Matthew Boruff

MSDA D208

Dr Middleton

With healthcare costs rising and penalties for high patient readmissions, hospitals are being faced with finding possible ways to predict if a patient will return or not on related issues. High readmission rates not only effect the patient but can also financially harm the hospital with fines from the Centers for Medicare and Medicaid Services. In previous analysis reviews, there had been a correlation between the number of days a patient was initially hospitalized and readmission, possibly found. With this study, we will hope to expand and find more possible correlations that may contribute to a patient having to be readmitted.

Part I Research Question

1. Research Question:

What other factors can we correlate to a possible length of stay for the initial visit to the hospital?

1. Objectives for this Data Analysis

The goal of this Analysis will be to help the hospital recognize a possible pattern in lengthy stays and their cause. If a pattern can be determined for a predictable hospital stay, steps can be taken to reduce or mitigate the risk of a lengthy stay.

Part II: Method Justification

B. Multiple Regression Methods:

1. A Multiple regression is a statistical took used to study the dependence of continuous response of two or more linear predictions. If it is determined that a relationship exists, then the model can help predict future outcomes given certain independent factors.

Multiple regression model assumptions:

1. Multiple regression assumes normal distribution of residuals, tested by plotting a histogram of residuals from a regression model
2. Multicollinearity-Independent variables. Are not correlated with each other.
3. It requires at least two independent variables, nominal, ordinal or interval
4. Homoscedasticity – Distribution of errors in the independent variables is similar. Can be checked by plotting residuals

Benefits of Python for Data analysis:

The user-friendly syntax and robust libraries for data analysis along with a familiarity with the language, make this a personal best choice for this case. There is also a diverse visualization tools libraries to help with seeing a correlation easier or trends better for predictability.

Multiple Linear Regression as the tool for this analysis

Multiple linear regression can be used to estimate the relationship between two or more independent variables and one continuous quantitative dependent variable. We will be using the dependent variable, Initial days as a continuous data type. Can we see determine what other variables have a strong relationship? Examples would be age, Pre-existing conditions and complication risks. The value of the dependent variable, at a certain value of the independent variables.

Once we can establish whether these have a linear relationship through a multiple regression, then we may be able to predict who will be readmitted.

Part III Data Prep

Data Prep Summary

1. Goals and Techniques

The goal is to clean and transform the data in to a usable format to run the multiple regression testing on, to gather useful and meaningful data.

The preparation process will include:

* Loading the libraries
* Load the CSV into a dataframe and structure
* Examine columns and data types
* Detect and remove duplicates
* Review and transform data from categorical to numeric
* Eliminate format inconsistencies
* Fill in null values with median

1. Target Variable

We will be using the Initial\_days column for the target variable. This column represents the amount of time the patient stayed on their first hospitalization stay. After this we will be using several independent variables that describe different aspects of their health. Complication Risk, Overweight, High Blood pressure are among the examples

|  |  |  |  |
| --- | --- | --- | --- |
| Age | Continuous | Age of patient | 33 |
| ReAdmis | Categorical | If patient is readmitted | Yes/no |
| VitD\_levels | Continuous | Levels of vitamin d | 16.99 |
| Doc\_visits | Continuous | Number of visits between hospitializations | 4 |
| VitD\_supp | Continuous | Number of times Vit D was given during hospital stay | 0 to 5 |
| Initial\_Admin | Categorical | Days of first stay | 0 to 10 |
| High\_Blood | Categorical | If patient has high blood pressure | Yes/no |
| Stroke | Categorical | Has patient had a stroke | Yes/no |
| Complication\_risk | Categorical | Level of complication risk | High/medium/Low |
| Overweight | Categorical | Is patient overweight | Yes/no |
| Arthritis | Categorical | Does patient have arthritis | Yes/no |
| Diabetes | Categorical | Does patient have diabetes | Yes/no |
| Hyperlipidemia | Categorical | Does patient have hyperlipidemia | Yes/no |
| BackPain | Categorical | Does patient have back pain | Yes/no |
| Anxiety | Categorical | Does patient have Anxiety | Yes/no |
| Allergic\_rhinitis | Categorical | Does patient have allergic rhinitis | Yes/no |
| Reflux\_esophagitis | Categorical | Does patient have Reflux esophagitis | Yes/no |
| Asthma | Categorical | Does patient have Asthma | Yes/no |
| Item1 | Ordinal | Customer rating of timely admission | Scale of 1 - 8 |
| Item2 | Ordinal | Customer rating of timely service | Scale of 1 - 8 |
| Item3 | Ordinal | Customer rating of timely treatment | Scale of 1 - 8 |
| Item4 | Ordinal | Customer rating of timely visits | Scale of 1 - 8 |
| Item5 | Ordinal | Customer rating of reliability | Scale of 1 - 8 |
| Item6 | Ordinal | Customer rating of hours or treatment | Scale of 1 - 8 |
| Item7 | Ordinal | Customer rating of courteous staff | Scale of 1 - 8 |

