Data Cleaning — D206

PRFA — NUM2

Table of Contents

[Part I: Research Question 1](#_Toc121292008)

[Part II: Data-Cleaning Plan 7](#_Toc121292009)

[Part III: Data Cleaning 13](#_Toc121292010)

[Part IV: Supporting Documents 34](#_Toc121292011)

# Part I: Research Question

1. Describe ***one*** question or decision that you will address using the data set you chose. The summarized question or decision must be relevant to a realistic organizational need or situation.

If a patient is readmitted for observation or an emergency and they have low Vitamin D levels while also taking Vitamin D supplements, do they have any other conditions such as arthritis, Diabetes, hyperlipidemia, back pain, anxiety, reflux, esophagitis or asthma?

1. Describe the variables in the data set and indicate the specific type of data being described. Use examples from the data set that support your claims.

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable Name** | **Data Type** | **Description** | **Example** |
| CaseOrder | Ordinal Categorical | A variable used to maintain the order of the patient records. | 20 |
| Customer\_id | Qualitative | A variable that is a unique ID for identifying a particular patient. | C412403 |
| Interaction | Qualitative | A variable that is a unique identifier (GUID) that marks an interaction with a patient/customer and the hospital. | 8cd49b13-f45a-4b47-a2bd-173ffa932c2f |
| UID | Qualitative | A variable that is a unique identifier (GUID) that marks a procedure or admission with a patient/customer and the hospital. | 176354c5eef714957d486009feabf195 |
| City | Nominal Categorical | A variable that represents which city the patient lives in. | Sioux Falls |
| State | Nominal Categorical | A variable that represents which state the patient lives in | FL |
| County | Nominal Categorical | A variable that represents which county the patient lives in | Jackson |
| Zip | Nominal Categorical | A variable that represents which zip code the patient lives in. | 57110 |
| Lat | Continuous Numerical | A variable that represents the latitudinal value of the patient’s location. | 34.3496 |
| Lng | Continuous Numerical | A variable that represents the longitudinal value of the patient’s location. | -86.72508 |
| Population | Discrete Numerical | A variable that represents the population based on a patient’s zip code and includes a one-mile radius. | 11303 |
| Area | Nominal Categorical | A variable that represents which type of area a patient lives in. This can be rural, urban, or suburban. | Suburban |
| Timezone | Nominal Categorical | A variable that represents the time zone the patient lives within. | America/Chicago |
| Job | Nominal Categorical | A variable that holds the job title of a patient or of the primary insurance policy holder. | Computer games developer |
| Children | Discrete Numerical | A variable that represents the number of children a patient has. | 10 / NA |
| Age | Discrete Numerical (Whole numbers are used)  NOTE: Would be continuous if decimal of exact age used. | A variable that represents the age of the patient30 | 78 |
| Education | Ordinal Categorical | A variable that contains the patients highest earned education level. | Regular High School Diploma |
| Employment | Nominal Categorical | A variable that indicates type of employment of the given patient. | Full Time |
| Income | Continuous Numerical | A variable that holds the income of the given patient. | 55586.48 / NA |
| Marital | Nominal Categorical | A variable that contains the marital status of a given patient. | Never Married |
| Gender | Nominal Categorical | A variable that contains the gender of a given patient. | Male/ Female/ Prefer not to answer |
| ReAdmins | Nominal Categorical | A variable that represents if the patient has had a re-admission into the hospital. | Yes/No |
| VitD\_levels | Continuous Numerical | A variable that holds the amount of a patient’s vitamin D level at nanograms per milliliter. | 18.99463952 |
| Doc\_visits | Discrete Numerical | A variable that holds the number of doctor visits a patient has had. | 6 |
| Full\_meals\_eaten | Discrete Numerical | A variable that holds the number of full meals eaten upon admission by a patient. | 3 |
| VitD\_supp | Discrete Numerical | A variable that represents how many times vitamin D supplements were given to a patient during treatment. | 2 |
| Soft\_drink | Nominal Categorical | A variable that represents if the patient drinks soft drinks or not. | Yes/No/NA |
