Western Governors University

D206 – Data Cleaning Performance Assessment

By: Krista Moik

Table of Contents

Part I: Research Question

Part II: Data-Cleaning Plan

Part III: Data Cleaning

Part IV: Supporting Documents

Part I: Research Question

A. Question or Decision

Using the provided medical_raw_data.csv file, which was collected to predict readmission based on other conditions and factors, my research question is: **Which variables have the most meaningful relationship with patient readmission?**

B. Required Variables

Describe *all* variables in the data set (regardless of the research question) and indicate the data type for *each* variable. Use examples from the data set to support your claims.

By using the code df.info() in our IDE, we can see that the data has 53 columns and 10,000 lines. It is important to use this function in our IDE to confirm the columns instead of just going by the provided data dictionary for the data, which only states there are 50 columns. The columns are our variables. I will use both the provided data dictionary and the IDE to describe all the variables.

1. The first variable, or column, is listed as Unnamed. This variable is a place holder to maintain the original order of the data. The columns consist of numbers. As we are using Python, counting starts at 0, so this is considered Column #0 and looking at our provided CSV medical_raw_data, row 2 is the number 1, row 3 is the number 2. The numbers ascend in order and end at row 10,001. This type of variable is quantitative as it provides a count.
2. Variable 2 is CaseOrder. This variable is quantitative as well and provides the same count and preservation function as our Unnamed first variable. An example of this variable from row 2 is: 1.
3. Variable 3 is Customer_id. An example of this variable from row 2 is: C412403. This column also appears to provide a unique ID in this CSV. This type of variable would be considered qualitative as it is categorical.
4. Variable 4 is Interaction. An example of this variable from row 2 is: 8cd49b13-f45a-4b47-a2bd-173ffa932c2f. This variable provides another type of unique ID related to patient transactions, procedures and admissions. This type of variable is qualitative.
5. Variable 5 is UID. This provides our last column of unique IDs related to patient transactions, procedures, and admissions. An example of this variable from row 2 is: 3a83ddb66e2ae73798bdf1d705dc0932. This type of variable is qualitative.
6. Variable 6 is City, which provides the name of the city the patient resides in, per the billing statement. An example of this variable from row 2 is: Eva. This variable is categorical and thus qualitative.
7. Variable 7 is State, which provides the name of the state the patient resides in, per the billing statement. An example of this variable from row 2 is: AL. This type of variable is qualitative.
8. Variable 8 is County, which provides the name of the county the patient resides in, per the billing statement. An example of this variable from row 2 is: Morgan. This type of variable is qualitative.
9. Variable 9 is Zip, which is the zip code where the patient resides, per the billing statement. An example of this variable from row 2 is: 35621. This type of variable is qualitative.
10. Variable 10 is Lat, which is the latitudinal GPS coordinates of the patient's residence, per the billing statement. An example of this variable from row 2 is: 34.3496. This type of variable is qualitative.
11. Variable 11 is Lng, which is the longitudinal GPS coordinates of the patient's residence, per the billing statement. An example of this variable from row 2 is: -86.72508. This type of variable is qualitative.

12. Variable 12 is Population. This the total number of people within a one-mile radius of the patient's residence, per unofficial census data. An example of this variable from row 2 is: 2951. This type of variable is quantitative.
13. Variable 13 is Area. This is the classification of the area using three descriptors: rural, urban, or suburban. An example of this variable from row 2 is: Suburban. This type of variable is qualitative.
14. Variable 14 is TimeZone. This is the actual time zone where the patient's residence is based on the admission information. An example of this variable from row 2 is: America/Chicago. This variable is qualitative.
15. Variable 15 is Job. This is the job of the patient or the primary insurance holder, per the information obtained during admission. An example of this variable from row 2 is: Psychologist, sport and exercise. This variable is qualitative.
16. Variable 16 is Children. This variable provides the number of children in the patient's household, as obtained during admission. An example of this variable from row 2 is: 1. This variable is quantitative.
17. Variable 17 is Age. This is the reported age of the patient at the time of admission. An example of this variable from row 2 is: 53. This variable is quantitative.
18. Variable 18 is Education. This is the highest earned degree the patient reported obtaining during the admissions process. An example of this variable from row 2 is: Some College, Less than 1 Year. This variable is qualitative.
19. Variable 19 is Employment. This is the employment status of the patient as obtained during the admissions process. An example of this variable from row 2 is: Full Time. This variable is qualitative.
20. Variable 20 is Income. This is the reported income of the patient, or primary insurance holder. An example of this variable from row 2 is: 86575.93. This variable is quantitative.
21. Variable 21 is Marital. This is the marital status of the patient or primary insurance holder as obtained during the admission process. An example of this variable from row 2 is: Divorced. This variable is qualitative.
22. Variable 22 is Gender. This is the patient's self-identification as male, female, or nonbinary. An example of this variable from row 2 is: Male. This variable is qualitative.
23. Variable 23 is ReAdmis. This column indicates yes or no as to whether the patient has been readmitted to the hospital within a month of release. An example of this variable from row 2 is: No. This variable is qualitative.
24. Variable 24 is VitD_levels. This is the patient's vitamin D levels that are measured in and written as ng/ml. An example of this variable from row 2 is: 17.80233049. This variable is quantitative.
25. Variable 25 is Doc_visits. This is a count of the number of times the primary physician visited the patient during the initial hospitalization. An example of this variable from row 2 is: 6. This variable is quantitative.
26. Variable 26 is Full_meals_eaten. This provides account of the number of full meals that patient consumed during their hospitalization. It is noted that partial meals are counted as 0 and indicates that patients could request and consume more than three meals a day. An example of this variable from row 2 is: 0. This variable is quantitative.
27. Variable 27 is VitD_supp. This variable counts the number of times that Vitamin D supplements were given to the patient. An example of this variable from row 2 is: 0. This variable is quantitative.
28. Variable 28 is Soft_drink. This variable shows whether the patient regularly drinks three or more soft drinks per day using either yes or no responses. An example of this variable from row 2 is: NA. This variable is qualitative.
29. Variable 29 is Initial_admin. This variable shows the method in which the patient was initially admitted to the hospital using the classifiers emergency admission, elective admission, and

