Part 1: Research Question

1. Question or Decision

The question I am going to be determining is whether there is any correlation between a person’s age, gender, and income when it comes to hospital re-admissions before comparing these results against two different states.

1. Describe the variables

This data set is a list of 10,000 patients and 50 columns all containing information regarding each patient.

The main information we are looking at are patients that were readmitted to the hospital within a month of release. This is stored in the column “ReAdmis” and is our dependent variable.

The rest of the data are independent variables that consist of patient medical conditions (high blood pressure, stroke, weight, arthritis, diabetes, etc.), patient information (services they received while hospitalized, days spent in the hospital, type of initial admission, etc.) and patient demographic information (gender, age, job, income, etc.) as well as the responses to an eight-question survey given to patients.

The data set contains both numeric/continuous variables such as the patients’ age, number of children, or income as well as categorical/nominal variables such as the patient’s job and the “yes” or “no” entries in columns such as stroke or asthma.

As for the variables and their types that will be used to answer the research question, we’ll be using age as an integer, income as a float, and both gender and state as strings. The remaining variables consist of various integers (CaseOrder, Customer\_id, Interaction, UID, ZIP, Population, Children, Doc\_visits, Full\_meal, Overweight, Anxiety, and the eight survey questions), strings (City, County, Area, Timezone, Job, Education, Employment, Martial, ReAdmis, Soft\_drink, Initial\_admin, HighBlood, Stroke, Complication\_risk, Arthritis, Diabetes, Hyperlipidemia, BackPain, Allergic\_rhinitis, Reflux\_esophagitis, Asthma, and Services), and floats (Lat, Lng, VitD\_levels, Initial\_days, TotalCharge, and Additional\_charges).

Part 2: Data-Cleaning Plan

C1. Plan for identifying anomalies

My approach to cleaning this data set was to keep it simple and straightforward as follows:

1. Import the data with Python
2. Check for which columns had missing values using the isnull().sum() function to see which columns include nulls, this will let me know precisely which columns I need to focus on cleaning
3. Create histograms using the .hist() function of those columns to see the distribution, by seeing the distribution of the data I will be able to determine which method of imputation we’ll be using for each column that needs to be cleaned
4. Fill those missing values via imputation with the .fillna() function and using either the mean, median or mode depending on the distribution from the histograms created in the previous step
5. Create additional histograms and boxplots using seaborn to get a visual representation of remaining outliers, if any. Once we see those we can determine how to deal with them whether we retain them and make a note of their existence, exclude them from upcoming data exploration, replace them with the median values or to simply remove them in the case of a confirmed error

C2. Justification of approach

Given the number of records in our data set it was likely that there would be missing data regardless of the source of the data. Reading the data file, it seems to be missing completely at random with no pattern to which fields are null or not. Considering which columns had variables that were missing, I felt that a rather simple approach would work best for cleaning variables like number of children and age.

C3. Justification of tools

While both Python and R have their own sets of advantages, I have chosen to use Python for this assessment. Since Python is a general-purpose language, it has numerous packages to enhance its usability. I have previous experience in object-oriented programming with Java and C#, so the structure in Python felt fairly familiar. The packages I will be using to support the data-cleaning process are NumPy, pandas, seaborn, SciPy, matplotlib and sklearn. These packages will give functionality needed to import my datasets and work with them making various graphs and so on.

C4. Provide the code

*import numpy as np*

*import pandas as pd*

*import seaborn as sns*

*from pandas import DataFrame*

*import scipy.stats as stats*

*from matplotlib import pyplot as plt*

*%matplotlib inline*

*df = pd.read\_csv('medical\_raw\_data.csv')*

*df.info()*

*df.isnull().sum()*

*df[['Children', 'Age', 'Income', 'Overweight', 'Anxiety', 'Initial\_days']].hist(ec = "black", figsize = (14, 11))*