| Initial\_admin | Nominal Categorical | A variable that represents the type of admission into the hospital of a patient. | Emergency Admission |
| HighBlood | Nominal Categorical | A variable that indicates if a patient has high blood pressure or not. | Yes/No |
| Stroke | Nominal Categorical | A variable that indicates if a patient has had a previous stroke or not. | Yes/No |
| Complication\_risk | Ordinal Categorical | A variable that represents if a patient falls into a particular complication category. | Medium |
| Overweight | Nominal Categorical | A variable that indicates if a patient is overweight or not.  Will convert to Yes/No after missing data treatment for uniformity. | 0/1/NA  Converting to Yes/NO |
| Arthritis | Nominal Categorical | A variable that indicates if a patient has arthritis or not. | Yes/No |
| Diabetes | Nominal Categorical | A variable that indicates if a patient has Diabetes or not. | Yes/No |
| Hyperlipidemia | Nominal Categorical | A variable that indicates if a patient has Hyperlipidemia or not. | Yes/No |
| BackPain | Nominal Categorical | A variable that represents if a patient has existing back pain or not. | Yes/No |
| Anxiety | Nominal Categorical | A variable that indicates if a patient has anxiety or not  Will convert to Yes/No after missing data treatment for uniformity. | 0/1/NA |
| Allergic\_rhinitis | Nominal Categorical | A variable that indicates if a patient has Allergic rhinitis or not. | Yes/No |
| Reflux\_esophagitis | Nominal Categorical | A variable that indicates if a patient has reflux esophagitis or not. | Yes/No |
| Asthma | Nominal Categorical | A variable that indicates if a patient has Asthma or not. | Yes/No |
| Services | Nominal Categorical | A variable that represents which services category a patient falls into. | Intravenous |
| Initial\_days | Continuous Numerical | A variable which it’s unclear on its meaning and will need investigation. | 15.12956221 |
| TotalCharge | Continuous Numerical | A variable that represents the total charges incurred by the patient. | 2575.607671 |
| Additional\_charges | Continuous Numerical | A variable that represents the additional charges that go beyond the normal total charges incurred by a patient. | 8099.185127 |
| Item1 | Ordinal Categorical | This variable is an item within a survey given to patients.  This item represents ‘Timely admission’.  The allowable scores include: 1 = most important through 8 = least important. | 1 |
| Item2 | Ordinal Categorical | This variable is an item within a survey given to patients.  This item represents ‘Timely treatment’.  The allowable scores include: 1 = most important through 8 = least important. | 2 |
| Item3 | Ordinal Categorical | This variable is an item within a survey given to patients.  This item represents ‘Timely visits’.  The allowable scores include: 1 = most important through 8 = least important. | 3 |
| Item4 | Ordinal Categorical | This variable is an item within a survey given to patients.  This item represents ‘Reliability’.  The allowable scores include: 1 = most important through 8 = least important. | 4 |
| Item5 | Ordinal Categorical | This variable is an item within a survey given to patients.  This item represents ‘Options’.  The allowable scores include: 1 = most important through 8 = least important. | 5 |
| Item6 | Ordinal Categorical | This variable is an item within a survey given to patients.  This item represents ‘Hours of treatment’.  The allowable scores include: 1 = most important through 8 = least important. | 6 |
| Item7 | Ordinal Categorical | This variable is an item within a survey given to patients.  This item represents ‘Courteous staff’.  The allowable scores include: 1 = most important through 8 = least important. | 4 |
| Item8 | Ordinal Categorical | This variable is an item within a survey given to patients.  This item represents ‘Evidence of active listening from doctor’.  The allowable scores include: 1 = most important through 8 = least important. | 2 |

# Part II: **Data-Cleaning Plan**

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

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

1.  Propose a plan that includes the relevant techniques and specific steps needed to identify anomalies in the data set.

**Detecting Anomalies Plan:**

Below is a plan for three types of detections which include duplicates, missing values, and outliers. The detection plans will be used on the file ‘medical\_raw\_data.csv’ which will be loaded into a Pandas Data Frame using Python.

