Performance Assessment D206 - Data Cleaning

Doug Haunsperger

College of IT, Western Governors University

September 5, 2023

# Part I. Research Question

## A. Question Description

As implied by the data dictionary provided, a likely research question for this set of data would be “What variables (patient conditions, demographics, attitudes, etc.) contribute the most to patient readmission to the hospital?” According to the scenario given, hospitals have a need to reduce readmissions in order to avoid CMS fines and would find this data analysis highly valuable.

## B. Variable Description

The data set ‘medical\_raw\_data.csv’ contains 53 columns. Naïvely importing the CSV into Python using the pandas read\_csv() function without specifying data types assigns ‘int64’, ‘float64’, or ‘object’ data types to each variable. Examining examples of each variable will allow us to choose more appropriate data types for each. Note that column 0 has identical values to column 1 (‘CaseOrder’) in all 10,000 rows, so I will plan to delete the unnamed column 0 as a duplicate. The below table lists each variable, its description, whether a particular variable is quantitative (Q) or categorical (C), and examples of each variable taken from the first two records of the data file. For quantitative variables, I then further describe them as discrete (D) or continuous (C) variables. For categorical variables, I further describe them as nominal (N) or ordinal (O) categorical variables. Columns 1-10 contain an index, patient ID information, and geographic information. I did not perform any cleaning on these variables. All further references to the data set only consider columns 11-52.

Table

*Variable Description for Columns of D206 Medical Data File*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Col # | Variable | Description | Quantitative/Categorical | Subtype | Examples | |
| 0 | “Unnamed: 0” (assigned by Python import) | Index (duplicate) | <delete> |  | 1 | 2 |
| 1 | CaseOrder | Index (1-n) | C | O | 1 | 2 |
| 2 | Customer\_id | Customer ID string | C | N | C412403 | Z919181 |
| 3 | Interaction | Unique ID | C | N | 8cd49b13-f45a-4b47-a2bd-173ffa932c2f | d2450b70-0337-4406-bdbb-bc1037f1734c |
| 4 | UID | Different unique ID | C | N | 3a83ddb66e2ae73798bdf1d705dc0932 | 176354c5eef714957d486009feabf195 |
| 5 | City | Geographic location | C | N | Eva | Marianna |
| 6 | State |  | C | N | AL | FL |
| 7 | County |  | C | N | Morgan | Jackson |
| 8 | Zip |  | C | N | 35621 | 32446 |
| 9 | Lat |  | Q | C | 34.3496 | 30.84513 |
| 10 | Lng |  | Q | C | -86.7251 | -85.2291 |
| 11 | Population | Number of people living within one mile of patient | Q | D | 2951 | 11303 |
| 12 | Area | Geographic type | C | O | Suburban | Urban |
| 13 | Timezone | TZ string | C | N | America/Chicago | America/Chicago |
| 14 | Job | Patient’s work role | C | N | Psychologist, sport and exercise | Community development worker |
| 15 | Children | Children in household | Q | D | 1 | 3 |
| 16 | Age | Patient age | Q | D | 53 | 51 |
| 17 | Education | Patient education level | C | O | Some College, Less than 1 Year | Some College, 1 or More Years, No Degree |
| 18 | Employment | Patient’s job status | C | N | Full Time | Full Time |
| 19 | Income | Patient income | Q | C | 86575.93 | 46805.99 |
| 20 | Marital | Patient marital status | C | N | Divorced | Married |
| 21 | Gender | Patient gender | C | N | Male | Female |
| 22 | ReAdmis | Was patient readmitted within one month? | C | O | No | No |
| 23 | VitD\_levels | Patient Vitamin D level (ng/mL) | Q | C | 17.80233 | 18.99464 |
| 24 | Doc\_visits | Number times physician visited patient in hospital | Q | D | 6 | 4 |
| 25 | Full\_meals\_eaten | Number full meals eaten by patient in hospital | Q | D | 0 | 2 |
| 26 | VitD\_supp | Number of Vitamin D supplements given to patient | Q | D | 0 | 1 |
| 27 | Soft\_drink | Does the patient drink 3+ soft drinks per day? | C | O | NA | No |
| 28 | Initial\_admin | Means of admission to hospital | C | N | Emergency Admission | Emergency Admission |
| 29 | HighBlood | Does the patient have high blood pressure? | C | O | Yes | Yes |
| 30 | Stroke | Has the patient had a stroke? | C | O | No | No |
| 31 | Complication\_risk | Assessed risk of complications | C | O | Medium | High |
| 32 | Overweight | Is the patient overweight? | C | O | 0 | 1 |
| 33 | Arthritis | Does the patient have arthritis? | C | O | Yes | No |
| 34 | Diabetes | Does the patient have diabetes? | C | O | Yes | No |
| 35 | Hyperlipidemia | Does the patient have hyperlipidemia? | C | O | No | No |
| 36 | BackPain | Does the patient have back pain? | C | O | Yes | No |
| 37 | Anxiety | Does the patient have anxiety? | C | O | 1 | NA |
| 38 | Allergic\_rhinitis | Does the patient have allergic rhinitis? | C | O | Yes | No |
| 39 | Reflux\_esophagitis | Does the patient have reflux esophagitis? | C | O | No | Yes |
| 40 | Asthma | Does the patient have asthma? | C | O | Yes | No |
| 41 | Services | Primary service received in hospital | C | N | Blood Work | Intravenous |
| 42 | Initial\_days | Days stayed in hospital | Q | C | 10.58577 | 15.12956 |
| 43 | TotalCharge | Average amount charged to patient per day | Q | C | 3191.049 | 4214.905 |
| 44 | Additional\_charges | Average amount charged for ancillary services | Q | C | 17939.4 | 17613 |
| 45 | Item1 | Survey responses | C | O | 3 | 3 |
| 46 | Item2 |  | C | O | 3 | 4 |
| 47 | Item3 |  | C | O | 2 | 3 |
| 48 | Item4 |  | C | O | 2 | 4 |
| 49 | Item5 |  | C | O | 4 | 4 |
| 50 | Item6 |  | C | O | 3 | 4 |
| 51 | Item7 |  | C | O | 3 | 3 |
| 52 | Item8 |  | C | O | 4 | 3 |

