**HOW DOES INCOME AFFECT MONTHLY CHARGES ON INTERNET SERVICES?**

NUM2 – NUM2 TASK1: DATA CLEANING

DATA CLEANING – D206

PRFA – NUM2

STUDENT NAME: IBRAHIM SULEIMAN

DATA ANALYTICS

STUDENT NUMBER: 001429984

COLLEGE OF INFORMATION TECHNOLOGY, WESTERN GOVERNORS UNIVERSITY

NAME OF INSTRUCTOR: KEIONA MIDDLETON

December 18th, 2021

**HOW DOES INCOME AFFECT MONTHLY CHARGES ON INTERNET SERVICES?**

Part I: **Research Question**

**A. 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 real organizational need or situation.**

**The description states a question or decision that can be addressed through analysis of the chosen data set.**

How does income affect monthly charges on internet services?

The question is relevant to a real organizational need or situation in creating an innovative solution for the need of having internet security based on the customer’s income and subscriptions with regards to the rate at which internet service is used.

**B.  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.**

***The description includes the variables in the data set and indicates the specific type of data being described, and includes examples from the data set to support claims.***

The data set is Churn CSV dataset which will be imported to describe the datasets to support the claims.

The below data sets contain the entire data sets variable that entails NAs.

**Data Table**

import pandas as pd

# Info of column names along with the number of non –null values in each column,

df = pd.read\_csv('churn\_data')

df.info()