|  |  |
| --- | --- |
| Duplicates | For the detection of duplicates, we will use the functionality provided by Pandas Python Library ‘.isduplicated()’[[1]](#footnote-1). If the results are greater than 0, we know that we have duplication as defined below.   1. The first style of duplication that we will be detecting is full observation (row) detection. The entire Data Frame will be used for the duplicated function to run on which will return if entire rows have been duplicated. 2. Unique identification duplicate detection:    1. Grouped as ‘Customer\_id’, ‘Interaction’, and ‘UID’       1. A given patient should have an Observation (Row) per interaction with the hospital. To detect this, we will once again call the duplicate function on the Data Frame for the subset columns ‘Customer\_id’, ‘Interaction’ and ‘UID’. We don’t want any duplicate interactions being logged.    2. CaseOrder       1. If we need to give cases in the correct order, we are making sure that no case is listed with a duplicate order number. We will again be executing duplicate function on the Data Frame for the subset column ‘CaseOrder’ |
| Missing Values | For missing values, we will be using the Pandas Python Library functionality. The overall documentation can be found [here](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.isna.html).  A custom Python function will be created to loop through each column of the ‘medical\_data’ Data Frame and check it for missing data via the isna()*[[2]](#footnote-2)* function and the sum()[[3]](#footnote-3) function. The function will return a Data Frame with two columns Column and NA\_Count. If missing data is found the column name will be logged with the number of missing values in the NA\_Count column.  Using the Data Frame returned from the custom Python function we will visualize the missing data in a Histogram and verify that there are indeed gaps in the bar value. The library that will help with the visualization is the Missingno[[4]](#footnote-4) package. |
| Outliers | For the detection of outliers, we’ll be collecting the columns with data types of int64 and float64 (numerical) and using the library Seaborn to generate Boxplots. For each boxplot we will visually check if there are outliers.  **Note:** There were a few columns that needed to be converted from 0/1 to Yes/No for data uniformity reasons. These will be excluded. |
| Re-expression | There were two variables (columns) that were using 0’s and 1’s to represent Yes/No in a categorical way. The rest of the variables of a Boolean nature are expressed as Yes/No so these misrepresented values will be converted into Yes/No and re-expression takes place after the missing data process was completed.  Variables that were re-expressed from 0/1’s to Yes/No’s are ‘Overweight’ and ‘Anxiety’ |

2.  Justify your approach for assessing the quality of the data, include:

•  characteristics of the data being assessed,

•  the approach used to assess the quality data

|  |  |
| --- | --- |
| Duplicates | After reviewing the medical data set the variables (columns) chosen to define duplicates was done so as these were the ways to determine entire observation (row) duplicates and if customers were duplicated. |
| Missing Values | The best approach for finding missing values within the ‘medical data’ is using the ‘.isna()’ per variable (column). If the results from each column contain any True indicating missing data, the column name is collected into a list.   It's also should be noted that using ‘.isna()’ as the missing data function is great as it can find missing values for numeric, object and datetime objects. |
| Outliers | For the detection of outliers, I chose to use the Seaborn package as it gave the ability to chart the numeric data as Box Plot. Box plots created a very easy to understand chart with left and right whiskers and any dots that appear beyond the left and right whisker indicates an outlier.  This approach is nice as it can be shown to nontechnical people and the intent is very clear. |
| Re-Expression | Data uniformity is very important to keep things consistent and easier to understand. So, for the variables (columns) whom values were 0/1 to clearly mean a Yes or No I re-expressed them from numerical to a categorical datatype with the values Yes or No AFTER the missing values were imputed.  Objective with this approach was clarity. |

1. Justify your selected programming language and any libraries and packages that will support the data-cleaning process.  
     
   I will be using Python and the packages Pandas, NumPy, Missingno, Matplotlib and Seaborn to perform the data-cleaning process.   
     
   One major reason I started with Python for this project is that I am a developer by trade and the transition to Python was less steep than a transition to R where more advanced functionalities are more difficult to master.  
     
   Another major reason I selected Python was I really enjoyed the power and ease of manipulating data via Pandas Data Frames/Series. I felt it was also better to start with Pandas as the Data Frame library of choice as their documentation has a wonderful comparison guide to make learning R later a bit easier. See Pandas comparison with R guide[[5]](#footnote-5).
2. Provide the code you will use to identify the anomalies in the data.  
     
