Matthew E. Heino

D206 Data Cleaning

Introduction

This document discusses the methods and procedures for cleaning data. The data cleaning process is a stage in data analysis that prepares the data for future analysis. Data often is saved in a state that makes it unsuitable for direct analysis. Errors in data collection, errors in the format, and missing data hinder the proper analysis of the data.

This document seeks to lay out steps that can be utilized to transform data into data from a mass of unstructured data to a structured format that can be used to answer an organizational question.

The dataset that is used is the **medical_raw_data.csv.** This dataset is composed of numerous errors and omissions. These will be addressed over the course of this document. Other elements of this document will include a brief discussion of the data. There will be a discussion of the types of data as well as their intended purpose.

There will be an examination of the various components of the dataset. The analysis will be a principal component analysis that attempts to reduce the number of components that are worth analyzing. This will be discussed in a section later in this document.

At the end of this document, you will find a complete code listing that has been annotated with each step and it will correspond to sections covered in this document.

Note: There will be a Jupyter Notebook included with this submission. You will find that the Jupyter Notebook is broken into the steps as they are described in this document. All code that is discussed in this assessment can be found in the Jupyter Notebook.

Part I: Research Question

The medical dataset can be used to ask a wealth of questions that are relevant to the medical organization. The main research question is described in the next section.

A. The Research Question

The question that is being considered for research is the following, "What are the factors that can lead to readmission?" The data in the given CSV can be used to answer the question. It is believed that all the necessary features are available although the state of the features may need to be cleaned to make the data contained in the feature columns usable for analysis. The data cleaning process and how anomalies are remedied are the major focus of the document.

B. The Variables of the Dataset

In the data set, there are many different types of data stored in the CSV file. The CSV file is composed of 10,000 rows along with 53 columns. Code to view the number of rows and columns is (after loading the data from the CSV into a pandas data frame)[1]:

```
print(medical_df.shape)
```

The table on subsequent pages will discuss the variables or features that are found in the dataset.

[1] A full listing of the code for this assessment can be found in Jupyter Notebook.

Num3 – NUM3 Task 1 Data Cleaning

The following is a table that will describe the types of data that were found in the CSV along with a description of the data as given in the supplied data dictionary. This data dictionary was supplied in the course materials.[2] Unless otherwise noted all data was taken from the first row of the CSV file.

Variable name	Example	Description	Qualitative or Quantitative
CaseOrder	1	This is a placeholder to keep the original order of the table	Quantitative
Interaction	8cd49b13-f45a-4b47-a2bd-173ffa932c2f	This is a unique patient ID for transactions and procedures.	Qualitative
Customer_id	C412403	Patient ID.	Qualitative
UID	3a83ddb66e	This is a unique patient ID for transactions and the associated ID	Qualitative

[2]The examples for the variables were taken directly from the CSV file no code was used to create this table.

	2ae73798bd f1d705dc09 32	related to the patient	
City[3]	Eva	City of residence of the patient.	Qualitative
State[4]	AL	State of residence of the patient.	Qualitative
County[5]	Morgan	County of patient	Qualitative
Zip[6]	35621	ZIP code of the patient	Qualitative
Lat	34.3496	Latitude of the patient.	Quantitative

[3] The following is categorical value: City, State, County, and Zip depend on the calculation that needs to be performed
[4] The following is a categorical value: City, State, County, and Zip depend on the calculation that needs to be performed
[5] The following is a categorical City, State, County, and Zip depending on the calculation that needs to be performed
[6] The following is a categorical value: City, State, County, and Zip depends on the calculation that needs to be performed

Lng	-86.72508	Longitude of the patient	Quantitative
Population	2951	Population of the patient's town within one square mile.	Quantitative
Area	Suburban	Type of area that the patient lives in.	Qualitative
Timezone	America/ Chicago	The time zone.	Qualitative
Job	Psychologist, sport and exercise	Type of job the patient has.	Qualitative
Children	1	Number of children the patient has living with them.	Quantitative
Age	53	Age of the patient	Quantitative

Education	Some College, Less than 1 Year	Years of education the patient has.	Qualitative
Employment	Full Time	Type of employment (e.g. full-time, part-time)	Qualitative
Income	86575.93	Patient's income	Quantitative
Marital	Divorced	Marital status of the patient.	Qualitative
Gender	Male	Gender	Qualitative
ReAdmis	No	Was the patient readmitted within the last month.	Qualitative
VitD_levels	17.8023304	Vitamin D levels measured in ng/l.	Quantitative

	9		
Doc_visits	6	Number of visits the patient's primary care physician visited during hospitalization.	Quantitative
Full_meals_eaten	0	Number of full meals eaten during hospitalization.	Quantitative
VitD_supp	0	How many vitamin D supplements were given during hospitalization.	Quantitative
Soft_drink	NA	Does the patient consume more than three sodas a day?	Qualitative
Initial_admin	Emergency Admission	Type of initial hospital admission.	Qualitative
HighBlood	Yes	Does the patient have high blood pressure?	Qualitative

Stroke	No	Does the patient have a history of stroke?	Qualitative
Complication_risk	Medium	The level of risk associated with the patient.	Qualitative
Overweight[7]	0	Is the patient considered to be overweight?	Qualitative
Arthritis	Yes	Does the patient have arthritis?	Qualitative
Diabetes	Yes	Does the patient have diabetes?	Qualitative
Hyperlipidemia	No	Does the patient have hyperlipidemia?	Qualitative
BackPain	Yes	Does the patient have back pain?	Qualitative

[7]Overweight should be changed to "Yes" or "No" to better reflect the similar data in other columns. This change will make it fall inline with what is recorded in the data dictionary.

Anxiety[8]	1	Does the patient have anxiety?	Qualitative
Allergic_rhinitis	Yes	Does the patient have allergic rhinitis?	Qualitative
Reflux_esophagitis	No	Does the patient have reflux esophagitis?	Qualitative
Asthma	Yes	Does the patient have asthma?	Qualitative
Services	Blood Work	Main service the patient received during hospitalization.	Qualitative
Initial_days	10.58577	Number of days the patient was in the hospital.	Quantitative
TotalCharge	3191.048774	Amount charged to the patient on a daily basis – the average.	Quantitative

[8] Anxiety will need to be changed from zero or one to Yes or No to keep consistent with other data in the file.

Num3 – NUM3 Task 1 Data Cleaning

Additional_charges	17939.4034 2	Average amount charged for miscellaneous service while in the hospital.	Quantitative
Item1	3	Timely Admission	Qualitative
Item2	3	Timely treatment	Qualitative
Item3	2	Timely visits	Qualitative
Item4	2	Reliability	Qualitative
Item5	4	Options	Qualitative
Item6	3	Hours of treatment	Qualitative

Item7	3	Courteous staff	Qualitative
Item8	4	Evidence of active listening.	Qualitative

Table 1. Listing of the variables in the dataset.

For this paper the following columns or features will be omitted from data cleaning: CaseOrder, Interaction, UID, address related information. These would need to be checked by a means that is outside the scope of this paper as they require specialized resources.

Part II: Data Cleaning Plan

This section of the document will propose a plan for cleaning the given dataset. It will describe the techniques and the procedure that will be used to accomplish the task of cleaning the data. This section will discuss the language that was used to accomplish the data cleaning. Code will be presented to justify the quality of the code.

C1. Data cleaning plan. The plan for cleaning this for based on generally accepted steps for performing a thorough and regimented cleaning of a data set. The plan will be composed of the following steps ((Guide to Data Cleaning: Definition, Benefits, Components, and How to Clean Your Data, n.d.))

1. Remove Duplicate and irrelevant rows.
2. Fix formatting and inconsistencies in the data.
3. Remove outliers from the dataset.
4. Impute or drop rows with missing data.
5. Validate the data for quality assurance.

This five-step plan should provide a regimented way to clean a dataset while addressing the initial data quality issues that might arise. The justification for each of these steps will be discussed in the following section.