**<class 'pandas.core.frame.DataFrame'>**

**RangeIndex: 10000 entries, 0 to 9999**

**Data columns (total 52 columns):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Variable Name** | **Non-Null Count** | **Variable Data type** | **Descriptions of the Variables** | **Examples** |
| 0 | Unnamed: 0 | 10000 non-null | int64  **quantitative** | Unnamed means it's a variable that maintains the actual order of the unprocessed data file that could be updated at some point. This contains 0 NA and the data is complete and has integers numeric. | 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20. |
| 1 | CaseOrder | 10000 non-null | int64  **quantitative** | Case order means the categorical sequence of the products customers purchase, this contains 0 NA and data is complete, it has integers and numeric values. | 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20. |
| 2 | Customer\_id | 10000 non-null | Object **qualitative**  **categorial data** | Customer\_id is the unique identifier for each customer, contains 0 NA and data is complete and contains Var and integers. | K409198,S120509,K191035,D90850,K662701,W303516 |
| 3 | Interaction | 10000 non-null | Object **qualitative**  **categorial data** | The customer's interaction is two-way communication by chats, phone or recorded contact center, and technical support. The interaction shows the behavior of one variable depends on the value of another variable, contains 0 NA and data is complete in this column and it contains Var and int values. | Aa90260b-4141-4a24-8e36-b04ce1f4f77b, fb76459f-c047-4a9d-8af9-e0f7d4ac2524, 344d11c-3736-4be5-98f7-c72c281e2d35, abfa2b40-2d43-4994-b15a-989b8c79e311. |
| 4 | City | 10000 non-null | Object **qualitative**  **categorial data** | The city describes where the customer resides permanently and is densely settled with administrative defined boundaries.  , it contains 0 NA, Char, and the data is complete. | Names of cities: point baker, west branch, yamhill, del mar, needville, fort valley. |
| 5 | State | 10000 non-null | Object **qualitative**  **categorial data** | The customer’s state is an organized political community under one government, It contains 0 NA, CHAR duplicates and the data is complete in this column. | TX,TX,GA,CA,AZ,GA,MI,VT,ID,MN,FL,CA,VA,PA,LA,NV,MA |
| 6 | County | 10000 non-null | Object **qualitative**  **categorial data** | The customer’s county is a political and administrative division in a state providing specifics local government services, contains 0 NA and data is complete and data is complete in this column CHAR. | Prince of wales-hyder, ogemaw, yamhill, san diego, fort bend, peach, scott, oklahoma, osceola. |
| 7 | Zip | 10000 non-null | int64  **quantitative** | Zip code of customer is a series of numbers used to communicate information of postal addresses sing five numerical digits, it contains 0 NA and data is complete in this column and it has integer values. | 99927,48661,97148,92014,77461,31030,37847,73109,34771,45237 |
| 8 | Lat | 10000 non-null | float64  **quantitative** | The latitude of the customer is the geographical coordinate that specifies the North-South position of a point on the earth's surface. contains 0 NA and data is complete in this column, this data type entails integer mixed with fraction presented in decimal format. | 56.251, 44.32892, 45.35589,32.96687, 29.38012, 32.57032, 36.4342 |
| 9 | Population | 10000 non-null | int64  **quantitative** | The population of the customer means the number of people in a city measured in quantity. Population variable contains 0 NA and data is complete in this column, this data type entails stored values outside the range of the progress integer data type. | 38, 10446, 3735, 13863, 11352, 17701, 2535, 23144, 17351, 20193, 555, 0. |
| 10 | Lng | 10000 non-null | float64  **quantitative** | The longitude of the customer is a geographical coordinate that specifies the East-West position of a point on the earth's surface. It contains 0 NA and data is complete in this column, this data type comprises real numbers and decimal points showing real values dividing the integer and fractional parts of the data with negative digression values. | -33.3757,-84.2408,-123.2466,-117.248,-95.80673,-83.8904,-81.27892 |
| 11 | Area | 10000 non-null | Object **qualitative**  **categorial data** | An area describes where the customer lives which is considered a unit of purpose or classification part of a town, country, or region. It contains 0 NA and data is complete in this column with unique identities that create meaning as a whole, it has CHAR. | Urban, suburban, rural |
| 12 | Timezone | 10000 non-null | Object **qualitative**  **categorial data** | A Time zone is a customer's area that observes a uniform standard time for legal, commercial, and social purposes. It contains 0 NA and data is complete which gives identified the meaning of different times of diverse locations, it contains duplicates and CHAR. | America/New\_york, America/Chicago, America/Denver, America/New\_york, America/New\_york. |
| 13 | Job | 10000 non-null | Object **qualitative**  **categorial data** | A customer's job refers to a task or works an individual does regularly to earn money. It contains 0 NA, CHAR. The data is complete and it reflects job types in different fields of professions separated by commas. | Environment health practitioner, chief financial officer, solicitor, medical illustrator, immunologist,  immigration officer  Engineer, electrical.  programmer, multimedia. |
| 14 | Children | 7505 non-null | float64  **quantitative** | This describes the number of children of the customer which refers to a person between the stage of birth and puberty, it contains 2495 nulls and missing not at random with number values. | NA, 7, 2, 0, 5, NA, NA, NA, NA, 3, 3, 4, 3, NA ,2 ,3.  Missing data is missing based on the missing column which is variable. |
| 15 | Age | 7525 non-null | float64  **quantitative** | The customer's age refers to the length of time a person has lived or the number of years, it contains 2475 NAs and missing completely at random with number values. | 83, 83, NA, NA, 49, 86, 23, 56, 83, NA, 30, 39, 63  There is no relationship between the missingness of data and any values observed or missing. |
| 16 | Education | 10000 non-null | Object  **qualitative**  **categorial data** | The customer's educational qualifications reflect the acquired knowledge and skills, morals, values, personal development, it contains 0 NA and data is complete and it shows the type of education completed. It contains Varchar and int values | Nursery school to 8th grade, 9th grade to 12th grade no diploma, associate's degree, bachelor's degree, master's degree, regular high school diploma, doctorate, GED or alternative credential, some college, less than 1 year, some college, 1 or more years, no degree |
| 17 | Employment | 10000 non-null | Object  **qualitative**  **categorial data** | This reflects the condition of having paid work and the status of the condition of the customer, it contains 0 NA and data is complete. It has CHAR with duplicates of employment status. | Part time, retired, student, student, fulltime, fulltime  ,fulltime, fulltime, fulltime. unemployed, unemployed, unemployed |
| 18 | Income | 7510 non-null | float64  **quantitative** | Customer's income is the money received regularly for work or task, It contains 2490nulls and missing completely at random. | 28561.99,21704.77,NA,18925.23,40074.19,NA  There is no relationship between the missingness of data and any values, observed or missing |
| 19 | Marital | 10000 non-null | Object **qualitative**  **categorial data** | Marital means the marital status of the customer, It contains 0 NA and has Char, | Married, widowed, separated, never married, divorced |
| 20 | Gender | 10000 non-null | Object **qualitative**  **categorial data** | Gender is a nominal variable reflecting how the customer identifies as male or female or prefers not to answer, it contains 0 NA and has Char. | Male, female, prefer not to answer |
| 21 | Churn | 10000 non-null | Object **qualitative**  **categorial data** | Churn reflects whether the customer stopped using the service during a certain time frame. It contains 0 NA and has Char. | No, yes, no, yes, no |