   **NOTE:** the code provided in the code column assumes that the data has already been loaded into *‘medical\_data’* Data Frame.

|  |  |
| --- | --- |
| **Anomaly** | **Code** |
| Duplicates | If the assertions in the code below throw and exception, we know we have duplicates for the defined meaningful columns for duplication.  all\_columns\_duplicates = medical\_data.duplicated()  medical\_duplicate\_rows = medical\_data[all\_columns\_duplicates]  #Assert that we have 0 full row duplicates  assert len(medical\_duplicate\_rows.values) == 0  duplicate\_customer\_ids\_groups = medical\_data.duplicated(subset = ['Customer\_id', 'Interaction', 'UID'])  customer\_id\_duplicate\_rows = medical\_data[duplicate\_customer\_ids\_groups]  #Asset that we do not have any of the 3 customer id types as duplicated  assert len(customer\_id\_duplicate\_rows) == 0  duplicate\_caseorder = medical\_data.duplicated(subset = ['CaseOrder'])  caseorder\_duplicates = medical\_data[duplicate\_caseorder]  #Assert that we do not have any duplicated Case Orders  assert len(caseorder\_duplicates) == 0  See attached code in file: *‘D206-PA-AndréDavis .ipynb’* |
| Missing Data | For the code below the repeatable Missing Data functions/logic were wrapped in Python functions for reusability and to be run on each column of the ‘medical\_data’ Data Frame. Any missing data detected will be added to the Data Frame that will be rendered to the screen as a histogram for visual verification.  #Creating helper functions for detection of missing data  def checkForMissingValues(series, columnName):  naCount = 0  try:  naInfo = series[columnName].isna()  naCount = naInfo.sum()  #print(f'{columnName} total missing count: {naCount}\n')  assert naCount == 0  except AssertionError:  print(f'Missing data detected for {columnName} column of Data Frame')  except:  print(f'Unknown column found {columnName}')  return naCount  def checkForMissingValuesByColumns(series, columns=[]):  naCounts = []  for column in columns:  naCount = checkForMissingValues(series, column)  naCounts.append(naCount)  missingValuesResult = {  'Column': columns,  'NA\_Count': naCounts  }  naDataFrame = pd.DataFrame(missingValuesResult)  return naDataFrame  print('\nChecking for missing data in each column. \n')  naReport = checkForMissingValuesByColumns(medical\_data, medical\_data.columns)  missing\_data = naReport[naReport['NA\_Count'] > 0]  missing\_data  missing\_data\_columns = missing\_data['Column']  msno.matrix(medical\_data[missing\_data\_columns], fontsize = 12, labels = True)  plt.title('Columns with Missing Data')  plt.show()  See attached code in file: *‘D206-PA-AndréDavis .ipynb’* |
| Outliers | Seaborn library is being used to generate Boxplots for Data Frame columns that are quantitative, excluding numerical columns that need converting into categorical (qualitative) datatypes.  #Get all numerical columns  numerical\_columns = medical\_data.select\_dtypes(include = ['int64', 'float64']).columns  numerical\_quantitative = numerical\_columns.get\_indexer\_for(['CaseOrder', 'Overweight', 'Anxiety'])  numerical\_columns = numerical\_columns.delete(numerical\_quantitative)  for column in numerical\_columns:  sb.boxplot(x = medical\_data[column])  plt.title(f"Boxplot of {column}")  plt.show()  See attached code in file: *‘D206-PA-AndréDavis .ipynb’* |
| Re-Expression | Detecting if columns need to be re-expressed from 0/1 to Yes/No the code below was used.  noyes\_as\_bits = [0, 1]  columns\_to\_reexpress = []  numerical\_columns = medical\_data.select\_dtypes(include=[np.int64, np.float64]).columns  for column in numerical\_columns:  if imputated\_medical\_data[column].isin(noyes\_as\_bits).all():  columns\_to\_reexpress.append(column)  print(f"Re-Expression needed for {columns\_to\_reexpress}")  See attached code in file: *‘D206-PA-AndréDavis .ipynb’* |

