D206 Data Cleaning Performance Assessment

Joe Paris

School of IT, Western Governors University

Dr. Middleton

February 15, 2024

D206 Data Cleaning Performance Assessment

Part I: Research Question and Variables

A: Research Question

For this performance assessment the `medical_raw_data` dataset will be used to explore the question: is there any correlation between pre-existing conditions and patient readmission?

B: Variables

The `medical_raw_data` dataset contains 10,000 observations consisting of fifty-two variables each (Table 1).

Table 1 Variables in the medical_raw_data dataset

Variable	Data Type	Example	Description
Additional_charges	quantitative	7361.105691	Average amount charged to the patient per day for procedures/treatments.
Age	quantitative	45	Patient's age as reported in their admissions information.
Allergic_rhinitis	qualitative	No	Whether or not the patient has been diagnosed with allergic rhinitis.
Anxiety	qualitative	Yes	Whether or not the patient has been diagnosed as suffering from anxiety.
Area	qualitative	Suburban	Type of area the residence on the billing statement is in.
Arthritis	qualitative	No	Whether or not the patient has been diagnosed with arthritis.
Asthma	qualitative	No	Whether or not the patient has been diagnosed with asthma.
BackPain	qualitative	Yes	Whether or not the patient has back pain.
CaseOrder	qualitative	1	Dummy variable used to maintain the original order of the table.

Variable	Data Type	Example	Description
Children	quantitative	1	Number of children in the patient's household as reported in their admissions information.
City	qualitative	West Point	The city on the billing statement.
Complication_risk	qualitative	medium	Assessed level of risk for complications.
County	qualitative	Geauga	The county on the billing statement.
Customer_id	qualitative	Z919181	Unique identifier for the patient.
Diabetes	qualitative	Yes	Whether or not the patient has been diagnosed with diabetes.
Doc_visits	quantitative	6	Number of times the patient's primary care physician saw the patient during their initial hospitalization.
Education	qualitative	Bachelor's Degree	Patient's highest level of education as reported in their admissions information.
Employment	qualitative	Part Time	Patient's employment status as reported in their admissions information.
Full_meals_eaten	quantitative	1	Number of meals completely eaten by the patient during their hospitalization.
Gender	qualitative	Female	Patient's self-identified gender.
HighBlood	qualitative	Yes	Whether or not the patient has been diagnosed as having high blood pressure.
Hyperlipidemia	qualitative	Yes	Whether or not the patient has been diagnosed with hyperlipidemia.
Income	quantitative	61723.87	Patient's annual income as reported in their admissions information.
Initial_admin	qualitative	Emergency Ad...	The way the patient was initially hospitalized.
Initial_days	quantitative	11.74786866	Number of days the patient was admitted during their initial hospitalization.
Interaction	qualitative	a2057123-abf5...	Unique identifier for each interaction with the patient.
Item1	quantitative	5	Survey response ranking the importance of timely admission.
Item2	quantitative	3	Survey response ranking the importance of timely treatments.

Variable	Data Type	Example	Description
Item3	quantitative	3	Survey response ranking the importance of timely visits.
Item4	quantitative	5	Survey response ranking the importance of reliability.
Item5	quantitative	3	Survey response ranking the importance of options.
Item6	quantitative	3	Survey response ranking the importance of hours of treatment.
Item7	quantitative	2	Survey response ranking the importance of courteous staff.
Item8	quantitative	4	Survey response ranking the importance of evidence of active listening by the attending physician.
Job	qualitative	Actuary	Patient's job as reported in their admissions information.
Lat	quantitative	30.20097	Latitude of the residence on the billing statement.
Lng	quantitative	-94.71424	Longitude of the residence on the billing statement.
Marital	qualitative	Separated	Marital status of the patient or the primary insurance holder as reported in their admissions information.
Overweight	qualitative	Yes	Whether or not the patient is considered overweight.
Population	quantitative	426	Population within a one-mile radius of the residence on the billing statement.
ReAdmis	qualitative	No	Whether or not the patient was readmitted within a month of initially being released.
Reflux_esophagitis	qualitative	No	Whether or not the patient has been diagnosed with reflux esophagitis.
Services	qualitative	Blood Work	Primary services provided to the patient during their hospitalization.
Soft_drink	qualitative	Yes	Whether or not the patient normally drinks three or more soft drinks per day.
State	qualitative	OK	The state on the billing statement.

Variable	Data Type	Example	Description
Stroke	qualitative	No	Whether or not the patient has suffered a stroke.
Timezone	qualitative	America/Chicago	Time zone of the residence on the billing statement.
TotalCharge	quantitative	3574.732199	Average amount charged to the patient per day for typical expenses.
UID	qualitative	cd17d7b6d152...	Unique identifier combining patient, admission, and procedures.
VitD_levels	quantitative	25.51463527	The patient's vitamin D level at time of admittance.
VitD_supp	quantitative	0	Number of vitamin D supplements given to the patient during their hospitalization.
Zip	qualitative	22641	The zip code on the billing statement.

Part II: Planning

C1: Quality Assessment Plan

The following outlines the methods used to detect duplicate and missing values, outliers, and the re-expression of categorical variables:

1. Install all the Python modules to be used to clean the data and import them. Placing all imports at the top of the file is considered to be a best practice (van Rossum, Warsaw, & Coghlan, 2001).
2. Make a copy of the dataset to work on so it can be reverted to its original state if necessary (Nelli, 2023).
3. Build a DataFrame from the file using pandas' `.read_csv()` method using the first, unnamed column in the CSV file as the index to avoid creating a redundant variable.

