Dameia Brewster

Chi-Squared Analysis on Hospital Readmission Data Set

D214 Data Analytics Graduate Capstone

Task 2: Data Analytics Report and Executive Summary

A. Research Question

The research question for this capstone project is “What specific factors statistically influence hospital readmissions?”

The contribution of this study to the field of Data Analytics and the MSDA program is to create a predictive model which will provide an assessment of the relationship between the variables that significantly relate to the readmission of a patient. Inadequate or incomplete treatment of the diagnosed condition was the most common cause leading to readmission (Shalchi et al., 2009). Stakeholders in the healthcare policy system could benefit from the results of the analysis that could determine strategies or solutions needed to improve the healthcare system to avoid any penalties or fines for readmissions.

This study will utilize multivariate analysis to visually analyze the significant difference between variables and identify which variables mostly influence patient readmission. Multivariate analysis is any type of statistical analysis that reviews more than one variable, more specifically a logistic regression model will be created to analyze the relationship between binary dependent variables and independent variables to predict outcomes in the analysis.

Logistic regression accomplishes binary classification tasks by predicting the probability of a relationship between one or more independent variables (Researcher et al., Logistic regression: Equation, assumptions, types, and best practices 2022). A chi-square test is a statistical test used to compare observed results with expected results. The purpose of this test is to determine if a difference between observed data and expected data is due to chance, or if it is due to a relationship between the variables you are evaluating (Chi Square | Practical Applications of Statistics in the Social Sciences| University of Southampton, n.d.). By understanding the relationship between the two variables, we can conclude if a relationship exists or not.

For the purpose of this analysis, we have established the following null and alternative hypotheses:

Null Hypothesis: There is *no* significant difference in hospital readmission among variables studied.

Alternative Hypothesis: There is a significant difference in hospital readmission among variables studied.

The null hypothesis assumes there is no significant difference in hospital readmission among variables studied. This means that variables such as are not associated with an increase or decrease in hospital readmission. The hypothesis suggests that any observed relationship has no significant underlying connection.

The alternative hypothesis assumes there is a significant difference in hospital readmission among variables studied. This means that variables such as are associated with an increase or decrease in hospital readmission.

By conducting a multivariate logistic regression model along with the chi-squared technique analysis will allow for the significant variables to be examined and as the chi-independence test

states if the p-value <= alpha value of 0.05, the null hypothesis is rejected and the alternative hypothesis is accepted, which means the two variables are dependent on one another.

B. Data Collection

The data will be downloaded publicly as a csv file from kaggle.com which shows a reduced data set of history of hospital readmission data delineated by various measures of diabetes diagnosis. This data has been encoded based on the UC Irvine Machine Repository provided by the BioMed Research International for the history of clinical care at 130 US hospitals and readmission data. The data set before removing the records contains 50 attributes and 101,766 instances whereas the dataset from Kaggle contains 17 attributes and 25,000 instances. The data is a ten year history of hospital readmission data delineated by various measures of diabetes diagnosis and has 0% sparseness.

The original dataset represents ten years (1999-2008) of clinical care at 130 US hospitals and integrated delivery networks. The BioMed Research International has made this information publicly available. There is no personal patient information that could personally identify an individual. Overall the dataset is of good quality, as it has been collected and published publicly for public use. The data set includes continuous and categorical variables. Any columns that are inconsistent and are not needed in executing the analysis will be considered to be removed. Clean data simplifies analyses and can accelerate model training. Algorithms converge faster and require less computational power when trained on clean datasets (Tate, 2023).

The kaggle data set includes the following variables:

Field	Description	Data Type
Age	age bracket of the patient	continuous
Time_in_hospital	days (from 1 to 14)	continuous
N_procedures	number of procedures performed during the hospital stay	continuous
n_lab_procedures	number of laboratory procedures performed during the hospital stay	continuous
n_medications	number of medications administered during the hospital stay	continuous
n_outpatient	number of outpatient visits in the year before a hospital stay	continuous
n_inpatient	number of inpatient visits in the year before the hospital stay	continuous

n_emergency	number of visits to the emergency room in the year before the hospital stay	continuous
medical_specialty	the specialty of the admitting physician	categorical
diag_1	primary diagnosis (Circulatory, Respiratory, Digestive, etc.)	categorical
diag_2	secondary diagnosis	categorical
diag_3	additional secondary diagnosis	categorical
glucose_test	whether the glucose serum came out as high, normal, or not performed	categorical
A1Ctest	whether the A1C level of the patient came out as high, normal, or not performed	categorical
change	whether there was a change in the diabetes medication ('yes' or 'no')	categorical
diabete_med	whether a diabetes medication was prescribed ('yes' or 'no')	categorical
readmitted	if the patient was readmitted at the hospital ('yes' or 'no')	categorical

One advantage of the kaggle data set is that it includes a 10 year history of hospital readmission data. In chi-squared analysis, sensitivity to sample size can be a limitation. If the sample size is too large, this could raise problems with hypothesis testing, which could make the overall results less valuable. This analysis will focus on whether the primary diagnosis of a patient and the readmission of a patient are related.

Some disadvantages do exist in the intended study. Exploratory data analysis reveals that hospital readmission is a huge concern across the United States. This analysis will focus on the primary diagnosis of patient readmissions rather than the intended concern of the origin of the data regarding diabetes diagnosis. The chi-square test is an approximate test, and the approximation can be poor when the cell frequencies are low (Limitations to Chi-Square and Exact Alternatives - Categorical Data and Chi-Square Tests | Biostatistics for the Health Sciences, n.d.).

Regarding challenges, there were not any challenges during the data collection process.