**Explain and show Code:**

Below are the libraries that are going to be needed to perform this analysis and pull in the local CSV file to the data frame for manipulation. After the libraries are imported, we will use the Pandas pd.set\_option to see all the columns at once and not a “redacted” version and pd.read\_csv to pull in the medical csv document. Then the last command of data to view the new data DataFrame.

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline

import scipy.stats as stats

import seaborn as sns

from sklearn.decomposition import PCA

import plotnine as p9

from statsmodels.formula.api import ols

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

data = pd.read\_csv('medical\_clean.csv', index\_col='CaseOrder')

data

Next we will be using a few commands to check the data frame for duplicates and nulls.

data.head()

data.info()

data.columns

Index(['Customer\_id', 'Interaction', 'UID', 'City', 'State', 'County', 'Zip',

'Lat', 'Lng', 'Population', 'Area', 'TimeZone', 'Job', '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', 'Initial\_days', 'TotalCharge', 'Additional\_charges',

'Item1', 'Item2', 'Item3', 'Item4', 'Item5', 'Item6', 'Item7', 'Item8'],

#Look for Duplicates

data.duplicated().sum()

0

#Check for Nulls

data.isnull().sum()

Customer\_id 0

Interaction 0

UID 0

City 0

State 0

County 0

Zip 0

Lat 0

Lng 0

Population 0

Area 0

TimeZone 0

Job 0

Children 0

Age 0

Income 0

Marital 0

Gender 0

ReAdmis 0

VitD\_levels 0

Doc\_visits 0

Full\_meals\_eaten 0

vitD\_supp 0

Soft\_drink 0

Initial\_admin 0

...

Item5 0

Item6 0

Item7 0

Item8 0

dtype: int64

Per the code and output of that code this data is in a very clean position to use. There are 0 nulls and no duplicates apparent in the data frame.

**Rename columns and convert columns in prep for the analysis.**

#setting new names for columns Item1-8

new\_col\_names={'Item1':'T\_Admission', 'Item2':'T\_Treatment', 'Item3':'T\_Visits', 'Item4':'Reliability', 'Item5':'Options', 'Item6':'Hours\_Treatment', 'Item7':'Staff', 'Item8':'Active\_Listening'}

data.rename(columns=new\_col\_names, inplace=True)

data.columns

Output:

Index(['Customer\_id', 'Interaction', 'UID', 'City', 'State', 'County', 'Zip',

'Lat', 'Lng', 'Population', 'Area', 'TimeZone', 'Job', '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', 'Initial\_days', 'TotalCharge', 'Additional\_charges',

'T\_Admission', 'T\_Treatment', 'T\_Visits', 'Reliability', 'Options',

'Hours\_Treatment', 'Staff', 'Active\_Listening'],

dtype='object')

#data columns converted from yes/no to 0/1

d = data.replace({'Yes':1, 'No':0})

d

We will now save this data with the converted columns to a new csv, medical\_clean2.csv

data.to\_csv('medical\_clean2.csv')

**Univariate and Bivariate Analysis:**

**Univariant Analysis:**

Using the response variable of Initial\_days (how many days the patient spent in the hospital the first stay) has no outliers. There are no outliers however the data is not normally distributed according to the following histograms.