4. Next, perform basic exploratory analysis of the data using the DataFrame's `.shape` and `.columns` properties and its `.describe()` and `.head()` methods to get a basic feel for the dataset.
5. Check for duplicated rows using the `.duplicated()` method of the DataFrame.
6. Additionally, check for duplicate values in variables that should be unique (such as *CaseOrder*, *Customer_Id*, *Interaction*, and *UID*) using the `.duplicated()` method and supplying the `subset` parameter.
7. Missing qualitative and quantitative values can be found using the DataFrame's `.info()` method and isolated using a combination of `isna()` and `.sum()`. The type of missingness will be explored using `missingno.matrix()`, `msno.heatmap()`, and `msno.dendrogram()`.
8. Using scipy's `stats.zscore()` (scipy, n.d.) method, find outliers in quantitative variables in the dataset (scipy, n.d.). Seaborn will also be useful to generate graphs of the data for easier visual comparison.
9. Categorical variables will be re-expressed numerically using the DataFrame's `.replace()` method and appropriate new encodings (Larose & Larose, 2019).

C2: Quality Assessment Justification

The techniques chosen for identifying data quality issues are industry-standard methods. Making a backup of the data before modifying it helps guard against accidental loss of data (Nelli, 2023).

Duplication in a dataset, whether it be duplicated rows or duplicate values in a column whose data should be unique, can skew the analysis results and lead to incorrect conclusions. Duplicated data can be a sign of defective data and predictions made are only as good as the data they are based on.

Additionally, duplicated data increases the size of the dataset leading to less efficient use of computing resources.

Missing values in the dataset represent a loss of information, in the least, and can lead to bias in the analysis. The ability to do meaningful statistical analysis on the data will be reduced due to the reduction in sample size. Furthermore, many machine learning algorithms cannot work with datasets that are missing data. For all these reasons, missing data must be dealt.

Outliers may have a similar impact on the data, skewing it and reducing the accuracy of the analysis or predictions. While outliers are not necessarily bad data, the oversized effect they can have on the analysis needs to be guarded against and adjusted for.

The re-expression of categorical variables enables more effective data analysis and modeling, leading to better insights and predictions. Most machine learning models operate on numerical data only necessitating the expression of categorical variables as numeric values (Sonoda, 2024). Reduced dimensionality is another benefit of re-expressing categorical variables. In lieu of encoding techniques such as one-hot encoding, categorical values can be grouped into a reduced number of numerical values.

C3: Language and Libraries Justification

Python was chosen as the language for this performance assessment for several reasons. In February, 2024 it was ranked as the number one most popular programming language on the TIOBE index and has been in the top ten for over a decade (TIOBE, 2024). It is an approachable language that is both easy to learn and easy to read with its English-like syntax. It has extensive support for data analysis due to the libraries that are available for it including NumPy, pandas, matplotlib, scipy and scikit-learn among thousands of others (McKinney, 2022). Because of Python's popularity and the popularity of the libraries that are available for it, support is easy to find when problems arise.

C4: Detection Code

See the accompanying `d206_paris_performance_assessmsnt.ipynb` file.

Part III: Data Cleaning

D1: Findings

`CaseOrder` is a generated dummy variable used to track the original order of the observations in the data set. As such, it is not truly part of the data, and its presence will be ignored for purposes of looking for duplicate observations. No duplicated observations were found in the dataset. The `CaseOrder`, `Customer_id`, `Interaction`, or `UID` variables each should be unique and were also checked for duplicate values. A case can be made for checking `CaseOrder` because as a sort of indexing field, duplicate values are not allowed. No duplicate values were found in these variables in the dataset.

The `Children`, `Age`, `Income`, `Soft_drink`, `Overweight`, `Anxiety`, and `Initial_days` variables all had missing values. Table 2 shows number of missing values in each. The nullitiy matrices, heatmap, and dendrogram show no correlation between the missing values. `Children`, `Age`, `Income`, `Soft_drink`, `Overweight`, `Anxiety`, and `Initial_days` are all missing completely at random.

Table 2 Missing values per variable

Variable	Number of Missing Values
Age	2414
Anxiety	984
Children	2588
Income	2464
Initial_days	1056
Overweight	982
Soft_drink	2467

An outlier is an unusually large or small value. A common definition of an outlier is a value that is greater than three times the standard deviation of the dataset it is in (Nelli, 2023). Simply being an outlier does not necessarily mean the data is faulty. However, outliers should be identified and their reason for existing in the data understood to avoid lowering the reliability of any analysis done. Table 3 shows the how many outliers were found using this definition across the dataset.

Table 3 Outliers per variable

Variable	Number of Outliers
Additional_charges	0
Age	0
Children	146
Doc_visits	8
Full_meals_eaten	33
Income	113
Initial_days	0
Item1	11
Item2	12
Item3	12
Item4	12
Item5	13
Item6	10
Item7	11
Item8	12
Lat	144
Lng	98
Population	218
TotalCharge	276
VitD_levels	500
VitD_supp	70

It's worth noting that some zip codes in the United States start with a zero and those that do are commonly written without the leading digit; i.e. 01234 would be written as 1234. Therefore, three-digit zip codes are, in fact, valid. Three-digit zip codes are common in Puerto Rico and all such zip codes in the dataset are associated with the island. Negative latitude and longitude are also valid values.

D2: Treatment

Table 2 lists the number of missing values per variable, and there are significant amounts of data missing for some variables in the dataset. The quantity of missing values (Table 4) prohibits simply dropping observations and dropping entire variables is impractical. Instead, we will consider the distribution (Figure 1) and type (Table 1) of each variable and select the most appropriate imputation method.

Table 4 Analysis of missing variables detail

Variable	Number of Missing Values	Percent Missing
Age	2414	24.14%
Anxiety	984	9.84%
Children	2588	25.88%
Income	2464	24.64%
Initial_days	1056	10.56%
Overweight	982	9.82%
Soft_drink	2467	24.67%

As shown in Figure 1, `Anxiety`, `Overweight`, and `Soft_drink` have bimodal distributions which makes sense given they are categorical in nature. `Inital_days` is also bimodal but not to the extreme of either `Anxiety` or `Overweight`. `Children` and `Income` skew positively to the right. `Age` has a remarkably even distribution.