observation. An example of this variable from row 2 is: Emergency Admission. This variable is qualitative.

30. Variable 30 is HighBlood. This is a yes or no classification as to whether the patient has high blood pressure. An example of this variable from row 2 is: Yes. This variable is quantitative.
31. Variable 31 is Stroke. This provides a yes or no response as to whether the patient has had a stroke. An example of this variable from row 2 is: No. This variable is qualitative.
32. Variable 32 is Complication_risk. This variable uses the classifiers high, medium, and low to indicate the level of complication risk for the patient. An example of this variable from row 2 is: Medium. This variable is qualitative.
33. Variable 33 is Overweight. This is a yes or no indicator as to whether the patient is considered overweight using age, gender, and height. An example of this variable from row 2 is: 0. It is important to note that while the provided data dictionary indicates these are yes or no responses, the yes and no responses are displayed on the CSV as 0s and 1s. While displayed as a number, this variable is in fact qualitative.
34. Variable 34 is Arthritis. This is a yes or no response indicating whether the patient has arthritis. An example of this variable from row 2 is: Yes. This variable is qualitative.
35. Variable 35 is Diabetes. This variable is a yes or no response indicating whether the patient has diabetes. An example of this variable from row 2 is: Yes. This variable is qualitative.
36. Variable 36 is Hyperlipidemia. This variable uses yes or no to indicate if the patient has hyperlipidemia. An example of this variable from row 2 is: No. This variable is qualitative.
37. Variable 37 is BackPain. This variable uses a yes or no response to indicate if the patient has chronic back pain. An example of this variable from row 2 is: Yes. This variable is qualitative.
38. Variable 38 is Anxiety. This provides a yes or no response indicating if the patient has an anxiety disorder. An example of this variable from row 2 is: 1. Even though this variable is displayed numerically, the variable is qualitative.
39. Variable 39 is Allergic_rhinitis. This provides a yes or no indication showing if the patient has allergic rhinitis. An example of this variable from row 2 is: Yes. This variable is qualitative.
40. Variable 40 is Reflux_esophagitis. This is a yes or no indicator as to whether the patient has allergic rhinitis. An example of this variable from row 2 is: No. This variable is qualitative.
41. Variable 41 is Asthma. This variable uses yes or no to indicate if the patient has asthma. An example of this variable from row 2 is: Yes. This variable is qualitative.
42. Variable 42 is Services. This variable indicates the primary services given to the hospitalized patient as broken down into the following services: blood work, intravenous, CT scan, and MRI. An example of this variable from row 2 is: Blood Work. This variable is qualitative.
43. Variable 43 is Initial_days. This is a count of the number of days the patient was hospitalized for during the initial visit. An example of this variable from row 2 is: 10.58576971. This variable is quantitative.
44. Variable 44 is TotalCharge. This variable provides the average amount charged to the patient per day based on the total charge divided by the total number of days the patient was hospitalized. The data dictionary does indicate that this average does not include specialized treatments. An example of this variable from row 2 is: 3191.048774. This variable is quantitative.
45. Variable 45 is Additional_charges. This variable provides the average amount a patient is charged for things like miscellaneous procedures, treatments, medicines, anesthesiology, etc. An example of this variable from row 2 is: 17939.40342. This variable is quantitative.

The next variable are all responses to an 8 question survey given to customers/patients to rate on a scale of 1 to 8 (where 1 is the most important and 8 is least) the importance of 8 different factors.

46. Variable 46 is Item1. Item1 asks the patient to rate the timeliness of admission. An example of this variable from row 2 is: 3. This variable is qualitative.
47. Variable 47 is Item2. This survey question asks the patient to rate the timeliness of treatment. An example of this variable from row 2 is: 3. This variable is qualitative.

48. Variable 48 is Item3. This survey question asks the patient to rate the timeliness of visits. An example of this variable from row 2 is: 2. This variable is qualitative.
49. Variable 49 is Item4. This survey questions asks the patient to rate the reliability. An example of this variable from row 2 is: 2. This variable is qualitative.
50. Variable 50 is Item5. This survey question asks the patient to rate options. An example of this variable from row 2 is: 4. This variable is qualitative.
51. Variable 51 is Item6. This survey question asks the patient to rate the hours of treatment. An example of this variable from row 2 is: 3. This variable is qualitative.
52. Variable 52 is Item7. This survey question asks patients to rate courteous staff. An example of this variable from row 2 is: 3. This variable is qualitative.
53. Variable 53 is Item8. This survey question asks the patient to rate the evidence of active listening from the doctor. An example of this variable from row 2 is: 4. This variable is qualitative.