#Univariate Visuals

sns.boxplot('Initial\_days', data=dMedSorted)

plt.show()

sns.displot(data['Initial\_days'])

Chart, histogram

Description automatically generated

Chart, histogram

Description automatically generated

fig, ax=plt.subplots(1,3)

sns.countplot(x='Services', data=dMedSorted, ax=ax[0])

sns.countplot(x='Initial\_admin', data=dMedSorted, ax=ax[1])

sns.countplot(x='Complication\_risk', data=dMedSorted, ax=ax[2])

plt.show()

Services, Initial\_admin and Complication risk were categorical values that were transformed to numeric.

Chart, bar chart

Description automatically generated

dMedSorted[['Age', 'VitD\_levels', 'Initial\_days', 'Doc\_visits', 'Full\_meals\_eaten',]].hist()

plt.show()

Chart

Description automatically generated

Next, we will see that the distribution of the Survey responses that we renamed look to be normal

dMedSorted[['T\_Admission', 'T\_Treatment', 'T\_Visits', 'Reliability', 'Options', 'Hours\_Treatment', 'Staff', 'Active\_Listening']].hist()

plt.show()

Chart

Description automatically generated

fig, ax=plt.subplots(1,3)

sns.countplot(x=dMedSorted['Overweight'], ax=ax[0])

sns.countplot(x=dMedSorted['Diabetes'], ax=ax[1])

sns.countplot(x=dMedSorted['Anxiety'], ax=ax[2])

plt.tight\_layout()

fig.show(0)

Chart, bar chart

Description automatically generated

**Bivariant Analysis:**

#Bivariant and Reduced sample size

data\_sample=dMedSorted.head(100)

sns.regplot(x='Age', y='Initial\_days', data=data\_sample)

plt.title('Dist of Hospialization Days vs. Age')

plt.show()

**Chart, scatter chart

Description automatically generated**

There really does not seem to be a correlation between Age and the patient’s initial stay.

#Corelation between ReAdmis chance and Initial Days

sns.regplot(x='ReAdmis', y='Initial\_days', data=d)

Chart, line chart

Description automatically generated

However, there does seem to be a strong correlation between the longer your Initial stay is to a risk of a readmission.

sns.scatterplot(x='Complication\_risk', y='Initial\_days', hue='Gender', data=d)

plt.title('Hopitalization Days Vs Complication Risk')

plt.show()

Chart

Description automatically generated

The possible correlation between complication risk and the initial hospital stay is very ambiguous.

sns.scatterplot(x='Initial\_admin', y='Initial\_days', data=d)

plt.title('Hospitalization Days Vs Admission')

plt.show()

Chart

Description automatically generated

sns.scatterplot(x='T\_Treatment', y='Initial\_days', data=d)

plt.title('Hospitalization vs Timely Treatment')

plt.show()

A picture containing chart

Description automatically generated

T\_Treatment(Timely Treatment) looks to be equally distributed across hospitalization days, except at score 7 that seems to be more associated with a higher number of days on the initial stay.

#Emergency Admis = 3 Elective = 2 Obs = 1

sns.boxplot(x='Initial\_admin', y='Initial\_days', data=d)

plt.title('Observation 1, Elective 2, Emergency 3')

plt.show()

Chart, box and whisker chart

Description automatically generated

With this boxplot, we can see that Emergency Admissions have a lower median than Observation or Elective admission stays.

**Initial Multiple regression model with all predictors identified in C2**

#Starting the mulitple regression

m\_data=ols('Initial\_days~ Children + Age + VitD\_levels + Doc\_visits + Full\_meals\_eaten + vitD\_supp + Initial\_admin + Complication\_risk + Services + T\_Admission + T\_Treatment + T\_Visits + Reliability + Options + Hours\_Treatment + Staff + Active\_Listening + Soft\_drink + HighBlood + Stroke + Overweight + Arthritis + Diabetes + Hyperlipidemia + BackPain + Anxiety + Allergic\_rhinitis + Reflux\_esophagitis + Asthma', data = d).fit()

m\_data.summary()