# Part II. Data-Cleaning Plan

## C1/C2. Discussion and Justification of Detection Methods

### Duplicates

I will use the methods described by Dr. Middleton in the course video on detecting duplicates, namely using value\_counts() on the Series returned by the duplicated() method. Any True results will indicate an entirely duplicated row of data. I will also drill down with the subset parameter and see if any Customer\_ids are duplicated, but with different observations.

Duplicate data can bias analysis, therefore it is necessary to remove them. (Middleton, n.d., *Duplicates*)

### Missing Values

The info() method that I used to show the column names in section B also produced non-null counts for each column. These values can simply be subtracted from the total number of records (10,000) to give the number of missing values for each column. Alternately, one could count the nulls with df.isnull().sum(). I also visualize the “missingness” of the data using the missingno package.

Data scientist Nasima Tamboli gives three reasons why it is important to detect and treat missing values:

* Many machine learning algorithms fail if the dataset contains missing values. However, algorithms like K-nearest and Naive Bayes support data with missing values.
* You may end up building a biased machine learning model, leading to incorrect results if the missing values are not handled properly.
* Missing data can lead to a lack of precision in the statistical analysis. (Tamboli, 2019)

Missingness visualizations can be useful to detect patterns in the missingness, that is, to discover whether the missing data is Missing Not At Random (MNAR), Missing At Random (MAR), or Missing Completely At Random (MCAR) (Donthi, n.d.).

### Outliers

To detect outliers, I consider only the non-geographic quantitative variables: 'Population', 'Children', 'Age', 'Income', 'VitD\_levels', 'Doc\_visits', 'Full\_meals\_eaten', 'VitD\_supp', 'Initial\_days', 'TotalCharge', and 'Additional\_charges'. I look at histograms and box plots for each variable using the seaborn package (Middleton, n.d., *Outliers*).

I chose to use visualizations rather than a more numerical method such as Z-scores because visualizations can provide additional insight into the data. For instance, the histogram of 'VitD\_levels' shows two discrete distributions, perhaps indicating that there was a systematic data entry or definition error.

### Categorical Variables

The medical data set contains several ordinal categorical variables. Namely, these are 'Area', 'Education', 'ReAdmis', 'Soft\_drink', 'HighBlood', 'Stroke', 'Complication\_risk', 'Arthritis', 'Diabetes', 'Hyperlipidemia', 'BackPain', 'Allergic\_rhinitis', 'Reflux\_esophagitis', and 'Asthma'. 'Overweight', 'Anxiety' and the survey responses 'Item1' through 'Item8' already are expressed in this manner. I plan to re-express the rest to match for consistency. To accomplish this, I create dictionaries based on the unique values of each variable, following the method shown in the class text (Larose & Larose, 2019).

The reason for re-expressing categorical variables as integer data is to capture their underlying order and to allow for numerical and statistical analysis to be performed on the data.