The standard pandas **read.csv()** method was used to read the data into the data frame. Other miscellaneous functions are **head()**, **dtypes**, and **shape** to get a view of the data and the original data types. The following table will discuss the methods used to detect anomalies in the data and subsequent sections there is a discussion about how these anomalies were mitigated.

Python Method	Description of Use	Step[9]

[9] Indicates what step in the annotated document where you will find the method first used. The steps were laid out earlier in section C1.

duplicated()	Method to determine if any of the values in the data frame are duplicates in the data frame.	1
unique()	To look for uniqueness for the feature columns. Will be used for columns in the dataset.	2
min(), max()	To see if the range of values is within a suitable range of values. For example, are there any issues with age range?	2
nunique()	Check the number of unique values in this column. Also used to check for missing values in the data frame.	2
describe()	Statistical summary of the data in the columns to better understand how the outliers fit in.	3
boxplot()	Visualization of the quartiles, max, min, and median.	3
stripplot()	Visualization of the data points to see how they lie in relation to the boxplot.	3
value_counts()	To get a count of the values to see if there are unique values and see if there are duplicates.	1

Table 2. Functions and methods used to detect.

The first step was to look for the duplicates involved the use of the **duplicate()** method to see if there were any duplicates in the dataset as well as the **value_counts()** method.

The second step required that any improperly formatted columns be corrected. This required an investigation into what each of these columns has. Ideally, the columns should be composed of unique descriptive values. Any values that did not meet quality standards should be remedied. An example would be an overuse of categories or spelling errors in the values that are contained within the column.

To find out what values were contained in the columns the **unique()** and **nunique()** methods were used to extract these values. The methods allowed an inspection of the values and to look for values that were not in a proper format like the Anxiety or Overweight columns. These columns stored values that were contrary to what was expected, 0's and 1s as opposed to Yes or No values.

Using the functions stated earlier returned values that indicated if there were any missing values. Missing values showed up as "nan" or something similar when the unique values were returned. These aided in seeing that these columns contained missing values.

Other methods that were used to detect anomalies were the **min()** and **max()** functions. This was to look for anomalies in columns that may contain data that may be out of bounds. For example, the Zip column does not hold the data correctly. To adequately store this information it should be stored as a string to allow for proper encoding of the five-digit Zip code. The current method of storing as numeric truncates the current value stored.

For example, if the Zip has leading zeros the zeros will be truncated. The resulting observation will only contain only digits 1 through 9. No leading zeros will be recorded.

During the third step in the process, the requirement is to look for outliers among the quantitative variables. To find the outliers, the use of the **boxplot** and **stripplot** that was found

in the Seaborn package and the use of the **describe()** function will give statistics about the data. It will give the 25% and 75% quartiles, the mean, max, and the mean. This will help decide about the outliers of the column.

To help get a better understanding of what these values mean the **boxplot()** along with a **stripplot()** functions were used to see how the data points were dispersed alongside a statistical visualization that was provided by the boxplot. An example of the plot is shown below.

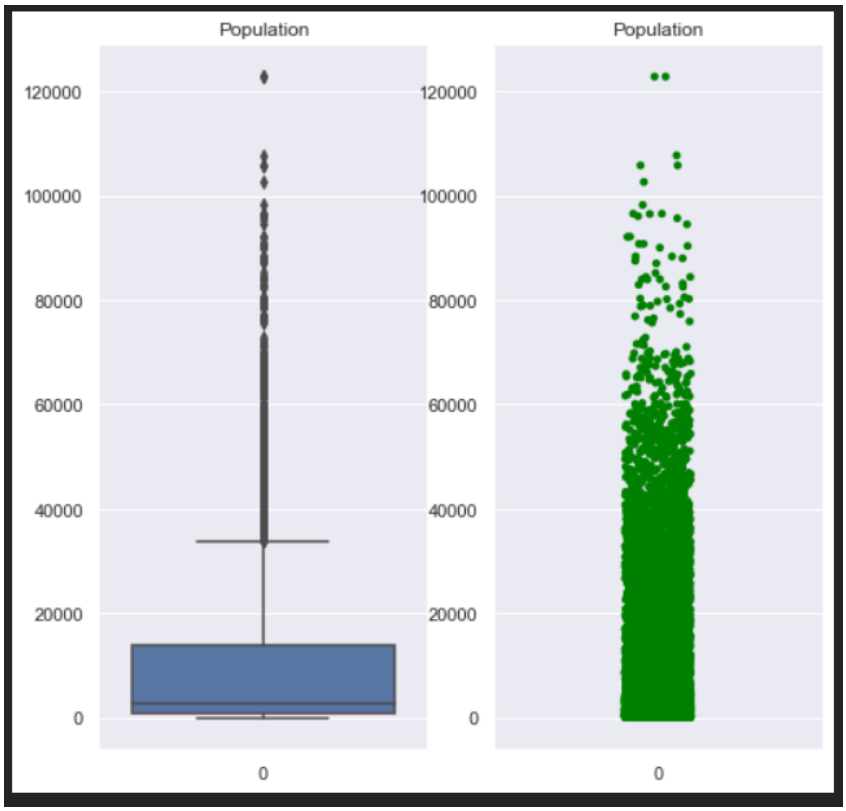


Figure 1. Example of boxplot to show the data layout.[10]

The methods and strategies to deal with the outliers will be discussed in section D2 of this assessment.

[10] To view the rest of the graphs please execute the Jupyter Notebook that has been included with the submission.

C2. Justification of approach. When a data scientist or analyst needs to clean data there needs to be a systematic approach to how the dataset is cleaned. The procedure should follow well-defined steps and the procedure should yield results that do not allow the inclusion of bias into the dataset.

This bias could be introduced in the way the analyst handles missing data or how they view the data. For instance, if we drop too many rows the dataset may no longer be a good representation of the sample population.

When we drop rows we may need to look at the exclusion of these rows and how will they affect the analysis. Does the exclusion of these rows bias the insight? For instance, with the medical dataset, will the exclusion of people of a certain age or group force an errant view of who is admitted to the hospital? This dilemma is something that could happen if rows that are missing data are "dropped" from consideration for analysis.

This dataset is composed of many different types of data. It has both numeric and string-type data. The understanding of these data types is important because the data types will help guide how this data is to be cleaned. The data types aid in understanding the purpose and possible calculations that may be performed on the data. If we understand the data we can clean it while preserving the information contained within it.

For example, we know that a column like Income is supposed to be stored as a float or if applicable a currency data type. If this was stored as anything else we know that we will need to remedy this by correcting for this error in the data type. We will need to keep in mind that if we change the data type we have to be cognizant that any changes may have a profound effect

on the meaning of the data and what we can do with it. That is why I have proposed the five-step approach to cleaning the data.

The step-by-step course of action is the most logical route to achieve a clean set of data. For the first step, the analyst wants to pare down the number of columns/features in the data set. There are a few columns that will not be used as they duplicate data in the data frame. The methods that were used to detect duplicates were **duplicate()** and **value_counts()** to detect the number of row values that were duplicated in the table. These methods are designed to acquire this information and are aptly suited for this task.

The second step will look at inconsistencies in the data and how it is represented. This could be things like the datatype of the columns, the types of data contained within the columns (is there a numeric where a string should be), and formatting of currency (e.g. TotalCharge, Additional_charge, etc.). Using methods like **unique()** allowed inspection of the values that were stored in the column. It allowed for the inspection of the values for their appropriateness as well as looking for possible missing values. The **min()** and **max()** functions allow for the inspection of values that may be out of range.

There will be a collapse of the number of categories used in the data frame. It was noted that there may be more categories than is necessary to do a suitable analysis. This is apparent in the Timezone column where there is too much granularity in the Timezone column. These time zones can be condensed to a few time zones with no major loss of information or context. This was accomplished using **unique()** method to look for unique values in the column. This method will list any unique values for the column. This method was employed for all the

columns in the dataset. Any anomalies will be mitigated and these will be discussed in the following sections of the paper.

