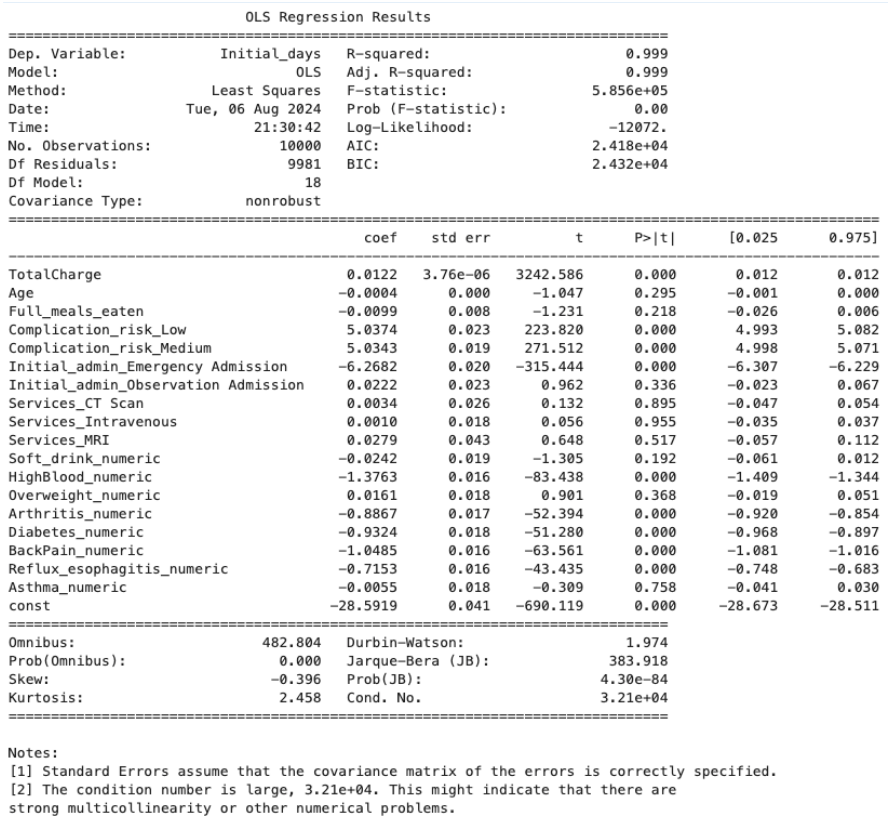
The output above shows moderate multicollinearity between 'TotalCharge' and 'Age.' Since the VIF threshold we set is 10 and these are below 10, we will keep them for now. Most notably, we know that 'TotalCharge' has a high linear correlation with 'Initial_days,' so keeping it in the model is useful.

**D2. Linear Model Reduction Justification**

After fitting the reduced set of variables with a new model, we get the summary in Figure 20.

Figure 20

*Model After VIF Reduction*



OLS Regression Results

| | | | |
|---|---|---|---|
| Dep. Variable: | Initial_days | R-squared: | 0.999 |
| Model: | OLS | Adj. R-squared: | 0.999 |
| Method: | Least Squares | F-statistic: | 5.856e+05 |
| Date: | Tue, 06 Aug 2024 | Prob (F-statistic): | 0.00 |
| Time: | 21:30:42 | Log-Likelihood: | -12072. |
| No. Observations: | 10000 | AIC: | 2.418e+04 |
| Df Residuals: | 9981 | BIC: | 2.432e+04 |
| Df Model: | 18 | | |
| Covariance Type: | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| TotalCharge | 0.0122 | 3.76e-06 | 3242.586 | 0.000 | 0.012 | 0.012 |
| Age | -0.0004 | 0.000 | -1.047 | 0.295 | -0.001 | 0.000 |
| Full_meals_eaten | -0.0099 | 0.008 | -1.231 | 0.218 | -0.026 | 0.006 |
| Complication_risk_Low | 5.0374 | 0.023 | 223.820 | 0.000 | 4.993 | 5.082 |
| Complication_risk_Medium | 5.0343 | 0.019 | 271.512 | 0.000 | 4.998 | 5.071 |
| Initial_admin_Emergency Admission | -6.2682 | 0.020 | -315.444 | 0.000 | -6.307 | -6.229 |
| Initial_admin_Observation Admission | 0.0222 | 0.023 | 0.962 | 0.336 | -0.023 | 0.067 |
| Services_CT Scan | 0.0034 | 0.026 | 0.132 | 0.895 | -0.047 | 0.054 |
| Services_Intravenous | 0.0010 | 0.018 | 0.056 | 0.955 | -0.035 | 0.037 |
| Services_MRI | 0.0279 | 0.043 | 0.648 | 0.517 | -0.057 | 0.112 |
| Soft_drink_numeric | -0.0242 | 0.019 | -1.305 | 0.192 | -0.061 | 0.012 |
| HighBlood_numeric | -1.3763 | 0.016 | -83.438 | 0.000 | -1.409 | -1.344 |
| Overweight_numeric | 0.0161 | 0.018 | 0.901 | 0.368 | -0.019 | 0.051 |
| Arthritis_numeric | -0.8867 | 0.017 | -52.394 | 0.000 | -0.920 | -0.854 |
| Diabetes_numeric | -0.9324 | 0.018 | -51.280 | 0.000 | -0.968 | -0.897 |
| BackPain_numeric | -1.0485 | 0.016 | -63.561 | 0.000 | -1.081 | -1.016 |
| Reflux_esophagitis_numeric | -0.7153 | 0.016 | -43.435 | 0.000 | -0.748 | -0.683 |
| Asthma_numeric | -0.0055 | 0.018 | -0.309 | 0.758 | -0.041 | 0.030 |
| const | -28.5919 | 0.041 | -690.119 | 0.000 | -28.673 | -28.511 |