## C3. Choice of Programming Language

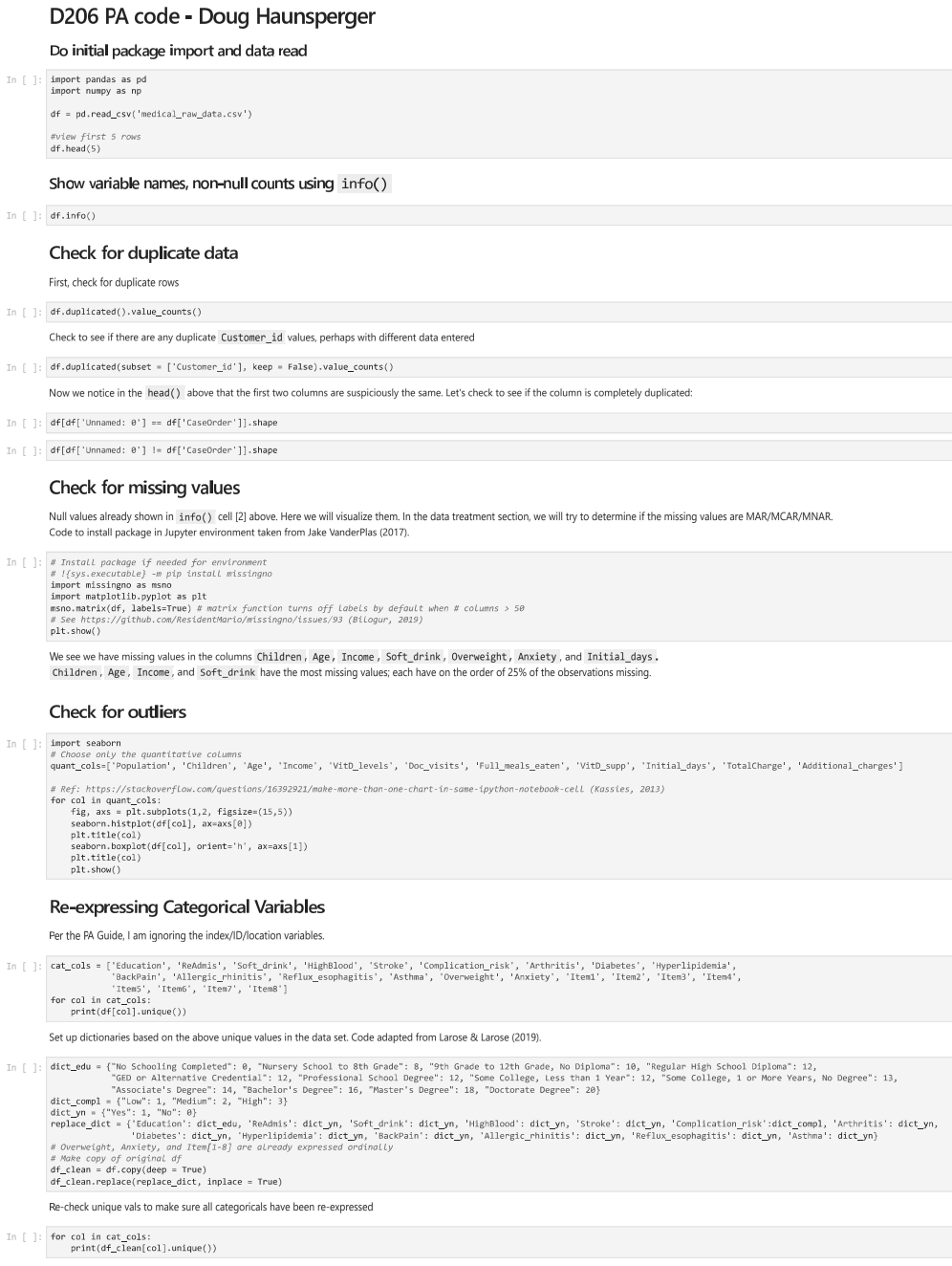
I chose to use Python for this course. While both Python and R are well-suited to data cleaning, I was already somewhat familiar with Python, and based both on my own experience and the WGU *R or Python* page, Python is more widely used in machine learning which is a field that interests me.

In my code, I used the packages pandas, numpy, seaborn, missingno, matplotlib, fancyimpute, and sklearn (also known as scikit-learn). The pandas package is used to “easily manipulate data” (Western Governors University, n.d.). numpy has several useful scientific computing functions, among which is the dot product numpy.dot function, used to calculate eigenvectors for principal component analysis (Middleton, n.d., *PCA*). I generated visualizations of missingness using missingno, and pair plots, histograms and boxplots using seaborn and matplotlib. The fancyimpute package provides several useful tools for imputing missing data, including the MICE function IterativeImputer and K-nearest-neighbor function KNN (Donthi, n.d.). Scikit-learn was used for performing PCA as recommended by Dr. Middleton.

Each of these packages greatly speed and simplify the process of doing data analysis. As one example, writing a function to produce a missingness matrix visualization would take dozens of lines of code and be prone to error. The missingno.matrix function call reduces that to a single line of code that works the same way every time.

## C4. Detection Code

See attached D206.ipynb Jupyter notebook. The detection and re-expression code used is in cells 1 – 11.



Figure

*Input code screenshot*

# Part III. Data Cleaning

## D1. Findings

### Duplicates

There are no completely duplicated rows in the medical\_raw\_data set. I also checked for partial duplicates by keying on the Customer\_id field. There were no records with the same Customer\_id. As discussed in section B, inspection of the head() of the data frame shows that the first (unnamed) column is likely a duplicate of the CaseOrder column, both of which appear to be 1-n indices of the observations. I verified this by looking at the shape() of data frame slices where the first column is either equal to or not equal to the CaseOrder column. All 10,000 records have identical values in both columns.

### Missing Values

The seven columns with missing values and their count are:

Table

*Medical Data Set Variables with Missing Values*

|  |  |
| --- | --- |
| Children | 2588 |
| Age | 2414 |
| Income | 2464 |
| Soft\_drink | 2467 |
| Overweight | 982 |
| Anxiety | 984 |
| Initial\_days | 1056 |

I used the matrix() function from the missingno package as suggested by Dr. Middleton to visualize the data frame’s missingness. This shows the missing values in the columns 'Children', 'Age', 'Income', 'Soft\_drink', 'Overweight', 'Anxiety', and 'Initial\_days'. 'Children', 'Age', 'Income', and 'Soft\_drink' have the most missing values; each have on the order of 25% of the observations missing.