C. Data Extraction and Preparation

The programming language used to prepare and clean the medical data set for logistic regression was Python. Python is an interpreted, object-oriented, high-level programming language

with dynamic semantics (What is python? executive summary, n.d.). Even though it is a high-level programming language, the primary disadvantage is that it can be slower than other languages when it comes to execution. Python has very diverse functionalities with its extensive libraries and packages that are designed for every stage of statistical analysis. The main libraries used for data processing and mining are Pandas and Numpy. Sklearn library provides advanced analytics tools for machine learning. Matplotlib and Seaborn libraries provide descriptive data visualization. Stats model and SciPy provide statistical modeling and testing. Pylab is a module used that bulk-imports Matplotlib and Numpy for convenience. Also Scikit-learn is used to build machine learning models and model evaluation.

```
[195]: import pandas as pd
       import numpy as np
       import matplotlib.pyplot as plt
       import seaborn as sns
       %matplotlib inline
       pd.set_option('display.max_columns', None)
       df=pd.read_csv('hospital_readmissions 3.csv')
```

```
[197]: import pylab
       from pylab import rcParams
       import statsmodels.api as sm
       import statistics
       from scipy import stats
```

```
[199]: import sklearn
       from sklearn.preprocessing import StandardScaler
       from sklearn.feature_selection import RFE
       from sklearn.linear_model import LogisticRegression
       from sklearn.model_selection import train_test_split
       from sklearn import metrics
       from sklearn.metrics import classification_report,confusion_matrix
       from scipy.stats import chisquare
       from scipy.stats import chi2_contingency
```

The beginning steps of data preparation involved examination of the data set and identifying if there are any missing, duplicated, or outliers and either replacing them with zero, mean, or median, or removing them from the data set.

```
[201]: #examine datas structure, size, and columns
       df.head()
       df.info()
       df.columns
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25000 entries, 0 to 24999
Data columns (total 17 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   age                25000 non-null  object
 1   time_in_hospital   25000 non-null  int64 
 2   n_lab_procedures   25000 non-null  int64 
 3   n_procedures       25000 non-null  int64 
 4   n_medications      25000 non-null  int64 
 5   n_outpatient       25000 non-null  int64 
 6   n_inpatient        25000 non-null  int64 
 7   n_emergency        25000 non-null  int64 
 8   medical_specialty  25000 non-null  object
 9   diag_1             25000 non-null  object
 10  diag_2             25000 non-null  object
 11  diag_3             25000 non-null  object
 12  glucose_test       25000 non-null  object
 13  A1Ctest            25000 non-null  object
 14  change             25000 non-null  object
 15  diabetes_med       25000 non-null  object
 16  readmitted         25000 non-null  object
dtypes: int64(7), object(10)
memory usage: 3.2+ MB
```

```
[201]: Index(['age', 'time_in_hospital', 'n_lab_procedures', 'n_procedures',
       'n_medications', 'n_outpatient', 'n_inpatient', 'n_emergency',
       'medical_specialty', 'diag_1', 'diag_2', 'diag_3', 'glucose_test',
       'A1Ctest', 'change', 'diabetes_med', 'readmitted'],
      dtype='object')
```

```
[203]: #duplicates
       df.duplicated().sum()
```

```
[203]: 0
```

To further examine the distribution of the dataset, the descriptive statistics were viewed by examining various statistical measures.

[205]:
```
#missing values
df.isnull().sum()
```

[205]:
```
age                  0
time_in_hospital     0
n_lab_procedures     0
n_procedures         0
n_medications        0
n_outpatient         0
n_inpatient          0
n_emergency          0
medical_specialty    0
diag_1               0
diag_2               0
diag_3               0
glucose_test         0
A1Ctest              0
change               0
diabetes_med         0
readmitted           0
dtype: int64
```

[207]:
```
df.describe()
```

[207]:

	time_in_hospital	n_lab_procedures	n_procedures	n_medications	n_outpatient	n_inpatient	n_emergency
count	25000.00000	25000.00000	25000.000000	25000.000000	25000.000000	25000.000000	25000.000000
mean	4.45332	43.24076	1.352360	16.252400	0.366400	0.615960	0.186600
std	3.00147	19.81862	1.715179	8.060532	1.195478	1.177951	0.885873
min	1.00000	1.00000	0.000000	1.000000	0.000000	0.000000	0.000000
25%	2.00000	31.00000	0.000000	11.000000	0.000000	0.000000	0.000000
50%	4.00000	44.00000	1.000000	15.000000	0.000000	0.000000	0.000000
75%	6.00000	57.00000	2.000000	20.000000	0.000000	1.000000	0.000000
max	14.00000	113.00000	6.000000	79.000000	33.000000	15.000000	64.000000

[209]:
```
df.head()
```

[209]:

	age	time_in_hospital	n_lab_procedures	n_procedures	n_medications	n_outpatient	n_inpatient	n_emergency	medical_specialty
0	[70-80)	8	72	1	18	2	0	0	Missing
1	[70-80)	3	34	2	13	0	0	0	Other
2	[50-60)	5	45	0	18	0	0	0	Missing
3	[70-80)	2	36	0	12	1	0	0	Missing
4	[60-70)	1	42	0	7	0	0	0	InternalMedicine

[211]:
```
df.columns
```

[211]:
```
Index(['age', 'time_in_hospital', 'n_lab_procedures', 'n_procedures',
       'n_medications', 'n_outpatient', 'n_inpatient', 'n_emergency',
       'medical_specialty', 'diag_1', 'diag_2', 'diag_3', 'glucose_test',
       'A1Ctest', 'change', 'diabetes_med', 'readmitted'],
      dtype='object')
```

[254]:
```
CategoricalData = df.select_dtypes(include = "object").columns
print(CategoricalData)
```

```
Index(['change', 'diabetes_med', 'readmitted'], dtype='object')
```

Data transformation was performed by re-expressing the categorical variables into numerical variables which was done by replacing yes and no to represent the values of 1 and 0 so that a logistic regression can be conducted properly. The one-hot-encoding method was used to create dummy variables for the categorical variables to be expressed as binary values. One disadvantage of one-hot encoding is that it produces multicollinearity among the various variables, lowering the model's accuracy (*What is one-hot encoding* 2021). The df.head() function is used to confirm that all categorical values were transformed to numerical values with the help of the Sklearn library.