| 22 | Outage\_sec\_perweek | 10000 non-null | float64  **quantitative** | This variable reflects the amount of time of seconds per week an outage occurred in the area of customers, it contains 0 NA, integers and the data is complete in this column. This data type comprises of integer mixed with fraction presented in decimal format. | 6.9725661, 12.014541, 10.245616, 15.206193, 8.9603164 |
| 23 | Email | 10000 non-null | int64  **quantitative** | This reflects the total number of emails received by customers in the previous year for marketing, it contains 0 NA, integers and the data is complete in this column. | 10, 12, 9, 15, 16, 15, 10, 16, 20, 18, 9, 17, 9, 14 |
| 24 | Contacts | 10000 non-null | int64  **quantitative** | This describes the overall technical support contacts a customer made, it contains 0 NA, integers and the data appears complete although it has duplicate values. | 0, 0, 0, 2, 2, 3, 0, 0, 2, 1, 0, 1 |
| 25 | Yearly\_equip\_failure | 10000 non-null | int64 **quantitative** | This describes the number of times the customer experienced a service failure yearly, it contains 0 NA, it has integers and the data appears to be complete although it has duplicate values. | 1, 1, 1, 0, 1, 1, 1, 0, 3, 0, 2, 1 |
| 26 | Techie | 7523 non-null | Object **qualitative**  **categorial data** | This reflects whether the customer is highly proficient technically wise, contains 2477 Nulls, and missing completely at random. | yes, yes, NA, NA, no, no, no, no ,no, no, no, no, yes, no, NA, no  There is no relationship between the missingness of data and any values, observed or missing. |
| 27 | Contract | 10000 non-null | Object **qualitative**  **categorial data** | This reflects the duration of contract agreement of the customer, contains 0 NA, Has Char, data appears complete although it has duplicate values. | One year, month-to-month, two years, two years, one year, month-to-month |
| 28 | Port\_modem | 10000 non-null | Object **qualitative**  **categorial data** | This reflects if the customer uses a portable modem, contains 0 NA, Has Char, data appears complete. | Yes, no, yes, no, yes, yes, no, no, yes, yes |
| 29 | Tablet | 10000 non-null | Object **qualitative**  **categorial data** | This reflects if the customer uses a tablet device like iPad, tablets, contains 0 NA, Has Char, data appears complete. | Yes, yes, no, no, no, no, no, no, no, no, no, no, no, no, no |
| 30 | Internet service | 10000 non-null | Object  **qualitative**  **categorial data** | This reflects the internet service type used by the customer, contains 0 NA, Has Char, data appears complete. | Fiber optic, fiber optic, DSL, DSL, fiber optic, none, none |
| 31 | Phone | 8974 non-null | Object **qualitative**  **categorial data** | This reflects if the customer has service on phone devices, contains 1026Nulls, and missing completely at random, Has Char. | Yes, yes, NA, yes, yes, yes, yes, yes, yes, NA, yes, yes, no, yes, yes, NA.  There is no relationship between the missingness of data and any values, observed or missing. |
| 32 | Multiple | 10000 non-null | Object **qualitative**  **categorial data** | This reflects whether the customer uses more than one line, contains 0 NA, Char, data appears complete. | No, yes, yes, no, no, yes, no, no, no, yes |
| 33 | online security | 10000 non-null | Object **qualitative**  **categorial data** | This shows whether the customer uses online security for the protection of their information, contains 0 NA, Has Char, data appears complete. | Yes, yes, no, yes, no, yes, no, no, yes |
| 34 | Online Backup | 10000 non-null | Object **qualitative**  **categorial data** | This reflects if the customer has online backup, contains 0 NA, Has Char, data appears complete. | Yes, no, no, no, no, yes, no, yes, yes |
| 35 | DeviceProtection | 10000 non-null | Object **qualitative**  **categorial data** | This reflects whether device protection is used by the customer, contains 0 NA, Has Char, data appears complete. | No, no, no, no, no, yes, no, no, no, yes, no |
| 36 | TechSupport | 9009 non-null | Object **qualitative**  **categorial data** | This reflects whether the customer uses tech support, contains 991Nulls and Missing at random | No, no, no, NA, no, no, no, no, no, yes, no, NA, no, no, yes, no, no, NA, no, yes. Missing at random because missing data is affected only by the complete observed variables and not by the characteristics of the missing data itself. |
| 37 | StreamingTV | 10000 non-null | Object **qualitative**  **categorial data** | This reflects if the customer has streaming TV, contains 0 NA, Has Char, data appears complete. | No, yes, no, yes, no, yes, no, no, no, yes |
| 38 | StreamingMovies | 10000 non-null | Object **qualitative**  **categorial data** | This reflects if the customer has streaming movies, contains 0 NA, Has Char, data appears complete. | Yes, yes, yes, no, no, yes, yes, no, no, yes, no, no, yes |
| 39 | Paperless Billing | 10000 non-null | Object **qualitative**  **categorial data** | This reflects if the customer uses a paperless billing option, contains 0 NA, Has Char, data appears complete. | Yes, yes, yes, yes, no, no, no, yes, yes, yes |
| 40 | Payment Method | 10000 non-null | Object **qualitative**  **categorial data** | This reflects if the customer uses a specific method of payment of bills, contains 0 NA, Has Char, data appears complete. | Payment method, credit card (automatic),  bank transfer (automatic), mailed check, mailed check, electronic check, mailed check, electronic check, electronic check, mailed check, bank transfer (automatic) |
| 41 | Tenure | 9069 non-null | float64  **quantitative** | This reflects the duration the customer uses the service provided by the specific service provider, contains 931Nulls, and missing not at random. | 13.236774, NA, 8.2206864, 3.4220861, 19.267262, NA,  13.011492, 16.87922, 10.060199, 13.870013, 15.78215, NA, 1  7.109956. Missing at random because missing data is affected only by the complete observed variables and not by the characteristics of the missing data itself. |
| 42 | MonthlyCharge | 10000 non-null | float64  **quantitative** | This reflects the customer’s monthly payments for services received , contains 0 NA ,has integers, data appears complete. | 171.4497621, 242.9480155, 159.4403984, 120.2494934, 150.7612159, 184.4015581, 200.0648859, 114.7541111 |
| 43 | Bandwidth\_GB\_Year | 8979 non-null | float64  **quantitative** | This reflects the customer’s data (GB)consumption annually, contains 1021Null and missing not at random . | 631, NA, 777, 139, 393, NA, 213, 443, 248, 223, NA, 188, 229, 826, NA.  There is no relationship between the missingness of data and any values, observed or missing |
| 44 | item1  Timely Response | 10000 non-null | int64  **quantitative** | Contains 0 NA ,has integers, data appears complete. | 5, 3, 4, 4, 4, 3, 6, 2, 5, 2, 4, 4, 1, 5, 3, 3,  These (Items 1 to 8) variables shows responses to questions based on relevant factors scaling 1 to 8, with 1 = Most Agreed and 8 = Least Agreed, |
| 45 | item2  Times fixes | 10000 non-null | int64  **quantitative** | Contains 0 NA ,has integers, data appears complete. | 5, 4, 4, 4, 4, 4, 4, 3, 5, 2 |
| 46 | item3  Timely replacements | 10000 non-null | int64  **quantitative** | Contains 0 NA, ,has integers, data appears complete | 5, 3, 2, 4, 4, 3, 6, 2, 4, 2 |
| 47 | item4 Reliability | 10000 non-null | int64  **quantitative** | Contains 0 NA ,has integers, ,has integers, data appears complete. | 3, 3, 4, 2, 3, 2, 4, 5, 3, 2 |
| 48 | item5  Options | 10000 non-null | int64  **quantitative** | Contains 0 NA ,has integers, data appears complete. | 4, 4, 4, 5, 4, 4, 1, 2, 4, 5 |
| 49 | item6  Respectful response | 10000 non-null | int64  **quantitative** | Contains 0 NA ,has integers, , data appears complete. | 4, 3, 3, 4, 4, 3, 5, 3, 3, 2 |
| 50 | item7  Courteous exchange | 10000 non-null | int64  **quantitative** | Contains 0 NA ,has integers, data appears complete. | 3, 4, 3, 3, 4, 3, 5, 4, 4, 3, 3 |
| 51 | item8  Evidence of active listening | 10000 non-null | int64  **quantitative** | Contains 0 NA ,has integers, data appears complete. | 4, 4, 3, 3, 5, 3, 5, 5, 4, 3, 4 |