Because of their distributions, `Age`, `Children` and `Income` were treated using univariate statistical imputation and replacing missing values with the median. The K-Nearest Neighbor technique was used to impute the missing values for `Initial_days`, and `Soft_drink` to generate data that is as accurate as possible using the other data in the dataset to guide the process. `Anxiety` and `Overweight` were imputed using scikit-learn's `SimpleImputer` and the `most_frequent` strategy. Imputation techniques are summarized in Table 5.

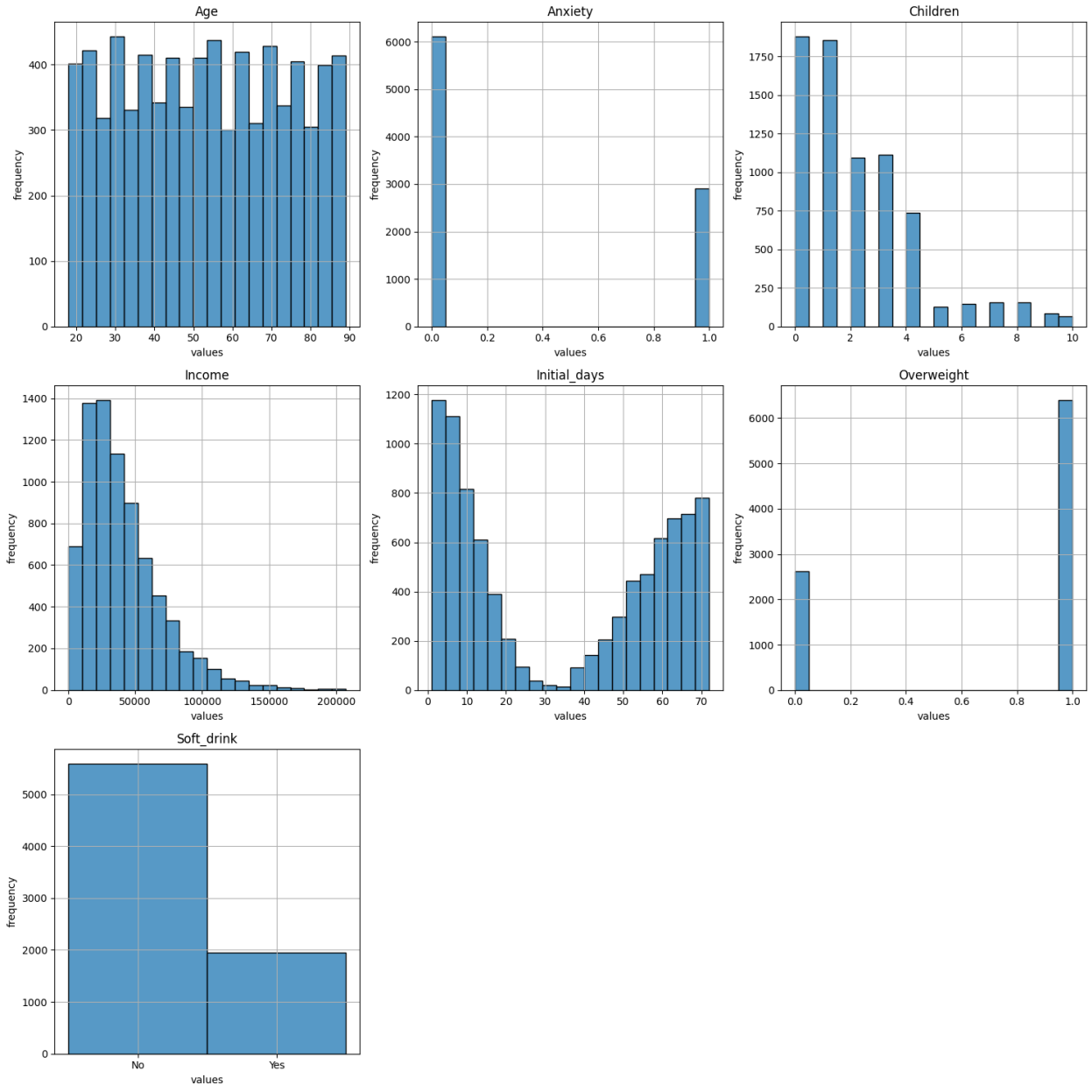
Table 5 Imputation methods

Variable	Imputation Method
Age	Univariate Statistical Imputation (mean)
Anxiety	Simple Imputer (most frequent)
Children	Univariate Statistical Imputation (mean)
Income	Univariate Statistical Imputation (mean)
Initial_days	K-Nearest Neighbor
Overweight	Simple Imputer (most frequent)
Soft_drink	K-Nearest Neighbor

Table 6 shows the number of identified outlying values per variable along with their minimum and maximum values. In some cases domain knowledge was required to identify whether the values were truly out-of-range. The United States covers a range of latitude from 18.9131° N to 49.3844° N and longitude from 179.7887° W to 66.9548° W (OpenAI, 2024). The values for longitude (`Lng`) in the dataset were within range but the values for latitude (`Lat`) were not. To correct this, latitude values were clamped to be between 49.3844 and 18.9131. Upon inspection, all other values (`Children`, `Doc_visits`, `Full_meals_eaten`, `Income`, `Item1`, `Item2`, `Item3`, `Item4`, `Item5`, `Item6`, `Item7`, `Item8`, `Population`, `TotalCharge`, `VitD_levels`, and `VitD_supp`) were within a reasonable range.

Table 7 lists the categorical variables in the dataset that needed to be re-encoded as numeric. In all cases, a new variable was added to the dataset using the name of the original variable with "`_num`" appended to it. As none of these variables had any intrinsic ordering and `OrdinalEncoder` was used.