*#plt.savefig('Histograms Before Imputation.jpg')*

*df['Children'].fillna(df['Children'].median(), inplace = True)*

*df['Income'].fillna(df['Income'].median(), inplace = True)*

*df['Anxiety'].fillna(df['Anxiety'].mean(), inplace = True)*

*df['Age'].fillna(df['Age'].mean(), inplace = True)*

*df['Overweight'].fillna(df['Overweight'].mean(), inplace = True)*

*df['Initial\_days'].fillna(df['Initial\_days'].mean(), inplace = True)*

*df['Soft\_drink'] = df['Soft\_drink'].fillna(df['Soft\_drink'].mode()[0])*

*df.isnull().sum()*

*df[['Children', 'Age', 'Income', 'Overweight', 'Anxiety', 'Initial\_days']].hist(ec = "black", figsize = (14, 11))*

*#plt.savefig('Histograms After Imputation.jpg')*

*df['Children\_Z\_Scores'] = stats.zscore(df['Children'])*

*df['Income\_Z\_Scores'] = stats.zscore(df['Income'])*

*df['Age\_Z\_Scores'] = stats.zscore(df['Age'])*

*df['Initial\_days\_Z\_Scores'] = stats.zscore(df['Initial\_days'])*

*df[['Children\_Z\_Scores', 'Age\_Z\_Scores', 'Income\_Z\_Scores', 'Initial\_days\_Z\_Scores']].hist(ec = "black", figsize = (14, 11))*

*boxplot = sns.boxplot(x = 'Children\_Z\_Scores', data = df)*

*boxplot = sns.boxplot(x = 'Income\_Z\_Scores', data = df)*

*df.to\_csv('medical\_data\_clean.csv')*

Part 3: Data Cleaning

D1. Describe Findings

After importing all my packages, the first thing I did was to use the *isnull().sum()* function to see what columns contained NA or NaN’s. Through that function I discovered that seven columns (Children, Income, Anxiety, Age, Overweight, Initial\_days and Soft\_drink) all had multitudes of missing values. To clean the data, I used imputation to fill in the missing values for these columns using the mean (Anxiety, Age, Overweight, Initial\_days) median (Children, Income) and mode (Soft\_drink). After the data set was cleaned, I searched for and found a few outliers within a few of the columns that were cleaned, which will be discussed more below.

D2. Justify Methods

After finding which columns contained missing values, I created histograms to get a better view of the distribution of variables. I went with imputation using mean, median and mode because it utilized the data distribution and would not reduce my sample size.