|  |  |  |  |
| --- | --- | --- | --- |
| OLS Regression Results | | | |
| Dep. Variable: | Initial\_days | R-squared: | 0.004 |
| Model: | OLS | Adj. R-squared: | 0.001 |
| Method: | Least Squares | F-statistic: | 1.317 |
| Date: | Sat, 25 Mar 2023 | Prob (F-statistic): | 0.119 |
| Time: | 15:06:33 | Log-Likelihood: | -46869. |
| No. Observations: | 10000 | AIC: | 9.380e+04 |
| Df Residuals: | 9970 | BIC: | 9.401e+04 |
| Df Model: | 29 |  |  |
| Covariance Type: | nonrobust |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | coef | std err | t | P>|t| | [0.025 | 0.975] |
| Intercept | 37.7926 | 3.845 | 9.829 | 0.000 | 30.256 | 45.329 |
| Children | 0.2716 | 0.122 | 2.232 | 0.026 | 0.033 | 0.510 |
| Age | 0.0203 | 0.013 | 1.593 | 0.111 | -0.005 | 0.045 |
| VitD\_levels | -0.0315 | 0.131 | -0.241 | 0.810 | -0.288 | 0.225 |
| Doc\_visits | -0.1798 | 0.252 | -0.714 | 0.475 | -0.673 | 0.314 |
| Full\_meals\_eaten | -0.4421 | 0.261 | -1.692 | 0.091 | -0.954 | 0.070 |
| vitD\_supp | 0.6671 | 0.419 | 1.592 | 0.111 | -0.154 | 1.488 |
| Initial\_admin | -0.2902 | 0.319 | -0.909 | 0.364 | -0.916 | 0.336 |
| Complication\_risk | -0.4770 | 0.361 | -1.323 | 0.186 | -1.184 | 0.230 |
| Services | 0.0673 | 0.316 | 0.213 | 0.832 | -0.553 | 0.687 |
| T\_Admission | -0.8244 | 0.379 | -2.174 | 0.030 | -1.568 | -0.081 |
| T\_Treatment | 0.2611 | 0.350 | 0.746 | 0.455 | -0.425 | 0.947 |
| T\_Visits | 0.0347 | 0.323 | 0.107 | 0.914 | -0.598 | 0.667 |
| Reliability | -0.3598 | 0.288 | -1.251 | 0.211 | -0.923 | 0.204 |
| Options | 0.0161 | 0.303 | 0.053 | 0.958 | -0.578 | 0.610 |
| Hours\_Treatment | -0.0538 | 0.313 | -0.172 | 0.863 | -0.667 | 0.559 |
| Staff | 0.3519 | 0.294 | 1.195 | 0.232 | -0.225 | 0.929 |
| Active\_Listening | -0.0575 | 0.277 | -0.208 | 0.836 | -0.601 | 0.486 |
| Soft\_drink | 0.1644 | 0.603 | 0.273 | 0.785 | -1.017 | 1.346 |
| HighBlood | -0.3438 | 0.536 | -0.642 | 0.521 | -1.394 | 0.706 |
| Stroke | -0.1264 | 0.659 | -0.192 | 0.848 | -1.418 | 1.165 |
| Overweight | -0.5901 | 0.580 | -1.018 | 0.309 | -1.727 | 0.547 |
| Arthritis | 1.0286 | 0.550 | 1.872 | 0.061 | -0.049 | 2.106 |
| Diabetes | -0.1239 | 0.591 | -0.210 | 0.834 | -1.283 | 1.035 |
| Hyperlipidemia | -0.1959 | 0.557 | -0.352 | 0.725 | -1.288 | 0.897 |
| BackPain | 0.9117 | 0.535 | 1.703 | 0.089 | -0.138 | 1.961 |
| Anxiety | 0.6454 | 0.564 | 1.145 | 0.252 | -0.460 | 1.750 |
| Allergic\_rhinitis | 0.2157 | 0.539 | 0.400 | 0.689 | -0.841 | 1.272 |
| Reflux\_esophagitis | 0.6709 | 0.535 | 1.254 | 0.210 | -0.378 | 1.719 |
| Asthma | -0.7957 | 0.581 | -1.370 | 0.171 | -1.934 | 0.343 |

|  |  |  |  |
| --- | --- | --- | --- |
| Omnibus: | 41652.924 | Durbin-Watson: | 0.166 |
| Prob(Omnibus): | 0.000 | Jarque-Bera (JB): | 1270.719 |
| Skew: | 0.070 | Prob(JB): | 1.17e-276 |
| Kurtosis: | 1.259 | Cond. No. | 892. |