# Part III: Data Cleaning

D.  Summarize the data-cleaning process by doing the following:

1.  Describe the findings, including all anomalies, from the implementation of the data-cleaning plan from part C.

|  |  |  |
| --- | --- | --- |
| Duplicates | There were 3 types of duplication being searched for:   1. Entire observation (row) duplicates:  No duplicates were detected 2. Customer Unique Identification via ‘Customer\_id’, ‘Interaction’, and ‘UID’:  No duplicates were detected. 3. CaseOrder as this column was used for ordering the data and duplicate numbers don’t make sense:  No duplicates were detected. | |
| Missing Data | Scanning all the columns of the ‘medical\_data’ Data Frame it was detected that seven columns contained missing data, and this was confirmed visually. Below is a table that has the columns and total count of missing data per column.     |  |  | | --- | --- | | **Column** | **NA\_Count** | | Children | 2588 | | Age | 2414 | | Income | 2464 | | Soft\_drink | 2467 | | Overweight | 982 | | Anxiety | 984 | | Initial\_days | 1056 |   Visualizing Missing Data Findings:    Figure - Visualizing Missing Data by displaying gaps  Displaying Distribution of Numerical Missing Data:    Figure - Children variable with a Right-Skew    Figure - Age variable with a U-Shaped Distribution    Figure - Income variable with a Right-Skew    Figure - Initial\_days variable with a U-shaped distribution | |
| Outliers | The below table list the name of the column and findings about any outliers detected based on the Boxplot that was generated by Seaborn.  NOTE: Columns with no outliers will not be listed in the table.   |  |  | | --- | --- | | **Column** | **Findings** | | Lat | Lat has outliers both below the lower limit and above the max. The data falls within the range of values for a latitudinal line and are valid.  **Values Samples:**  Total: 150 Lowest: 17.96719  Largest: 70.56099 | | Lng | Lng has outliers below the lower limit. The data falls within the range of values for a longitudinal line and are valid.  **Values Samples:**  Total: 237 Lowest: -174.20969  Largest: -122.72547 | | Population | Population has a few extreme outliers.   Example: TX, Katy enters have 122000+ as their population. After a quick internet search for population data for TX, Katy it can be noted that even without knowing the year this data was entered from that the population has never been anywhere that high.  **Values Samples:**  Total: 855 Lowest: 33894  Largest: 122814  Population data from 1970 – 2022 used to check outlier value.[[6]](#footnote-6) | | Children | Children has outliers to over the right (upper limit) whisker. Some families can be big so unable to determine if this is erroneous.  **Values Samples:**  Total: 303 Lowest: 8.0  Largest: 10.0 | | Income | Income has outliers outside of the right (upper limit) whisker once incomes start getting outside of the $100k range.  **Values Samples:**  Total: 252 Lowest: 106220.51  Largest: 207249.13 | | VitD\_levels | Vitamin D Levels has outliers on both the left (lower limit) and right (upper limit) whisker.  Vitamin D levels can be converted to a range and become categorical. Reference: [Vitamin D Levels](https://www.medicalnewstoday.com/articles/normal-vitamin-d-levels#normal-levels)   |  |  | | --- | --- | | Level | Blood Test Result | | Low | 30 nmol/l or 12 ng/ml or below | | Adequate | 50 nmol/l or 20 ng/ml or above | | High | 125 nmol/l or 50 ng/ml or above |   **Values Samples:**  Total: 534 Lowest: 9.519012  Largest: 53.019124 | | Full\_meals\_eaten | Full meals eaten has a few outliers to the right (upper limit) whisker of the boxplot.  Some people do overeat or eat for competitive purposes. These do not seem to be erroneous.  **Values Samples:**  Total: 8 Lowest: 6  Largest: 7 | | VitD\_supp | Vitamin D Supplements has outliers on the right (upper limit) whiskers.  People with extremely low vitamin d levels can be asked to take higher dosages. Unable to determine if these are erroneous entries.  **Values Samples:**  Total: 70 Lowest: 3  Largest: 5 | | TotalCharge | Total charges have outliers to the right (max) whisker of the boxplot.  **Values Samples:**  Total: 466 Lowest: 14159.65973  Largest: 21524.22421 | | Additional\_charges | Additional charges have outliers to the right(max) whisker of the boxplot.  **Values Samples:**  Total: 424 Lowest: 27088.14922  Largest: 30566.07313 | | |
| Re-Expression | | After the missing data treatment phase, we then set to detect if any numerical columns only had 0/1’s as values. This indicated that they could be represented as Yes/No to become uniform with the rest of the Boolean columns. The variables (columns) ‘Overweight’ and ‘Anxiety’ were both found to only include 0/1’s.  DataFrame ‘.isin()[[7]](#footnote-7)’ and ‘.all()[[8]](#footnote-8)’ in combination allowed use to determine if a numerical column was indeed being used as a Boolean Yes/No column. | |