### Outliers

Of the quantitative variables given, the box plots show outliers in all but 'Age', 'Doc\_visits', and 'Initial\_days'. Prior to performing any treatment on the outliers, only 'VitD\_levels' shows outliers to the low side. All other variables with outliers are right-skewed and show outliers to the high side only.

## D2. Treatment

### Re-expressing Categorical Variables

I executed the code discussed in section C2.

### Duplicates

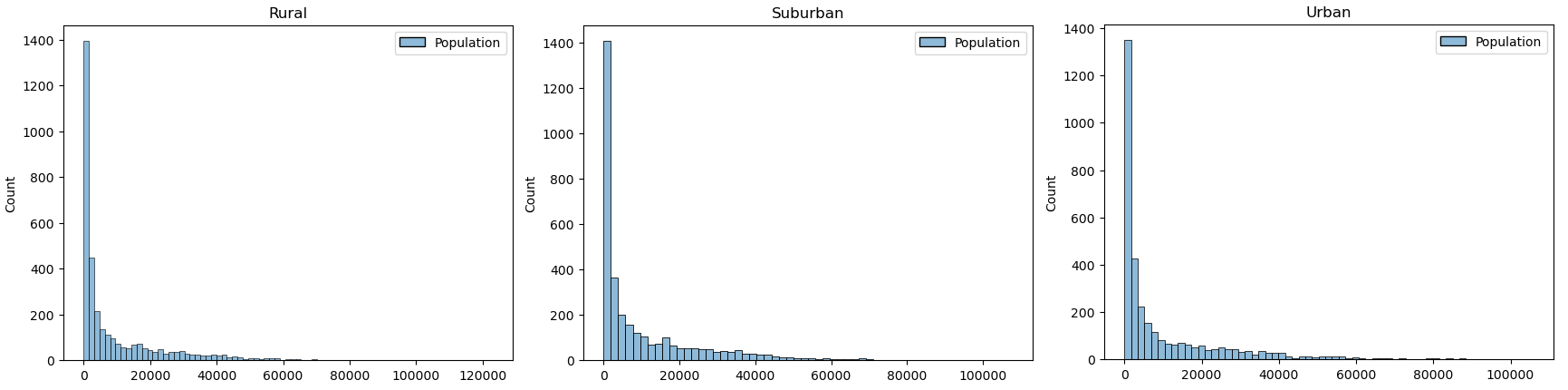
I dropped the duplicate index column from the data frame. No further action was required.

### Outliers

Before imputing missing values, it is first necessary to treat outliers in the data. I plan to use a multivariate method of imputation (MICE) and the presence of outliers in the data would degrade the effectiveness of that method. The variables with outliers are presented below.

*Population*

The 'Population' variable shows a very right-skewed distribution. There are 109 records with zero population recorded within one mile, but this is not outside the range of the distribution, given that the 25th percentile value is only 694. We cannot assume that a zero value recorded indicates missing data in this case.

There is, however, a significant issue with the distribution of the 'Population' variable given its definition in the data dictionary. One would expect to see a correlation between the 'Population' variable (which indicates the population in a one-mile radius of the patient address) and the 'Area' variable (which indicates whether a patient lives in a rural, suburban, or urban area). In this case, though, the distributions of the 'Population' variable are similar for all three possible values of 'Area' – see Figure 2. Were this an actual data analysis for a business, I would recommend re-evaluating the 'Population' and 'Area' variables using the address fields and up-to-date US Census data. For the purposes of this exercise, we will ignore this discrepancy and move on to evaluating whether to keep the outliers. The maximum value of the 'Population' variable is 122,814, which implies a local population density of 39,093 people per square mile (122814 people divided by π square miles in a 1-mile radius). This is well within the maximum county-level population density for urban areas per the U.S. Census (2010). Therefore, we will retain these outliers, even though they are outside the standard 1.5x inter-quartile range (IQR).

Figure

*Distribution of Population Counts by Area Type*

*Children*

The next quantitative variable I considered is the 'Children' variable. This distribution is also right-skewed. A boxplot indicates that values above 7 can be considered outliers. I was unable to find a good reference for the number of children in a household. The Census data I found simply reported values of 0, 1, 2, 3, or “4 or more” children, so I’m unable to determine if the tail of this distribution makes sense. For this reason, I have chosen to exclude the rows with 8 or more children reported from the cleaned data frame.

*Income*

The distribution of 'Income' in the data set is right-skewed, and not unreasonable. Even though there are outliers shown in the box plot, the maximum value of $207,249 in the data set is at approximately the 90th percentile of US annual income (Semega & Kollar, 2002, Table A-4a). All outliers for this variable have been retained.

*Vitamin D Levels*

The distribution of 'VitD\_levels' in the data set is bimodal, as can be seen in the histogram plot, Figure 3.

**Figure 3**

*Histogram of VitD\_levels in the original data set*

A graph of a number of levels

Description automatically generated

This separate distribution could indicate an error in data collection. For instance, a measurement device could be set to a different scale or set of units. For Vitamin D blood serum levels, the data dictionary indicates that the values should be expressed in ng/mL. According to the National Institutes of Health, “Serum concentrations of [Vitamin D] are reported in both nanomoles per liter (nmol/L) and nanograms per milliliter (ng/mL). … 1 ng/mL is equal to 2.5 nmol/L.” (Office of Dietary Supplements, 2022). As it happens, the mean of the right-side distribution (recorded values > 35, **x̄** = 47.1) is roughly 2.5 times the mean of the left-side distribution (recorded values < 35, **x̄** = 18.0). I therefore assume that all values above 35 were incorrectly expressed as nmol/L rather than ng/mL. I have scaled these values by dividing them by 2.5. The revised histogram and box plot are shown in Figure 4.