| | | | |
|---|---|---|---|
| Omnibus: | 482.804 | Durbin-Watson: | 1.974 |
| Prob(Omnibus): | 0.000 | Jarque-Bera (JB): | 383.918 |
| Skew: | -0.396 | Prob(JB): | 4.30e-84 |
| Kurtosis: | 2.458 | Cond. No. | 3.21e+04 |

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.21e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

The figure above shows that we have reduced multicollinearity yet still get an error. Since 'TotalCharge' and 'Age' have VIFs of about 7, we can move on from the multicollinearity note and proceed with model reduction. 'Age' has a high p-value, so that we can remove it during the backward stepwise elimination process.

Since we have identified p-values higher than the established alpha of 0.05, we can perform a feature selection method. My choice of model reduction is to do backward stepwise elimination and use the p-value as a guide for the inclusion or exclusion of variables (Larose & Larose, 2019). Backward stepwise elimination removes one variable with a high p-value, re-runs the model, and identifies how the model improves or changes. Reducing variables in the model avoids overfitting with too many parameters that can cause noise with a different dataset instead of the training dataset (Larose & Larose, 2019). We can identify and select only the statistically relevant variables. You can see the recursive outputs of the

backward stepwise elimination process by referring to the "Backward Stepwise Elimination" section in the Jupyter Notebook titled "D208_MultiLinearRegression.ipynb."

### D3. Reduced Model

Figure 21 below shows the reduced model after performing backward stepwise elimination. The Model B has eight variables plus the constant. The adjusted $R^2$ value stayed the same at .99.

Figure 21

*Model B: After Backward Stepwise Elimination*

OLS Regression Results

| | | | |
|---|---|---|---|
| Dep. Variable: | Initial_days | R-squared: | 0.999 |
| Model: | OLS | Adj. R-squared: | 0.999 |
| Method: | Least Squares | F-statistic: | 1.171e+06 |
| Date: | Wed, 07 Aug 2024 | Prob (F-statistic): | 0.00 |
| Time: | 17:19:21 | Log-Likelihood: | -12075. |
| No. Observations: | 10000 | AIC: | 2.417e+04 |
| Df Residuals: | 9990 | BIC: | 2.424e+04 |
| Df Model: | 9 | | |
| Covariance Type: | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| TotalCharge | 0.0122 | 3.76e-06 | 3244.808 | 0.000 | 0.012 | 0.012 |
| Complication_risk_Low | 5.0384 | 0.022 | 223.993 | 0.000 | 4.994 | 5.082 |
| Complication_risk_Medium | 5.0354 | 0.019 | 271.799 | 0.000 | 4.999 | 5.072 |
| Initial_admin_Emergency Admission | -6.2797 | 0.016 | -385.404 | 0.000 | -6.312 | -6.248 |
| HighBlood_numeric | -1.3761 | 0.016 | -83.486 | 0.000 | -1.408 | -1.344 |
| Arthritis_numeric | -0.8869 | 0.017 | -52.419 | 0.000 | -0.920 | -0.854 |
| Diabetes_numeric | -0.9332 | 0.018 | -51.363 | 0.000 | -0.969 | -0.898 |
| BackPain_numeric | -1.0486 | 0.016 | -63.635 | 0.000 | -1.081 | -1.016 |
| Reflux_esophagitis_numeric | -0.7152 | 0.016 | -43.461 | 0.000 | -0.747 | -0.683 |
| const | -28.6076 | 0.029 | -993.699 | 0.000 | -28.664 | -28.551 |

| | | | |
|---|---|---|---|
| Omnibus: | 482.377 | Durbin-Watson: | 1.974 |
| Prob(Omnibus): | 0.000 | Jarque-Bera (JB): | 384.450 |
| Skew: | -0.397 | Prob(JB): | 3.29e-84 |
| Kurtosis: | 2.459 | Cond. No. | 2.35e+04 |