2.  Justify your methods for mitigating each type of discovered anomaly in the data set.

|  |  |
| --- | --- |
| Duplicates | No meaningful duplicates were found. There is no need for treatment in this category. |
| Missing Data | For each of the missing data columns we are going to use univariate imputation so that the missing data is replaced with one of the statistical elements of mean, median or mode.  Imputing was chosen as we do not want to reduce and damage the dataset size and accuracy for all other observations (rows).   The columns that required imputation are Children, Age, Income, Soft\_drink, Overweight, Anxiety, and Initial\_days as they all contain missing data.   * Right Skewed distributions were found in Children and Income. The imputation method for this is replacement with the ‘Median Value’ * Categorical variables (columns) with missing data included Soft\_drink, Overweight, and Anxeity. The imputation method for this is replacement with ‘Mode Value’. * U-Shaped distributions were found in Age and Initial\_days. The imputation method for this is replacement with the ‘Mode Value’ as U-shaped are considered bimodal.   NOTE: After ‘Overweight’ and ‘Anxiety’ were imputed they were converted from 0/1’s to Yes/No’s for data uniformity reasons. |
| Visualization of missing data after treatment/mitigation:    Figure – Visualizing Detected Missing Data after Imputation. Gaps filled.    Figure - Children variable after imputation, maintained Right-Skew    Figure - Age after imputation, maintained U-Shaped distribution    Figure - Income after imputation, maintained Right-Skew    Figure - Initial\_days after imputaiton, maintained U-shaped Distribution | |
| Outliers | After reviewing the Boxplots generated using the Seaborn package it is noted that the outlier values themselves are valid within the context of the variable(column) they represent. Because the variables were value the treatment method in this case is simply ***‘RETAIN’***.  However, there was one column in which we added an additional column to help represent the decimal values as categorical which makes the Outliers themselves understandable. The VitD\_level values seemed to have odd outliers. So, a new column was created that categorized the values of VitD\_levels into ‘VitD\_levels\_level’ with the bins ‘Low’, ‘Adequate’, and ‘High’. This representation of the values hows that there truly only 3 ranges of values in meaning/category.  ‘VitD\_levels’ additional column as categories ‘VitD\_levels\_level’ to help visualize that outliers (High) are a valid value. |
| Re-expression | After the missing data treatment stage, it was safe to detect if any numerical columns only contained 0/1’s. If this was detected the column was then re-expressed using the DataFrame ‘.map()[[9]](#footnote-9)’ function. |

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

|  |  |
| --- | --- |
| Duplicates | There were no duplicates to treat. |
| Missing Data | All missing data columns were imputed with data as to no delete or loss data at an observation (row) level. |
| Outliers | Outliers had meaningful values and were retained as to not damage the meaning of the datasets. |
| Re-Expression | Two columns were found to be in the from datatype and were re-expressed from 0/1s to Yes/No to address uniformity issues. |

4.  Provide the code used to mitigate anomalies.

|  |  |
| --- | --- |
| Duplicates | The definitions used in the detection of duplicates found no duplicates. There was no treatment needed. |
| Missing Values | Imputation was used for missing values. Code:  imputations = {  'Children': int(medical\_data['Children'].median())  ,'Age': int(medical\_data['Age'].mode())  ,'Income': medical\_data['Income'].median()  ,'Soft\_drink': medical\_data['Soft\_drink'].mode()[0]  ,'Overweight': int(medical\_data['Overweight'].mode()[0])  ,'Anxiety': int(medical\_data['Anxiety'].mode()[0])  ,'Initial\_days': medical\_data['Initial\_days'].mode()[0]  }  imputated\_medical\_data = medical\_data.fillna(imputations)  #Visualization check  msno.matrix(imputated\_medical\_data[missing\_data\_columns], fontsize=12, labels=True)  plt.title('Missing Data columns after imputation.')  plt.show()  for numerical\_column in numerical\_nas:  df = medical\_data[numerical\_column]  df.hist()  plt.title(f"Distribution of '{numerical\_column}' after imputation.")  plt.show()  convert\_to\_int64 = {  'Age': np.int64,  'Children': np.int64  }  imputated\_medical\_data = imputated\_medical\_data.astype(convert\_to\_int64)  See attached code in file: *‘D206-PA-AndréDavis .ipynb’* |
| Outliers | Additional Levels column added since outliers were retained.  # https://www.medicalnewstoday.com/articles/normal-vitamin-d-levels#function  vitamin\_d\_bins = [0, 20, 50, np.inf]  vitamin\_d\_categories = ['Low', 'Adequate', 'High']  vitd\_levels\_group = pd.cut(imputated\_medical\_data['VitD\_levels'], bins=vitamin\_d\_bins, labels=vitamin\_d\_categories)  vitd\_level\_index = imputated\_medical\_data.columns.get\_loc('VitD\_levels')  imputated\_medical\_data.insert((vitd\_level\_index + 1), 'VitD\_levels\_level', vitd\_levels\_group)  See attached code in file: *‘D206-PA-AndréDavis .ipynb’* |
| Re-Expression | Detected columns re-expressed from 0/1’s to Yes/No’s code:  # Detected values of pure 0/1's should be Re-Expressed into Yes/No.  for column in columns\_to\_reexpress:  imputated\_medical\_data[column] = imputated\_medical\_data[column].map({1: 'Yes', 0: 'No'})  See attached code in file: *‘D206-PA-AndréDavis .ipynb’* |