dtypes: float64(9), int64(15), object(28)

memory usage: 4.0+ MB

Part II: **Data-Cleaning Plan**

**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.**

***The proposal includes a detailed description of the techniques and steps needed for identifying anomalies in the selected data set.***

In the stage of data cleaning, we will perform multiple analytical techniques which goes a long way in examining the variables, analytically detecting missing values, outliers and cleaning the data sets to be error free.

**Plan for detecting missing Values**

**Step 1**. The entire data set with the NAs will be imported.

**Step 2**. The isnull().sum() function will be executed and this will populate the visualized variables with missing values. The missing values will be detected (NAs) by performing the is isnull().sum() function (on python).

**Plan for detecting Outliers**

**Step 1. Histogram:** This will visualize the isolated bars within the distribution.

Use of histogram to detect outlier by applying the function below;

|  |
| --- |
| # Matplotlibis deployed to plot and Visualize bars  %matplotlib inline  import matplotlib.pyplot as plt  plt.hist(df['column'])  plt.show()  df.hist(column) |

**Step 2. Boxplot**: This will be executed to visualize the men values, dispersion of dataset, a sign of skewness, and summary of the data to show the outliers, therefore values that fall outside the two fences (minimum and maximum) are considered outliers. The below is the function:

|  |
| --- |
| # Using Seaborn to provides visualization  import seaborn  # Outliers are not dictated  boxplot=seaborn.boxplot(x='column',data=df) |

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

**•  characteristics of the data being assessed,**

**•  the approach used to assess the quality.**

***The justification includes the characteristics of the data being assessed and references the approach used to assess the quality of the data. The justified approach aligns with the selected data set.***

**The Reason for using isnull()sum Function**

The approach used to assess the quality of the data was to find missing values (NA)s within the data set using codes on python which were in the form of NAs. Therefore, the function used to assess the NAs values in the data was isnull()sum function respectively. Therefore working with dirty data is unavoidable, so the cleaning of the data is relevant (John Sullivan, 2018).

Therefore, the reason for using the isnull function is to make identify and locate all missing data in the churn dataset, using the children variable which contains NAs which could be misleading during the several steps of the data cleaning process therefore, we can easily replace the NAs with mean values and make sure that the original dataset is maintained.it is better to replace the NAs instead of deleting the NAs. Because deleting the NAs in bulk could affect the size and quality of the entire data sets being analyzed (Larose & Larose 2019).

**The Reason for using Boxplot**

The income and children columns show several outliers clearly. This is achieved with the use of a boxplot which vividly shows the distribution of that data with more detailed information. It also reveals the outliers more clearly, maximum, minimum, quartile (Q1), third quartile(Q3), interquartile range (IQR), and median. The children column appears to have approximately four outliers at maximum (Q3), therefore the children and income column requires to be treated using treatment technique.

The boxplot gives a consistent and uniform approach of visualizing the distribution of data established (minimum, first quartile, median, third quartile, and maximum) upon how the values are spread and have the advantage of utilizing a small portion of space size (Michael Galarnyk 2018).

**The Reason for using Histogram**

The use of a histogram in the data detection stage helps provide a visual representation of the distribution of the children variable and income variable because it reflects the location, spread, and skewness of the data, it also helps to visualize whether the distribution is symmetric or skewed left or right. Additionally, if it is unimodal, bimodal, or multimodal. It has the capability of presenting outliers and gaps in the data. Therefore, accessing data quality and finding outliers is a continuous process that allows the identification of unusual data points within a data set (Angelica Lo Duca, 2021).