**Figure 4**

*Histogram and box plot of VitD\_levels after treatment of outliers*

A graph of a function

Description automatically generated

The revised data is distributed normally. There are now a few outliers outside of the 1.5\*IQR whiskers in the box plot of the revised distribution, but these appear to be valid data.

*Full Meals Eaten*

There are a few outliers in this right-skewed distribution, but the data dictionary specifically indicates that some patients request more than 3 meals in a day. Therefore, we will retain these outliers.

*Vitamin D Supplements*

Similarly, with 'VitD\_supp' there are a few records that indicate more than 2 supplements given, which could be considered outliers. The histogram shows that there are few records like this, though, and no reason to think that this data is inaccurate.

*Length of initial stay in hospital*

'Initial\_days' has a bimodal distribution. There are no apparent outliers that need to be treated.

*Average Daily Total Charge*

The distribution of values of 'TotalCharge' is interesting – two disjoint bimodal distributions. It appears that the smaller right-hand distribution is similar to the larger left-hand distribution but shifted up by approximately 11,700. Unlike the factor of 2.5 that was present between the two disjoint distributions of 'VitD\_levels', which could be explained by an error in units, I did not find a satisfactory explanation for the apparent systematic error in 'TotalCharge'. I performed some Exploratory Data Analysis, and I did find that these two errors were completely correlated – only those records with the Vitamin D levels recorded in nmol/L had the higher average daily total charge. I also found that whether or not a patient was a readmit explained the bimodality of each half of the distribution, but did not appear to correlate to whether the recorded 'TotalCharge' was in the left-hand or right-hand distribution – See Figure 5. To treat these outliers, I subtracted the difference in means of the two distributions from the values in the right-hand set.

**Figure 5**

*Pair plot of original Vitamin D levels in data set with Daily Average Total Charge*

A screenshot of a computer screen

Description automatically generated

*Additional Charges*

Finally, I address the 'Additional\_charges' variable. During EDA, I discovered an interesting relationship between 'Additional\_charges', 'Age', and 'HighBlood'. I created a variable called 'high\_line' to capture the distinct population with the greater slope on the age vs. additional charges graph. As can be seen in Figure 6, the rows where this is true have a distinct, mostly uniform distribution within 'Additional\_charges'. This population accounts for nearly all the outliers seen in the box plot of 'Additional\_charges'. There are no records in that 'high\_line' population without high blood pressure. Given that outliers can be explained by a population difference, I chose to retain all values for this variable.

**Figure 6**

*Pair plots of Age, Additional Charges, and High Blood Pressure*

A screenshot of a computer screen

Description automatically generated

### Missing Values

Having dealt with outliers, I proceeded to use the Multiple Imputation by Chained Equations (MICE) technique to impute missing values. Per the Getting Started with Missing Values course material, MICE is useful to “impute mixes of … data [types]” such as we have in this data set (Middleton, n.d.). I isolated the columns with quantitative or ordinal categorical values for the MICE algorithm to run on. Since MICE typically generates floating point output, and all the columns with missing values identified in Table 2 should be of type integer, I then rounded the output values. However, upon inspection of the imputed data, there were several problems with out-of-bounds data (i.e., negative days stayed in hospital, 80 children present, etc.). Upon finding this, I chose to switch to using the K-Nearest Neighbor (KNN) methodology.

## D3. Summary

At the end of the cleaning process, I have a data frame with 9697 rows. 303 rows were excluded due to outliers. All remaining rows have complete data population; there are no remaining missing values. There are still some outliers in the data; their retention was justified in the section above. See cell 34 of the attached D206 Jupyter notebook for histograms and box plots of the cleaned data.

## D4. Treatment Code

The treatment code is below and can also be found in the attached Jupyter notebook, cells 12-26. The revised code to treat missing values with KNN rather than MICE is found in cells 30-32.

*#Data Cleaning*

*#Treating Duplicates*

*#Drop the duplicated column*

df **=** df**.**drop(columns**=**['Unnamed: 0'])

df**.**tail(5)

*# Treating Outliers*

*# Population*

*# See if Population distribution makes sense given Area descriptions*

print(df['Population']**.**describe())

seaborn**.**histplot(df**.**loc[(df['Area'] **==** 1),['Population']])**.**set(title**=**'Rural')

plt**.**show()

seaborn**.**histplot(df**.**loc[(df['Area'] **==** 2),['Population']])**.**set(title**=**'Suburban')

plt**.**show()

seaborn**.**histplot(df**.**loc[(df['Area'] **==** 3),['Population']])**.**set(title**=**'Urban')

plt**.**show()

df**.**loc[(df['Population'] **<** 100),['Population', 'Area']]