**Regression Formula:**

Y=37.79(Intercept) + .2716(Children) + .02(Age) - .031(VitD\_levels) - .179(Doc\_visits) -.442(Full\_meals\_eaten) + .667(vitD\_supp) - .477(Complication\_risk) + .067(Services) - .824(T\_Admission) + .261(T\_Treatment) + .034(T\_Visits) - .359(Reliability) + .016(Options) - .053(Hours\_Treatment) + .351(Staff) - .057(Active\_Listening) + .164(Soft\_drink) - .343(HighBlood) -.126(Stroke) - .590(Overweight) + 1.028(Arthritis) - .123(Diabetes) - .195(Hyperlipidemia) + .911(BackPain) + .645(Anxiety) + .215(Allergic\_rhinitis) + .670(Reflux\_esophagitis) - .795(Asthma)

**Validating the Initial Model:**

Looking at the coefficient determination, it is the factor that the variations in the dependent variable that comes from the independent variable. The R2 score is ranging 0 to 1, the closer to 1 it is the greater the linear relationship is between the variables.

print('R2 score:', m\_data.rsquared)

R2 score: 0.003815829743797683

The R2 score translates to a .4% of the variation in our dependent variable can be explained using the independent variable. Looking at the adjusted R square we can see it is lower at .1%, this is a very low R score and is likely that none of the independent variables influence the Initial\_days in a linear relationship.

Next the F Test checks for independent variable combined as related to the dependent variable when compared to the state when all the independent variables are 0. If F-Stat p value is greater than .05 there is no evidence of co-linearity between combined independent variable with the output.

#Run the F test and check for independent variables

print('F-statistic:', m\_data.fvalue)

print('probability of observing value at least as high as F-Statistic:', m\_data.f\_pvalue)

F-statistic: 1.3168809433230266

probability of observing value at least as high as F-Statistic: 0.11864182587747099

The F-P value is higher than .05, means that our independent variables combine have no correlation to the dependent variable.

We can analyze the p-Values to evaluate those with a higher than .05 value to remove for the reduced model. The higher value variables can be very influential on the target variable.

print('p-values:', m\_data.pvalues)

p-values: Intercept 1.066882e-22

Children 2.565722e-02

Age 1.112780e-01

VitD\_levels 8.095092e-01

Doc\_visits 4.752569e-01

Full\_meals\_eaten 9.073976e-02

vitD\_supp 1.113649e-01

Initial\_admin 3.636255e-01

Complication\_risk 1.860026e-01

Services 8.315262e-01

T\_Admission 2.969167e-02

T\_Treatment 4.554369e-01

T\_Visits 9.144763e-01

Reliability 2.108317e-01

Options 9.575797e-01

Hours\_Treatment 8.633913e-01

Staff 2.320992e-01

Active\_Listening 8.356199e-01

Soft\_drink 7.849952e-01

HighBlood 5.210159e-01

Stroke 8.478636e-01

Overweight 3.089070e-01

Arthritis 6.127904e-02

Diabetes 8.340309e-01

Hyperlipidemia 7.252017e-01

...

Allergic\_rhinitis 6.889268e-01

Reflux\_esophagitis 2.097387e-01

Asthma 1.707427e-01

Only Children and T\_Admission variables have a p value below .05, so the rest of the variables in the data set do not affect the dependent variable.

The RESIDUALS are the difference between the observed value and predicted value of the data set. This is often used to assess whether a linear regression model is fit for the dataset. Evenly distributed residual points between the predicted and actual values indicate linear relationship between the dependent and independent variables.