**Reasons for Missing Values and Outliers**

The data being assessed contains missing values and outliers due to the following reasons :

* The data creator mistakenly skipped the row.
* There was a loss of data during the process of migrating the data into a different database.
* Mistakes result in the programming operations.

**3.  Justify your selected programming language and any libraries and packages that will support the data-cleaning process.**

***The justification describes the benefits of using the programming language, including the libraries and packages used to clean the data, and includes specific examples of how these tools are ideal in this scenario as opposed to other available tools.***

As machine learning in data analysis is skyrocketing in the contemporary business environment, there is a strong urge to utilize effective programming tools and methods in

data analysis (Michael Galarnyk,2018).

Therefore, to achieve the goals of this assessment, a python programming language will be used from the inception of data cleaning to the production of a cleaned data set respectively.

**[1.] Anaconda navigator**: The anaconda navigator serves as a geographical platform for launching a python program without having to use common lines to install packages and manage the platform.

**The Reason for Using Anaconda Navigator this Report**

• This tool provided the Jupyter notebook (Python) programming language that was used in this report

**[2.] Python:** is an informal language that is well-used to read, assemble and organize code capable of execution on several interfaces, platforms designated by workload management systems due to its powerful indentation.

**The Reasons for using Python for this Report**

• Eco-friendly environment – The user-friendly environment python entails provided a fast and easy way to adapt to the functions and perform code execution easily.

• The python is effective for implementing, reading code, detecting missing values and outliers in the data set, and creating visual representations to spot animalities and identify outliers during the data cleaning execution

• Python made it systematic for the ‘Churn\_data.csv’ file to be imported for the cleaning operations to begin.

In addition, python allowed the importation of packages into the new environment such as Pandas, NumPy, Matplotlib, Sklearn, and Seaborn. These packages offer a variety of features such as creating visualizations of histograms, boxplots, and data tables. With no doubt, these packages alongside the programming languages are user-friendly, ideal, and intuitive in providing data analysts with efficiency and error-free output in an innovative presentation as opposed to other tools (Michael Galarnyk ,2018).

**4.  Provide the code you will use to identify the anomalies in the data.**

***The submission provides the complete and executable code, which could be used to identify anomalies in the data set.***

|  |  |
| --- | --- |
| Code |  |
| import pandas as pd  df = pd.read\_csv('churn\_data')  df.info()  df.isna().sum() | This is a relevant technique that is needed to identify anomalies in the data set which is the NAs |
| import numpy as np  import pandas as pd  from pandas import DataFrame  # Matplotlibis deployed to plot and Visualize bars  %matplotlib inline  import matplotlib.pyplot as plt  df.hist()  plt. df['Age'].hist(figsize=(8,5))  df['Children'].hist(figsize=(8,5))  df['Income'].hist(figsize=(8,5))  df['Tenure'].hist(figsize=(8,5))  df['Bandwidth\_GB\_Year'].hist(figsize=(8,5)) | This will create a histogram to visualizing the outliers. |
| import seaborn  boxplot=seaborn.boxplot(x='Age',data=df)  boxplot=seaborn.boxplot(x='Children',data=df)  boxplot=seaborn.boxplot(x='Income',data=df)  boxplot=seaborn.boxplot(x='Tenure',data=df)  boxplot=seaborn.boxplot(x='Bandwidth\_GB\_Year',data=df) | Using Seaborn package to provides boxplot visualization  This will create boxplots to visualize the outliers present. |

**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.**

***The description accurately includes all of the anomalies found by running the code from part C4.***

Findings achieved by running the code from part c4 indicates that we have NAs missing values in the data after executing the below isnull function.

|  |
| --- |
| df.isnull().sum()  Unnamed: 0 0  CaseOrder 0  Customer\_id 0  Interaction 0  City 0  State 0  County 0  Zip 0  Lat 0  Lng 0  Population 0  Area 0  Timezone 0  Job 0  Children 2495[***anomalies found***]  Age 2475[***anomalies found***]  Education 0  Employment 0  Income 2490[***anomalies found***]  Marital 0  Gender 0  Churn 0  Outage\_sec\_perweek 0  Email 0  Contacts 0  Yearly\_equip\_failure 0  Techie 2477[***anomalies found***]  Contract 0  Port\_modem 0  Tablet 0  InternetService 0  Phone 1026[***anomalies found***]  Multiple 0  OnlineSecurity 0  OnlineBackup 0  DeviceProtection 0  TechSupport 991[***anomalies found***]  StreamingTV 0  StreamingMovies 0  PaperlessBilling 0  PaymentMethod 0  Tenure 931[***anomalies found***]  MonthlyCharge 0  Bandwidth\_GB\_Year 1021[***anomalies found***]  item1 0  item2 0  item3 0  item4 0  item5 0  item6 0  item7 0  item8 0  dtype: int64 |