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.35e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

The figure above shows a multicollinearity note, and rechecking the variance inflation factors shows that the constant has a high VIF (Figure 22). Since we must keep the Intercept ('const') in the model so that the fitted line is not biased, we ignore the multicollinearity note this time.

Figure 22

*Final Multicollinearity Check*

```
# Create dataframe to load variable name and VIF scores
vif_df = pd.DataFrame()

# Create column called feature to list all variables
vif_df["feature"] = X.columns
# Creates column called 'VIF' to list all corresponding VIF scores

vif_df['VIF'] = [variance_inflation_factor(X.values, i) for i in range(len(X.columns))]

# vif_df = vif_df.set_index('feature')
print(vif_df)
```

| | feature | VIF |
|---|---|---|
| 0 | TotalCharge | 1.023009 |
| 1 | Complication_risk_Low | 1.290980 |
| 2 | Complication_risk_Medium | 1.296056 |
| 3 | Initial_admin_Emergency Admission | 1.011844 |
| 4 | HighBlood_numeric | 1.001273 |
| 5 | Arthritis_numeric | 1.002434 |
| 6 | Diabetes_numeric | 1.000648 |
| 7 | BackPain_numeric | 1.002556 |
| 8 | Reflux_esophagitis_numeric | 1.001217 |
| 9 | const | 12.637069 |

## E1. Models Comparison

Figure 23

*Model A (Initial) vs. Model B (Reduced)*

OLS Regression Results

| | | | |
|---|---|---|---|
| Dep. Variable: | Initial_days | R-squared: | 0.999 |
| Model: | OLS | Adj. R-squared: | 0.999 |
| Method: | Least Squares | F-statistic: | 5.271e+05 |
| Date: | Thu, 01 Aug 2024 | Prob (F-statistic): | 0.00 |
| Time: | 00:17:00 | Log-Likelihood: | -12070. |
| No. Observations: | 10000 | AIC: | 2.418e+04 |
| Df Residuals: | 9979 | BIC: | 2.433e+04 |
| Df Model: | 20 | | |
| Covariance Type: | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| VitD_levels | 0.0024 | 0.004 | 0.599 | 0.549 | -0.005 | 0.010 |
| TotalCharge | 0.0122 | 3.76e-06 | 3242.730 | 0.000 | 0.012 | 0.012 |
| Age | -0.0004 | 0.000 | -1.066 | 0.287 | -0.001 | 0.000 |
| Doc_visits | 0.0135 | 0.008 | 1.746 | 0.081 | -0.002 | 0.029 |
| Full_meals_eaten | -0.0100 | 0.008 | -1.238 | 0.216 | -0.026 | 0.006 |
| Complication_risk_Low | 5.0380 | 0.023 | 223.842 | 0.000 | 4.994 | 5.082 |
| Complication_risk_Medium | 5.0346 | 0.019 | 271.520 | 0.000 | 4.998 | 5.071 |
| Initial_admin_Emergency Admission | -6.2691 | 0.020 | -315.380 | 0.000 | -6.308 | -6.230 |
| Initial_admin_Observation Admission | 0.0213 | 0.023 | 0.922 | 0.357 | -0.024 | 0.067 |
| Services_CT Scan | 0.0028 | 0.026 | 0.108 | 0.914 | -0.048 | 0.053 |
| Services_Intravenous | 0.0013 | 0.018 | 0.072 | 0.943 | -0.035 | 0.037 |
| Services_MRI | 0.0292 | 0.043 | 0.678 | 0.498 | -0.055 | 0.114 |
| Soft_drink_numeric | -0.0246 | 0.019 | -1.328 | 0.184 | -0.061 | 0.012 |
| HighBlood_numeric | -1.3766 | 0.016 | -83.457 | 0.000 | -1.409 | -1.344 |
| Overweight_numeric | 0.0156 | 0.018 | 0.874 | 0.382 | -0.019 | 0.051 |
| Arthritis_numeric | -0.8867 | 0.017 | -52.397 | 0.000 | -0.920 | -0.854 |
| Diabetes_numeric | -0.9326 | 0.018 | -51.270 | 0.000 | -0.968 | -0.897 |
| BackPain_numeric | -1.0486 | 0.016 | -63.570 | 0.000 | -1.081 | -1.016 |
| Reflux_esophagitis_numeric | -0.7150 | 0.016 | -43.416 | 0.000 | -0.747 | -0.683 |
| Asthma_numeric | -0.0050 | 0.018 | -0.279 | 0.781 | -0.040 | 0.030 |
| const | -28.7021 | 0.091 | -315.271 | 0.000 | -28.881 | -28.524 |