Within this set, there will be an evaluation of the names of the columns. The columns that correspond to the items of the survey will have the column names changed to something more appropriate. This observation of the columns was not done using code but a review of the columns given within the data dictionary and looking at the raw data CSV file. There is a discussion about changing these values in another section of this paper.

The third step will be to remove any outliers from the dataset. This method will be done via statistical means and any outlier will be removed from the dataset. This step needs to be completed before moving on the Step 4 – the imputation of the missing values. If the dataset needs to have imputed values for NAs then we need to have an accurate measure of the mean of the column.

Including outliers can skew the mean and change the reliability of any calculation that may be performed in the future. There will be a few methods to help detect the outliers within the dataset. These methods are the **describe()**, **boxplot()**, and **stripplot()**. The two graphing methods will help with the visualization of the data and its associated statistics.

While the **describe()** method will give the data numerical statistics. Using this information it can be decided on what values to drop. The chosen method will be the interquartile range (IQR) method. How this method was used will be discussed in a subsequent section.

The fourth step is to impute or drop the rows. There will be columns that will need to have their missing data imputed. The method for imputing the value will be based on the

column as well as the amount of data that is missing. This step will be discussed in further detail in section D of the paper.

The fifth step is to look at the dataset for the quality of the data. The finished cleaned dataset should facilitate easy data analysis without compromising the information within it. Data cleaning should maintain as much of the original information as possible. The goal of data cleaning is to make the dataset ready for analysis.

Any reduction in the features in the data set should be left to the individual analyst as deleting data or dropping features may hinder any future processing. There will be further elucidation on these topics in subsequent sections of this document.

The Python methods that were chosen for the data cleaning allowed the data to be viewed and dealt with simply and straightforwardly. Utilizing the packages made the data-cleaning process easier. The packages were proven and they yielded results that are expected.

C3. Language and packages used. The language that will be used for this assessment will be Python. Python provides a robust selection of packages that will aid in the cleaning of the data. The packages that were used are best suited for this type of assessment. The table below briefly explains what each did and how they contributed to cleaning the data. The packages that will be used are summarized in the following table.

Python Package	Description
pandas	Used to read from the CSV file; store the data in a data frame.

missingno	Used to help visualize the missing data elements
numpy	Used to sort output to look for unique or spelling errors in the columns.
seaborn	For graphing the outliers for a visual look at the outliers.
matplotlib	Simple graphing of plots for outliers.

Table 3. Python packages used.

C4. Data Anomaly Detection Code

The chart below shows the code methods used along with the step you will find them in the Jupyter Notebook located within the **In[]** line makers within Jupyter Notebook.

Python Method	Step[11]	Places of Usage[12]
duplicated()	1	3
unique()	2	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19
min(), max()	2	4, 7, 10, 13, 14, 19
nunique()	2	10
describe()	3	21, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 43,

[11] Indicates what step in the annotated document where you will find this first used. The steps were laid out earlier in section C1.

[12] Refers to the cell(s) in the Jupyter Notebook where it is used. Please run the notebook to get the In[] values.

		45, 47, 49, 51, 53, 54, 56, 57, 59, 63, 64, 65, 67, 71, 72
boxplot()	3	22, 25, 27, 29, 31, 33, 35, 37, 39, 41, 44, 46, 48, 50, 52, 55, 58, 60
stripplot()	3	22, 25, 27, 29, 31, 33, 35, 37, 39, 41, 44, 46, 48, 50, 52, 55, 58, 60
value_counts()	1	3

Table 4. Location and step of the used methods and code.

****** See attached Jupyter Notebook for full code with comments/annotations. These comments explain what is the purpose of the code snippet and what it is accomplishing. ******

Part III: Data Cleaning

The following sections will describe findings that involve data quality issues. There will be justification for the methods that were used to mitigate the data quality issues in the data set. A summary will be provided of the outcome of the data cleaning at each step that was proposed in the previous section.

The code that was used to mitigate the data quality issues encountered while cleaning the data can be found in the accompanying Jupyter Notebook. There will be a brief discussion about the limitations of data cleaning and how these limitations may hamper the question that was stated in section A of this document.

D1. Description of findings regarding quality issues.

Num3 – NUM3 Task 1 Data Cleaning

The dataset exhibited a few data quality issues. There were discrepancies between what was recorded in the CSV file and the data dictionary. There were no duplicates found using the methods that were utilized in Step 1.

There were some discrepancies in how data was stored in some of the columns. For example, the columns of Anxiety and Overweight are recorded in the data dictionary as taking "Yes" or "No" as values. This is not recorded in the CSV. These values are recorded as zero or one.

While this may not cause an immediate issue. It might not become obvious until there are categorizations implemented or other types of calculations that may undertaken. There were 984 missing values at the beginning of the data cleaning regiment. The reasons for mitigating this are discussed in the following sections of this part of the document.

The Population is of the right type and there is no further need to check this column for valid data. The Population column had a few outliers that fell outside of the range of the IQR method. The total number of outliers was 855 or about 8.5% of the frame. Please note the size of the dataset going forward will be the following 9145. The outliers were below a lower bound of 694.75 and above 13945.00. Any data points above these values were considered extreme and will be discussed in section D2.

Checking the Area column yielded unique values that afford an analyst the ability to categorize patients based on their area. No mitigation will be needed for this feature.

The Timezone column exhibited an undue amount of categorization. There is a count of twenty-six for this column. The number of categories can be reduced to the standard time zones. It is the opinion of this author that the current amount of categories is too many. This

column will undergo categorization collapse to make it easier to categorize patients based on their time zones. These were converted to a more easily read and more common reference. The conversions were created with the help of this website (*know about the Time zones and weather divisions in the USA, 2021,* n.d.):

https://www.y-axis.com/blog/time-zones-and-weather-divisions-in-usa/

Inspecting the Job feature returned an astounding number of job titles. The total number of job titles was 639. While this is excessive it might be necessary to have this level of granularity when it comes to the wealth of available occupations. For the moment these job titles will be left alone. They are not germane to the question that was proposed in Section A.

The next two columns are the Children and Age columns. These store the correct data values outside of a few missing values that can be treated later in later steps. The total number of values that were noted as missing was 2588 (before steps 3 and 4) in the Children column. The number of outliers in this frame was 275. This was after dropping some rows based on the findings of the Population column. The lower bound for the column was 0 and the upper bound was 3.

Age there were an initial 2414 missing values (before steps 3 and 4) and the range of values was within acceptable limits. The outlier lower bound was 35 and the upper bound was 71. The total number of outliers was 0. This was after the outliers were removed from the Children column.

The Education and Employment columns have no deficiencies in them. The data is sound and there is no need to mitigate these columns. These columns contained no missing values.

Num3 – NUM3 Task 1 Data Cleaning

The Income was inspected and was found that the values may be within a reasonable range. There is an extremely low value within the data set it is $154.08. This finding will need further investigation. It had an initial count of 2464 missing values. There were 221 outliers in the income column. This was after removing outliers from the Age column. The lower bound of this set was $19479.99 and the upper bound was $54173.13.

The Soft_drink has some missing values that will be addressed in the following step as indicated earlier. The amount of missing initial values was 2467.

The Overweight column is not in the right format. Based on the data dictionary and the other columns that are in the dataset. This column is supposed to hold only "Yes" or "No" values it is currently composed of 1, 0, and nan values. This column contained 982 missing values and methods of mitigation will be discussed in a subsequent section.

Gender did contain any missing values but did not have the right value. The "nonbinary" value should be entered for the recorded entry where “preferred … “ is currently entered. This anomaly will need to be mitigated and will be discussed in a subsequent section.

The Marital column contained no missing values and the data was consistent with the data dictionary. The data dictionary was not declared in what values this column is supposed to be composed, so it assumed that the data was correct.

The ReAdmis column contained no missing values and was consistent with the data dictionary.

VitD_levels column contained no missing values and was consistent with the data dictionary. The bound for this column was ~16.49 for the lower bound and 19.78 for the upper

bound. The total number of outliers was 465. These outliers reflect the previous removal of values in the other columns.

Doc_visits column contained no missing values and was consistent with the data dictionary. There were no outliers based on the IQR method.