**# Results found during Detecting Outliers**

***Findings achieved by running the code from part c4 indicates that we have outliers in the data after executing the below functions***

|  |
| --- |
| import numpy as np  import pandas as pd  from pandas import DataFrame  # Matplotlibis deployed to plot and Visualize bars  %matplotlib inline  import matplotlib.pyplot as plt  df['Age'].hist(figsize=(8,5))  <AxesSubplot:>    # Using Seaborn to provides visualization  import seaborn  **# Outliers are not dictated**  boxplot=seaborn.boxplot(x='Age',data=df)  Chart  Description automatically generated  df['Children'].hist(figsize=(8,5))  <AxesSubplot:>    **# Outliers are dictated**  boxplot=seaborn.boxplot(x='Children',data=df)  Chart  Description automatically generated  plt.hist(df['Income'])  plt.show()    **# Outliers are dictated**  boxplot=seaborn.boxplot(x='Income',data=df)  Chart  Description automatically generated  plt.hist(df['Tenure'])  plt.show()    **# Outliers are not dictated**  boxplot=seaborn.boxplot(x='Tenure',data=df)  Chart  Description automatically generated  plt.hist(df['Bandwidth\_GB\_Year'])  plt.show()    **# Outliers are not dictated**  boxplot=seaborn.boxplot(x='Bandwidth\_GB\_Year',data=df)  Chart, histogram  Description automatically generated |

From the above findings, there are NAs values in these columns; Children[2495], Age[2475], Income[2490], Techie [2477], Phone [1026], TechSupport [991], Tenure [931]and Bandwidth\_GB\_Year [1021].

However, it also appears that there are outliers in the income variable and children variable and require treatment in the next phase of the data cleaning process.

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

***The justification includes the specific mitigation methods for each type of anomaly listed in part D1.***

In this stage, we are going to perform methods and techniques to treat missing values and outliers to arrive at a cleaned and corrupt-free dataset.

**Why we used the Two Treatment Methods**

**- Univariate Imputation:** The reason for applying this method in the treatment of data is because this method is effective in replacing the missing values of specific variables having animalities by applying to mean, median, and mode because this method is highly relevant when the percentage of the missing data is less. Therefore, this loss of data can be avoided using this method. It is chosen because it is the best technique if the data size appears to be less and it goes a long way in analytically avoiding the depreciation of data in the deletion of rows and columns respectively.

**- Deletion/Dropping method:** This method will be used to treat missing data actively present in the rows and columns. The reason this method is chosen is that the fact that dropping missing data could lead to an enormous error-free output. Therefore dropping a specific row or column with certain data is preferred because the value isn't containing an inclined weightage.

**Before Imputation Treatment**

Examination of the distribution and treating missing values

|  |
| --- |
| # TREATING MISSING VALUES  # EXAMINE THE DISTRIBUTION  df.hist()  array([[<AxesSubplot:title={'center':'Unnamed: 0'}>,  <AxesSubplot:title={'center':'CaseOrder'}>,  <AxesSubplot:title={'center':'Zip'}>,  <AxesSubplot:title={'center':'Lat'}>,  <AxesSubplot:title={'center':'Lng'}>],  [<AxesSubplot:title={'center':'Population'}>,  <AxesSubplot:title={'center':'Children'}>,  <AxesSubplot:title={'center':'Age'}>,  <AxesSubplot:title={'center':'Income'}>,  <AxesSubplot:title={'center':'Outage\_sec\_perweek'}>],  [<AxesSubplot:title={'center':'Email'}>,  <AxesSubplot:title={'center':'Contacts'}>,  <AxesSubplot:title={'center':'Yearly\_equip\_failure'}>,  <AxesSubplot:title={'center':'Tenure'}>,  <AxesSubplot:title={'center':'MonthlyCharge'}>],  [<AxesSubplot:title={'center':'Bandwidth\_GB\_Year'}>,  <AxesSubplot:title={'center':'item1'}>,  <AxesSubplot:title={'center':'item2'}>,  <AxesSubplot:title={'center':'item3'}>,  <AxesSubplot:title={'center':'item4'}>],  [<AxesSubplot:title={'center':'item5'}>,  <AxesSubplot:title={'center':'item6'}>,  <AxesSubplot:title={'center':'item7'}>,  <AxesSubplot:title={'center':'item8'}>, <AxesSubplot:>]],  dtype=object) |
| # Skewed distribution positively skewed right  df.hist(column='Children')  array([[<AxesSubplot:title={'center':'Children'}>]], dtype=object)    # Equally Spread (uniform distribution)  df.hist(column='Age')  array([[<AxesSubplot:title={'center':'Age'}>]], dtype=object)    # Categorical variables  # Skewed distribution positively skewed right  df.hist(column='Income')  array([[<AxesSubplot:title={'center':'Income'}>]], dtype=object)    # Bi modal distribution with two modes non-symmetric  df.hist(column='Tenure')  array([[<AxesSubplot:title={'center':'Tenure'}>]], dtype=object)    # Bi modal distribution non-symmetric  df.hist(column='Bandwidth\_GB\_Year')  array([[<AxesSubplot:title={'center':'Bandwidth\_GB\_Year'}>]], dtype=object) |