Part II: Data-Cleaning Plan

Note: You may use Python or R for implementing your coding solutions, manipulating the data, and creating visual representations.

C1. Plan to Assess Quality of Data

Explain the plan for cleaning the data by doing the following:

Propose a plan that includes the relevant techniques and specific steps needed to assess the quality of the data in the data set.

My plan for cleaning the data is to first review the provided data dictionary to get an understanding of the data, what it is showing, and how it is showing it. Then I will import the medical_raw_data CSV into a Jupyter Notebook so I can use Python to further explore, identify, and clean the data as needed. After importing the CSV, I will add the libraries I believe I will be using, like the pandas as pd and numpy as np. As needed, I will add additional libraries. I will use the Python function df.info() to obtain a broad view of the data. I can also use df.describe() to view statistical information of the entire dataset as well as for individual columns as needed. This will allow to me to get a different view of the data and potentially locate outliers, missing values, etc., as well as determine if the datatype is appropriate for the variable.

To determine if duplicate values exist and how many, I can use the Python df.duplicated().value_counts() to see where any duplicates exist and count the duplicates that exist in the data.

For missing values, I will use the Python function df.isnull() and the df.isnull().sum(). This will allow me to locate null, or missing values, and also provide a count of the number of how many exist within the dataset. I will use imputation for the missing values.

Using the data dictionary and my review of variables, including which are quantitative or qualitative. I can add visualization packages like seaborn as sns and matplotlib.pyplot as plt to obtain a visual of the data to locate outliers in the columns that contain quantitative data.

During my current exploration of the data, I know that 1 variable will need to be re-expressed. Variable 28 Soft_drinks is represented by Yes or No instead of 1 or 0. To re-express these variables, I will use ordinal encoding to use 0 to indicate No and 1 to indicate Yes.

Finally, I will use the library from sklearn.decomposition to import PCA to complete my Principal Component Analysis. Once I normalize my quantitative data points, I can define eigenvalues and create a scree plot to visualize the results.

C2. Justification of Approach

Justify your approach for assessing the quality of the data, including the following:

- **characteristics of the data being assessed**
- **the approach used to assess the quality of the data**

The approach I used to assess the characteristics and quality of the data is one of many different possible approaches. My approach involved looking at the data as a whole and then narrowing it down as needed to make sure I had a clear understanding of what the data was showing and whether and what kind of cleaning the data may need. Using the data dictionary along with the exploration of the data using functions such as describe() and visualization like boxplots allowed me to be able to identify the datatypes, and identify duplicates, missing values, and outliers, and complete imputation as needed.

Additionally, while there are many imputation choices, my choice to impute the quantitative variables via the median and the mean was the best method based on the type of distribution. Likewise, imputation via the mode is acceptable for qualitative values. Since the qualitative column responses could only be yes or no/0 or 1, using this technique involved minimal guesswork so there was less chance of data distortion.

I primarily ended up using visualization and boxplots to address the outliers and ultimately chose to retain the outliers as I found the data and outliers to be reasonable. However, this is an instance where my own biases and experiences may be influencing my insight and interpretation into the data.

C3. Justification of Tools

Justify your selected programming language and any libraries and packages that will support the data-cleaning process.

I chose to use Python as the programming language and Jupyter Notebook as the IDE as I do have prior experience using them. Additionally, per Western Governors University Information Technology (2024), Python has easy-to-access libraries, is quick to return results, and has mathematical capabilities useful for data analysis. I used the pandas library as it provides a data frame that allows us to store our CSV. I used numpy for its mathematical function. I used seaborn and matplotlib.plyplot to help visualize the data. I also used the math and scipy.stats as stats for additional mathematical functions to view and clean the data. Finally, I used the library from sklearn.decomposition to complete the PCA analysis and create a scree plot.

C4. Provide the Code

Provide the annotated code you will use to assess the quality of the data in an executable script file.

Please see the attached .ipynb and pdf Jupyter Notebook files titled KMoikD206 code for the full code.

```
#Code used to locate and count duplicates
    df.duplicated().value_counts()
```

```
#Code used to locate and count missing values
    df.isnull().sum()

#Code used to locate outliers using boxplots, histograms, and the describe() function
    sns.boxplot(df, x='Children')
    plt.hist(df[‘Children’])
    plt.show()
    df.Children.describe()

    sns.boxplot(df, x='Age')
    plt.hist(df[‘Age’])
    plt.show()
    df.Age.describe()

    sns.boxplot(df, x='Income')
    plt.hist(df[‘Income’])
    plt.show()
    df.Income.describe()

    sns.boxplot(df, x='Initial_days')
    plt.hist(df[‘Initial_days’])
    plt.show()
    df.Initial_days.describe()
```

Part III: Data Cleaning

D1. Cleaning Findings

Summarize the data-cleaning process by doing the following:
Describe the findings for the data quality issues found from the implementation of the data-cleaning plan from part C.