5.  Provide a copy of the cleaned data set.  
  
Clean data file attached to the Performance Assessment is named *‘medical\_cleaned\_data.csv’.*

6.  Summarize the limitations of the data-cleaning process.

|  |  |
| --- | --- |
| ***Detection/Treatment of*** | ***Disadvantages*** |
| Duplicates | Working with the *‘medical dataset’* it was not found to have duplicates.  Some disadvantages to the variations used to check for duplicates is that fuzzy and not-exact duplicates have the potential to sneak through. This can skew data results later down the line. GIGO (Garbage in, Garbage out) could be triggered if there are enough duplicates present. |
| Missing Values | When cleaning the medical data for missing values we went with the Univariate Imputation (Mean, Median, and Mode) as replacement values for the missing data.  Limitations Include:   * “Could possibly distort data/distribution of data” (Middleton, 2022). |
| Outliers | When cleaning the *‘medical dataset’* it was obvious that the outliers were meaningful data and not skewing data due to incorrect entries or other invalid forms.  The RETAIN methodology in this case. The disadvantage of this is that we need to make notes that this is the case for outliers as it may be needed to understand the results of the study later. If we fail to make these notes the interpretation of the results could be misunderstood, especially during the PCA process. |

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

After the treatment and mitigation of data quality an analysis may run into issues if the question grows to include a few more variables that had extreme outliers such as Income, TotalCharge, and Additional\_charges. It was noted that outliers were retained but if the note is missed and they want to add additional factors to their question to see if income is a factor affecting the question the analytics maybe be skewed unexpectedly.

After missing data has been treated via imputation an analyst can run into problems when the incorrect mean, median or mode is used as the imputation value. This means that the incorrect value was selected for the type of distribution a variable (column) shows when charted as a histogram. If the wrong value was selected the distribution will not be distorted and not the same as pre-imputation and this means that the outcome of the analysis has been altered.

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

1.  List the principal components in the data set.  
  
**Discrete/Continuous variables used in calculating the Loading Matrix for PCA:**  
Lat, Lng, Population, Age, Children, Income, VitD\_levels, Full\_meals\_eaten, VitD\_supp, Initial\_days, TotalCharge, Additional\_charges

**PCA Loadings Matrix:**  
  
Graphical user interface, text

Description automatically generated

2.  Describe how you identified the principal components of the data set.

For the selection of PCs, we used the Kaiser Rule. This generates a scree graph of eigenvector values. Any value that is above 1 is a principal component (PC) that should be selected.

The Principal Components that are equal to or greater than 1 includes PC1, PC2, PC3, PC4, PC5, and PC6. According to the Kaiser Rule these are the selected and kept Principal Components. PC7 was close at 0.996 but in the end starts the list of components not selectable by the Kaiser Rule.

Chart, line chart

Description automatically generated

Figure - Kaiser Rule Visualization

3.  Describe how the organization can benefit from the results of the PCA

“PCA can help us uncover the underlying trends in our data” (Yiu, 2021).

