Adetutu B.

Data Cleaning Task

Part I

A: Question or Decision

What factors affect the total daily charge of a patient’s stay?

B: Required Variables

CaseOrder: quantitative – row 2 – example: 2

- This is a dummy variable column used to help keep the original raw dataset organized in its initial order.

Customer_id: qualitative – row 2 – example: Z919181

- This is a unique ID used to identify each individual patient.

Interaction: qualitative – row 2 – example: d2450b70-0337-4406-bdbb-bc1037f1734c

UID: qualitative – row 2 – example: 176354c5eef714957d486009feabf195

- These are encoded unique IDs used to identify a patient’s transaction, procedure, and admission information from their records.

City: qualitative – row 2 – example: Marianna

State: qualitative – row 2 – example: FL

County: qualitative – row 2 – example: Jackson

Zip: qualitative – row 2 – example: 32446

Lat: quantitative – row 2 – example: 30.84513

Lng: quantitative – row 2 – example: -85.22907

- These columns include the city, state, county, zip code, and latitude and longitude coordinate data from the patient’s billing information to identify where they reside.

Population: quantitative – row 2 – example: 11303

Area: qualitative – row 2 – example: Urban

- These columns provide census data of the population within 1 mile of the patient’s residence and identifies if the area is a rural, urban, or suburban one.

Timezone: qualitative – row 2 – example: America/Chicago

- This provides the patient’s registered time zone.

Job: qualitative – row 2 – example: Community development worker

- This provides the patient's, or the primary policyholder's, listed job in the patient's admissions information.

Children: quantitative – row 2 – example: 3

- This provides information of the number of children listed as living in the same household patient's admissions information.

Age: quantitative – row 2 – example: 51

- This provides the patient's reported age from their admissions information.

Education: qualitative – row 2 – example: Some College, 1 or More Years, No Degree

- This provides the patient's highest level of education reported in their admissions information.

Employment: qualitative – row 2 – example: Full Time

- This provides the patient's reported employment status from their admissions information.

Income: quantitative – row 2 – example: 46805.99

- This provides the patient's, or the primary policyholder's, reported annual salary in the patient's admissions information.

Marital: qualitative – row 2 – example: Married

- This is the reported marital status of the patient, or the primary policyholder, from the patient's admissions information.

Gender: qualitative – row 2 – example: Female

- This provides information on the patient's self-identified gender. Their options are limited to male, female, or non-binary.

ReAdmis: qualitative – row 2 – example: No

- A binary (yes or no) variable for whether the patient was readmitted within 1 month of release.

VitD_levels: quantitative – row 2 – example: 18.99463952

- This provides information on the patient's vitamin D levels in nanograms per milliliter.

Doc_visits: quantitative – row 2 – example: 4

- This provides information on the number of times the patient was visited by their primary physician while hospitalized.

Full_meals_eaten: quantitative – row 2 – example: 2

- This provides information on how many full meals the patient consumed during their time in the hospital.

VitD_supp: quantitative – row 2 – example: 1

- Information on how many times vitamin D supplements were given to the patient.

Soft_drink: qualitative – row 2 – example: No

- A binary (yes or no) variable for whether the patient habitually drank at least 3 or more sodas on a daily basis.

Initial_admin: qualitative – row 2 – example: Emergency Admission

- Identifies whether the patient was admitted as an emergency admission, elective admission, or observation.

HighBlood: qualitative – row 2 – example: Yes

- A binary (yes or no) variable for whether the patient has high blood pressure or not.

Stroke: qualitative – row 2 – example: No

- A binary (yes or no) variable for whether the patient had a stroke or not.

Complication_risk: qualitative – row 2 – example: High

- Risk of complication for a patient reported at either a high, medium, or low level.

Overweight: qualitative – row 2 – example: 1

- A binary (yes or no) variable for whether the patient is considered overweight for their gender, age, and height.

Arthritis: qualitative – row 2 – example: No

- A binary (yes or no) variable indicating if the patient has arthritis or not.

Diabetes: qualitative – row 2 – example: No

- A binary (yes or no) variable indicating if the patient has diabetes.

Hyperlipidemia: qualitative – row 2 – example: No

- A binary (yes or no) variable indicating if the patient has hyperlipidemia or not.

BackPain: qualitative – row 2 – example: No

- A binary (yes or no) variable indicating if the patient has chronic back pain.

Anxiety: qualitative – row 2 – example: NA

- A binary (yes or no) variable indicating if the patient has an anxiety disorder.

Allergic_rhinitis: qualitative – row 2 – example: No

- A binary (yes or no) variable indicating if the patient has allergic rhinitis.