| | | | |
|---|---|---|---|
| Omnibus: | 481.369 | Durbin-Watson: | 1.975 |
| Prob(Omnibus): | 0.000 | Jarque-Bera (JB): | 383.476 |
| Skew: | -0.396 | Prob(JB): | 5.36e-84 |
| Kurtosis: | 2.459 | Cond. No. | 6.48e+04 |

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.48e+04. This might indicate that there are strong multicollinearity or other numerical problems.

OLS Regression Results

| | | | |
|---|---|---|---|
| Dep. Variable: | Initial_days | R-squared: | 0.999 |
| Model: | OLS | Adj. R-squared: | 0.999 |
| Method: | Least Squares | F-statistic: | 1.171e+06 |
| Date: | Wed, 07 Aug 2024 | Prob (F-statistic): | 0.00 |
| Time: | 17:19:21 | Log-Likelihood: | -12075. |
| No. Observations: | 10000 | AIC: | 2.417e+04 |
| Df Residuals: | 9990 | BIC: | 2.424e+04 |
| Df Model: | 9 | | |
| Covariance Type: | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| TotalCharge | 0.0122 | 3.76e-06 | 3244.808 | 0.000 | 0.012 | 0.012 |
| Complication_risk_Low | 5.0384 | 0.022 | 223.993 | 0.000 | 4.994 | 5.082 |
| Complication_risk_Medium | 5.0354 | 0.019 | 271.799 | 0.000 | 4.999 | 5.072 |
| Initial_admin_Emergency Admission | -6.2797 | 0.016 | -385.404 | 0.000 | -6.312 | -6.248 |
| HighBlood_numeric | -1.3761 | 0.016 | -83.486 | 0.000 | -1.408 | -1.344 |
| Arthritis_numeric | -0.8869 | 0.017 | -52.419 | 0.000 | -0.920 | -0.854 |
| Diabetes_numeric | -0.9332 | 0.018 | -51.363 | 0.000 | -0.969 | -0.898 |
| BackPain_numeric | -1.0486 | 0.016 | -63.635 | 0.000 | -1.081 | -1.016 |
| Reflux_esophagitis_numeric | -0.7152 | 0.016 | -43.461 | 0.000 | -0.747 | -0.683 |
| const | -28.6076 | 0.029 | -993.699 | 0.000 | -28.664 | -28.551 |

| | | | |
|---|---|---|---|
| Omnibus: | 482.377 | Durbin-Watson: | 1.974 |
| Prob(Omnibus): | 0.000 | Jarque-Bera (JB): | 384.450 |
| Skew: | -0.397 | Prob(JB): | 3.29e-84 |
| Kurtosis: | 2.459 | Cond. No. | 2.35e+04 |

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.35e+04. This might indicate that there are strong multicollinearity or other numerical problems.

The adjusted $R^2$ values (Figure 23) remain unchanged at .99, meaning that the predictor variables can explain 99% of the variation in 'Initial_days.' The F-statistic probability is 0 on both models, which also tells us that overall, the variables as a group in each regression model are significant. The AIC and BIC differences between the two models are extremely minimal, with Model B being the 'better' model with slightly fewer values. The residual standard error and residuals are also very similar in value

between the two models, as seen in Figure 24. A closer look at the numbers shows Model B edging out Model A for better RSE at the hundredth thousandth place, yet this insight is negligible.

The main differences in the model summaries lie within the individual variable's statistical significance in combination. In the initial model, there are variables with p-values well above the alpha, adding to the noise and risking overfitting the model. The final model includes only individual statistically significant variables, making it more reliable. Because Model B consists of only statistically significant variables compared to Model A, we will move forward with the reduced model (Model B) because of the significance of its explanatory variables.

Let us revisit the four assumptions of a multiple linear regression model mentioned in section B1. So far, we know neither model passes the assumption of linearity due to the overview of the correlation matrix generation in section C4. Only one explanatory variable shows a linear correlation to the predictor variable. In contrast, the rest of the variables that were added for the exercise denote a very low or nearly no linear relationship to the response variables. Both Model A and Model B violate the linearity assumption.