[258]:
```
df['change'] = df['change'].replace({'yes': 1, 'no': 0})
df['diabetes_med'] = df['diabetes_med'].replace({'yes': 1, 'no': 0})
df['readmitted'] = df['readmitted'].replace({'yes': 1, 'no': 0})
df
```

[258]:

	age	time_in_hospital	n_lab_procedures	n_procedures	n_medications	n_outpatient	n_inpatient	n_emergency	medical_specialty	diag
0	4	8	72	1	18	2	0	0	1	
1	4	3	34	2	13	0	0	0	2	
2	2	5	45	0	18	0	0	0	1	
3	4	2	36	0	12	1	0	0	1	
4	3	1	42	0	7	0	0	0	3	
...	...	...	...	...	...	...	...	...	...	
24995	5	14	77	1	30	0	0	0	1	
24996	5	2	66	0	24	0	0	0	1	
24997	4	5	12	0	6	0	1	0	1	
24998	4	2	61	3	15	0	0	0	5	
24999	2	10	37	1	24	0	0	0	1	

25000 rows × 17 columns

[217]:
```
data={'age':['[40-50)','[50-60)','[60-70)','[70-80)','[80-90)','[90-100)']}
data = pd.DataFrame(data)
from sklearn.preprocessing import OneHotEncoder
encoder = OneHotEncoder()
encoded_results = encoder.fit_transform(df).toarray()
print(encoded_results)
```

```
[[0. 0. 0. ... 1. 1. 0.]
 [0. 0. 0. ... 1. 1. 0.]
 [0. 1. 0. ... 1. 0. 1.]
 ...
 [0. 0. 0. ... 0. 0. 1.]
 [0. 0. 0. ... 1. 1. 0.]
 [0. 1. 0. ... 0. 0. 1.]]
```

[219]:
```
data={'medical_specialty':['Missing','Other','InternalMedicine','Surgery','Family/GeneralPractice','Cardiology','Emergency/Traum
data = pd.DataFrame(data)
encoder = OneHotEncoder()
encoded_results = encoder.fit_transform(df).toarray()
print(encoded_results)
```

```
[[0. 0. 0. ... 1. 1. 0.]
 [0. 0. 0. ... 1. 1. 0.]
 [0. 1. 0. ... 1. 0. 1.]
 ...
 [0. 0. 0. ... 0. 0. 1.]
 [0. 0. 0. ... 1. 1. 0.]
 [0. 1. 0. ... 0. 0. 1.]]
```

[221]:
```
data={'diag_1':['Circulatory','Diabetes','Digestive','Injury','Missing','Other','Musculoskeletal','Respiratory']}
data = pd.DataFrame(data)
encoder = OneHotEncoder()
encoded_results = encoder.fit_transform(df).toarray()
print(encoded_results)
```

```
[[0. 0. 0. ... 1. 1. 0.]
 [0. 0. 0. ... 1. 1. 0.]
 [0. 1. 0. ... 1. 0. 1.]
 ...
 [0. 0. 0. ... 0. 0. 1.]
 [0. 0. 0. ... 1. 1. 0.]
 [0. 1. 0. ... 0. 0. 1.]]
```

[223]:
```
data={'diag_2':['Circulatory','Diabetes','Digestive','Injury','Missing','Other','Musculoskeletal','Respiratory']}
data = pd.DataFrame(data)
encoder = OneHotEncoder()
encoded_results = encoder.fit_transform(df).toarray()
print(encoded_results)
```

```
[[0. 0. 0. ... 1. 1. 0.]
 [0. 0. 0. ... 1. 1. 0.]
 [0. 1. 0. ... 1. 0. 1.]
 ...
 [0. 0. 0. ... 0. 0. 1.]
 [0. 0. 0. ... 1. 1. 0.]
 [0. 1. 0. ... 0. 0. 1.]]
```

[225]:
```
data={'diag_3':['Circulatory','Diabetes','Digestive','Injury','Missing','Other','Musculoskeletal','Respiratory
data = pd.DataFrame(data)
encoder = OneHotEncoder()
encoded_results = encoder.fit_transform(df).toarray()
print(encoded_results)
```

```
[[0. 0. 0. ... 1. 1. 0.]
 [0. 0. 0. ... 1. 1. 0.]
 [0. 1. 0. ... 1. 0. 1.]
 ...
 [0. 0. 0. ... 0. 0. 1.]
 [0. 0. 0. ... 1. 1. 0.]
 [0. 1. 0. ... 0. 0. 1.]]
```

[227]:
```
data={'A1Ctest':['no','high','normal']}
data = pd.DataFrame(data)
encoder = OneHotEncoder()
encoded_results = encoder.fit_transform(df).toarray()
print(encoded_results)
```

```
[[0. 0. 0. ... 1. 1. 0.]
 [0. 0. 0. ... 1. 1. 0.]
 [0. 1. 0. ... 1. 0. 1.]
 ...
 [0. 0. 0. ... 0. 0. 1.]
 [0. 0. 0. ... 1. 1. 0.]
 [0. 1. 0. ... 0. 0. 1.]]
```

[229]:
```
data={'glucose_test':['no','high','normal']}
data = pd.DataFrame(data)
encoder = OneHotEncoder()
encoded_results = encoder.fit_transform(df).toarray()
print(encoded_results)
```