Reflux_esophagitis: qualitative – row 2 – example: Yes

- A binary (yes or no) variable indicating if the patient has reflux esophagitis or not.

Asthma: qualitative – row 2 – example: No

- A binary (yes or no) variable indicating whether the patient does or does not have asthma.

Services: qualitative – row 2 – example: Intravenous

- This denotes the main service received by the patient when they were hospitalized. The service options for this column are limited to blood work, intravenous, CT scan, and MRI.

Initial_days: quantitative – row 2 – example: 15.12956221

- Information about the number of days the patient spent in the hospital during their initial visit.

TotalCharge: quantitative – row 2 – example: 4214.905346

- The daily average cost a patient is charged during their hospital stay for non-specialized treatments (in USD).

Additional_charges: quantitative – row 2 – example: 17612.99812

- The daily average cost a patient is charged during their hospital stay for miscellaneous treatments, procedures, medicines, and more.

Item1 Timely admission: qualitative – row 2 – example: 3
Item2 Timely treatment: qualitative – row 2 – example: 4
Item3 Timely visits: qualitative – row 2 – example: 3
Item4 Reliability: qualitative – row 2 – example: 4
Item5 Options: qualitative – row 2 – example: 4
Item6 Hours of treatment: qualitative – row 2 – example: 4
Item7 Courteous staff: qualitative – row 2 – example: 3
Item8 Evidence of active listening from doctor: qualitative – row 2 – example: 3

- These columns represent survey responses given by each patient rating how important various factors of their patient experience were on a scale of most important (1) to least important (8).

Part II

C1: Quality Assessment Plan

I will be utilizing python for this project. My first step would be to import the NumPy and Pandas packages and load the data from the medical data CSV file into the kernel. I will run the code ‘info( )’ to get a better view of my dataset and use the information to check for anomalies. I would then check for duplicate columns using code like ‘df.duplicated( )’, and also check for values that may require uniqueness (e.g. UID) or categorical classification using code like ‘value_counts( )’. I will check for existing null values in all my columns by using the ‘isnull( ).sum( )’ function and to detect possible outliers I can import the matplotlib, seaborn, and SciPy packages to determine which values are outliers. Finally, I will check for consistency in my categorical variables to determine if re-expression is necessary.

C2: Justify Approach

I began by loading and visually inspecting my database in order to make it easier to read, clean, and manipulate my data. Before checking for anomalies, I returned the code 'info( )' in order to better understand my dataset and its qualities. By checking for anomalies in the relevant places in the dataset I can optimize my process and ensure that the information is as accurate as possible to minimize a biased analysis later down the line. The approach in my assessment plan enables me to detect things like duplicates, missing values, etc in a more straightforward way, so I only have to reorganize and manipulate what is necessary.

C3: Justify Tools

I would consider this to be a fairly large dataset, so I chose Python as my programming language to complete this task. As previously stated, I wanted to fix this data in a more straightforward way, and python is known for its optimization. With it I was able to treat and detect anomalies quickly without using an excess amount of code. I downloaded NumPy and SciPy to enable me to perform many mathematical operations and pandas so I could easily import the CSV file data. By importing matplotlib and seaborn I was able to visualize and graph my data, which aided in detecting any possible outliers/abnormalities within the dataset.

C4: Code File

I have provided an Executable File (Attached Separately) of the data that will be used to check for anomalies and data quality. The code for this section is under the 'C4 Quality Check' header. It is named: **project_code.ipynb**

Part III

D1: Code Results

Upon visual inspection of the data frame and doing an inspection using the code from C4, I found multiple issues that need to be fixed before analyzing the data. Multiple columns are not in correct python casing and follow inconsistent naming rules. This will need to be addressed. There are many columns that require re-expression/categorization to better represent the value entries. There was a column that produced results based on the data frame that were not fully aligned with the data dictionary. There were also columns that required standardization to either accurately present the column description/title or the proper notation. Some columns had missing data, and a few had skewed data. I plan to address all the null values and any skew that results from null values. However, I will not address every skewed column because in some cases the intensity of the skew was not the result of an outlier or data error, but with the way the x-axis was scaled in the histogram. Real data points will not be automatically assumed as anomalies unless they are relevant errors.

D2: Justify Mitigation Methods

After identifying which anomalies/quality issues are present, I have gone into detail about what these issues are and where these issues are located. I have also explained how and why I chose to solve them the way I did.

Any misleading/inconsistent/non-python cased column will be renamed/fixed.