There is a difference in the multicollinearity assumption between Model A and Model B. Model A exhibited severe multicollinearity compared to Model B. Part of the reduction method for Model B was to remove variables that exhibited high VIFs. Thus, the reduced model consists of variables with VIFs less than 5 to prevent severe multicollinearity, which meets the second assumption of this exercise.

The third assumption is homoscedasticity, plotted for Model B in Figure 25. The residual plot shows horizontal bands evenly distributed across the graph. Although the plot does not exhibit a random pattern, the variance between 0 and the points does not increase as it does with heteroscedasticity (*Qualtrics*, n.d.). The placement of points is constant around 0 between 1 and -2, making the variance predictable. Model B meets the third assumption of homoscedasticity.

The QQ plot of predicted values against Model B's residuals is shown in Figure 26. It does not demonstrate normality because quite a few points do not fall on the 45-degree line. These points are very close and appear horizontal, so I checked the top ten largest and smallest residual values to make sure that the values are continuous. The calculation in Figure 26 showed that they are continuous but incremented very slightly from one another. The generated histogram in Figure 27 confirms that the residuals are not normally distributed, so Model B does not meet this normality assumption.

**E2. Screenshots of Outputs and Calculation**

Figure 24

*Residual Standard Error Calculations (Model A and Model B)*

```
# Calculate Residual Standard Error
model_A_rse = np.sqrt(model_A.mse_resid)
model_B_rse = np.sqrt(model_B.mse_resid)


print('Model A RSE: {}'.format(model_A_rse))
print('Model B RSE: {}'.format(model_B_rse))
```

```
Model A RSE: 0.8098814229317172
Model B RSE: 0.8098483722308781
```

Figure 25

*Model B Residual Plot*

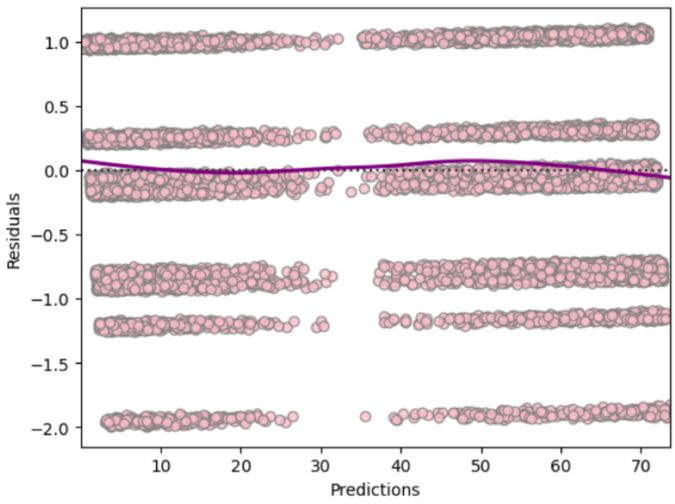
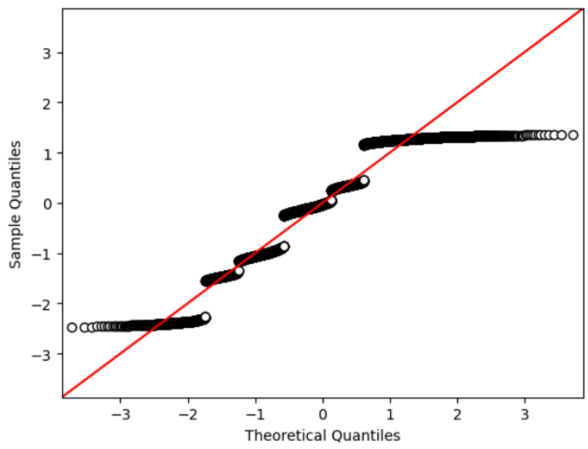