Figure 1 Distribution of variables with missing values



Other miscellaneous clean-up was performed including rounding converting the `Children`, `Doc_visits`, and `Full_meals_eaten` columns to integer as fractional values for those variables do not make sense. `Income`, `TotalCharge`, and `Additional_charges` were rounded to two decimal places as they are all currency values. `Age` was rounded to a whole number. `VitD_levels` was rounded to two decimal places and `Initial_days` to one, both for the sake of clarity and consistency after having been imputed. Finally, for clarity, the survey response columns were renamed from `Item1`, `Item2`, … to `Question1`, `Question2`, and so forth.

Table 6 Outliers, count and range

Variable	Number of Outliers	Min	Max
Children	146	0	10
Doc_visits	8	1	9
Full_meals_eaten	33	0	7
Income	113	154.08	122814
Item1	11	1	8
Item2	12	1	7
Item3	12	1	8
Item4	12	1	7
Item5	13	1	7
Item6	10	1	7
Item7	11	1	7
Item8	12	1	7
Lat	144	17.96719	70.56099
Lng	98	-174.20969	-65.29017
Population	218	0	122814
TotalCharge	276	1256.751699	21524.22421
VitD_levels	500	9.519011638	53.01912416
VitD_supp	70	0	5

Table 7 Categorical variables

Variable
Allergic_rhinitis
Area
Arthritis
Asthma
BackPain
Complication_risk
Diabetes
Education
Employment
Gender
HighBlood
Hyperlipidemia
Initial_admin
Job
Marital
ReAdmis
Reflux_esophagitis
Services
Soft_drink
Stroke
Timezone

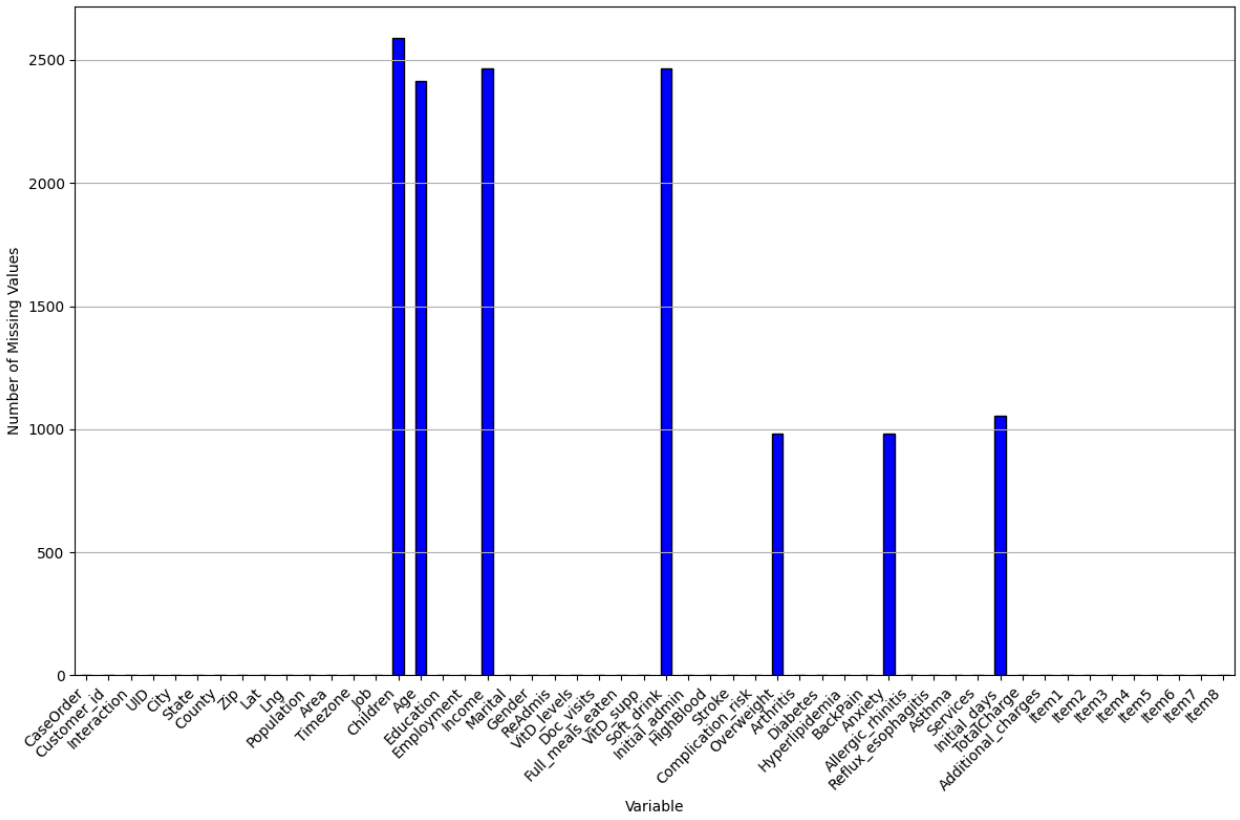
D3: Work Performed

The dataset was checked for duplicate observations, then for duplicate values in a number of key variables which by necessity should hold unique values. No duplicates were found.

There were 12,955 values missing from the dataset pre-treatment (Figure 2). Missing values were imputed using techniques including K-Nearest Neighbor, replacing the missing data with the most

frequently found existing value, or the mean of the existing values (Table 5). Post-treatment there were no missing values (Figure 3). Figure 4 shows the distribution of the variables with missing values pre- and post-treatment.