#Create predicted values and residuals, adding two columns.

d['Initial\_days\_predict']=m\_data.predict()

d['residual']=m\_data.resid

d.head()

#scatter plot of residuals then plotting Observed vs Predicted

sns.lmplot(x='Initial\_days', y='Initial\_days\_predict', data=d, fit\_reg=False, height=5)

#lets add the diagonal line too while we are at it

line\_coords=np.arange(d[['Initial\_days\_predict']].max().max()+10)

plt.plot(line\_coords, line\_coords, color='orange', linestyle='--')

plt.ylabel('Predicted Initial', fontsize=12)

plt.xlabel('Actual Initial', fontsize=12)

plt.title('Linearity Assumption')

plt.show()

Chart

Description automatically generated

The predicted initial days do not correlate to the actual initial days. There is not linear distribution for predicted days across the plot.

#Normal Dist of residual values

#plotting Resid of Dist

plt.subplots(figsize=(8,4))

plt.title('Distribution of Residuals', fontsize=14)

sns.displot(d['residual'])

plt.show()

Chart, histogram

Description automatically generated

We have a twin peak skew of the residual distribution. This correlates to the indication we had in the Univariate analysis histogram of initial days. We will now move on to doing the statistical analysis of the p-value in residuals, a P Value of .05 or higher will mean a normal distribution.

First we will need a specific import from statsmodel.stats.diagnostic

from statsmodels.stats.diagnostic import normal\_ad

#Testing the residuals

p\_value = normal\_ad(d['residual'])[1]

print('p-value from the test Anderson-Darling test below .05 generally means non-normal:', p\_value)

#reporting the normality of resid's

if p\_value <.05:

print('Residuals are not Normally Distributed')

else:

print('Residuals are distributed normally')

p-value from the test Anderson-Darling test below .05 generally means non-normal: 0.0

Residuals are not Normally Distributed

With this confirming it, the residuals are not normally distributed.

Next, we will plot the residuals vs predicted to see how may correct predictions the model made. A value near 0 are the closest to correct predictions.

#residual plots

sns.scatterplot(x='Initial\_days\_predict', y='residual', data=d)

plt.show()

Chart, scatter chart

Description automatically generated

And with that the model is unable to predict correctly most of the actual values.

**Homoscedasticity**

The assumption of homoscedasticity (meaning “same variance”) is central to [linear regression](https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/what-is-linear-regression/) models.  Homoscedasticity describes a situation in which the error term (that is, the “noise” or random disturbance in the relationship between the independent variables and the dependent variable) is the same across all values of the independent variables.  Heteroscedasticity (the violation of homoscedasticity) is present when the size of the error term differs across values of an independent variable.  The impact of violating the assumption of homoscedasticity is a matter of degree, increasing as heteroscedasticity increases. (1)

#Homoscedasticity The variation in the errors acorss variables is similar

#Heteroscedasticity, the violation of homo, occurs when we do not have even variance across error terms. to detect homo, we can plt residual and iss if the vairance is uniform

#Plotting the residuals

plt.subplots(figsize=(8,4))

plt.scatter(x=d.index, y=d.residual, alpha=.8)

plt.plot(np.repeat(0, len(d.index)+2), color='orange', linestyle='--')

plt.ylabel('Residual', fontsize=12)

plt.xlabel('Case Order', fontsize=12)

plt.title('Homoescedacticity Assumption', fontsize=18)

plt.show()

Chart, scatter chart

Description automatically generated

There appears to be two groups in the dataset that are distributed differently, apparently showing us Heteroscedascity across the data.

**Multicollinearity**

Multicollinearity is the occurrence of high intercorrelations among two or more independent variables in a multiple regression model. Multicollinearity can lead to skewed or misleading results when a researcher or analyst attempts to determine how well each independent variable can be used most effectively to predict or understand the dependent variable in a statistical model.

#Checking for multicolinearity by correlation

corr=d.corr()

plt.figure(figsize=(16,9))

sns.heatmap(corr, xticklabels=corr.columns, yticklabels=corr.columns, cmap='RdBu')

Chart, scatter chart

Description automatically generated

Starting to isolate the independent variables that appear to have a possible relationship

#Attempt to isolate independent variables that have some relationships beetween themselves

sns.pairplot(d[['Initial\_days', 'T\_Treatment', 'T\_Admission', 'T\_Visits']])

plt.show()

Chart

Description automatically generated

Some collinearity between the Survey questions, T\_Admissions, T\_Visits and T\_Treatment, has shown to be evident. We can then reduce the T columns to one as they seem to all behave the same.