Figure 26

*Model B QQ Plot & Residual Values*

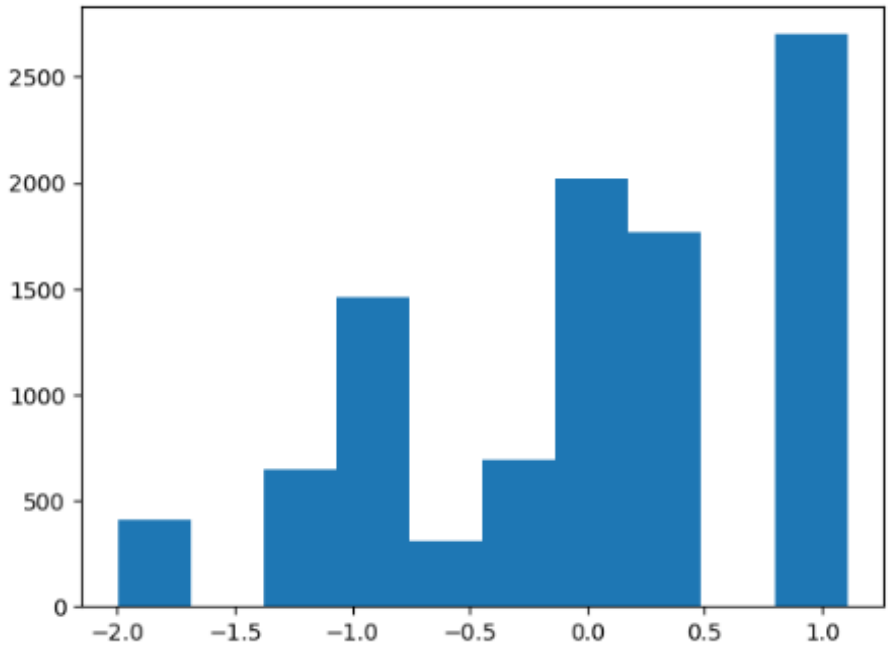



```
print('Largest values: {}'.format(model_B.resid.nlargest(10)))
print('Smallest values: {}'.format(model_B.resid.nsmallest(10)))
```

```
Largest values: 6731    1.107522
8663    1.104873
9292    1.103105
8560    1.102719
7413    1.100985
5016    1.100251
9876    1.099150
9444    1.096202
7949    1.092874
9545    1.092588
dtype: float64
Smallest values: 2286   -1.996872
2689   -1.993989
4032   -1.992500
3134   -1.991052
2694   -1.988102
4586   -1.987583
3713   -1.986554
2829   -1.983804
3003   -1.983390
3386   -1.983378
dtype: float64
```

Figure 27

*Model B Residual Histogram Plot*




## E3. Executable Code

See the file titled "Multiple_Linear_Regression_Notebook.ipynb" to see all the code generated for this paper.

## Part V: Data Summary and Implications

### F1. Results of Data Analysis

Figure 28

*Regression Equation*

**Hospitalization Days = -28.6076+ 0.0122 (Average Daily Charge) + 5.0384 (Low Complication Risk) + 5.0354 (Medium Complication Risk) – 6.2797 (Emergency Admission) – 1.3761 (High Blood) – 0.8869 (Arthritis) – 0.9332 (Diabetes) – 1.0486 (Back Pain)**

The equation above (Figure 28) reflects Model B. The naming convention has been changed slightly for improved readability. 'Hospitalization Days' is the response variable 'Initial_days' or y that equals the following expression. The regression coefficient interpretations are as follows:

- Holding all other variables constant, one unit *increas*e in average daily charge is associated with 0.0122 days *increase* in a patient's hospitalization stay.
- Holding all other variables constant, patients who have low complication risk, on average, stay at the hospital 5.0384 *more* days.
- Holding all other variables constant, patients who have medium complication risk, on average, stay at the hospital 5.0354 *more* days.
- Holding all other variables constant, patients who have emergency admission status, on average, stay at the hospital 6.2797 *fewer* days.
- Holding all other variables constant, patients who have high blood pressure, on average, stay at the hospital 1.3761 *fewer* days than patients who don't.
- Holding all other variables constant, patients who have arthritis stay at the hospital 0.8869 *fewer* days than patients who don't.
- Holding all other variables constant, patients who have diabetes stay at the hospital 0.9332 *fewer* days than patients who don't.
- Holding all other variables constant, patients who have back pain stay at the hospital 1.0486 *fewer* days than patients who don't.

This reduced model and combined explanatory variables demonstrate a vital statistical significance in helping us find some patient factors for the research question. Model B's low F-statistic probability and meaningful p-values signal statistical significance within the overall model.

However, I would not call Model B practically significant for the reasons below. Some are reiterations of insights taken from other sections in this paper.

The first reason Model B is not practically significant is because Model B does not meet all four assumptions of linear regression. To echo section E1, Model B meets the multicollinearity and homoscedastic assumptions but fails to meet linearity and normality assumptions. Issues with homoscedasticity and multivariate normality, as mentioned in section B1, make models unreliable. Not meeting all the conditions lowers the confidence that this model is helpful in a practical sense (Middleton, 2022b).