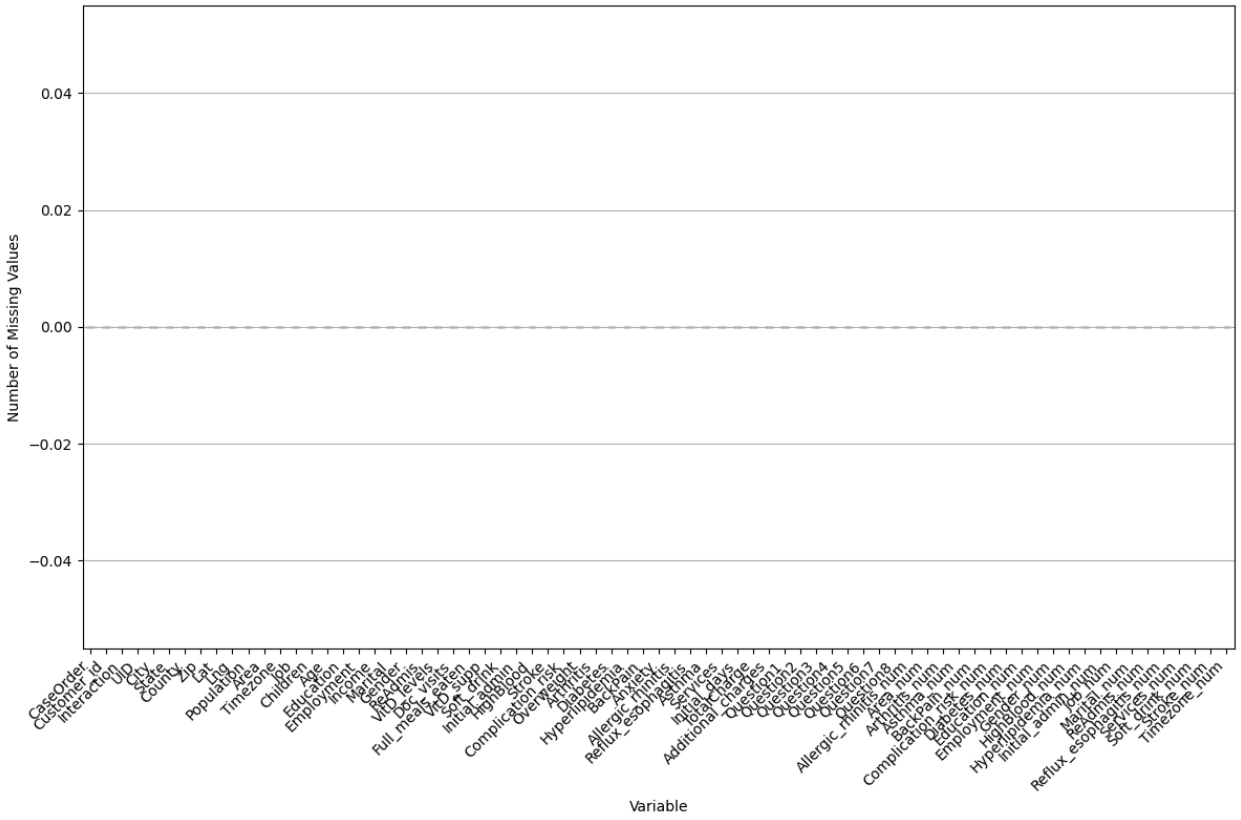
Figure 2 Number of missing values per variable (pre-treatment)



Outliers were examined and only latitude (`Lat`) was found to have values that were out-of-range. Values in the variable were clamped to the minimum and maximum latitude that bounds the United States (Figure 5, Figure 6).

Other miscellaneous clean-up was performed including rounding and renaming certain values as previously described.

Figure 3 Number of missing values per variable (post-treatment)



D4: Treatment Code

See the accompanying `d206_paris_performance_assessmsnt.ipynb` file.

D5: Clean Data

See the accompanying `medical_treated_data.csv` file.

D6: Clean Data Disadvantages

Univariate statistical imputation assumes the missing values are missing at random or missing completely at random. If this is not true, the imputed values will distort the distribution of the data leading to biased analysis of the data. Univariate imputation by its nature ignores interactions between variables which can also lead to inaccurate analysis due to added bias in the data. Additionally, statistical

imputation is sensitive to outliers in the data which can bias the imputed values and, again, lead to inaccurate analysis.