Column(s) that *are stored as string objects but* ***can be re-expressed as category*** *because there are only a few different value options that can be made into categories:* Area, Education, Employment, Marital, Gender, Initial_admin, Complication_risk, Services

- Additionally, the code output in the Gender column expressed one of the results from the data frame as "Prefer not to answer", but the data dictionary states that the self-identification options should be "male", "female", and "non-binary". The "Prefer not to answer" option should be re-expressed as "non-binary" to properly reflect the data dictionary.

Column(s) that *are stored as integers but* ***can be re-expressed as ordinal categorical*** *because they have value options that can be made into rank-able categories:* Item1, Item2, Item3, Item4, Item5, Item6, Item7, Item8

Column(s) that *are stored as integers but* ***should be stored as strings****:* Zip

- Note: Zip codes are not technically numbers. In order to restore the leading 0's of the zip codes, they will need to be properly recast.

Column(s) that *are stored as float point numbers but* ***can be stored as integers*** *because their values should all be whole numbers:* Children, Age, and Initial_days

- Additionally, all Initial_days values seem to be greater than or equal to 1, so NaNs can be replaced by the value '0'.

Column(s) that *are stored as string objects but* ***are better stored as booleans*** *instead due to their nature as binary (yes/no - true/false) variables:* ReAdmis, Soft_drink, HighBlood, Stroke, Arthritis, Diabetes, Hyperlipidemia, BackPain, Allergic_rhinitis, Reflux_esophagitis, Asthma

Column(s) that *are stored as float point numbers but* ***are better stored as booleans*** *instead due to their nature as binary (yes/no - true/false) variables:* Overweight, Anxiety

Column(s) that *need standardization/adjustment:* Timezone, TotalCharge, Additional_charges

- Timezone is currently listed by cities but can be standardized to reflect the recognized US time zones and stored as category instead (Time Zones in the United States, 2024).
- TotalCharge and Additional_charge can be rounded down to 2 decimal places. Since this is US data, the money should realistically only go out to 2 decimal places to properly reflect USD.

Column(s) that currently have under 10,000 values (**have present null values**) that could be replaced for a more accurate analysis down the line: Children, Age, Income, Overweight, Anxiety, Initial_days, Soft_drink

- Note 1: The replacement of null values in the **Initial_days** column was addressed above
- Note 2: The null values in the **Children, Overweight,** and **Anxiety** columns will be replaced with the value ‘0’. This was under the assumption that a patient is more likely to leave that space unanswered or respond with N/A if the situation was not applicable to them than if it was. However, I am aware of the possibility of other reasons the space contains an N/A value.
- Note 3: The null values in the **Age** column and the **Income** column will be replaced with the relevant median for its column (since the distribution from the histogram was not a normal distribution). I noticed that all the present age entries were values of 18 and above, so if I wanted to replace the values in a different way, I could possibly replace all nulls with the lowest age entry, however I believe that would unjustly skew the data. Similar to my reasoning, the income values already show skew and I worry that replacing the nulls with zero would introduce more skew, so I thought it was better to take the average of the available values.

- The NA values in the **Soft_drink** column will be replaced with ‘No’, under the assumption that those who filled this column with NA either did not, or were not, allowed to drink soda during their stay and that is why it was not applicable to them.

D3: Outcome Summary
In addressing the data quality issues, I can alleviate the analysis process that occurs later on. I reformatted columns to a proper title to work well within the python environment and present a more accurate depiction of the column description. By recasting certain data types to booleans, categorical, strings, integers, etc I can place proper constraints on what entries are allowed in each column, which will help reduce the number of entries incorrectly input later on. In standardizing and adjusting certain things the data is easier to sort through and is more accurate/applicable to the description found in the data dictionary. I replaced null values and addressed outliers with the intent to give as accurate a database as possible and fill the presence of missing values. As I mentioned before, my goal is to make the data more straightforward to work with. My mitigation approach was done in the hope of standardizing the data and making it as accurate, clear, and consistent as possible. I hope my steps will help ensure the integrity and honesty of the data analysis process.

D4: Mitigation Code
The code used to mitigate the issues with the quality of the data is in the same file presented in part C4. In the executable file, the mitigation code begins below the ‘D4 Code Mitigation’ header. The file name for this code is still: **project_code.ipynb**

D5: Clean Data
I have provided a CSV File (Attached Separately) of the cleaned data that is named:
clean_data.csv

D6: Limitations
In this section I will list a few of the limitations I faced/was worried about while cleaning this data.

- Changing/altering the ‘prefer not to say’ response in the **gender** column.
- Determining whether **initial_days** should be a float or integer
- Some decimal rounding/missing value replacement had to be assumed based on my current limited knowledge which may have introduced bias or altered the accuracy of the data.