**Performing Treatment of Null using Imputation method**

|  |
| --- |
| # Median imputation approach is applied to preserve the integrity of the data, the variable children appears to be a skewed distribution, therefore all the NAs will be replaced with median.  df['Children'].fillna(df['Children'].median(), inplace=True) |
| #Mean imputation will be applied because age appears to be a uniform distribution that why we are using imputation method to replace the NAs with mean.  df['Age'].fillna(df['Age'].mean(), inplace=True) |
| # The income variable is a skewed distribution that is positively skewed to the right and the best way to treat the NAs is by using the mean because this would impute and replace the NAs.  df['Income'].fillna(df['Income'].mean(), inplace=True) |
| # The tenure variable appears to be Bi modal distribution with two modes non-symmetric. therefore the mean will be imputed in order to treat the NAs in this variable.  df['Tenure'].fillna(df['Tenure'].mean(), inplace=True) |
| # The bandwidth\_GB\_year appears to be Bi modal distribution non-symmetric, therefore the best way to treat it is by imputation of the mean to replace the NAs.  df['Bandwidth\_GB\_Year'].fillna(df['Bandwidth\_GB\_Year'].mean(), inplace=True) |
| **Performing Treatment of NA Using Deletion/Dropping Method**  df['TechSupport'].fillna(df['TechSupport'].mode()[0])  0 No  1 No  2 No  3 No  4 Yes  ...  9995 No  9996 No  9997 No  9998 Yes  9999 No  Name: TechSupport, Length: 10000, dtype: object  df.dropna(subset=['TechSupport'])  df.dropna()  df.dropna(subset = ["TechSupport"], inplace=True)  df.replace('TechSupport', " ", inplace=True)  nan\_value = float("NAN")  df.replace("", nan\_value, inplace=True)  df.dropna(subset = ["TechSupport"], inplace=True) |
| df['Techie'].fillna(df['Techie'].mode()[0])  0 No  1 Yes  2 Yes  3 Yes  4 No  ...  9995 No  9996 No  9997 No  9998 No  9999 No  Name: Techie, Length: 10000, dtype: object  df.dropna(subset=['Techie'])  df.dropna()  df.dropna(subset = ["Techie"], inplace=True)  df.replace('Techie', " ", inplace=True)  nan\_value = float("NAN")  df.replace("", nan\_value, inplace=True)  df.dropna(subset = ["Techie"], inplace=True) |
| df['Phone'].fillna(df['Phone'].mode()[0])  0 Yes  1 Yes  2 Yes  3 Yes  4 No  ...  9995 Yes  9996 Yes  9997 Yes  9998 No  9999 Yes  Name: Phone, Length: 10000, dtype: object  df.dropna(subset=['Phone'])  df.dropna()  df.dropna(subset = ["Phone"], inplace=True)  df.replace('Phone', " ", inplace=True)  nan\_value = float("NAN")  df.replace("", nan\_value, inplace=True)  df.dropna(subset = ["Phone"], inplace=True) |

**The Treatment of Outliers**

This is achieved by dropping the outliers systematically using the outlierfilter=df[‘column’] function in other to clean the outliers from the children and income columns. Therefore, we could detect duplicates using df.duplicate() in rows and columns and treat them using the df.drop\_duplicates(inplace = True) respectively

|  |
| --- |
| Dropping outliers systematically  outlierFilter=df['Income'] < 65000  df = df[outlierFilter]  boxplot=seaborn.boxplot(x='Income',data=df)  outlierFilter=df['Children'] < 4  df = df[outlierFilter]  boxplot=seaborn.boxplot(x='Children',data=df)  df.head()  df.duplicated()  # Performing Treatment of Outliers by Dropping Rows containing duplicates  df.drop\_duplicates(inplace = True)  df.head(1000) |

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

***The summary details the outcome from the implementation of each data-cleaning step. The summarized expected outcomes are plausible given the interventions.***

**After Imputation and Deleting/Dropping Method has been applied**

The outcome shows the result of the NAs were treated using the imputation and Deletion/Dropping method. By performing the isna().sum() function, we can examine and observe that the NAs have been treated using the imputation and Deletion/Dropping method successfully.

**Cleaned data showing 0 NA**

|  |
| --- |
| df.isna().sum()  Unnamed: 0 0  CaseOrder 0  Customer\_id 0  Interaction 0  City 0  State 0  County 0  Zip 0  Lat 0  Lng 0  Population 0  Area 0  Timezone 0  Job 0  Children 0  Age 0  Education 0  Employment 0  Income 0  Marital 0  Gender 0  Churn 0  Outage\_sec\_perweek 0  Email 0  Contacts 0  Yearly\_equip\_failure 0  Techie 0  Contract 0  Port\_modem 0  Tablet 0  InternetService 0  Phone 0  Multiple 0  OnlineSecurity 0  OnlineBackup 0  DeviceProtection 0  TechSupport 0  StreamingTV 0  StreamingMovies 0  PaperlessBilling 0  PaymentMethod 0  Tenure 0  MonthlyCharge 0  Bandwidth\_GB\_Year 0  item1 0  item2 0  item3 0  item4 0  item5 0  item6 0  item7 0  item8 0  dtype: int64 |