Full_meals_eaten column contained no missing values and the data was consistent with the data dictionary. The number of outliers for this column is 6. The lower bound was 0 and the upper bound was 2.

Soft_drink had missing values of 2467. The other values in the column are what is expected.

VitD_suppl column contained no missing values and was consistent with the data dictionary. 52 outliers were calculated using the IQR method. The bounds that were derived were 0 for the lower bound and 1 for the upper bound.

Initial_admin, HighBlood, Stroke, Diabetes, Complication_risk, Arthritis, Hyperlipidemia, BackPain, Allergic_rhinitis, Reflux_esophagitis, Asthma, and Services, columns contained no missing values and were consistent with the data dictionary.

Initial_days contained 1056 missing values. There were no outliers detected using the IQR method. The bounds that were used were ~7.9 for the lower and the upper was ~61.2.

The Total_charge column had no missing values and the range of values is expected. Using the IQR method some rows could be considered outliers. The bounds that were calculated were ~3205.88 for the lower and ~7460.75 for the upper bound.

Additional_charges columns contained no missing values and the initial review showed no missing values. 363 outliers fell outside the range. The bounds for ~7959.05 for the lower bound and for the upper bound it was ~15533.28.

The columns that were used to store the survey data (Item1 → Item8) were found to be not logically named and will have their names changed. How this task was done will be discussed in the following section.

The mitigation of the missing values will be discussed in a subsequent section. There will be an explanation of the methods and reasoning behind their uses.

D2. Justification of the methods used for mitigation of quality issues.

To mitigate some of the quality issues that were encountered while cleaning the dataset there will be a need to ascertain a listing of viable cities and counties to match against what is given in the dataset. It will assumed that the data for these columns were not manually entered, but were loaded from a drop-down in an application. This will greatly reduce the chance that this information may have been recorded incorrectly.

The following columns will not be discussed in this section since they were found not to have any errors or they do not have the concept of an outlier associated with them – the qualitative columns. The columns are all address-related, patient information, Area, Job, Education, Marital, ReAdmis, HighBlood, Complication_risk, Arthritis, Diabetes, Hyperlipidemia, Allergic_rhinitis, Asthma, Services, and BackPain.

Looking at the information that is stored in the Zip column using Python you can see that before cleaning some of the ZIP codes are not properly formatted. There are a few that have less than the required number of five digits. If you run the **min()** function of the column

you will get a value of 610. This value is only three digits long when a correct ZIP code should be five digits long. The zeroes have been removed from the left-hand side. This is not a legal ZIP code and needs to be changed to a standard length of five characters. This will be accomplished by changing the data type to a string and then padding the left with the appropriate amount of zeros. This will yield the standard length for a ZIP code. To fill the missing values the **zfill()** function is used to fill the left of the Zip to its appropriate length. As a side note, this will need to be done every time the CSV will be imported.

The Timezone column was reduced from the twenty-six columns to seven to aid in categorization. There is little difference between the timezone so there is little need to be as granular as the twenty-six categories that were in the original dataset. The names that were used for the time zones are also easier to remember since they are standard names that most people are familiar with and will make it easier. The strategy that was used was to use a dictionary that would map the current time zones to the new time zones. In cell 6 of the notebook, there is the code that will make the replacement. It utilizes the **replace()** function along with a Python dictionary.

The Overweight column was changed back to the format that was expected. The current values are not in line with the data dictionary. These values have been changed to better match what was given in the data dictionary. The process was the same as the Timezone. It used the **replace()** method along with a dictionary. The values of 1 and 0 were replaced with Yes and No values.

The Anxiety column was handled in the same manner. The **replace()** function along with a dictionary was used to replace the values of 1 and 0 with Yes and No.

Num3 – NUM3 Task 1 Data Cleaning

The Items columns had their columns changed to a more descriptive name this was accomplished using a dictionary with the new names mapped to the old ones. The rename() function was used to accomplish this task. The code for this can be found in cell 20 of the Jupyter Notebook.

Outliers were found in the quantitative columns. The Lat and Lng columns were left alone. These values will have outliers that may be needed. The data frame will have values from states like Alaska, Hawaii, and the territory of Puerto Rico. These could easily be seen as outliers if the latitude and longitude are used. Excluding these from the data frame will not allow for these observations to be used. As these values will not fall within the boundaries of the continental United States. And will most likely fall outside the "fences" or boundaries that would normally be employed by the IQR method that will be used for finding the outliers.

These outliers were dealt with using the IQR method. It uses the same concept as the boxplot discussed earlier. It uses the concept of lower and upper bounds. These correspond roughly to the bounds that are shown on a boxplot. A sample calculation for the Population column is shown below.

```
lower_bound = medical_df['Population'].quantile(0.25)
upper_bound = medical_df['Population'].quantile(0.75)
IQR = upper_bound - lower_bound
# Identify the outliers in the data frame
threshold = 1.5
outliers = medical_df[(medical_df['Population']  < lower_bound – threshold * IQR)
        |(medical_df['Population']  > upper_bound + threshold * IQR)]
```

The previous code shows how the bounds were calculated. These were used as cutoff points in determining the outliers. Points above or below using the following formula

lower_bound - threshold * IQR or **upper_bound + threshold * IQR**

were dropped from the frame.

This procedure was done for each of the quantitative columns in the frame. These columns are Children, Age, Income, Vitamin D Levels, Doc visits, Full Meals Eaten, Vitamin D Supplements, Initial Days, Total Charge, and Additional Charges[13].

A quick summary of the Python functions that were used to mitigate anomalies and the steps that they were used is given below:

Python Method(s)	Step	Column	Places of Usage [14]
zfill()	2	Zip	4
.replace()	2	Timezone	6
.replace()	2	Overweight	15
.replace()	2	Anxiety	17
IQR method, drop()	3	Population	23 & 24
IQR method, drop()	3	Children	28

[13] Code for the mitigation can be found in Step 3 and under a heading for each of the columns named above. The code is found in the Jupyter Notebook.

[14] Refers to the cell(s) in the Jupyter Notebook where it is used. Please run the notebook to get the In[] values.

IQR method, drop()	3	Age	32
IQR method, drop()	3	Income	34
IQR method, drop()	3	Vitamin D Levels	38
QR method, drop()	3	Doc visits	42
QR method, drop()	3	Full Meals Eaten	45
QR method, drop()	3	Vitamin D Supplements	49
QR method, drop()	3	Initial Days,	53
QR method, drop()	3	Total Charge	56
QR method, drop()	3	Additional Charges	59

Table 5. Location and step of the used methods and code.

This was how the outliers were handled before the missing values that were found were treated based on the data type. The data types that were found to have missing data were both quantitative and qualitative. The columns that were missing values were the following: Children, Age, Income, Soft_drink, Overweight, Anxiety, and Initial_days. Each of these types will be handled differently based on the type and the type of distribution that the data is exhibiting. The **missingno** graph shows visually the columns that have missing values after the outliers have been removed.