To check for duplicates, I used the function df.duplicated().value_counts() to locate and count the duplicates. The result was: False 10000. This means no duplicates were found out of all 10,000 rows of our data set.

To check for missing values, I used the function df.isnull().sum() to view location of and obtain a count of null values. This function yielded the following count of null values:

Column: Children	Number of Nulls: 2588
Column: Age	Number of Nulls: 2414
Column: Income	Number of Nulls: 2464
Column: Soft_drink	Number of Nulls: 2467
Column: Overweight	Number of Nulls: 982
Column: Anxiety	Number of Nulls: 984
Column: Initial_days	Number of Nulls: 1056

To detect outliers for the quantitative data, I used seaborn and matplotlib.pyplot libraries to create visualizations after imputing the missing values. Using the function sns.boxplot(df, x=’Children’) to make a boxplot of the Children column, I could now see that the Children column has 4 outliers and has a right skewed distribution. The Income column had many outliers and also has a right skewed distribution. To obtain a count of the outliers in Income, I used the function df['Income'][df['Income'] > 80229.88].count() to determine there were 705 outliers. There were no other outliers located and the other quantitative columns have normal distributions.

D2. Justification of Mitigation Methods

Justify your methods for mitigating the data quality issues in the data set.

As the dataset contained no duplicate values, no additional cleaning was needed in regards to that.

For the missing values, I chose to use imputation using the mean, median, and mode. Imputation via the mean was used for the Age column as it had a normal distribution. Imputation via the median was used for the Children, Income, and Initial_days columns as both Children and Income had right skewed distribution and Initial_days had a bimodal skew. Imputation via the median is sufficient for both of those types of skew and thus was chosen as the imputation method. The remaining columns with nulls were Soft_drink, Overweight, and Anxiety. As these columns contained qualitative variables, I used the imputation via mode function. WGU course videos were used to determine the best imputation method (Western Governors University, n.d.).

Of the missing values, Soft_drink was the only column to need re-expression as responses were indicated as Yes or No instead of 1 or 0 like Overweight and Anxiety. I used ordinal encoding to easily convert the categorical responses to numeric ones.

For the outliers, the Children column showed 4 outliers for the number of children per patient – those who had 7, 8, 9, or 10 children. The count() function determined that 457 patients had more than 6 children and up to 10 children total. The Income column had 705 outliers which means 705 patients had an income greater than $80,229.88. No other outliers were found. In dealing with outliers, it is important to determine whether the outliers are reasonable or not (Western Governors University, n.d.). I found that a small number of patients having up to 10 children and a salary up to $207,249 were reasonable and justifiable based on my own experiences, views on diversity, and potential bias. Due to me finding these outliers justifiable and reasonable, I chose to retain these outliers within the data. By retaining what I felt were reasonable outliers, I preserved the sample size and diversity of the data.

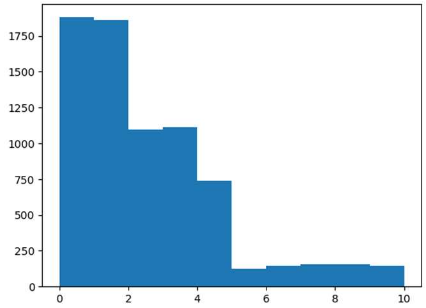
D3. Summary of the Outcomes

Summarize the outcome from the implementation of *each* data-cleaning step.

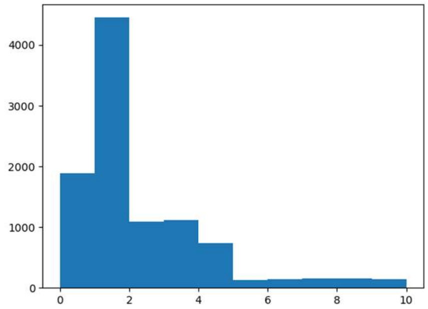
After cleaning the dataset, the new data frame still does not have any duplicate values, and missing values and outliers were treated appropriately.

The dataset prior to cleaning the missing values appeared as below:

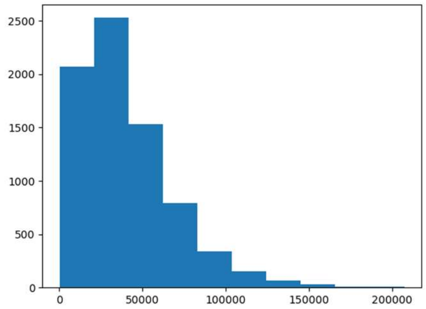
The Children column visualized like this:



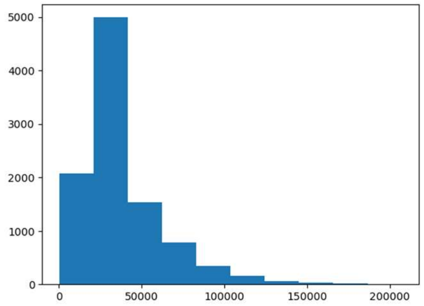
After imputation, the Children column appeared as below:



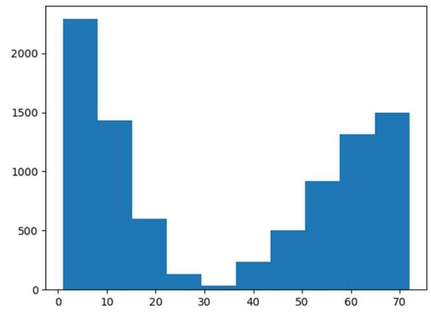
Income initially looked like this:



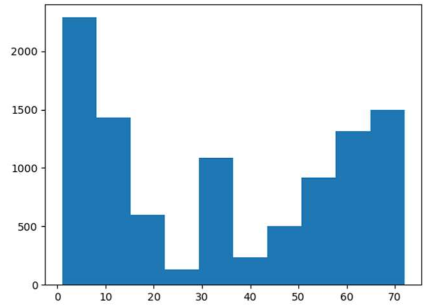
After Imputation, Income appeared like this:



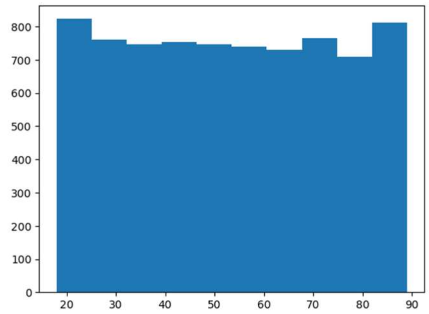
Initial_day originally looked like this:



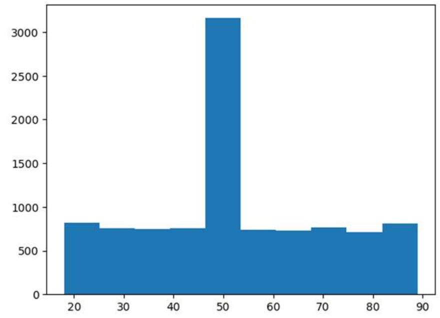
After Imputation, income appeared like this:



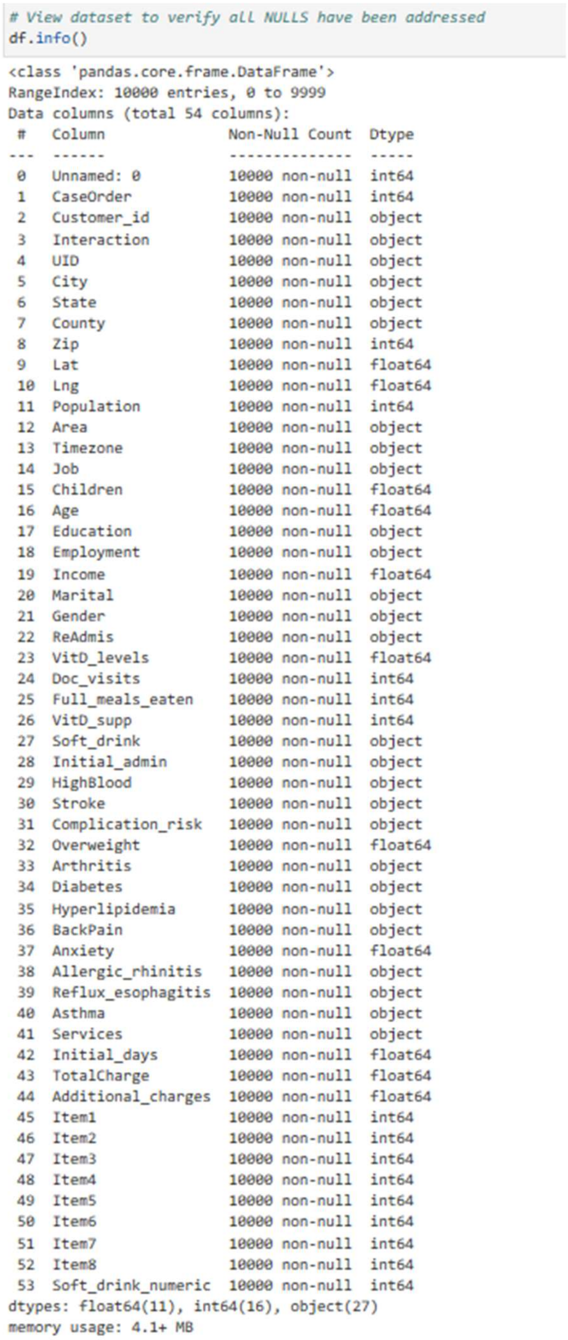
Finally, Age initially appeared like this:



After Imputation:



After imputing the missing values for the quantitative columns, the columns' visualizations changed slightly but retain similar shapes to the original, which confirms our choices as the best approach and that the data was not too changed by our imputation. After imputing via the mode for the qualitative columns, the df.isnull()sum() function returned zero null values. See below:



```
# View dataset to verify all NULLS have been addressed
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 54 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   Unnamed: 0          10000 non-null  int64  
 1   CaseOrder           10000 non-null  int64  
 2   Customer_id         10000 non-null  object 
 3   Interaction         10000 non-null  object 
 4   UID                 10000 non-null  object 
 5   City                10000 non-null  object 
 6   State               10000 non-null  object 
 7   County              10000 non-null  object 
 8   Zip                 10000 non-null  int64  
 9   Lat                 10000 non-null  float64
 10  Lng                 10000 non-null  float64
 11  Population          10000 non-null  int64  
 12  Area                10000 non-null  object 
 13  Timezone            10000 non-null  object 
 14  Job                 10000 non-null  object 
 15  Children            10000 non-null  float64
 16  Age                 10000 non-null  float64
 17  Education           10000 non-null  object 
 18  Employment          10000 non-null  object 
 19  Income              10000 non-null  float64
 20  Marital             10000 non-null  object 
 21  Gender              10000 non-null  object 
 22  ReAdmis             10000 non-null  object 
 23  VitD_levels         10000 non-null  float64
 24  Doc_visits          10000 non-null  int64  
 25  Full_meals_eaten    10000 non-null  int64  
 26  VitD_supp           10000 non-null  int64  
 27  Soft_drink          10000 non-null  object 
 28  Initial_admin       10000 non-null  object 
 29  HighBlood           10000 non-null  object 
 30  Stroke              10000 non-null  object 
 31  Complication_risk   10000 non-null  object 
 32  Overweight          10000 non-null  float64
 33  Arthritis           10000 non-null  object 
 34  Diabetes            10000 non-null  object 
 35  Hyperlipidemia      10000 non-null  object 
 36  BackPain            10000 non-null  object 
 37  Anxiety             10000 non-null  float64
 38  Allergic_rhinitis   10000 non-null  object 
 39  Reflux_esophagitis  10000 non-null  object 
 40  Asthma              10000 non-null  object 
 41  Services            10000 non-null  object 
 42  Initial_days        10000 non-null  float64
 43  TotalCharge         10000 non-null  float64
 44  Additional_charges  10000 non-null  float64
 45  Item1               10000 non-null  int64  
 46  Item2               10000 non-null  int64  
 47  Item3               10000 non-null  int64  
 48  Item4               10000 non-null  int64  
 49  Item5               10000 non-null  int64  
 50  Item6               10000 non-null  int64  
 51  Item7               10000 non-null  int64  
 52  Item8               10000 non-null  int64  
 53  Soft_drink_numeric  10000 non-null  int64  
dtypes: float64(11), int64(16), object(27)
memory usage: 4.1+ MB
```

For outliers, I used boxplots to visualize the quantitative columns and ultimately determined that all the outliers present were reasonable and thus decided to retain all outliers.

D4. Mitigation Code

Provide the annotated code you will use to mitigate the data quality issues—including anomalies—in the data set in an executable script file.

Please see the attached file with the code that was used titled KmoikD206code.ipynb. and Kmoik D206code.pdf

I used the following code, including code from the WGU course materials to mitigate the data quality issues:

As there were no duplicates, there is no code script for mitigation.

#Using code from WGU course materials for the missing values in the quantitative columns, I used the following functions (Western Governors University, n.d.):

```
df[‘Children’].fillna(df[‘Children’].median(), inplace=True)
df[‘Income’].fillna(df[‘Income’].median(), inplace=True)
df[‘Initial_days’].fillna(df[‘Initial_days’].median(), inplace=True)
Df[‘Age’].fillna(df[‘Age’].mean(), inplace=True)
```

#Using code from WGU course materials for the missing values in the qualitative columns, I used the following functions (Western Governors University, n.d.):

```
df[‘Soft_drink’].fillna(‘No’, inplace=True)
df[‘Overweight’].fillna(‘1’, inplace=True)
df[‘Anxiety’].fillna(‘0’, inplace=True)
```

#Using code from WGU course materials to re-express the Soft_drink column using 0s and 1s, I used ordinal encoding (Western Governors University, n.d.):

```
df[‘Soft_drink_numeric’] = df[‘Soft_drink’]
dict_soft_drink = [‘Soft_drink_numeric’ : [‘No’: 0, ‘Yes’: 1}}
df.replace(dict_Soft_drink, inplace=True)
```

As I chose to retain the outliers in the dataset, there is no mitigation code to show.

#Obtained a view of the now-mitigated data set to confirm the changes

```
df.info()
```

D5. Clean Data
Provide a copy of the cleaned data set as a CSV file.

Please see the attached CSV file with my cleaned data titled KMoikclean_medical.csv.

D6. Limitations
Summarize the limitations of the data-cleaning process.

Imputation using the median and mean were used for the quantitative columns with missing values. While this method is beneficial since it does not reduce our dataset like deletion does, it is ultimately a guess as to what the value is supposed to be or should be and can potentially distort the data and the insights we gain from it (Western Governors University, n.d.). There may be additional data available if this was an actual real-world assignment where we could reach out for clarification. Additionally, I chose the method I thought provided the best outcome, but there are many other methods that could provide more accurate replacements.

Imputation using the mode for the qualitative columns provides the same strengths and weaknesses as above. While it is ideal that it does not reduce our dataset, by using this function there is always the potential that we have distorted the data or the distribution of the data, which can result in inaccurate results and analysis.

D7. Impact of Limitations

Discuss how the limitations summarized in part D6 could affect the analysis of the question or decision from part A.