Figure 4 Variable distributions

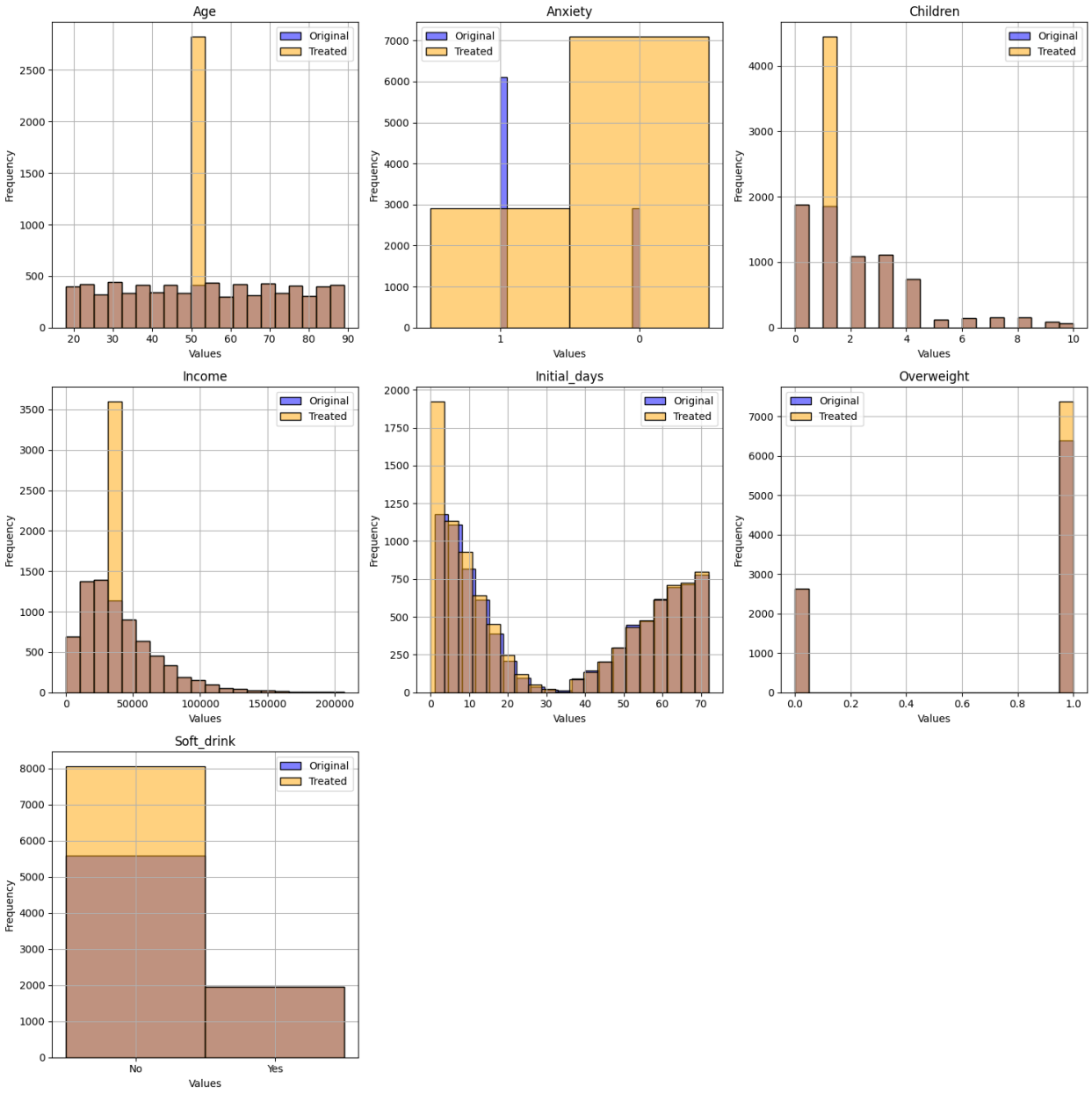


Figure 5 Latitude outliers boxplot

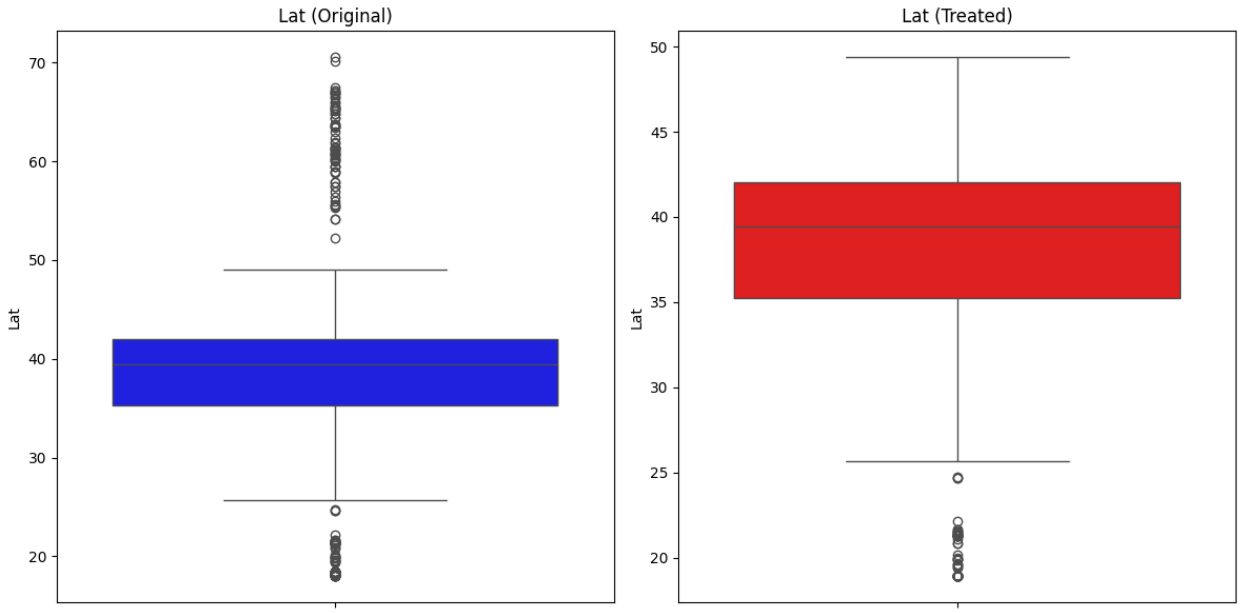
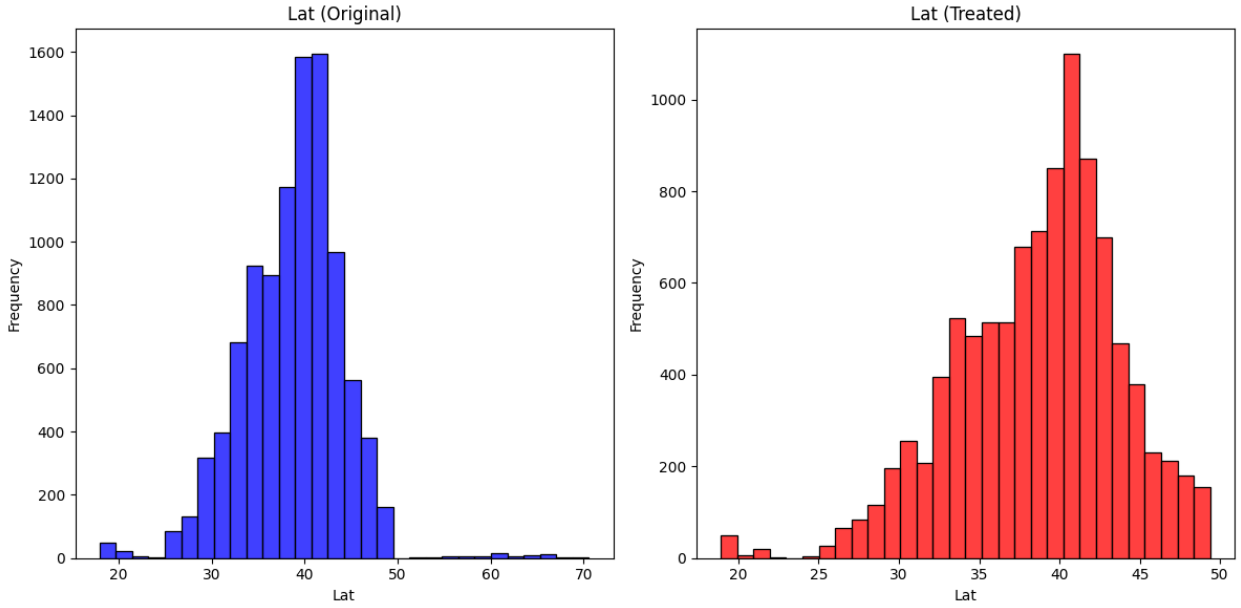