*# Leaving Population outliers in the data set*

*# Children*

*# Exclude rows reporting 8 or more children as outliers - move them to a separate data frame. (Middleton, n.d.)*

chil\_outliers **=** df[(df['Children'] **>=** 8)]

df**.**drop(df[(df['Children'] **>=** 8)]**.**index, inplace**=True**)

df**.**shape

*#Age*

*#Uniform distribution - no outliers*

*#Income*

*#Outlying values are within reason. Income levels of $207,000/year are expected in a population.*

df['Income']**.**describe()

# VitD\_levels

print(df**.**loc[(df['VitD\_levels'] **>** 35), 'VitD\_levels']**.**describe())

print(df**.**loc[(df['VitD\_levels'] **<** 35), 'VitD\_levels']**.**describe())

*# Scale recorded values above 35 by factor of 2.5 to convert from nmol/L to ng/mL*

df['VitD\_levels']**=**np**.**where(df['VitD\_levels'] **>** 35, df['VitD\_levels'] **/** 2.5, df['VitD\_levels'])

print(df**.**loc[(df['VitD\_levels'] **>** 35), 'VitD\_levels']**.**describe())

print(df**.**loc[(df['VitD\_levels'] **<** 35), 'VitD\_levels']**.**describe())

*# The statistical description of the revised VitD\_levels distribution is very close to the original left-hand (values < 35) distribution.*

col**=**'VitD\_levels'

fig, axs **=** plt**.**subplots(1,2, figsize**=**(15,5))

seaborn**.**histplot(df[col], ax**=**axs[0])

plt**.**title(col)

seaborn**.**boxplot(df[col], orient**=**'h', ax**=**axs[1])

plt**.**title(col)

plt**.**show()

*#Doc\_visits*

*#Normal distribution, no outliers*

*#Full\_meals\_eaten*

*#Right-skew distribution. Data dictionary indicates that more than 3 meals per day are sometimes requested by patients, so no need to remove these outliers.*

*#Initial\_days*

*#Bimodal distribution. No outliers on box plot.*

*#TotalCharge*

*#Two disjoint bimodal distributions. I looked at various combinations of variables to explain this and found a correlation between the original faulty VitD\_levels and the erroneous high TotalCharge. The bimodality is correlated to ReAdmis.*

seaborn**.**pairplot(df\_orig[['VitD\_levels','TotalCharge','ReAdmis']], hue**=**'ReAdmis')

*#As was done for VitD\_levels, compare the two halves of the distribution*

print(df**.**loc[(df['TotalCharge'] **>** 12000), 'TotalCharge']**.**describe())

print(df**.**loc[(df['TotalCharge'] **<** 12000), 'TotalCharge']**.**describe())

*#Shift the upper distribution down by the difference in means*

col**=**'TotalCharge'

meandiff **=** (df**.**loc[(df[col] **>** 12000), col]**.**mean()) **-** (df**.**loc[(df[col] **<** 12000), col]**.**mean())

print(meandiff)

df[col]**=**np**.**where(df[col] **>** 12000, df[col] **-** meandiff, df[col])

*#Check stats for revised distribution*

print(df**.**loc[(df[col] **>** 12000), col]**.**describe())

print(df**.**loc[(df[col] **<** 12000), col]**.**describe())

fig, axs **=** plt**.**subplots(1,2, figsize**=**(15,5))

seaborn**.**histplot(df[col], ax**=**axs[0])

plt**.**title(col)

seaborn**.**boxplot(df[col], orient**=**'h', ax**=**axs[1])

plt**.**title(col)

plt**.**show()

*#Additional\_charges*

*#Interesting relationship between Age, Additional\_charges, and HighBlood, but no reason to exclude the outliers.*

*# This is a key to distinguish between the population at the top of the Age vs Additional\_charges graph and those at the bottom*

df['high\_line'] **=** (df['Additional\_charges']**/**(df['Age']**+**20) **>** 150)

seaborn**.**pairplot(df[['Age','Additional\_charges','HighBlood','high\_line']], hue**=**'high\_line')

df**.**drop('high\_line', axis**=**1, inplace**=True**)

*#Treating Missing Values*

*#I will use MICE (IterativeImputer) from the fancyimputer package. First I will save a copy of the data frame now that I have removed outliers. MICE code adapted from DataCamp (Donthi, n.d.).*

df\_outl\_removed **=** df**.**copy(deep**=True**)

*# For MICE technique, only use the part of the dataset with quantitative & ordinal categorical variables*

*# Reduce dataset to remove survey, ID variables, Job, Employment, Services, Marital, Gender, Initial\_admin*

dfr **=** df**.**iloc[:,14:]

dfr **=** dfr[[c **for** c **in** dfr**.**columns **if** **not** c**.**startswith('Item')]]

dfr**.**drop(columns**=**['Employment', 'Services', 'Marital', 'Gender', 'Initial\_admin'], inplace**=True**)

dfr**.**info()

*# Install package if needed for environment (Vanderplas, 2017)*

*#import sys*

*#!{sys.executable} -m pip install fancyimpute*

*# MICE code adapted from Datacamp (Donthi, n.d.)*

**from** fancyimpute **import** IterativeImputer

mice **=** IterativeImputer()

dfr\_mice **=** dfr**.**copy(deep**=True**)

dfr\_mice**.**iloc[:,:] **=** mice**.**fit\_transform(dfr)