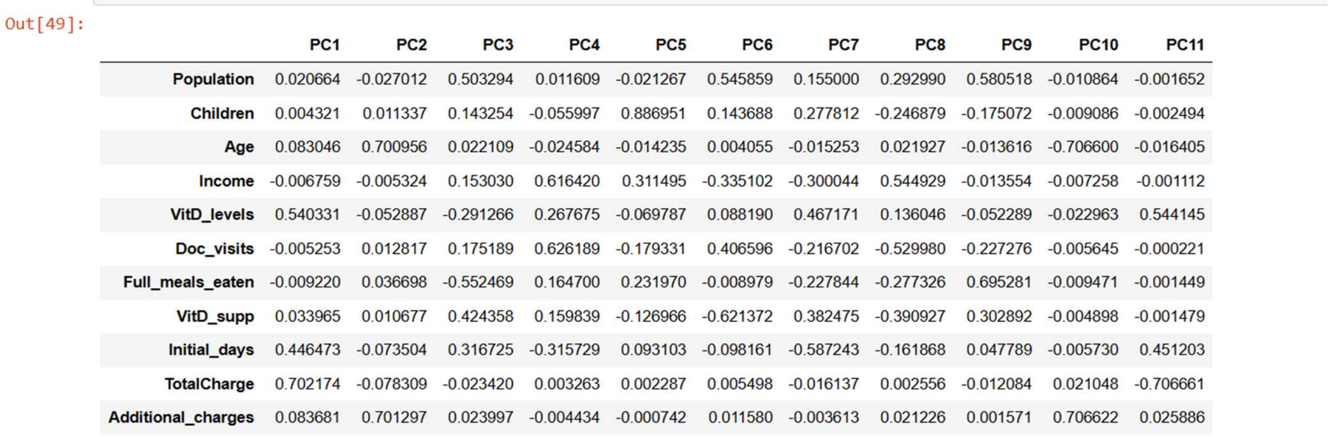
If a data analyst used my cleaned data for analysis, their results could be distorted and even biased due to the imputation methods I used. Specifically in relation to my original research question, the imputed columns may now have incorrect values as a result of the imputation that would affect the analysis of the relationship with readmission. This could result in inaccurate insights that could lead to belief in inaccurate relationships and even poor business decisions.

E1. Principal Components

Apply principal component analysis (PCA) to identify the significant features of the data set by doing the following:

Identify the total number of principal components and provide the output of the principal components loading matrix.

For the PCA, I included the quantitative variables: Population, Children, Age, Income, VitD_levels, Doc_visits, Full_meals_eaten, VitD_supp, Initial_days, TotalCharge, and Additional_charges. The resulting matrix showed 11 PCs. From this, I obtained the below PCA loadings matrix using code provided in WGU course materials (Western Governors University, n.d.):



Out[49]:

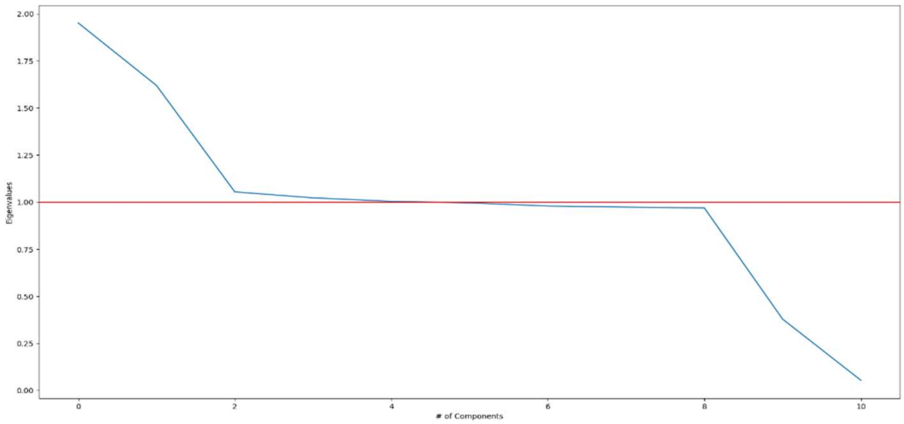
	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11
Population	0.020664	-0.027012	0.503294	0.011609	-0.021267	0.545859	0.155000	0.292990	0.580518	-0.010864	-0.001652
Children	0.004321	0.011337	0.143254	-0.055997	0.886951	0.143688	0.277812	-0.246879	-0.175072	-0.009086	-0.002494
Age	0.083046	0.700956	0.022109	-0.024584	-0.014235	0.004055	-0.015253	0.021927	-0.013616	-0.706600	-0.016405
Income	-0.006759	-0.005324	0.153030	0.616420	0.311495	-0.335102	-0.300044	0.544929	-0.013554	-0.007258	-0.001112
VitD_levels	0.540331	-0.052887	-0.291266	0.267675	-0.069787	0.088190	0.467171	0.136046	-0.052289	-0.022963	0.544145
Doc_visits	-0.005253	0.012817	0.175189	0.626189	-0.179331	0.406596	-0.216702	-0.529980	-0.227276	-0.005645	-0.000221
Full_meals_eaten	-0.009220	0.036698	-0.552469	0.164700	0.231970	-0.008979	-0.227844	-0.277326	0.695281	-0.009471	-0.001449
VitD_supp	0.033965	0.010677	0.424358	0.159839	-0.126966	-0.621372	0.382475	-0.390927	0.302892	-0.004898	-0.001479
Initial_days	0.446473	-0.073504	0.316725	-0.315729	0.093103	-0.098161	-0.587243	-0.161868	0.047789	-0.005730	0.451203
TotalCharge	0.702174	-0.078309	-0.023420	0.003263	0.002287	0.005498	-0.016137	0.002556	-0.012084	0.021048	-0.706661
Additional_charges	0.083681	0.701297	0.023997	-0.004434	-0.000742	0.011580	-0.003613	0.021226	0.001571	0.706622	0.025886