D7: Limitations Impacts

In the **gender** column, there was an issue with one of the values as it did not match the data dictionary. While the dataset had values that were listed as ‘prefer not to say’, ‘female’, and ‘male’ the data dictionary stated that the only options should be ‘non-binary’, ‘female’, and ‘male’. So, I made the decision to stick with the rules of the data dictionary and changed the ‘prefer not to say’ dataset responses to ‘non-binary’ to be consistent. However, I saw this as a limitation because these responses are not technically mutually exclusive. Someone who preferred not to state their gender is not necessarily non-binary. There could be many different reasons someone does not want to disclose their gender identity and giving them a label they did not explicitly state themselves is not only inaccurate but also ethically questionable, in my opinion. While I am unsure of the significance the variables in this column have on my research question, I still think it was worth mentioning as a plausible limitation.

While the data dictionary did not explicitly state that the **initial_days** column had to either be a float or integer, the original/uncleaned dataset had initial_days filled with float values. Truthfully it seemed unclear to me in this state because while you can mathematically quantify 2.8567 days, most people would either say they stayed somewhere for 2 or 3 days (maybe even 2 and a half) but they would rarely identify days by the decimal point. Since my knowledge is limited on whether it would have been best to represent the number of days the patient was there as a whole number or a mix of the days plus however many extra hours they were present at the hospital, I went ahead and changed the column to hold integers so only whole days should be placed/analyzed in the column. I saw this as a possible limitation as my knowledge is limited and changing the data type may have been the incorrect way to address this issue. Especially regarding my research question, the number of initial days a patient is admitted could have a significant impact on a patient’s daily charge, so changing the data type of this column could influence the results of my research question.

Similar to my decision with initial_days, I had to consider the decimal places in some of my other columns. I changed the columns with currency values to properly represent the 2 decimal places present in USD. However, I did not change the decimal places in the vitamin_d_levels columns as I did not have the proper knowledge to know how specific the column needed to be. I tried to do some research on how vitamin_d_levels results are usually presented in healthcare; however, the levels are usually given as integer ranges with no decimal place (U.S. Department of Health and Human Services, n.d.). I still did not change the decimal places or data type of this column because I still do not know much about how the actual vitamin levels are calculated to get the range and thought it was best to have information that was as accurate as possible. So, my lack of knowledge on this calculation/process is a limitation.

E1: Principal Components

In order to perform a Principal Component Analysis, I chose the columns from my dataset that only contain numeric values. Values from the latitude, longitude, population, children, age, income, vitamin d levels, amount of doctor visits, full meals, vit d supply, initial stay, daily charge, and additional charge columns will be used. After completing the PCA loading matrix, I had a total of 6 principal components.

The code and output of the loading matrix can be found in my executable script file named: **project_code.ipynb** under the section titled 'E1 PCA'.

E2: Criteria

To figure out I have 6 principal components, I used a scree plot to determine the eigenvalues of each of the PCs. Components with eigenvalues above one were identified for PCA and everything else was disregarded which enabled me to confirm my total count.

E3: Benefits

By using PCA, I was able to decrease the number of dimensions within the dataset and focus more on pertinent information. By focusing on important variables and finding which ones had a relationship I can test fewer elements of the data later in the analysis process (Chawla, 2023). To be more specific, the quantitative values used (such as daily charge, additional charge, amount of doctor visits, age, etc) can paint a picture of the patient's state of health; or can help explain what factors/variables are related to the cost of their stay/visits. The patterns kept within the variables that get combined using PCA can then be used to conserve the time it takes to train data models later.

Part IV

F: Web Code Sources

Indiana Time Zone: Visit Indiana: In Indiana: IDDC. Indiana Destination Development Corporation. (n.d.). https://www.visitindiana.com/about-indiana/time-zones/

G: In-Text Citation References

Chawla, A. (2023, May 10). *The advantages and disadvantages of PCA to consider before using it*. The Advantages and Disadvantages of PCA To Consider Before Using It. https://blog.dailydoseofds.com/p/the-advantages-and-disadvantages

Time and Date AS. (2024, July 6). *Time Zones in the United States*. Time and Date. https://www.timeanddate.com/time/zone/usa#:~:text=There%20are%209%20time%20zones,has%204%20standard%20time%20zones

U.S. Department of Health and Human Services. (n.d.). *Vitamin D Fact Sheet for Consumers*. NIH Office of Dietary Supplements. https://ods.od.nih.gov/factsheets/VitaminD-Consumer/#:~:text=Levels%20of%2050%20nmol%2FL,and%20might%20cause%20health%20problems