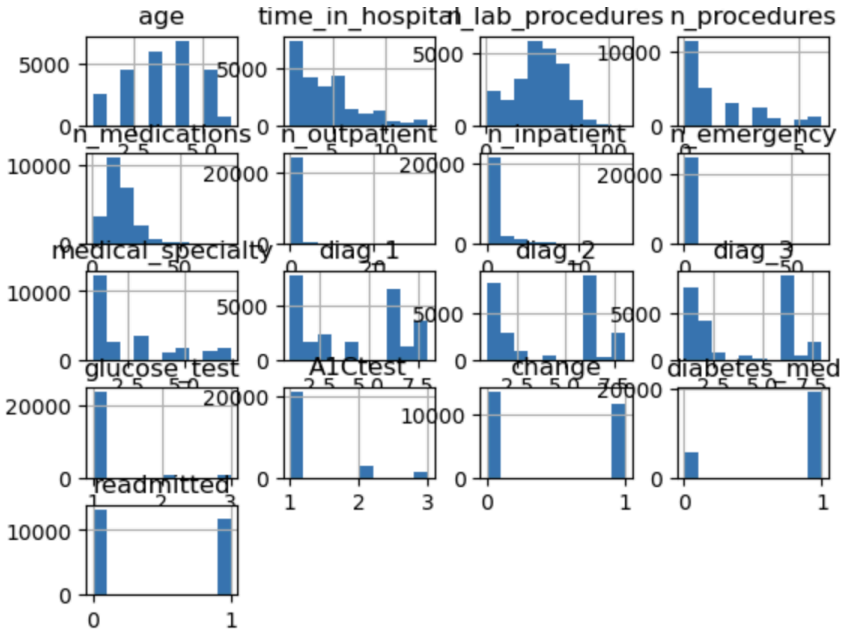
```
[[0. 0. 0. ... 1. 1. 0.]
 [0. 0. 0. ... 1. 1. 0.]
 [0. 1. 0. ... 1. 0. 1.]
 ...
 [0. 0. 0. ... 0. 0. 1.]
 [0. 0. 0. ... 1. 1. 0.]
 [0. 1. 0. ... 0. 0. 1.]]
```

[231]:
```
df['age']= df['age'].replace(['[40-50)','[50-60)','[60-70)','[70-80)','[80-90)','[90-100)'],[1,2,3,4,5,6])
df['medical_specialty'] = df['medical_specialty'].replace(['Missing','Other','InternalMedicine','Surgery','Family/GeneralPractic
df['diag_1'] = df['diag_1'].replace(['Circulatory','Diabetes','Digestive','Injury','Missing','Other','Musculoskeletal','Respirat
df['diag_2'] = df['diag_2'].replace(['Circulatory','Diabetes','Digestive','Injury','Missing','Other','Musculoskeletal','Respirat
df['diag_3'] = df['diag_3'].replace(['Circulatory','Diabetes','Digestive','Injury','Missing','Other','Musculoskeletal','Respirat
df['A1Ctest'] = df['A1Ctest'].replace(['no','high','normal'],[1,2,3])
df['glucose_test'] = df['glucose_test'].replace(['no','high','normal'],[1,2,3])
```

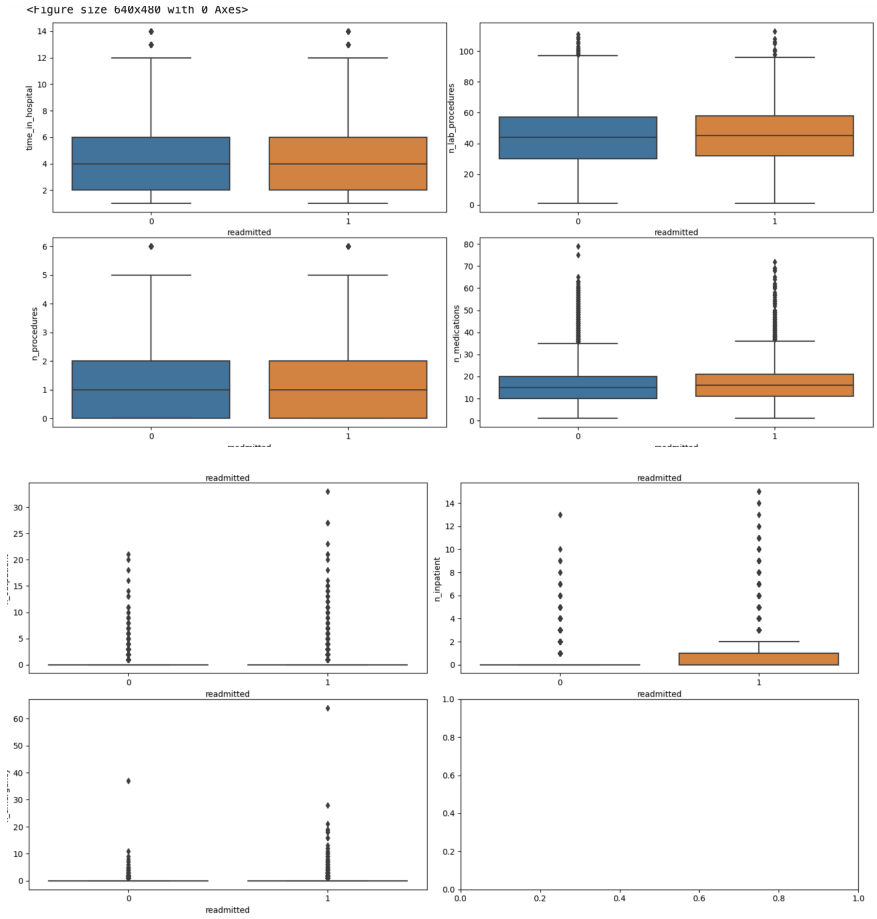
Exploratory Data Analysis, Multivariate Analysis, and the Chi-square statistical methods were used to determine the relationship between the individual hospital variables and the readmission of a patient. This analysis technique allows for pattern identification and understanding of the relationships between variables. Exploratory Data Analysis will help to visualize the relevant variables for a logistic regression model to perform multivariate analysis. In logistic regression, only the probability of an independent variable is being evaluated, against the dependent variable which in this case readmission. Examinations of the models were assessed by utilizing confusion matrices, p-values, variance inflation factors, and classification reports to ensure the validity of the regression analysis.