*# The variables with missing values were all encoded as integer - need to round of the float output of MICE*

cols\_with\_missing **=** ['Children', 'Age', 'Income', 'Soft\_drink', 'Overweight', 'Anxiety', 'Initial\_days']

dfr\_mice[cols\_with\_missing] **=** dfr\_mice[cols\_with\_missing]**.**round()

*# Some Initial\_days values were imputed as negative - clamp to 1 as min value*

dfr\_mice[dfr\_mice['Initial\_days'] **<** 1] **=** 0

*# Clamp minimum age to 20, maximum to 90 to match original data set*

dfr\_mice[dfr\_mice['Age'] **<** 20] **=** 20

dfr\_mice[dfr\_mice['Age'] **>** 90] **=** 90

*# Copy these back into larger data frame*

df[cols\_with\_missing] **=** dfr\_mice[cols\_with\_missing]

*# Verify no columns have missing values*

df**.**isna()**.**sum()

msno**.**matrix(df, labels**=True**)

*#Show data histograms / box plots after removal of unwanted outliers and imputation of missing values*

**for** col **in** quant\_cols:

fig, axs **=** plt**.**subplots(1,2, figsize**=**(15,5))

seaborn**.**histplot(df[col], ax**=**axs[0])

plt**.**title(col)

seaborn**.**boxplot(df[col], orient**=**'h', ax**=**axs[1])

plt**.**title(col)

plt**.**show()

*#MICE introduced some odd outliers, and it was necessary to clamp output to restrict invalid values. I will also try KNN to see if the output produced is more sane.*

*# KNN code adapted from Datacamp (Donthi, n.d.)*

**from** fancyimpute **import** KNN

knn **=** KNN()

dfr\_knn **=** dfr**.**copy(deep**=True**)

dfr\_knn**.**iloc[:,:] **=** knn**.**fit\_transform(dfr)

dfr\_knn**.**head(20)

*# The variables with missing values were all encoded as integer - need to round of the float output of KNN*

cols\_with\_missing **=** ['Children', 'Age', 'Income', 'Soft\_drink', 'Overweight', 'Anxiety', 'Initial\_days']

dfr\_knn[cols\_with\_missing] **=** dfr\_knn[cols\_with\_missing]**.**round()

*# Copy these back into larger data frame*

df[cols\_with\_missing] **=** dfr\_knn[cols\_with\_missing]

*#Show data histograms / box plots after removal of unwanted outliers and imputation of missing values*

**for** col **in** quant\_cols:

fig, axs **=** plt**.**subplots(1,2, figsize**=**(15,5))

seaborn**.**histplot(df[col], ax**=**axs[0])

plt**.**title(col)

seaborn**.**boxplot(df[col], orient**=**'h', ax**=**axs[1])

plt**.**title(col)

plt**.**show()

*#Outliers look reasonable, no out-of-bounds data - stay with KNN.*

*# Output dataframes to CSV files*

df**.**to\_csv('clean\_medical\_data.csv', index**=False**)

chil\_outliers**.**to\_csv('children\_outliers.csv', index**=False**)

## D5. CSV files

See attached files: 'clean\_medical\_data.csv' and 'children\_outliers.csv'.

## D6. Process limitations

*Duplicates –* Using the duplicated() method only finds entire rows that are duplicated. I attempted to account for this by also checking for duplicates just on the Customer\_id field.

*Missing Values –* My initial plan was to impute missing values with the MICE methodology. I found that for several variables, nonsensical or out-of-range data was generated. For this reason, I switched to the KNN method. With both methods, categorical variables are imputed as floats which must be converted to integers.

*Outliers –* For one variable, I chose to exclude the rows with detected outliers. This has the potential to throw away useful data from the analysis. For other variables, where I scaled or shifted values to address a likely erroneous set of data, I was relying on educated guesses about what was going on with the data. Imperfect domain knowledge on the part of the analyst can easily lead to incorrect treatment of outlier data.

## D7. Implications for research question

Excluding or modifying outlier data could mask a true relationship between the data in question and the research topic being studied. In this example, it could be that a large number of children at home, or an abnormally high Vitamin D blood serum level could have an impact on hospital readmissions. If that were the case, my treatment of the data would prevent that relationship from being discovered.

## E1. Principal Component Analysis – Setup

Principal Component Analysis (PCA) works best with continuous variables (Middleton, n.d.). There are five continuous variables in the medical data set. I have also chosen to include the discrete variables 'Age' and 'Population' since they each take on a wide range of values. The other discrete variables and the categorical variables are not good candidates for PCA. After normalization of the chosen variables, I ran the PCA.fit\_transform() method. The loadings matrix is reproduced below.

**Figure 7**

*Explained variance ratio and loadings matrix from PCA*

A screenshot of a computer

Description automatically generated

## E2. Principal Component Reduction – Justification

The Kaiser rule indicates keeping any principal component with an eigenvalue >= 1. The scree plot from our data is interesting – 3 PCs have an eigenvalue very close to 1 (see Figure 8).