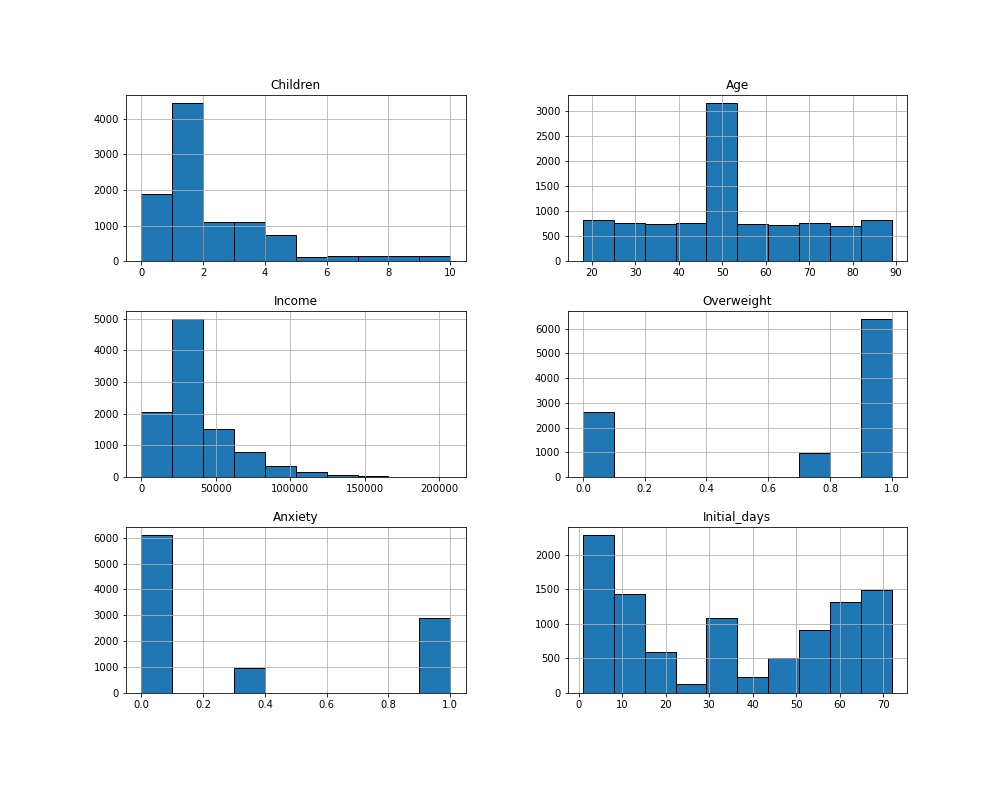
D3. Summarize Outcomes

Running the *isnull().sum()* function showed me the columns with missing values. Seeing that, I made histograms of each to see the data distributions.

Chart, box and whisker chart

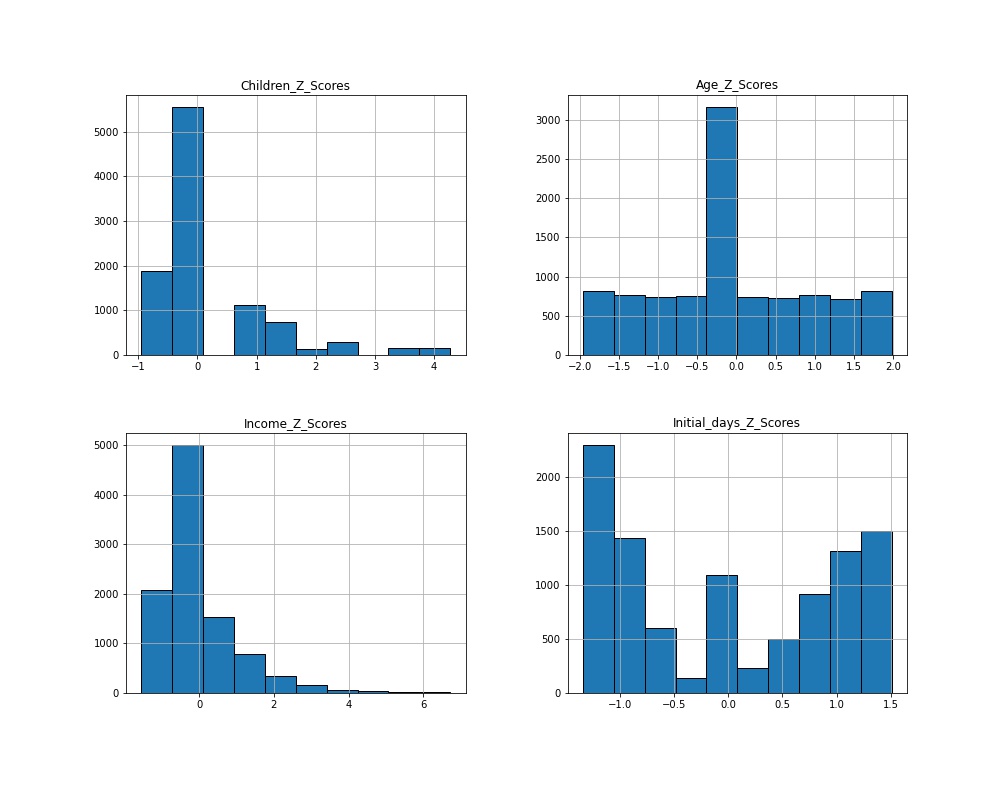
Description automatically generated

As you can see, the Children and Income columns are skewed, so when using imputation, I used the median for those two since it would be a better representation of most of the variables. The Overweight and Anxiety columns strictly contain 0’s and 1’s, the Age column is evenly distributed and the Initial\_days column is bimodal. For these four I decided to use the mean since the median would show nothing on the Overweight and Anxiety columns, Age would be nearly identical and for Initial\_days I felt that using the mean would still overall give the most consistent results. After imputation I ran *isnull().sum()* again to ensure that no columns contained any missing values, then I re-created the histograms to see if my data distribution was still similar to the previous ones.