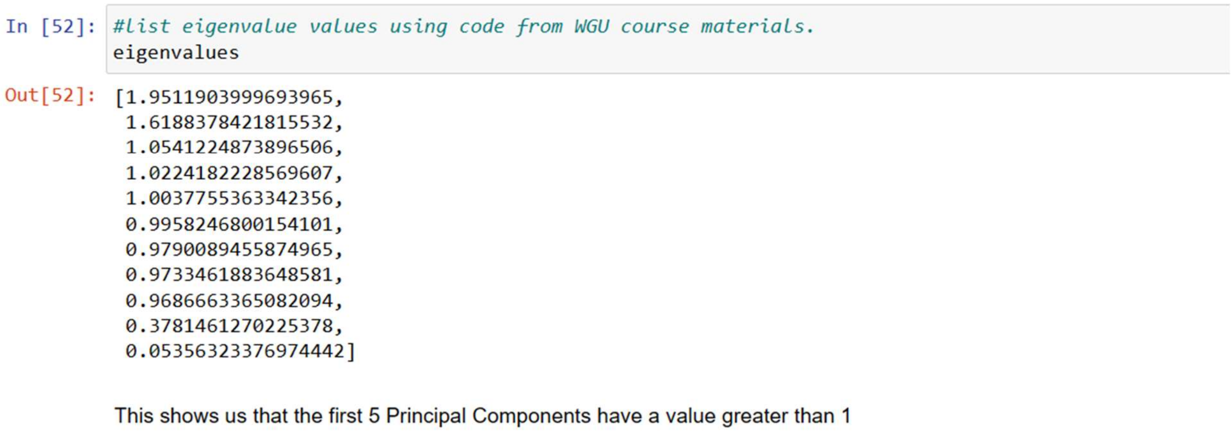
E2. Criteria Used

Justify the reduced number of the principal components and include a screenshot of a scree plot.

Abiding by the Kaiser Rule, which recommends retaining the PCs with eigenvalues greater than or equal to 1 as the most meaningful, I calculated the eigenvalues and then visualized them in a Scree Plot to determine that the most meaningful PCs were the first five as they were all greater than or equal to 1 (Western Governors University, n.d.).

Please see the below scree plot and eigenvalues confirming the most meaningful PCs:





```
In [52]: #list eigenvalue values using code from WGU course materials.
         eigenvalues

Out[52]: [1.9511903999693965,
          1.6188378421815532,
          1.0541224873896506,
          1.0224182228569607,
          1.0037755363342356,
          0.9958246800154101,
          0.9790089455874965,
          0.9733461883648581,
          0.9686663365082094,
          0.3781461270225378,
          0.05356323376974442]
```

This shows us that the first 5 Principal Components have a value greater than 1

E3. Benefits
Describe how the organization would benefit from the use of PCA.

PCA is a useful tool that attempts to locate the variables that possibly have the greatest impact on the data. This is beneficial to every organization as it allows the data analyst to reduce or group variables once patterns and relationships are found between the variables and determine which factors have the greatest impact on specific variable (Western Governors University, n.d.). This allows businesses to obtain more insight into how different variables interact with and affect each other so businesses can use these insights of relatedness to potentially make changes to increase and decrease specific variables relevant to their business goal.

Part IV. Supporting Documents

F. Video
Provide a Panopto video recording that includes the presenter and a vocalized demonstration of the functionality of the code used for the analysis of the programming environment.

Here is the link for my Panopto video recording:
https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=e23f24c8-3cbc-44ab-beba-b0f5014b6932

G. Sources of Third-Party Code
Acknowledge web sources, using in-text citations and references, for segments of third-party code used to support the application. Be sure the web sources are reliable.

The code I used came from the supplied Course Webinars and my own background in Python.

For D4 ordinal encoding: Western Governors University. (n.d.) *Getting Started With D206 Re-expression of Categorical Variables* [Video]. https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=bd7b8541-77ba-42e0-80a4-b059010bc790

For E1&2 PCA: Western Governors University. (n.d.) *Getting Started with D206 PCA* [Video]. https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=3bcc452f-fa35-43be-b69f-b05901356f95

H. Sources
Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized.

The only content summarized or paraphrased was from the WGU course webinars.

For C3: Western Governors University Information Technology. (2024). *R or Python.* Western Governors University. https://www.wgu.edu/online-it-degrees/programming-languages/r-or-python.html

For D2 outliers: Western Governors University. (n.d.) *Getting Started With D206 Outliers* [Video]. https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=19c24c56-0f37-408e-bb1f-b059002a77ac

For D2&6 imputation: Western Governors University. (n.d.) *Getting Started With D206 Missing Values* [Video]. https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=767749d2-ba19-4f94-bec8-b058017b2f5e

For E2&3 PCA: Western Governors University. (n.d.) *Getting Started with D206 PCA* [Video]. https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=3bcc452f-fa35-43be-b69f-b05901356f95

I. Demonstrate professional communication in the content and presentation of your submission.