**Model Conclusion**:

1.Through the process of checking the model we may have determined that the R2 value of the model was too low. The adjusted value was .1% meaning the data has a high variability.

2.The data has Heteroescedascity

3.Data does not distribute normally

4.Independent variables have no linearity

5.The p-values for Children and T\_ Admission have values below the .05 Alpha, continued use of these should be considered.

**Reduced Model:**

Now from the initial model, we can start to work on the reduced model using the same data, without the independent variables that were considered too low on the p-value. The reduced model is an attempt to become more accurate. We will be using T\_Admission and Children with the Initial\_days data.

model\_rev = model\_rev = ols('Initial\_days~Children + T\_Admission', data=d).fit()

print(model\_rev.summary())

print(model\_rev.summary())

OLS Regression Results

==============================================================================

Dep. Variable: Initial\_days R-squared: 0.001

Model: OLS Adj. R-squared: 0.001

Method: Least Squares F-statistic: 5.015

Date: Mon, 27 Mar 2023 Prob (F-statistic): 0.00665

Time: 15:00:49 Log-Likelihood: -46883.

No. Observations: 10000 AIC: 9.377e+04

Df Residuals: 9997 BIC: 9.379e+04

Df Model: 2

Covariance Type: nonrobust

===============================================================================

coef std err t P>|t| [0.025 0.975]

-------------------------------------------------------------------------------

Intercept 35.8822 0.968 37.061 0.000 33.984 37.780

Children 0.2738 0.122 2.252 0.024 0.035 0.512

T\_Admission -0.5687 0.255 -2.231 0.026 -1.068 -0.069

==============================================================================

Omnibus: 41245.176 Durbin-Watson: 0.160

Prob(Omnibus): 0.000 Jarque-Bera (JB): 1285.499

Skew: 0.071 Prob(JB): 7.20e-280

Kurtosis: 1.249 Cond. No. 16.9

==============================================================================

Notes:

...

==============================================================================

Validation of reduced model:

Need to check for multicollinearity, and ensure the data being used do not have to high of a value.

#Looking for Multicolinearity through correlation

x=d[['Children', 'Initial\_days', 'T\_Admission']]

corr=x.corr()

plt.figure(figsize=(12,6))

sns.heatmap(corr, xticklabels=corr.columns, yticklabels=corr.columns, cmap='RdBu')

Background pattern, square

Description automatically generated with medium confidence

Collinearity is absent.

The Reduced Model Statistics

print('R-Square:', model\_rev.rsquared)

R-Square: 0.0010023109063076463

R2 test result is .1% of the data variation explained by the chosen variables.

When comparing the results of the reduced model to the initial model, the results are nearly identical.

print('F-pvalue:', model\_rev.f\_pvalue)

F-pvalue: 0.006653823955992259

The F pvalue is below the .05 threshold of the Alpha. Therefore we can reject the null hypothesis. There is an improvement from the initial model, it also indicates that the chosen variables have a possible correlation to the dependent variable.

print(model\_rev.params)

Intercept 35.882234

Children 0.273766

T\_Admission -0.568682

print(model\_rev.pvalues)

Intercept 7.757493e-282

Children 2.433020e-02

T\_Admission 2.567716e-02

Children and T\_Admission have p-values below .05 identical to the initial model. Meaning we can still reject the null hypothesis.

Reduced Model Validation:

1. Testing the assumption of normality and homoscedascity of the model by looking at the residuals.
2. Checking for linearity, checking if the predictor variables in the regression of a straight line relationship with the observed/actual variable.