Univariate and bivariate statistical graphs were also used to help visualize the analysis of different continuous and categorical variables to determine the relationship of the two. Bivariate analysis looks at the relationship between two (‘bi’) variables (‘variates’). As bivariate analysis allows you to take a closer look at the relationship between your outcome (or dependent) variable and any potential explanatory (or independent) variables (Bivariate Analysis ,n.d.).

Univariate visualizations on all variables.



Bivariate visualizations on continuous variables.



Bivariate visualizations on categorical variables.

[276]:
```
fig, axs = plt.subplots(3, 3, figsize=(15, 15))
fig.tight_layout()
plt.subplots_adjust(wspace=0.5)

sns.countplot(data = df, y = 'age', hue = 'readmitted', ax = axs[0, 0])
sns.countplot(data = df, y = 'medical_specialty', hue = 'readmitted',ax = axs[0, 1])
# axs[0, 1].tick_params(rotation= 15)
sns.countplot(data = df, y = 'diag_1', hue = 'readmitted',ax = axs[0, 2])
sns.countplot(data = df, y = 'diag_2', hue = 'readmitted',ax = axs[1, 0])
sns.countplot(data = df, y = 'diag_3', hue = 'readmitted',ax = axs[1, 1])
sns.countplot(data = df, y = 'glucose_test', hue = 'readmitted',ax = axs[1, 2])
sns.countplot(data = df, y = 'A1Ctest', hue = 'readmitted',ax = axs[2, 0])
sns.countplot(data = df, y = 'change', hue = 'readmitted',ax = axs[2, 1])
sns.countplot(data = df, y = 'diabetes_med', hue = 'readmitted',ax = axs[2, 2])
```

[276]: <Axes: xlabel='count', ylabel='diabetes_med'>

D. Analysis

To determine the significant variables that statistically influence hospital readmission, this analysis utilized multivariate statistical analysis to visually analyze the significant difference between variables and identify which variables mostly influence patient readmission. Multivariate analysis sometimes requires more complex computations to arrive at an answer (*Multivariate analysis — definition, methods, and examples*, n.d). The multivariate analysis technique used was logistic regression, which is used to analyze the significance between a binary dependent variable and independent variable. Logistic regression will not perform well with independent variables that are not correlated to the target variable and are very similar or correlated to each other (Navlani, Python logistic regression tutorial with Sklearn & Scikit 2019).

The chi-square test was used to compare observed results with expected results to assess trained regression models. The Chi-square test does not give much information on the strength of the relationship between variables, so one could not assume the correlation of variables by only running a Chi-Square test.

The initial logistic regression model was constructed which contained 16 independent variables to be analyzed against the dependent variable, readmitted. The initial model has a Pseudo R squared value of 0.04790. This value is on a scale from 0 to 1 and when the value is closer to 1 equals a better fit. The value given could suggest that the majority of the variables are not strongly correlated. To visualize where multicollinearity is present and reduce which variables are more significant than others, a confusion matrix will be used. The LLR p-value is 0.000, and due to this value being less than 0.05 it could suggest there is evidence of the model being useful and there is a significant effect on the dependent variable.

```
[287]: df.to_csv(r'hospital_readmissions_d214final.csv')

[289]: data=pd.read_csv(r'hospital_readmissions_d214final.csv')

[461]: df['Intercept'] = 1

[463]: logit_model=sm.Logit(df['readmitted'],df[['age', 'time_in_hospital','n_lab_procedures','n_procedures','n_medications', 'n_outpat
           'medical_specialty', 'diag_1', 'diag_2', 'diag_3', 'glucose_test','Intercept',
           'A1Ctest', 'change', 'diabetes_med']]).fit()
       print(logit_model.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.658245
         Iterations 6
                           Logit Regression Results
==============================================================================
Dep. Variable:             readmitted   No. Observations:                25000
Model:                          Logit   Df Residuals:                    24983
Method:                           MLE   Df Model:                           16
Date:                Fri, 03 May 2024   Pseudo R-squ.:                0.04790
Time:                        23:33:32   Log-Likelihood:                -16456.
converged:                       True   LL-Null:                       -17284.
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================
                         coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
age                    0.0471      0.010      4.623      0.000       0.027       0.067
time_in_hospital       0.0163      0.005      3.190      0.001       0.006       0.026
n_lab_procedures       0.0020      0.001      2.671      0.008       0.001       0.003
n_procedures          -0.0478      0.009     -5.463      0.000      -0.065      -0.031
n_medications          0.0021      0.002      1.023      0.306      -0.002       0.006
n_outpatient           0.1233      0.013      9.342      0.000       0.097       0.149
n_inpatient            0.3848      0.014     26.725      0.000       0.357       0.413
n_emergency            0.2165      0.025      8.565      0.000       0.167       0.266
medical_specialty     -0.0028      0.007     -0.421      0.673      -0.016       0.010
diag_1                -0.0246      0.005     -4.761      0.000      -0.035      -0.014
diag_2                -0.0100      0.005     -1.950      0.051      -0.020    5.14e-05
diag_3                -0.0104      0.005     -1.971      0.049      -0.021   -5.75e-05
glucose_test           0.0140      0.037      0.376      0.707      -0.059       0.087
Intercept             -0.6611      0.086     -7.720      0.000      -0.829      -0.493
A1Ctest               -0.0677      0.026     -2.559      0.010      -0.120      -0.016
change                 0.0425      0.031      1.371      0.170      -0.018       0.103
diabetes_med           0.2432      0.036      6.676      0.000       0.172       0.315
==============================================================================
```