Figure 6 Latitude outliers histogram



Simple imputers suffer offer only a few available strategies which are appropriate for only certain types of data. These simple strategies may or may not capture underlying patterns present in the data. Likewise, simple imputers will ignore complex relationships between variables as, like univariate statistical imputation, they focus only on the data available for a single given variable. Also, much like univariate statistical imputers, they are sensitive to outliers and assume data is missing at random or completely at random.

Like simple and univariate statistical imputers, K-Nearest Neighbor imputers (KNNI) assume the missing values are missing at random or completely at random. Because their underlying theory requires that similar missing data points have similar present data points they are sensitive to both outliers and noise in the data. For this same reason, KNNI are vulnerable to patterns in the missing data. Finally, KNNI are computationally and memory intensive and may not be viable for very large datasets.

D7: Challenges

A recurring theme in the discussion of the disadvantages of the cleaning methods used was that of bias. An analyst using the now-cleaned data must be aware of the potential for bias in the data and work to minimize its impact on their analysis. Imputing data also carries certain assumptions about the missing data. While these assumptions should be clearly stated and controlled for the possibility that they do not hold true and therefore taking the imputed data exists. The is an amount of uncertainty inherent in imputed data, it is essentially a best guess at what the missing values are. There will inevitably be a difference between the imputed values and the actual values.

E1: Principal Component Analysis (PCA)

All continuous variables in the dataset were used to perform the principal component analysis (Table 8). The loading matrix (Figure 8) was generated based off code by Centellegher (2020).

Table 8 Variables for PCA

Variable
Lat
Lng
Population
Children
Age
Income
VitD_levels
Doc_visits
Full_meals_eaten
VitD_supp
Initial_days
TotalCharge
Additional_charges

E2: PCA Justification

The SCREE test and the Kaiser Rule are two methods to determine how many principal components to retain in a principal component analysis (Steiger, 2015). The Kaiser rule is fairly simplistic, simply stating that any principal component with an eigenvalue greater than 1 should be kept According to it and Figure 8 we would keep six principal components, PC1–PC6. By contrast, the SCREE test considers the principal components in descending order, attempting to identify where the impact of each “levels off” and thus the point where the components begin to have a diminished impact on the variance. A scree plot of our eigenvalues (Figure 9) suggests retaining four components. In order to reduce the number of dimensions being considered, the results of the SCREE test were chosen.