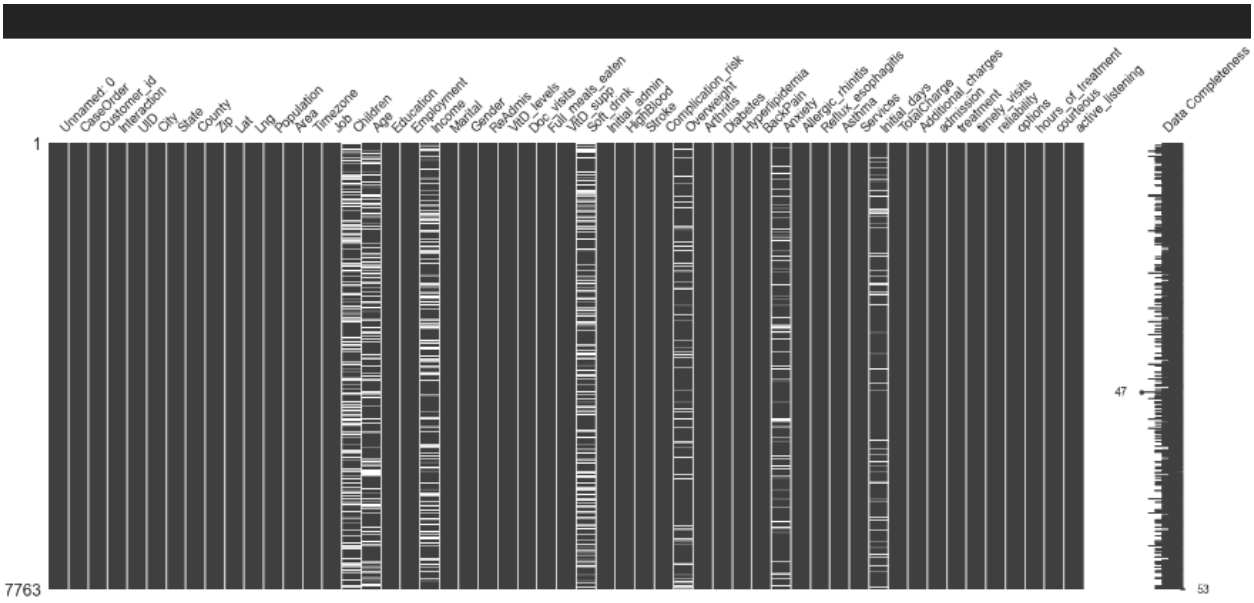


Figure 2. Missing matrix

The Children column is a quantitative column so the methods that could be implemented are the mean and the median. To figure out which method to use, a histogram was used to represent the distribution of the data after all the outliers had been removed from the columns of the frame. A table with a summary of the functions used, along with where to

find this in the Jupyter Notebook is included below the discussion of the treatment for the missing data (See Table 6).

For the Children column, the histogram is shown below.

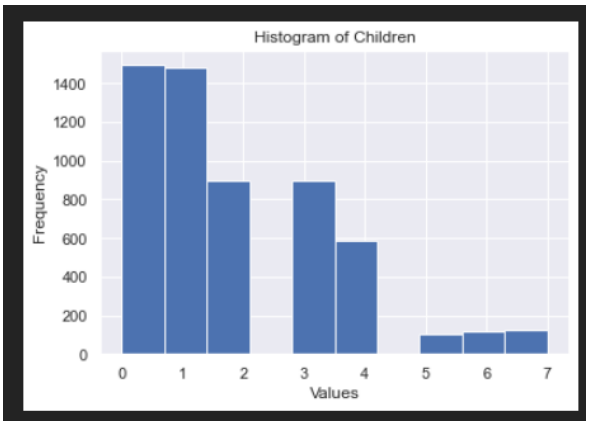


Figure 3. Histogram of Children

The graph shows that the data's distribution is right skewed so to treat the missing values the median will be used to fill in the missing values of the columns. The histogram below shows the distribution after the missing values have been imputed with the median. After imputation, the distribution has been mostly maintained.

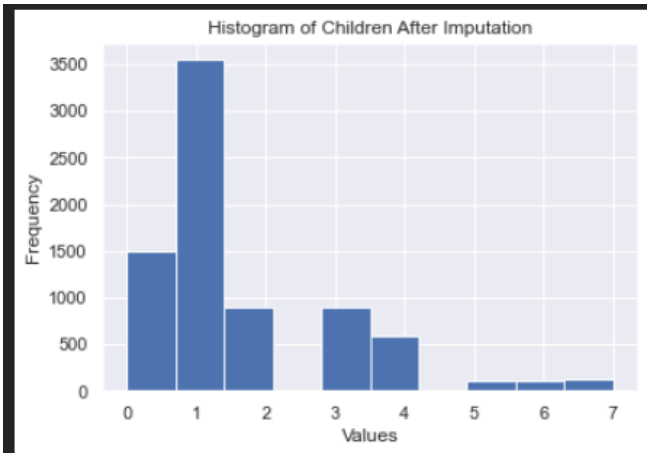


Figure 3. Histogram of Children After Imputation

The next column that needed missing values imputed was the Age column. The Age treatment was similar to the Children column but the histogram was different it followed an almost uniform distribution of the data. The graph below shows the distribution.

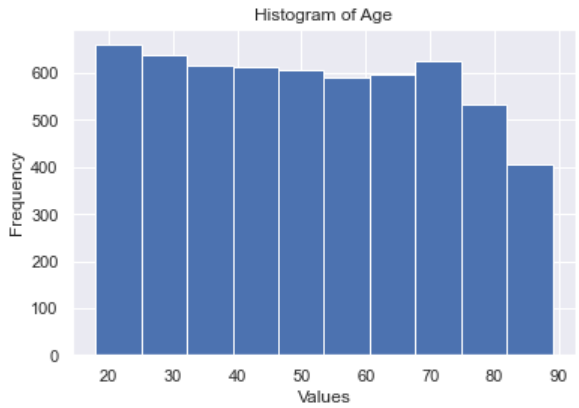


Figure 4. Histogram of Age

To impute the values for this there will be a need to use the mean for the Age column. Since it seems to be more uniform. The distribution has been maintained. The graph is shown below.

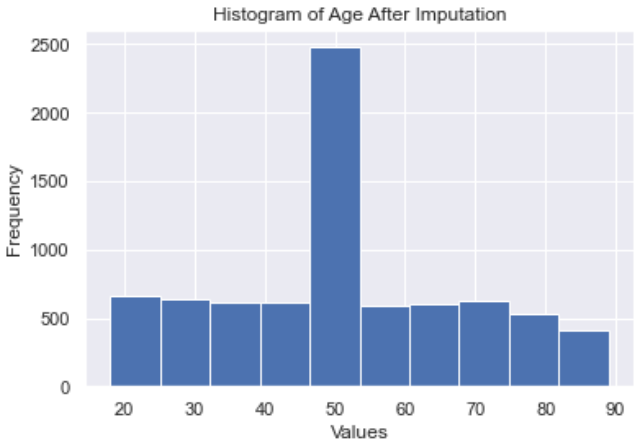


Figure 5. Histogram of Age After Imputation

The Income column is another qualitative column that needs to have values imputed. The distribution shown below shows that this distribution is right-skewed (like the Children column).

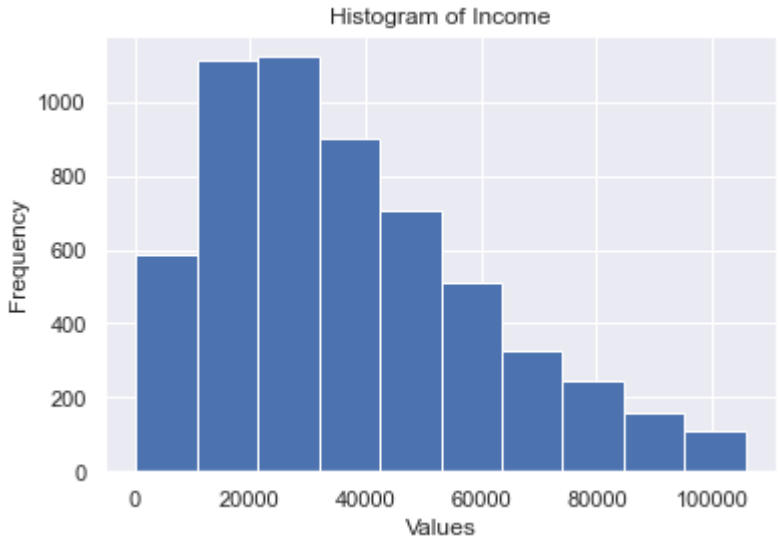


Figure 6. Histogram of Income

To impute these values there will be a need to use the median in this column. The histogram shows what the distribution is after imputation. The general shape of the distribution has been maintained.