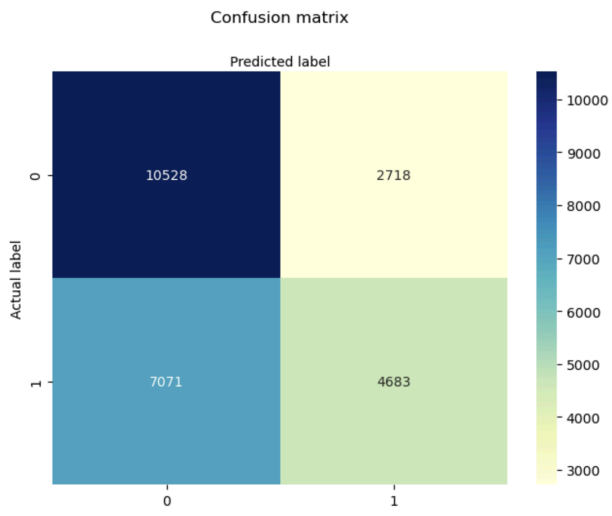
The p-values of each independent variable with p-values lower than 0.05 will be the most influential on the dependent variable. The p-values above 0.05 will most likely not be influential to

the analysis. When evaluating the p-values of the initial model, most of the chosen independent variables were above 0.05.

```
[465]: #split dataset in features and target variable
feature_cols = ['age', 'time_in_hospital','n_lab_procedures','n_procedures','n_medications', 'n_outpatient', 'n_inpatient', 'n_e
       'medical_specialty', 'diag_1', 'diag_2', 'diag_3', 'glucose_test',
       'A1Ctest', 'change', 'diabetes_med','Intercept']
X = df[feature_cols] # Features
y = df.readmitted # Target variable

[467]: logreg = LogisticRegression(random_state=16)
logreg.fit(X, y)
y_pred = logreg.predict(X)

/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/sklearn/linear_model/_logistic.py:460: ConvergenceWar
ning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(

[469]: cnf_matrix = metrics.confusion_matrix(y,y_pred)
cnf_matrix

[469]: array([[10528,  2718],
       [ 7071,  4683]])

[471]: print('Actual values', list(y[:16]))
print('Predictions :', list(y_pred[:16]))

Actual values [0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1]
Predictions : [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
```

For model reduction, a confusion matrix was created to see whether a logistic regression model is fit for the data set. If the predicted values are evenly distributed, this will reveal that there is a relationship between the dependent and independent variables. Once the statistical analysis has been performed, the accuracy calculation will need to be calculated to identify the precision and accuracy on the reduced model.

```
[473]: fig, ax=plt.subplots(1)
sns.heatmap(pd.DataFrame(cnf_matrix), annot=True, cmap="YlGnBu" ,fmt='g')
ax.xaxis.set_label_position("top")
plt.tight_layout()
plt.title('Confusion matrix', y=1.1)
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
plt.show();
```



[475]:
```
print('p- values :',logit_model.pvalues)
```

```
p- values : age                   3.780001e-06
time_in_hospital      1.421380e-03
n_lab_procedures      7.558302e-03
n_procedures          4.688182e-08
n_medications         3.061393e-01
n_outpatient          9.497383e-21
n_inpatient          2.426027e-157
n_emergency           1.077239e-17
medical_specialty     6.734053e-01
diag_1                1.930040e-06
diag_2                5.118181e-02
diag_3                4.874052e-02
glucose_test          7.066386e-01
Intercept             1.160078e-14
A1Ctest               1.048710e-02
change                1.702506e-01
diabetes_med          2.448278e-11
dtype: float64
```

[477]:
```
#check for multicollinearity
from statsmodels.stats.outliers_influence import variance_inflation_factor
vif = pd.DataFrame()
vif['VIF'] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
vif['variable'] = X.columns
vif
```

The variance inflation factor (VIF) will also be utilized to detect the multicollinearity among the independent variables and examine the strength correlation of the independent variables. When evaluating the VIF of the initial model, the variables that were close to 1 could suggest that the variables are more likely to not be correlated. One of the main limitations of VIF is that it only measures pairwise correlation between predictors, and it does not capture higher-order multicollinearity (Limitations of VIF and alternative solutions ,n.d)

[477]:

	VIF	variable
0	1.045067	age
1	1.378221	time_in_hospital
2	1.268237	n_lab_procedures
3	1.309840	n_procedures
4	1.579067	n_medications
5	1.038675	n_outpatient
6	1.088477	n_inpatient
7	1.067978	n_emergency
8	1.028873	medical_specialty
9	1.095634	diag_1
10	1.054998	diag_2
11	1.022906	diag_3
12	1.049511	glucose_test
13	1.083678	A1Ctest
14	1.404825	change
15	1.360286	diabetes_med
16	42.640759	Intercept

The reduced logistic regression model was constructed which contained 16 independent variables to be analyzed against the dependent variable, readmitted. The initial model has a Pseudo R squared value of 0.04790. This value is on a scale from 0 to 1 and when the value is closer to 1 equals a better fit. The value given could suggest that the majority of the variables are not strongly correlated. To visualize where multicollinearity is present and reduce which variables are more significant than others, a confusion matrix will be used. The LLR p-value is 0.000, and due to this

value being greater than 0.05 it could suggest there is no evidence of the model being useful and there no significant effect on the dependent variable. For further interpretation of the coefficients of the reduced model, the positive coefficients indicate a higher likelihood of a patient being readmitted. The negative coefficients indicate a lower likelihood of a patient being readmitted.

[480]:
```
reduced_logit_model=sm.Logit(df['readmitted'],df[[ 'age', 'time_in_hospital','n_lab_procedures','n_outpatient', 'n_inpatient', '
print(reduced_logit_model.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.659406
         Iterations 6
                           Logit Regression Results
==============================================================================
Dep. Variable:             readmitted   No. Observations:                25000
Model:                          Logit   Df Residuals:                    24991
Method:                           MLE   Df Model:                            8
Date:                Fri, 03 May 2024   Pseudo R-squ.:                0.04623
Time:                        23:34:10   Log-Likelihood:                -16485.
converged:                       True   LL-Null:                       -17284.
Covariance Type:            nonrobust   LLR p-value:                     0.000
==============================================================================
                     coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
age                0.0572      0.010      5.688      0.000       0.037       0.077
time_in_hospital   0.0112      0.005      2.413      0.016       0.002       0.020
n_lab_procedures   0.0017      0.001      2.453      0.014       0.000       0.003
n_outpatient       0.1249      0.013      9.548      0.000       0.099       0.151
n_inpatient        0.3931      0.014     27.437      0.000       0.365       0.421
n_emergency        0.2167      0.025      8.585      0.000       0.167       0.266
change             0.0433      0.030      1.420      0.156      -0.016       0.103
diabetes_med       0.2484      0.036      6.866      0.000       0.177       0.319
Intercept         -0.9543      0.053    -18.148      0.000      -1.057      -0.851
==============================================================================
```