The second reason is that when it comes to the individual explanatory variable significance, we know that 'TotalCharge' is the only variable with a linear correlation to the response variable 'Initial_days.' This is not a helpful variable when providing insights about patients' demographic information despite being counted as a patient-hospital interaction factor. Clearly, 'TotalCharge' and 'Initial_days' would be positively correlated since more extended hospital stays mean a patient would have to pay for more days. It is only helpful as a significant variable because it provides a correlation weight to the model and lends significance to the other explanatory variables. If we were to run the model without 'TotalCharge,' like in Figure 29, the model would be rendered useless with an extremely low adjusted $R^{2,}$ and the remaining variables' significance values inflate above the target alpha.

Figure 29

*Model Without 'TotalCharge'*

OLS Regression Results

| | | | |
|---|---|---|---|
| Dep. Variable: | Initial_days | R-squared: | 0.001 |
| Model: | OLS | Adj. R-squared: | 0.001 |
| Method: | Least Squares | F-statistic: | 1.667 |
| Date: | Fri, 09 Aug 2024 | Prob (F-statistic): | 0.101 |
| Time: | 21:54:00 | Log-Likelihood: | -46881. |
| No. Observations: | 10000 | AIC: | 9.378e+04 |
| Df Residuals: | 9991 | BIC: | 9.385e+04 |
| Df Model: | 8 | | |
| Covariance Type: | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| Complication_risk_Low | 1.1295 | 0.730 | 1.548 | 0.122 | -0.301 | 2.559 |
| Complication_risk_Medium | -0.1144 | 0.599 | -0.191 | 0.849 | -1.290 | 1.061 |
| Initial_admin_Emergency Admission | -0.6082 | 0.526 | -1.156 | 0.248 | -1.639 | 0.423 |
| HighBlood_numeric | -0.3236 | 0.535 | -0.605 | 0.545 | -1.373 | 0.726 |
| Arthritis_numeric | 1.0488 | 0.549 | 1.910 | 0.056 | -0.028 | 2.125 |
| Diabetes_numeric | -0.1484 | 0.590 | -0.251 | 0.801 | -1.305 | 1.008 |
| BackPain_numeric | 0.9163 | 0.535 | 1.713 | 0.087 | -0.132 | 1.965 |
| Reflux_esophagitis_numeric | 0.6218 | 0.534 | 1.164 | 0.245 | -0.425 | 1.669 |
| const | 33.7388 | 0.696 | 48.455 | 0.000 | 32.374 | 35.104 |

| | | | |
|---|---|---|---|
| Omnibus: | 41272.257 | Durbin-Watson: | 0.161 |
| Prob(Omnibus): | 0.000 | Jarque-Bera (JB): | 1284.428 |
| Skew: | 0.070 | Prob(JB): | 1.23e-279 |
| Kurtosis: | 1.250 | Cond. No. | 5.61 |

Alternatively, doing a simple linear regression with just 'TotalCharge' and 'Initial_days' would meet all four assumptions. However, to iterate, 'TotalCharge' does not help answer the entire research question because it is not a targeted factor of the patient medical or demographic background, we deem insightful.

There are several limitations to consider in this linear regression exercise. The overall dataset did not provide multiple variables that showed high correlation coefficients to the response variable. The model immediately could not meet the linearity assumption for linear regression, lowering the confidence in its usefulness. Regarding data cleaning and wrangling, I decidedly kept outliers, which all seemed realistic across all the variables. By including the outliers, I may have affected coefficients and potentially distorted residuals (Middleton, 2022b). Overall, the general dataset does not provide in-depth information about each variable. For example, a few categories are listed for 'Services.' 'TotalCharge,' while it increases as 'Initial_days' increase, could vary per patient's hospitalized treatments or insurance copay. Some vague dictionary definitions limit the dataset knowledge and consideration across variables. I also limited myself to backward stepwise elimination and VIF observations regarding the model reduction process. Some consider stepwise elimination faulty because individual variable selection may not necessarily help the model's overall significance. Smith (2018) gives an example of forward stepwise

elimination, where choosing a fifth explanatory variable based on the four selected variables' p-values does not tell us that these five variables will result in the highest $R^2$. However, other methods are meticulous and time-consuming, especially when there are many variables to examine.

**F2. Actionable Recommendations**

Our research question is: *What factors regarding a patient's hospital activities, medical history, and demographic background correlate with the length of their initial hospital stay?* Model B's statistical significance shows us that some demographic factors can influence the length of a patient's hospital stay. However, since it does not meet the linear regression assumptions, the usefulness of Model B ends here. Although we discovered some factors correlated with a hospital stay, we should also be able to use the model confidently with data that is not the training data. Patient demographic information changes and shifts, so moving forward with a definitive model will also provide a solid foundation that can be observed over time.