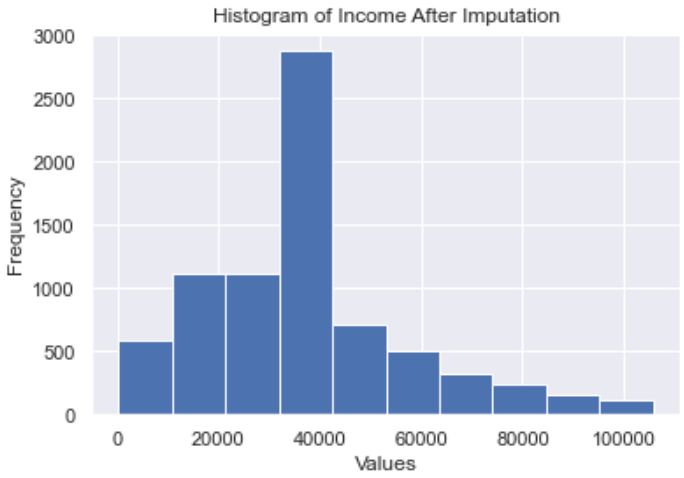


Figure 7. Histogram of Income After Imputation

The last of the quantitative columns is the Initial_days. The histogram for this column shows that it has a bimodal distribution.

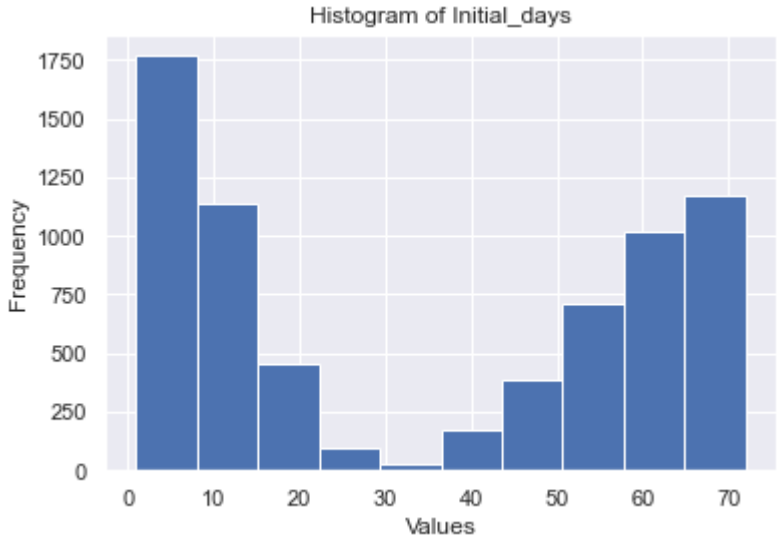


Figure 7. Histogram of Initial_days

To impute these values the median will be used. The histogram after imputation and the type of distribution that has been maintained.

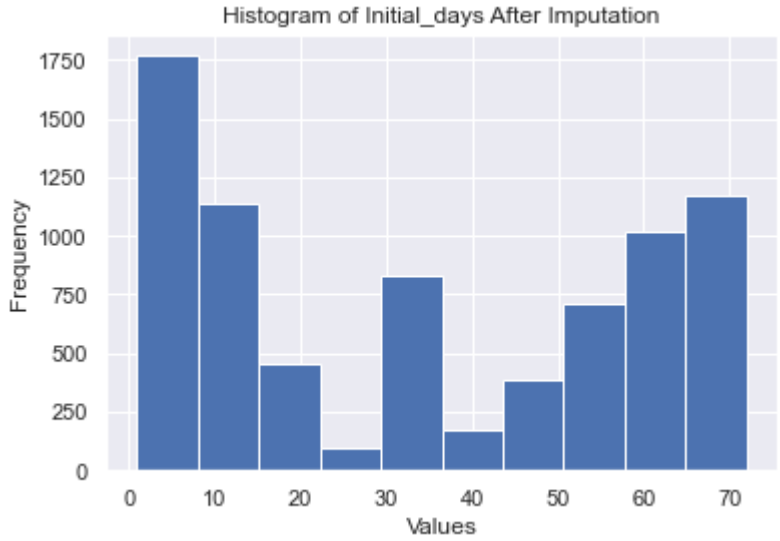


Figure 8. Histogram of Initial_days After Imputation

To treat the qualitative values the mode method was used to fill in the missing values for the Soft_drink, Overweight, and Anxiety. This was not the only method available, but it was

the best for the data on hand. If desired the data could have been filled with a value like "Missing." This would only be done if there were insight into what other analysis may be performed on the data using these categorical columns.

Python Method(s)	Step	Column	Places of Usage
Describe(), hist(), median()	4	Children	62, 63
Describe(), hist(), mean()	4	Age	64, 65
Describe(), hist(), median()	4	Income	66, 67
Describe(), hist(), median()	4	Initial_days	71, 72
mode()	4	Soft_drink, and	68
mode()	4	Overweight,	69
mode()	4	Anxiety	70

Table 6. Location and step of the used methods and code.

All the missing values have been filled and there are no more missing values in the data frame. The **missingno** matrix shows this visually.

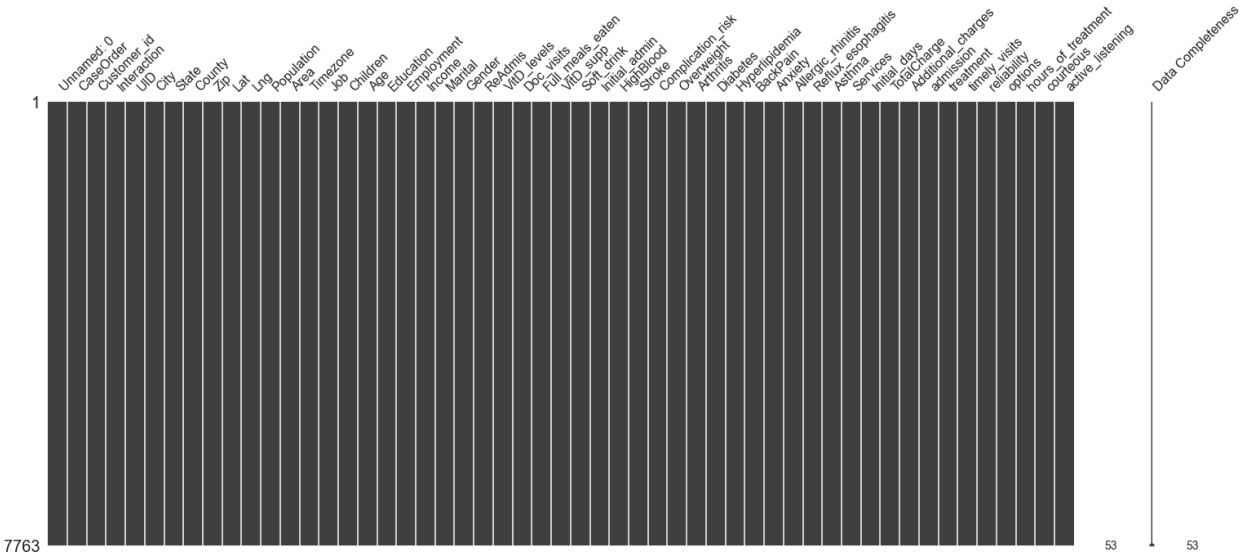


Figure 9. Missingno matrix showing no missing values

D3. Summary of outcomes. After cleaning the data there has been a reduction in the anomalies that were found in the raw dataset. There were a significant number of outliers within the dataset. 2237 outliers were removed from the dataset. This allowed for the majority of the data points to be saved.

There were errors in the encoding of the data. The Anxiety and Overweight columns were not saved in the same format as the other columns that store this type of data. They were converted into a format that matches the rest of the columns in the dataset. Converted from 1s and 0s to Yes and No.

Additionally, some of the values that were found in the categorical columns were changed to make them easier to understand. For example, the Timezone column had its number of categories reduced to make it easier to group locations by timezone without losing any information. The previous number of zones was not needed. This logic could be extended to the Job column but without a better understanding of the Job column's intended purpose. It will be difficult to re-express these current categories into categories that will permit analysis in

a later. Any assumptions made could lead to a loss of data and granularity within the Job category.