[482]:
```
print('p- values :',reduced_logit_model.pvalues)
```

```
p- values : age                 1.284023e-08
time_in_hospital     1.581612e-02
n_lab_procedures     1.415243e-02
n_outpatient         1.324720e-21
n_inpatient         1.005419e-165
n_emergency          9.096630e-18
change               1.557226e-01
diabetes_med         6.606174e-12
Intercept            1.323072e-73
dtype: float64
```

The reduced model now had a Pseudo R squared value of 0.04623. The reduced model's value is farther away from 1 which means then independent variables that were removed affected the reduced model in a negative way and may have had a more positive impact than the variables that were chosen.

The LLR p-value was stable at 0.000, same as the initial model. This could further explain that there is evidence of the model being useful.

[486]:
```
#split dataset in features and target variable
feature_cols = ['age', 'time_in_hospital','n_lab_procedures','n_outpatient', 'n_inpatient', 'n_emergency','change', 'diabetes_me
X = df[feature_cols] # Features
y = df.readmitted # Target variable
```

[488]:
```
logreg = LogisticRegression(random_state=16)
logreg.fit(X, y)
y_pred = logreg.predict(X)
y_pred = np.round(y_pred)
```

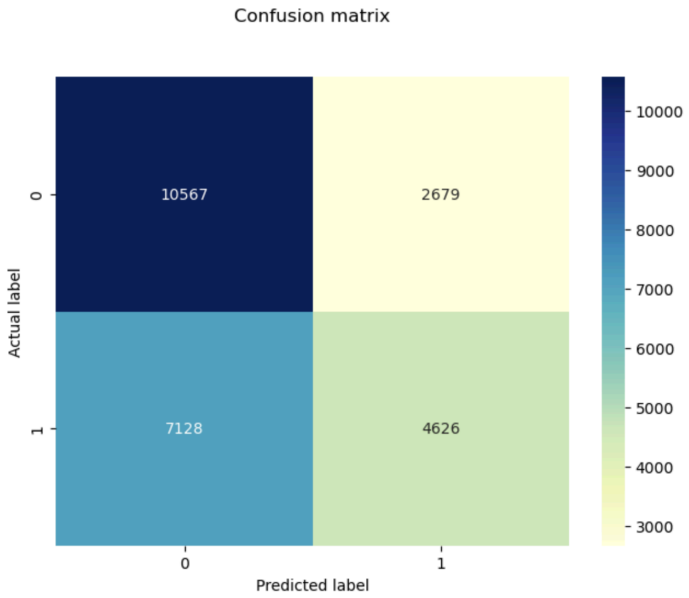
[490]:
```
cnf_matrix = metrics.confusion_matrix(y,y_pred)
cnf_matrix
```

[490]:
```
array([[10567,  2679],
       [ 7128,  4626]])
```

[492]:
```
print('Actual values', list(y[:8]))
print('Predictions :', list(y_pred[:8]))
```

```
Actual values [0, 0, 1, 1, 0, 1, 0, 1]
Predictions : [0, 0, 0, 0, 0, 0, 0, 0]
```

[494]:
```
sns.heatmap(pd.DataFrame(cnf_matrix), annot=True, cmap="YlGnBu" ,fmt='g')
ax.xaxis.set_label_position("top")
plt.tight_layout()
plt.title('Confusion matrix', y=1.1)
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
plt.show();
```



The classification report was generated to visualize the precision, recall, F1, and support scores for the reduced model. Precision reveals the accuracy of positive predictions. Recall reveals the fraction of positives that are correctly identified. F1 reveals a percentage of the positive predictions, and support is the number of actual occurrences of the class in the specified dataset (Kohli, 2019) . All of these performance metrics could be useful but can be misleading if the data is imbalanced.

[497]:
```
target_names = ['no readmitted', 'readmitted']
print(classification_report(y, y_pred, target_names=target_names))
```

```
                precision    recall  f1-score   support

no readmitted       0.60      0.80      0.68     13246
   readmitted       0.63      0.39      0.49     11754

     accuracy                           0.61     25000
    macro avg       0.62      0.60      0.58     25000
 weighted avg       0.61      0.61      0.59     25000
```

[499]:
```
#performing Chi-square test, calculation of p-value and degree of freedom
alpha=0.05
def chi_test(col_1, col_2):
    cont= pd. crosstab (col_1, col_2, margins=False)
    print(cont)
    c, p, dof, expected = chi2_contingency(cont)
    print('p-value = %.2f' %(p))
    print('dof value= %d' % (dof))
    print('expected= %s' %(expected))
    if p <= alpha:
        print('Dependent (Reject Null Hypothesis)')
        print('Independent (Null Hypothesis is True)')
```

[501]:
```
#chi squared method to assess trained regression model observed vs expected
result = chi_test(y, y_pred)
print(result)
```

```
col_0           0     1
readmitted
0           10567  2679
1            7128  4626
p-value = 0.00
dof value= 1
expected= [[9375.5188 3870.4812]
 [8319.4812 3434.5188]]
Dependent (Reject Null Hypothesis)
Independent (Null Hypothesis is True)
```

The Chi-independence test was performed on the reduced model for a further analysis. The final observed versus expected values were evaluated. The value of the chi-square test will be evaluated to determine it is large enough to reject the null hypothesis by analyzing the degree of freedom (dof) value. A chi-square variable with one degree of freedom is equal to the square of the standard normal variable (*Chi-square distribution* ,n.d.). The chi-independence test states if the p-value <= alpha value of 0.05, the null hypothesis is rejected and the alternative hypothesis is accepted, which means the two variables are dependent on one another.