**Figure 8**

*Scree plot of Eigenvalues*

A graph with a line

Description automatically generated

The eigenvalue for PC4 is 1.0003; the eigenvalue for PC5 is 0.9874. A strict application of the Kaiser rule would have me keep only the first 4 PCs. However, looking at the explained variance ratio, retaining the 5th PC explains over 95% of the variance; limiting to 4 PCs only explains 81.8%. Further, unless PC5 is retained, almost all the information from 'Population' and 'Additional\_charges' would be lost. In the loading matrix, neither of those variables have a load of greater than 0.1 in any of the first 4 PCs. Therefore, I would choose to retain the 5 most important principal components – PC1, PC2, PC3, PC4, and PC5.

## E3. Why PCA?

There are a number of benefits for data scientists and organizations to use PCA. In a Medium post, quantitative analyst Roland Jeannier identifies 5 main benefits: dimensionality reduction, increased performance, the ability to visualize higher dimensional data, the ability to obscure sensitive data, and to create independent features for use in linear regression (2017). One benefit of dimensionality reduction is to reduce the chance of models overfitting on extraneous variables (Boeye, n.d.). With fewer variables to analyze, computational models will come to solutions more quickly. High-dimensional spaces can be impractical or impossible to visualize; PCA can reduce the dimensionality down to something that can be plotted in 2-D or 3-D space. Additionally, organizations with requirements to protect sensitive data such as personal identifying information (PII) or payment card industry (PCI) data can obscure the data in their data sets while maintaining the full predictive value by transforming the data via PCA and retaining all the principal components.

# Part IV. Supporting Documentation

## F. Demonstration Video

## A video describing my methods and code can be found at:

## <https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=e170a4c4-aa00-4c1e-9ae2-b0700121a097>

## G. Third-party Code Sources

Bilogur, A. (2019, July 19). *Data labels missing on x-axis.* GitHub. <https://github.com/ResidentMario/missingno/issues/93>

Boeye, J. (n.d.) Lesson 4: Feature Extraction. *Dimensionality Reduction in Python* [MOOC]. DataCamp. <https://campus.datacamp.com/courses/dimensionality-reduction-in-python/feature-extraction?ex=6>

Donthi, S. (n.d.) Lesson 4: Advanced Imputation Techniques. *Dealing with Missing Data in Python* [MOOC]. DataCamp. <https://campus.datacamp.com/courses/dealing-with-missing-data-in-python/advanced-imputation-techniques>

Kassies, R. (2013, May 6). *Make more than one chart in same IPython Notebook cell.* StackOverflow. <https://stackoverflow.com/questions/16392921/make-more-than-one-chart-in-same-ipython-notebook-cell>

Vanderplas, J. (2017, December 5). *Installing Python Packages from a Jupyter Notebook*. Pythonic Perambulations. <https://jakevdp.github.io/blog/2017/12/05/installing-python-packages-from-jupyter/>

## H. References

Boeye, J. (n.d.) Lesson 2: The curse of dimensionality. *Dimensionality Reduction in Python* [MOOC]. DataCamp. [https://campus.datacamp.com/courses/dimensionality-reduction-in-python/feature-selection-i-selecting-for-feature-information?ex=1](https://campus.datacamp.com/courses/dimensionality-reduction-in-python/feature-selection-i-selecting-for-feature-information?ex=1%20)

Donthi, S. (n.d.) Lesson 2: Does Missingness Have a Pattern? *Dealing with Missing Data in Python* [MOOC]. DataCamp. <https://campus.datacamp.com/courses/dealing-with-missing-data-in-python/does-missingness-have-a-pattern>

Jeannier, R. (2017, September 8). *What’s the Point of PCA Anyway?* Medium. <https://medium.com/@rtjeannier/whats-the-point-of-pca-anyway-279cf0ef0683>

Larose, C., & Larose, D. (2019). *Data Science Using Python and R.* Wiley.

Middleton, K. (n.d.). *Getting Started with D206 | Detecting and Treating Duplicates.* Western Governors University. <https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=6eedfad4-240e-4c5c-8eab-b058003d3e6b>

Middleton, K. (n.d.). *Getting Started with D206 | Detecting and Treating Missing Values.* Western Governors University. <https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=767749d2-ba19-4f94-bec8-b058017b2f5e>

Middleton, K. (n.d.). *Getting Started with D206 | Detecting and Treating Outliers.* Western Governors University. <https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=19c24c56-0f37-408e-bb1f-b059002a77ac>

Middleton, K. (n.d.). *Getting Started with D206 | Principal Component Analysis.* Western Governors University. https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=3bcc452f-fa35-43be-b69f-b05901356f95

Office of Dietary Supplements (2022, August 12). *Vitamin D: Fact Sheet for Health Professionals.* National Institutes of Health, U.S. Dept. of Health & Human Services. <https://ods.od.nih.gov/factsheets/VitaminD-HealthProfessional/>

Semega, J. & Kollar, M. (2022, September 13). *Income in the United States: 2021.* U.S. Census Bureau. <https://www.census.gov/library/publications/2022/demo/p60-276.html>

Tamboli, N. (2021, October 29). *Effective Strategies for Handling Missing Values in Data Analysis*. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2021/10/handling-missing-value/>

United States Census Bureau (2010). *Population Density by County:2010.* <https://www.census.gov/library/visualizations/2010/geo/population-density-county-2010.html>

Western Governors University. (n.d.) *R or Python.* <https://www.wgu.edu/online-it-degrees/programming-languages/r-or-python.html>