The values that were missing in the raw data set were imputed with values that more closely match the sample population of the cleaned dataset. The image below is the matrix that was created using the matrix function from the missingno library.

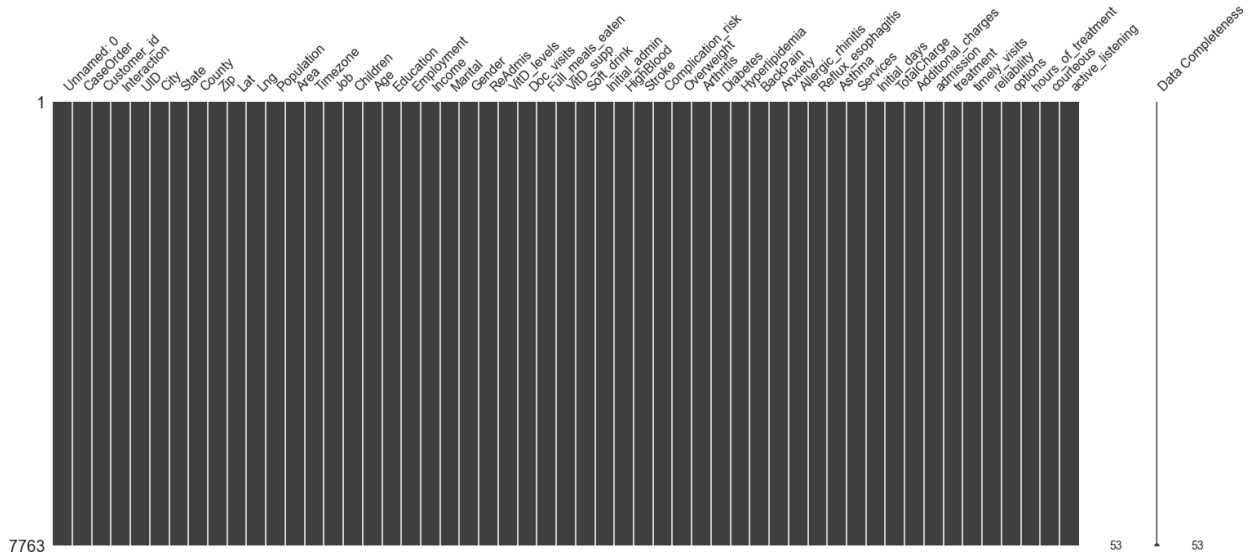


Figure 10. Matrix of the cleaned data showing no missing values.

D4. Annotated code to mitigate quality issues [15].

See the attached file for the annotated code. The file is called:

Heino_D206_Assessment.ipynb

It may be helpful to refer to the tables in Section D for the exact location of the code. Tables 5 and 6 should aid in referencing the code that was used to complete this section of the assessment.

[15] A complete listing of code can be found as a Jupyter Notebook file attachment.

D5. Copy of cleaned. A copy of the cleaned data is found as an attachment that accompanies this document. The file name is:

Heino_cleaned_medical.csv.

D6. Limitations of the data cleaning process. While cleaning data is an important task that must be carried it does have its limitations. Without sufficient resources, it may be difficult to assess whether the data contained in the columns is in a high state of reliability.

For example, the cites, latitude, longitude, and other columns may have errant data, but without the resources to check these, it has to be assumed that the data is correct and can be used for data analysis.

The dataset has been limited by the understanding of the one who has performed the data analysis. The data may have lost some relevant data points. This may be due to the lack of understanding of what the data set is trying to capture in its encoding and storing of the information.

Some columns that had a data value changed, the Gender column, may force observations into groups that have no bearing on the intended meaning of the column. An individual may not want to have their gender recorded. The non-binary category may not adequately represent a class or group of people appropriately. This category could be expanded upon to better categorize the different genders that the researcher may come across.

These are some of the observations of the cleaning of the data set. It is not an exhaustive list but these are some of the more salient observations. Individuals with more domain knowledge may be able to offer more insight on what can be done to make the data set "cleaner" and better versed in answering a wide girth of questions.

D7. How the limitations of data cleaning affect the question. The cleaning of the data set yielded 7,763 observations. This maintained about 77% of the data. This should have little effect on the question. The population sample that resulted has more than enough data points to be able to answer the question that was proposed in section A of this document. Most of the data has had values that are relevant to the column that contains them.

Categorical values with missing values were imputed with the mode for that given column. This was to preserve the integrity of the data that was present in the other features of the dataset. The missing numerical values like Children were given a median value. This was more of an objective choice. It was a quantitative feature and this was the appropriate method to impute the missing values. This should not have any bearing on the question that was stated earlier. Imputing this column to another value may cause observations to fall into a category that may have an unforeseen effect on the questions that may be based on this cleaned dataset.

The cleaned dataset will be able to be used since the vast majority of the data points were preserved. Missing data was imputed with values that should have very little impact on the analysis.

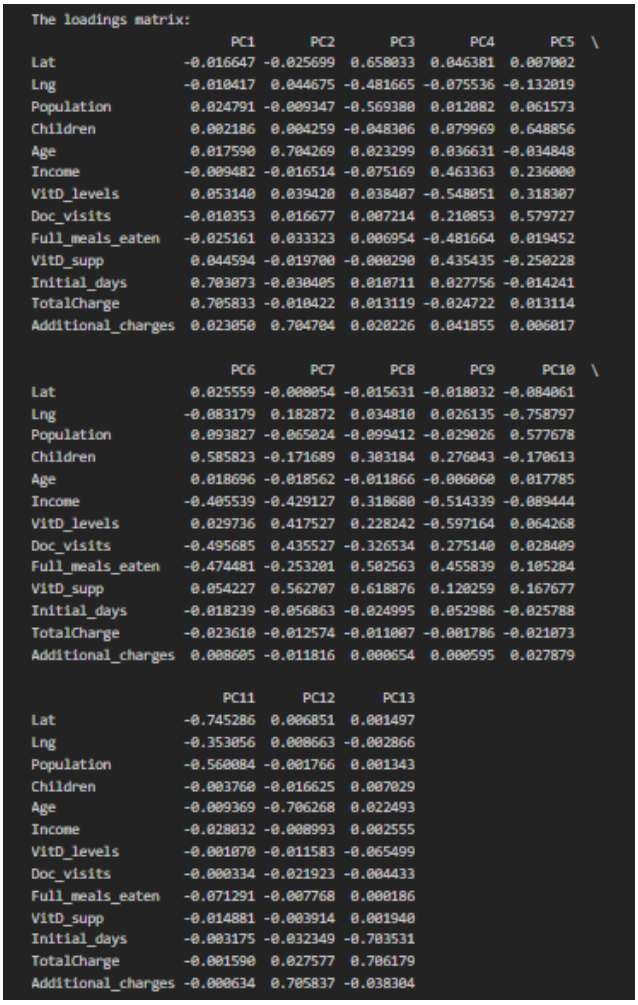
E. Applying Principal Component Analysis (PCA))

The PCA will be performed on the clean data set. The variables that were used were all the quantitative columns from the set.

E1. Total number of components. These variables are the following: Lat, Lng, Population, Children, Age, Income, VitD_levels, 'Doc_visits, Full_meals_eaten, VitD_supp,

Initial_days, TotalCharge, and Additional_charges. The code for the actual analysis can be found in section E of the Jupyter Notebook.