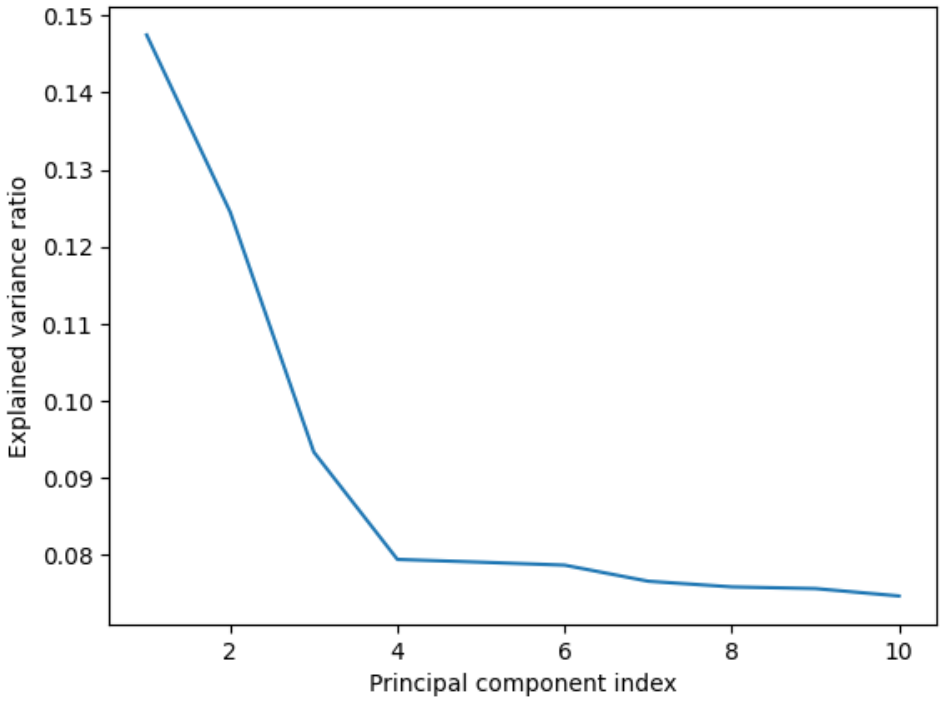
Figure 7 Loading matrix

```
         Lat       Lng  Population  Children       Age    Income  VitD_levels  \
0  -0.019326 -0.003172   -0.701053 -0.164124  0.025002  0.037355    -0.062295
1  -0.004858  0.015644    0.053433  0.272024 -0.471082 -0.572805     0.476184
2   0.022556 -0.026468    0.697467 -0.022039  0.100885  0.113616    -0.140962
3   0.003443  0.011313    0.028286 -0.218556  0.179824  0.395148     0.758700
4   0.094527  0.699613    0.009579 -0.010061 -0.026783  0.004733    -0.010307
5  -0.007685 -0.004929    0.049891 -0.274021  0.486418 -0.266845     0.337390
6   0.558440 -0.067035   -0.053560  0.276178  0.255529 -0.209664     0.052691
7  -0.006480  0.013137    0.016364 -0.181359  0.415418 -0.441428    -0.211570
8  -0.005377  0.035414   -0.100531  0.462984  0.387923  0.247312    -0.022906
9   0.033151  0.010255    0.033689 -0.580218 -0.035376 -0.232433    -0.090899
10  0.419273 -0.075375    0.029428 -0.339195 -0.328620  0.275760    -0.059050
11  0.701504 -0.090670   -0.021044  0.011857  0.003121 -0.002930     0.004371
12  0.095736  0.699647    0.011926 -0.016065  0.004331  0.012691    -0.010129

    Doc_visits  Full_meals_eaten  VitD_supp  Initial_days  TotalCharge  \
0    -0.066453         -0.044735  -0.057239      0.682291    -0.009112
1    -0.023043          0.178474   0.234882      0.241689     0.011718
2    -0.037953         -0.074836  -0.010975      0.679372    -0.013433
3    -0.304802         -0.237115   0.194552     -0.011347    -0.008912
4    -0.004968          0.008559  -0.029483      0.001237    -0.706605
5     0.385685          0.439452  -0.395829      0.045769    -0.007260
6     0.062985         -0.401897  -0.091646      0.007666    -0.023657
7    -0.669752          0.161316   0.273939     -0.069051    -0.005579
8     0.278037          0.332798   0.606504      0.077879    -0.009288
9     0.460737         -0.320518   0.530601     -0.029938    -0.004907
10   -0.098385          0.556697   0.111994     -0.010488    -0.006019
11   -0.011153          0.013711  -0.009524     -0.002144     0.021578
12   -0.002869         -0.000585  -0.023949      0.019780     0.706381

    Additional_charges
0             0.002484
1             0.000331
2            -0.000575
3             0.001346
4             0.017543
5             0.000325
6            -0.563890
7            -0.000758
8             0.005205
9            -0.000400
10           -0.427561
11            0.705816
12           -0.026444
```

Figure 8 Eigenvalues

```
[1.91728378 1.6182414  1.21337393 1.03220446 1.02754236 1.02254536
 0.99540785 0.98585912 0.98299461 0.97050486 0.77280339 0.37825169
 0.08428734]
```

Figure 9 SCREE plot



E3: PCA Benefits

As previously stated, PCA allows for reducing the dimensionality of a dataset while retaining most of the variation present in the data. This reduction simplifies the subsequent analysis of the data. The higher the dimensionality of the original dataset, the greater the simplification. Visualization are also made simpler as there are fewer dimensions that need to be considered. Relationships that might otherwise be difficult to see are made more clear. (Centellegher, 2020)

Part IV: Supporting Documents

F: Panopto Video

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=3075ea49-0ea4-48c5-9191-b1180024264f

G: Third-Party Code References

No third-party code sources were used.

H: References

Centellegher, S. (2020, January 27). *How to compute PCA loadings and the loading matrix with scikit-learn*. Retrieved February 14, 2024, from scentellegher.github.io: https://scentellegher.github.io/machine-learning/2020/01/27/pca-loadings-sklearn.html

Larose, C., & Larose, D. (2019). *Data ScienceL Using Pythong and R.* John Wiley & Sons, Inc.

McKinney, W. (2022). *Python for Data Analysis, 3rd Edition.* O'Reilly Media, Inc.

Nelli, F. (2023). *Python Data Analytics: With Pandas, NumPy, and Matplotlib.* Apress.

OpenAI. (2024, February 14). What range of latitude and longitude do the entire 50 states in the United States of America cover? [Response by ChatGPT]. Retrieved February 14, 2024, from https://www.chatgpt.com/

scipy. (n.d.). *scipy.stats.zscore*. Retrieved February 12, 2024, from docs.scipy.org: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.zscore.html

seaborn Tutorial. (n.d.). Retrieved February 13, 2024, from seaborn.pydata.org: https://seaborn.pydata.org/tutorial.html

Sonoda, R. (2024, February 8). *3 Key Encoding Techniques for Machine Learning: A Beginner-Friendly Guide with Pros, Cons, and Python Code Examples*. Retrieved February 8, 2024, from towardsdatascience.com: https://towardsdatascience.com/3-key-encoding-techniques-for-machine-learning-a-beginner-friendly-guide-aff8a01a7b6a

Steiger, J. H. (2015, February 16). *Principal Components Analysis*. Retrieved February 14, 2024, from https://www.statpower.net/Content/312/R%20Stuff/PCA.html: https://www.statpower.net/Content/312/R%20Stuff/PCA.html

TIOBE. (2024, February). *TIOBE Index*. Retrieved from TIOBE: https://www.tiobe.com/tiobe-index/

van Rossum, G., Warsaw, B., & Coghlan, A. (2001, July 5). *PEP 8 – Style Guide for Python Code*. Retrieved February 12, 2024, from python.org: https://peps.python.org/pep-0008/#imports

What is Principal Component Analysis? (n.d.). Retrieved February 14, 2024, from bigabid.com: https://www.bigabid.com/what-is-pca-and-how-can-i-use-it/