#Making model prediction and adding resid'set

d['Initial\_days\_predict']=model\_rev.predict(x)

d['residual']=model\_rev.resid

d.columns

#Plotting the Obs vs Pred for linearity

sns.lmplot(x='Initial\_days', y='Initial\_days\_predict', data=d, fit\_reg=False, size =5)

#adding the diagonal line

line\_coords = np.arange(x[['Initial\_days', 'Initial\_days\_predict']].min().min()-10, x[['Initial\_days', 'Initial\_days\_predict']].max().max()+10)

plt.plot(line\_coords, line\_coords, color='orange', linestyle='--')

plt.ylabel('Actual Initial Days', fontsize=12)

plt.xlabel('Linearity Assumption', fontsize=12)

plt.show()

Chart, line chart

Description automatically generated

Once again, as seen in the Initial Model, there is not linear relationship.

#Plotting residuals distribution for normalitu

plt.subplots(figsize=(8,4))

plt.title('Distribution of Residuals', fontsize=18)

sns.displot(x['residual'])

plt.show()

Chart, histogram

Description automatically generated

Like we have seen on all other test for reduced so far, the distribution of residuals is not normal.

#Performing the test on residuals

p\_value=normal\_ad(x['residual'])[1]

print('p vlaue from the Anderson-Darling test below .05 generally means the data is not normal:', p\_value)

#Normality of residuals

if p\_value<.05:

print('Residuals are not normally distributed')

else:

print('Residuals are normally distributed')

p vlaue from the Anderson-Darling test below .05 generally means the data is not normal: 0.0

Residuals are not normally distributed

#Plotting for homoscedacity

plt.subplots(figsize=(8,4))

plt.scatter(x=x.index, y=x.residual, alpha=.8)

plt.plot(np.repeat(0, len(x.index)+2), color='orange', linestyle='--')

plt.ylabel('Residual', fontsize=12)

plt.xlabel('Initial\_days', fontsize=12)

plt.title('Homoscedacity Assumption', fontsize=14)

plt.show()

Chart

Description automatically generated

Again, the plot shows heteroscedacity in the data, the residuals are bunched into two distinct groupings

Conclusion:

The reduced model, like the initial model, is not valid as it does not have any multiple regression assumptions.

1. Doing a reduced model did not improve the reliability of the model.
2. Based on the plots, neither model is robust in predicting correct numbers
3. The data is heteroscedatic
4. There is no multicollinearity in either model.
5. Data is no normally distributed.

Data Summary and implications

Reduced Model regression formula:

R-Square: .001 = .1% y=35.48 + .275(Children) - .568(T\_Admission)

The model was absent a lot of independent variables that had statistical relationships to the dependent variable. Children and T\_Admission were the only p\_values below the threshold of .05 alpha. When the independent factors are present, the hospitalization days are around 35.5. T\_Admission might reduce the stay by .56 days, pending all other variables are 0 and T\_Admission rating was a 1, Children may also increase the stay by .27 days for 1 child.

1. The prediction outcome of this model is not a good fit for this data set. With all the different variable only two would meet the thresholds required to be used in the reduced model.
2. There was no linear correlation in either model using the independent variables available.

Based on this model, the research question could not be answered, while two of the variables do show some indication or influence on the stay in the hospital, their significance is minimal.

Recommendations:

From the results or lack thereof in this model, either more data would need to be collected or a different model choice would be appropriate. Could we find a better result using a logistic regression on different variables? Using a non-linear based model may expose better fits in differing factors. Another recommendation would be to see if the research question is actually a good fit for this data set using this particular model. The research question may be hampering the efforts to understand what the data may be trying to tell us.

References:

1. <https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/homoscedasticity/>
2. <https://www.investopedia.com/terms/m/multicollinearity.asp#:~:text=Multicollinearity%20is%20a%20statistical%20concept,in%20less%20reliable%20statistical%20inferences>.
3. Introduction to Python for Data Analysis. Chapter 8: Simple Linear regression Linear Regression — Basic Analytics in Python (sfu.ca)

3rd party code use

1. Multiple Linear Regression using Python by Amrutha K; https://www.analyticsvidhya.com/blog/2022/03/multiple-linear-regression-using-python/
2. Multi Linear Regression using Python by Rafi Atha https://medium.com/swlh/multi-linear-regression-using-python-44bd0d10082d