E. Data Summary and Implications

The purpose of the analysis aimed to identify what specific factors statistically influence hospital readmissions using Multivariate and Chi-Squared analysis. This analysis indicated several variables were found to be significant in the prediction on hospital readmission indicated by a p-value less than 0.05 is typically considered to be statistically significant. The variables included: age, time_in_hospital, n_lab_procedures, n_outpatient, n_inpatient, n_emergency, change, diabetes_med. These findings provide insight and direction to the variables that contribute to hospital readmission.

The null hypothesis states there is no significant difference in hospital readmission among variables studied. The alternative hypothesis states there is a significant difference in hospital readmission among variables studied. Based on the p-values found from the logistic regression model and chi-squared techniques, there is a statistical significance in hospital readmission among variables studied. In this case, the null hypothesis is rejected.

The interpretation of coefficients also helped to indicate direction and insight on which variables contribute to patient readmission. The positive coefficients indicate a higher likelihood of a patient being readmitted. The negative coefficients indicate a lower likelihood of a patient being readmitted. The Chi-square was an additional technique added to evaluate the reduced logistic model of observed and expected results which had a p-value of 0.05, and in this case the null hypothesis is rejected.

The reduced logistic regression model did struggle to identify and predict classification. The confusion matrix's classification rate is 61%. The classification rate should be higher in order for it to be a dependable model. This could suggest that the selected variables may not be strong predictors of hospital readmission. Also, the model should be improved to better understand the significance between the independent variables and the dependent variable, readmitted. The limitations of the Chi-Square test is that it assumes random sampling and could possibly skew the data. The Chi-square test does not give much information on the strength of the relationship between variables, so one could not assume the correlation of variables by only running a Chi-Square test. The chi-square test is an approximate test, and the approximation can be poor when the cell frequencies are low (*Limitations to Chi-Square and Exact Alternatives - Categorical Data and Chi-Square Tests | Biostatistics for the Health Sciences*, n.d.).

The findings do provide valuable insights into the significance of hospital characteris and readmission, which could give guidance into future efforts to reduce hospital readmissions. Overall, this analysis would need further research and a better model to get a clearer understanding on which variables contribute to readmission.

Logistic regression is not able to handle a large number of categorical features/variables as it is vulnerable to overfitting which could indicate the low p-values of 0.000 for both models. One approach for a future study would be to complete an additional regression model, such as random forest modeling, that can improve model performance, prediction accuracy, and help discover what factors have an influence on how likely a patient is to be readmitted. Another approach for future study would be utilizing another statistical technique besides the chi-square analysis for further analysis, such an ANOVA test to assess any significant differences in mean values between variables.

F. Sources

What is one-hot encoding. Deepchecks. (2021, August 5). https://deepchecks.com/glossary/one-hot-encoding/#:~:text=Because%20this%20procedure%20generates%20several,variables%2C%20lowering%20the%20model%27s%20accuracy

Multivariate analysis — definition, methods, and examples. (n.d.). https://business.adobe.com/blog/basics/multivariate-analysis-examples

Limitations of VIF and alternative solutions. FasterCapital. (n.d.). https://fastercapital.com/topics/limitations-of-vif-and-alternative-solutions.html#:~:text

=One%20of%20the%20main%20limitations,among%20three%20or%20more%20predictors

Kohli, S. (2019, November 18). *Understanding a classification report for your machine learning model*. Medium. https://medium.com/@kohlishivam5522/understanding-a-classification-report-for-your-machine-learning-model-88815e2ce397

Chi-square distribution. StatsDirect. (n.d.). https://www.statsdirect.co.uk/help/distributions/chi_square_distribution.htm#:~:text=A%20chi%2Dsquare%20variable%20with,the%20central%20limit%20theorem%20dictates

Shalchi, Z., Saso, S., Li, H. K., Rowlandson, E., & Tennant, R. C. (2009, October). Factors influencing hospital readmission rates after Acute Medical Treatment. Clinical medicine (London, England). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4953449/#:~:text=Inadequate%20or%20incomplete%20treatment%20of,care%20to%20community%20services%2C%20as

Hayes, A. (n.d.). Chi-Square (X2) statistic: What it is, examples, how and when to use the test. Investopedia. https://www.investopedia.com/terms/c/chi-square-statistic.asp#:~:text=Limitations%20of%20the%20Chi%2DSquare,a%20causal%20relationship%20with%20another.

Clore,John, Cios,Krzysztof, DeShazo,Jon, and Strack,Beata. (2014). Diabetes 130-US Hospitals for Years 1999-2008. UCI Machine Learning Repository. https://doi.org/10.24432/C5230J.

Limitations to Chi-square and exact alternatives - categorical data and Chi-square tests: Biostatistics for the Health Sciences. pharmacy180.com. (n.d.). https://www.pharmacy180.com/article/limitations-to-chi-square-and-exact-alternatives-2976/

What is python? executive summary. Python.org. (n.d.). https://www.python.org/doc/essays/blurb/

Bivariate Analysis. Bivariate analysis | Practical Applications of Statistics in the Social Sciences | University of Southampton. (n.d.). https://www.southampton.ac.uk/passs/neighbourhood_policing_awareness/bivariate_analysis/index.page

Tate, A. (2023, October 26). The importance of data cleaning in EDA. Hex. https://hex.tech/blog/data-cleaning-exploratory-data-analysis/

GeeksforGeeks. (2023b, June 12). SAS vs R vs python. https://www.geeksforgeeks.org/sas-vs-r-vs-python/

Urach, C., Zauner, G., Wahlbeck, K., Haaramo, P., & Popper, N. (2016, November 18). Statistical methods and modelling techniques for analysing hospital readmission of Discharged Psychiatric Patients: A Systematic Literature Review. BMC psychiatry. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5116202/