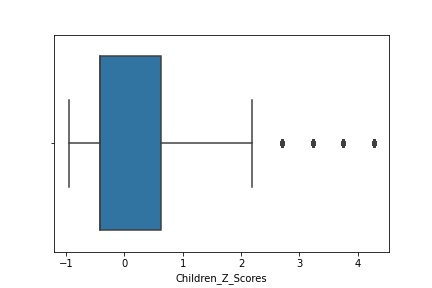


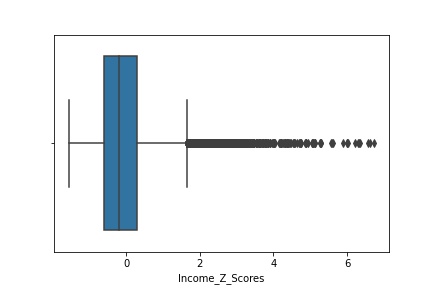
Looking at the new histograms made after imputation, we can see that the Children and Income graphs look nearly identical. The Overweight and Anxiety ones have slightly new values, although this makes sense considering it was the mean of only 1’s and 0’s. The Age column shifted in the middle more than I was expecting, perhaps using the median would have given better results. And for the Initial\_days graph I knew there would be a new bar growing in the middle using the mean, but it seems to still be a fair distribution. My next step was to make

more histograms using the Z-scores to check for any obvious outliers, while omitting the Overweight and Anxiety columns given that all the data for those columns are only 0’s and 1’s there will be no outliers.



With these histograms we can see that the Z-scores for both Age and Initial\_days are well contained within -3 and 3, so no outliers there. Children and Income look to have a few outliers, so I made two boxplots to get another view of them.





While it has become clear we indeed have outliers within those columns, for the moment I chose to leave them in and simply make a note that they exist. I made this decision by considering both the fact this is data for hospital admissions and the number of children someone has, and a person’s income can vary wildly. It could be possible that there are human errors in the data, but it is also very possible these are real results. Imagine a millionaire going into the hospital and putting that as their income, it would indeed skew the results.

D4. Provide the code

All the code, including mitigation, was put in section C4.

D5. Provide a copy of cleaned data set

See attached.

D6. Summarize limitations

When you look at the data-cleaning process, one of the main limitations is the time it takes to clean the data, especially on very large data sets. Human error is also a big limitation, whether the error came from missing values and outliers in the data set to begin with, or an error the analyst made while cleaning the data set. In the second case, this would wind up taking even more time to go back and fix it possibly delaying projects. Lastly, depending on how you clean the data, you could end up removing or losing what would be valuable data which then throws off results of analysis.

D7. How limitations affect analysis

Going back to the original question, in the real world it could be possible to avoid these limitations by being able to get in contact with these hospitals to see if there is a way to find the actual data missing from the set we were given. If not, the limitations on how we will fill in the missing data can affect the analysis by skewing results in a case such as income. In some cases, you might have income values far beyond the normal while also being factually correct, so deciding how to handle filling in the missing values with those still there can affect the analysis.

E. Principal Component Analysis

*#import packages*

*import numpy as np*

*import pandas as pd*

*import seaborn as sns*

*import matplotlib.pyplot as plt*

*from sklearn.decomposition import PCA*

*%matplotlib inline*

*#import clean data set*

*df = pd.read\_csv('medical\_data\_clean.csv', index\_col = 0)*

*df = df[['Population', 'Children', 'Age', 'Income', 'Doc\_visits', 'Initial\_days', 'TotalCharge', 'Additional\_charges', 'Item1', 'Item2', 'Item3', 'Item4', 'Item5', 'Item6', 'Item7', 'Item8']]*