The following is a screenshot of the loading matrix that was created.



```
The loadings matrix:
                          PC1       PC2       PC3       PC4       PC5  \
Lat                 -0.016647 -0.025699  0.658033  0.046381  0.007002
Lng                 -0.010417  0.044675 -0.481665 -0.075536 -0.132019
Population           0.024791 -0.009347 -0.569380  0.012082  0.061573
Children             0.002186  0.004259 -0.048306  0.079969  0.648856
Age                  0.017590  0.704269  0.023299  0.036631 -0.034848
Income              -0.009482 -0.016514 -0.075169  0.463363  0.236000
VitD_levels          0.053140  0.039420  0.038487 -0.548051  0.318307
Doc_visits          -0.010353  0.016677  0.007214  0.210853  0.579727
Full_meals_eaten    -0.025161  0.033323  0.006954 -0.481664  0.019452
VitD_supp            0.044594 -0.019700 -0.000290  0.435435 -0.250228
Initial_days         0.703073 -0.030405  0.010711  0.027756 -0.014241
TotalCharge          0.705833 -0.010422  0.013119 -0.024722  0.013114
Additional_charges   0.023050  0.704704  0.020226  0.041855  0.006017

                          PC6       PC7       PC8       PC9      PC10  \
Lat                  0.025559 -0.000054 -0.015631 -0.018032 -0.084061
Lng                 -0.083179  0.182872  0.034810  0.026135 -0.758797
Population           0.093827 -0.065024 -0.099412 -0.029026  0.577678
Children             0.585823 -0.171689  0.303184  0.276043 -0.170613
Age                  0.018696 -0.018562 -0.011866 -0.006060  0.017785
Income              -0.405539 -0.429127  0.318680 -0.514339 -0.089444
VitD_levels          0.029736  0.417527  0.228242 -0.597164  0.064268
Doc_visits          -0.495685  0.435527 -0.326534  0.275140  0.028409
Full_meals_eaten    -0.474481 -0.253201  0.502563  0.455839  0.105284
VitD_supp            0.054227  0.562707  0.618876  0.120259  0.167677
Initial_days        -0.018239 -0.056863 -0.024995  0.052986 -0.025788
TotalCharge         -0.023610 -0.012574 -0.011007 -0.001786 -0.021073
Additional_charges   0.008605 -0.011816  0.000654  0.000595  0.027879

                         PC11      PC12      PC13
Lat                 -0.745286  0.006851  0.001497
Lng                 -0.353056  0.008663 -0.002866
Population          -0.560084 -0.001766  0.001343
Children            -0.003760 -0.016625  0.007029
Age                 -0.009369 -0.706268  0.022493
Income              -0.028032 -0.008993  0.002555
VitD_levels         -0.001070 -0.011583 -0.065499
Doc_visits          -0.000334 -0.021923 -0.004433
Full_meals_eaten    -0.071291 -0.007768  0.000186
VitD_supp           -0.014881 -0.003914  0.001940
Initial_days        -0.003175 -0.032349 -0.703531
TotalCharge         -0.001590  0.027577  0.706179
Additional_charges  -0.000634  0.705837 -0.038304
```

Figure 11. Screenshot of the loading matrix.

E2. Justification for the component reduction. After the analysis of the dataset, the results were the following. If you want to account for 90% of the variance you will need to include ten of the principal components as they were calculated above. The Scree plot below shows the Eigenvalues along with the number of components.

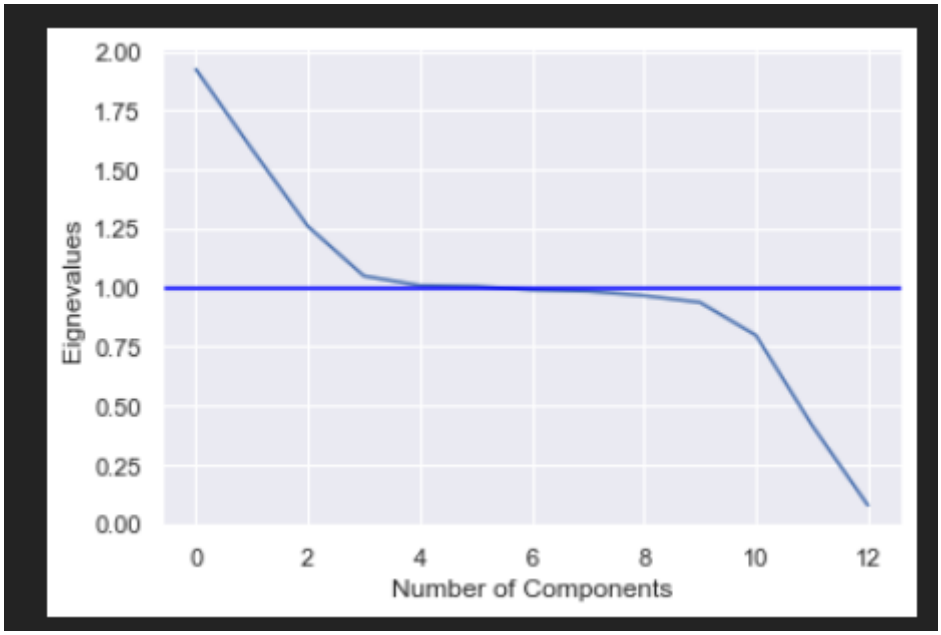


Figure 12. Screenshot of the Scree plot.

Viewing this graph and using the Kaiser rule we want to keep the values with Eigenvalues greater than or equal to one. Just using the Eigenvalues we will want to keep PC through PC6.

Yet if we look at the cumulative sum of explained ratio variance if we want to keep about 90% of the variance we will need to look at keeping 10 of the principal components. The output of the calculation is shown below.

```
Cummulative sum method:
 [0.14801397 0.27000883 0.36686138 0.44752083 0.52512239 0.60245635
 0.67856077 0.75424929 0.82849695 0.90058899 0.96193954 0.99404678
 1.        ]
```

Figure 13. Screenshot of the cumulative sum calculation.

The principal components that should be kept are PC1 through PC10, and this will allow for accounting for 90% of the variance.

E3. Benefits to the organization. The benefits to the organization will be that there will be reduced dimensionality in the amount of features within the dataset. It will lower the amount of data that is needed. The will be an increase in the performance of any future calculations. An organization will have a better "visualization" of the data when it is reduced from many dimensions to a few that adequately encapsulate the data (Jeannier, 2018).

PCA will allow the researcher to create better and more generalized models. The inclusion of too many features will be brought about by having too many features included in machine learning models. As an additional benefit, the amount of noise that can be introduced in a model will be reduced with the reduction in dimensionality (David, 2023).

F. Panopto Video

The link to the Panopto is the following:

References

G. Web Sources.

The resources that were used to create this paper are given below. Code was not used directly for the creation of this assessment but it was referenced to see how various library functions worked and concepts worked. The WGU course material was used as a reference.

Code resources:

Boeye, J. (n.d.). *Dimensionality Reduction in Python*. DataCamp. Retrieved October 29, 2023, from https://app.datacamp.com/learn/courses/dimensionality-reduction-in-python

GeeksforGeeks. (n.d.). *Boxplot using Seaborn in Python.* Retrieved October 27, 2023, from https://www.geeksforgeeks.org/boxplot-using-seaborn-in-python/

How to Count Duplicates in Pandas Dataframe? (n.d.). Retrieved October 27, 2023, from https://www.tutorialspoint.com/how-to-count-duplicates-in-pandas-dataframe

Saturn Cloud. (2023, June 19). *How to Detect and Exclude Outliers in a Pandas DataFrame | Saturn Cloud Blog*. Retrieved October 28, 2023, from https://saturncloud.io/blog/how-to-detect-and-exclude-outliers-in-a-pandas-dataframe/

H. In-text citations:

David. (2023, February 8). *What is Principal Component Analysis (PCA) & How to Use It? | Bigabid*. Bigabid. https://www.bigabid.com/what-is-pca-and-how-can-i-use-it/

Guide To Data Cleaning: Definition, Benefits, Components, And How To Clean Your Data. (n.d.). Tableau. Retrieved October 23, 2023, from https://www.tableau.com/learn/articles/what-is-data-cleaning

Jeannier, R. (2018, June 5). What's the point of PCA anyway? - Roland Jeannier - Medium. *Medium*. https://medium.com/@rtjeannier/whats-the-point-of-pca-anyway-279cf0ef0683

know about the Time zones and weather divisions in the USA, 2021. (n.d.). Immigration and Visa Consultants, India | Y-Axis Overseas Careers. Retrieved October 24, 2023, from https://www.y-axis.com/blog/time-zones-and-weather-divisions-in-usa/

Num3 – NUM3 Task 1 Data Cleaning