The model is not practically significant, and therefore, subsequent recommendations address the limitations mentioned in section F1. We must reconsider the dataset and expand on the variables to build a model that can assertively predict the length of hospital stay based on patient demographic information. For example, we need to find other variables with a linear correlation to 'Initial_days.' We should also consider the peculiarities of the bivariate statistics mentioned in Section C3 and how they might affect the regression model. 'Initial_days' is bimodal, with two different data sets dispersed on either end of its histogram. We should figure out why there seem to be two general groups of patients. A deep dive into the definitions of some variables is also important and we should ask questions about limitations on some of them. Such questions could include the following: Why is 'Services' limited to only four types? How come most patients seem only to eat one meal? How are Vitamin D levels relevant to this dataset, and what are other vitamin types?

Finally, given what we know about the correlation between the explanatory variables and the response variable, we can explore other models. For example, since the variables violate the linearity assumption of a linear regression model, we know that this model is not an appropriate one to apply to our research question. An excellent start would be considering nonlinear models where the variables do not violate the linearity assumption.

**Part VI: Demonstration**

**G. Sources for Third-Party Code**


1. *Assessing model fit - Visualizing model fit*. (n.d.). [Video]. DataCamp. https://campus.datacamp.com/courses/introduction-to-regression-with-statsmodels-in-python/assessing-model-fit-e78fd9fe-6303-4048-8748-33b19c4222fe?ex=4
2. *Pandas get_dummies()*. (n.d.). https://www.programiz.com/python-programming/pandas/methods/get_dummies
3. *Plotting many pie charts using a loop to create a single figure using matplotlib*. (n.d.). Stack Overflow. https://stackoverflow.com/questions/75176180/plotting-many-pie-charts-using-a-loop-to-create-a-single-figure-using-matplotlib
4. *Predicting again | Python*. (n.d.). DataCamp. https://campus.datacamp.com/courses/intermediate-regression-with-statsmodels-in-python/multiple-linear-regression-3?ex=10
5. *seaborn.residplot — seaborn 0.13.2 documentation*. (n.d.). https://seaborn.pydata.org/generated/seaborn.residplot.html
6. Sewell, W. (n.d.). *D208 Predictive Modeling Episode 1* [Video]. Panopto. Retrieved July 12, 2024, from

https://wgu.hosted.panopto.com/Panopto/Pages/Auth/Login.aspx?instance=WGU&Auth=SessionView&panoptoState=276ead8c-2e90-419f-aab6-b1cb001544e3

**H. References**


1. Bobbitt, Z. (2021, November 16). *The five assumptions of multiple linear regression*. Statology. https://www.statology.org/multiple-linear-regression-assumptions/
2. Bruce, P., Bruce, A., & Gedeck, P. (2020). *Practical Statistics for Data Scientists: 50+ essential concepts using R and Python*. https://openlibrary.telkomuniversity.ac.id/home/catalog/id/163886/slug/practical-statistics-for-data-scientists-50-essential-concepts-using-r-and-python-2-e-.html
3. Larose, C. D., & Larose, D. T. (2019). *Data science using Python and R*. John Wiley & Sons.
4. Lingsma, H. F., Bottle, A., Middleton, S., Kievit, J., Steyerberg, E. W., & De Mheen, P. J. M. (2018). Evaluation of hospital outcomes: the relation between length-of-stay, readmission, and mortality in a large international administrative database. *BMC Health Services Research*, *18*(1). https://doi.org/10.1186/s12913-018-2916-1
5. Middleton, K. (2022a, November). *D208 - Webinar: Getting Started with D208 Part I (November 2022)* [Video]. Panopto. Retrieved July 13, 2024, from https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=15e09c73-c5aa-439d-852f-af47001b8970
6. Middleton, K. (2022b, November). *D208 - Webinar: Getting Started with D208 Part II (November)* [Video]. Panopto. Retrieved July 28, 2024, from https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=39bbe2db-de7d-4bf5-913

7. *Qualtrics*. (n.d.). https://www.qualtrics.com/support/stats-iq/analyses/regression-guides/interpreting-residual-plots-improve-regression/
8. Smith, G. (2018). Step away from stepwise. *Journal of Big Data*, *5*(1). https://doi.org/10.1186/s40537-018-0143-6
9. Tipton, K., Leas, B. F., Mull, N. K., Siddique, S. M., Greysen, S. R., Lane-Fall, M. B., & Tsou, A. Y. (2021, September 1). *Introduction*. Interventions to Decrease Hospital Length of Stay - NCBI Bookshelf. https://www.ncbi.nlm.nih.gov/books/NBK574438/