An organization can benefit from the PCA done on the medical data set seen in the table below, which is broken down per Principal Component that was selected. Each PC can shed light on which coefficients have strong relationships with each other and reveal potential trends.

|  |  |
| --- | --- |
| **Principal Component** | **Findings** |
| PC1 | If a cutoff value of 0.4 and greater is used, we are presented with variables (columns) Initial\_days, TotalCharge and VitD\_levels.  This PC seems to be describing that a patient’s initial visits and total charges goes up as vitamin D levels increase. It’s worth noting that the outliers that were retained are probably contributing to this PC. The outliers were valid values that indicated someone with who has Vitamin D toxicity which is common with people with certain medical conditions. |
| PC2 | If we use a cutoff value of 0.4 or greater, we are presented with Age and Additiona\_charges. This PC seems to be describing as age increases additional charges increases. While it seems the initial days in the hospital goes down.  This may be describing as someone becomes elderly, they have more outpatients visits that rack up additional charges. |
| PC3 | This PC has two variables that have strong positive and negative values. As the Population variable increases the Lat (latitude) variable decreases.  This PC could be describing a situation in which as population rises the geographical location of that population tends to lean Southern. |
| PC4 | If a cutoff value of 0.25 or greater is used, we are presented with Population, Children, and VitD\_supp, and Initial\_days  It is unclear what this may be describing specifically but it seems indicate that the population and child count goes up so does your vitamin D supplements during an increasing initial hospital visit. |
| PC5 | If a cutoff value of 0.5 or greater as well as heading towards -.25 or greater (to -1), then we are presented with Full\_meals\_eaten, Initial\_days, and VitD\_supp.  As the number of meals eaten goes up the need for Vitamin D supplements and length of initial stay in hospital goes down. |
| PC6 | If a cutoff value of 0.4 or greater is used, we are presented with Income and VitD\_supp. The higher the income the more vitamin d supplements needed. This could be indicative of higher incomes being linked to working a lot indoors. |

# Part IV: Supporting Documents

F.  Provide a Panopto recording that demonstrates the warning- and error-free functionality of the code used to support the discovery of anomalies and the data cleaning process and summarizes the programming environment.

Note: For instructions on how to access and use Panopto, use the "Panopto How-To Videos" web link provided below. To access Panopto's website, navigate to the web link titled "Panopto Access", and then choose to log in using the “WGU” option. If prompted, log in using your WGU student portal credentials, and then it will forward you to Panopto’s website.

To submit your recording, upload it to the Panopto drop box titled “Data Cleaning – NUM2 \ D206” Once the recording has been uploaded and processed in Panopto's system, retrieve the URL of the recording from Panopto and copy and paste it into the Links option. Upload the remaining task requirements using the Attachments option.

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=8a6665e0-c73a-4474-80fd-af64010ad9c9>

G.  Reference the web sources used to acquire segments of third-party code to support the application. Be sure the web sources are reliable.  
  
Dr. Keiona Middleton (Instructor). (September 2022) D206 – Webinar 4: Getting Started with PCA - (September 2022) [Video Webinar].

Retrieved from https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=b627935e-73f8-41ad-b97b-af1e000d3015

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

Dr. Keiona Middleton (Instructor). (October 2022) D206 – Webinar 2: Getting Started Webinar Missing Values and Outliers (October 2022) [Video Webinar].  
 Retrieved from https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=d3af9533-0b7f-4d42-9db2-af48002f4799

Yiu, T. (2021, December 10). *Understanding PCA (Principal Components Analysis) - Towards Data Science*. Medium. https://towardsdatascience.com/understanding-pca-fae3e243731d

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

1. https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.duplicated.html [↑](#footnote-ref-1)
2. https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.isna.html [↑](#footnote-ref-2)
3. https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.sum.html [↑](#footnote-ref-3)
4. https://github.com/ResidentMario/missingno [↑](#footnote-ref-4)
5. https://pandas.pydata.org/pandas-docs/stable/getting\_started/comparison/comparison\_with\_r.html [↑](#footnote-ref-5)
6. <https://worldpopulationreview.com/us-cities/katy-tx-population> [↑](#footnote-ref-6)
7. https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.isin.html [↑](#footnote-ref-7)
8. https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.all.html [↑](#footnote-ref-8)
9. https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.Series.map.html [↑](#footnote-ref-9)