*df.head()*

*#normalize the data*

*df\_normalized = (df - df.mean()) / df.std()*

*pca = PCA(n\_components = df.shape[1])*

*pca.fit(df\_normalized)*

*df\_pca = pd.DataFrame(pca.transform(df\_normalized),*

*columns = ['Population', 'Children', 'Age', 'Income', 'Doc\_visits', 'Initial\_days', 'TotalCharge', 'Additional\_charges', 'Item1', 'Item2', 'Item3', 'Item4', 'Item5', 'Item6', 'Item7', 'Item8'])*

*#create scree plot*

*plt.plot(pca.explained\_variance\_ratio\_)*

*plt.xlabel('number of components')*

*plt.ylabel('explained variance')*

*#plt.savefig('PCA Scree Plot.jpg')*

*plt.show()*

*#get eigenvalues*

*cov\_matrix = np.dot(df\_normalized.T, df\_normalized) / df.shape[0]*

*eigenvalues = [np.dot(eigenvector.T, np.dot(cov\_matrix, eigenvector)) for eigenvector in pca.components\_]*

*#plot eigenvalues as screen plot*

*plt.plot(eigenvalues)*

*plt.xlabel('number of components')*

*plt.ylabel('eigenvalue')*

*#plt.savefig('Eigenvalue Plot.jpg')*

*plt.show()*

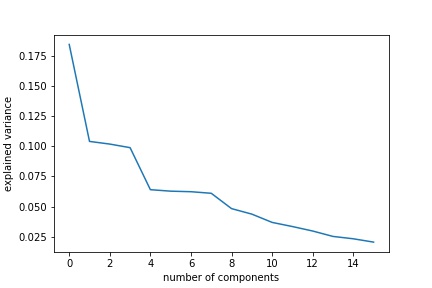
*#generate loading values*

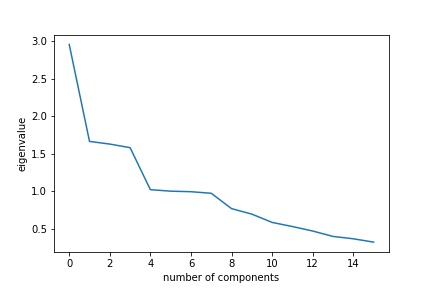
*loadings = pd.DataFrame(pca.components\_.T,*

*columns = ['PC1', 'PC2', 'PC3', 'PC4', 'PC5', 'PC6', 'PC7', 'PC8', 'PC9', 'PC10', 'PC11', 'PC12', 'PC13', 'PC14', 'PC15', 'PC16'],*

*index = df.columns)*

*loadings*





Graphical user interface, text

Description automatically generatedGraphical user interface, text

Description automatically generated

E1. Principal Components

Looking at the scree-plots, we can determine that the first six or seven components are the most important since the eigenvalues for them are above 1.

E2. Identified Components

When you go to identify principal components there are a few different ways to look at it which can depend on the analyst. For scree plots and the chart with eigenvalues, we can see the slope of the graph as a good indicator for which components to use, especially in the case of eigenvalues being equal to or greater than 1.

E3. Benefits

Organizations can benefit from the results of a PCA because it gives a method of obtaining the important variables from a large set of variables and grouping them together to better analyze the data. [4]

Part IV: Supporting Documents

F. Panopto Video

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=f94d4062-08c3-4685-9785-ad950169bf7e

G. Sources for third-party code

[1] Larose, C.D., & Larose, D.T. (2019). *Data science using Python and R.* John Wiley & songs. ISBN: 978-1-119-52684-1

[2] Ekker, R. (2018, May 24). *Pandas playbook: Manipulating data* [Video]. Pluralsight. <https://www.pluralsight.com/courses/pandas-playbook-manipulating-data>. (2h 15m)

[3] https://stackoverflow.com/questions/42172440/python-matplotlib-histogram-color

H. Sources

[4] “PCA: Practical Guide to Principal Component Analysis in R & Python.” *Analytics Vidhya*, 21 Mar. 2016, [www.analyticsvidhya.com/blog/2016/03/pca-practical-guide-principal-component-analysis-python/](http://www.analyticsvidhya.com/blog/2016/03/pca-practical-guide-principal-component-analysis-python/)