|  |
| --- |
| Before mean was applied to Age variable in the first stage and second stage after applied on the variable.  plt.hist(df['Age'])  plt.show()  It reflects a spread narrow range.  array([[<AxesSubplot:title={'center':'Age'}>]], dtype=object)  After  Before median was applied to Children variable in the first stage and second stage after applied on the variable.  df.hist(column='Children')  It is a skewed distribution positively skewed right.  array([[<AxesSubplot:title={'center':'Children'}>]], dtype=object)  After  Before mean was applied to Income variable in the first stage and second stage after applied on the variable.  df.hist(column='Income')  It is a skewed distribution positively skewed to the right.  array([[<AxesSubplot:title={'center':'Income'}>]], dtype=object)  After  Before mean was applied to Tenure variable in the first stage and second stage after applied on the variable.  This appears to be a Bi modal distribution with two modes non-symmetric.  df.hist(column='Tenure')  array([[<AxesSubplot:title={'center':'Tenure'}>]], dtype=object)  After  Before mean was applied to Bandwidth\_GB\_Year variable in the first stage and second stage after applied on the variable.  This appears to be a Bi modal distribution non-symmetric.  df.hist(column='Bandwidth\_GB\_Year')  array([[<AxesSubplot:title={'center':'Bandwidth\_GB\_Year'}>]], dtype=object)  After |

**The results after Treatment**

In the income column, it appears 200000 was the maximum income which appears to be abnormal compared to the remaining incomes < 65000. Therefore the income >65000 to 200000 are considered out of the normal income and are outliers. However, the children variable appears to contain outliers > 6.5 and the best way to get rid of the outliers is by dropping the outliers >6.5 to 10 which gave a result of 3.5, additionally treating every duplicate in rows and columns respectively.

|  |
| --- |
| Outliers appears in the income variable  boxplot=seaborn.boxplot(x='Income',data=df)  After    Outliers were found in the children variable  boxplot=seaborn.boxplot(x='Children',data=df)  After |

**4.  Provide the code used to mitigate anomalies**.

***The submission provides complete and executable code that could be used to mitigate the anomalies.***

The below codes were used in performing treatment of NAs by replacement of NAs in the Variables of children, age, income, techie, and tech support using univariate imputation method median, mean by the execution of df.dropna function and df[‘column’].fillna

|  |
| --- |
| df['Children'].fillna(df['Children'].median(), inplace=True) |
| df['Age'].fillna(df['Age'].mean(), inplace=True) |
| df['Bandwidth\_GB\_Year'].fillna(df['Bandwidth\_GB\_Year'].mean(), inplace=True) |
| df['Income'].fillna(df['Income'].mean(), inplace=True) |
| df['Tenure'].fillna(df['Tenure'].mean(), inplace=True) |
|  |

Using mode to replace the Techsupport and the Phone variables wasn't achieved, therefore the best way to solve this is by dropping the NAs the below functions df.dropna(subset=[‘column’]) respectively.

|  |
| --- |
| df['TechSupport'].fillna(df['TechSupport'].mode()[0])  df.dropna(subset=['TechSupport'])  df.dropna()  df.dropna(subset = ["TechSupport"], inplace=True)  df.replace('TechSupport', " ", inplace=True)  nan\_value = float("NAN")  df.replace("", nan\_value, inplace=True)  df.dropna(subset = ["TechSupport"], inplace=True)  df['Phone'].fillna(df['Phone'].mode()[0])  df.dropna(subset=['Phone'])  df.dropna()  df.dropna(subset = ["Phone"], inplace=True)  df.replace('Phone', " ", inplace=True)  nan\_value = float("NAN")  df.replace("", nan\_value, inplace=True)  df.dropna(subset = ["Phone"], inplace=True) |

**Code for Treatment of outliers**

|  |
| --- |
| Dropping outliers systematically  outlierFilter=df['Income'] < 65000  df = df[outlierFilter]  boxplot=seaborn.boxplot(x='Income',data=df)  outlierFilter=df['Children'] < 4  df = df[outlierFilter]  boxplot=seaborn.boxplot(x='Children',data=df)  df.head()  df.duplicated()  # Performing Treatment of Outliers by Dropping Rows containing duplicates  df.drop\_duplicates(inplace = True)  df.head(1000) |

**5.  Provide a copy of the cleaned data set.**

***The submission includes a clean data set created from the raw data.***

The provided data set includes the complete list of variables from the chosen data set in part A.

df.to\_csv("icu\_csv\_file.csv")

Output is saved as icu\_csv\_file.csv

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

***The submission accurately summarizes the limitations of the implemented data-cleaning process.***

With no doubt, the main goal is to have unbiased and error-free data, although there are certain setbacks entailed which include the following

- There is the possibility of data shrinkage during the process of cleaning the data which could result from improper usage of the programming languages.

- The data analyst could experience the possibility of consuming a larger amount of schedule if the data is enormous in size.

- The Univariate Imputation Technique also has certain limitations when it comes to the analysis because it could account for ambiguity and confusion as a result of lost data.

- There is the quantum of data decrease and disappearance in the process and the essential amount of the sample data is diminished.

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

***The submission includes a discussion of the impact of the limitations from part D6. The discussion logically aligns with the question or decision from part A.***

In contemporary business, environmental information erupts and data multiply without boundaries, and the need for data analysis and cleaning will always be harnessed to the highest capacity.

This report has come a long way in revealing the fact that the data set entailed missing values among the variables and several techniques were executed to those variables which helped to treat missing values however it is vehement to note that if these techniques were avoided the output will be biased and unpurified and could result in impracticable problem-solving operations.

**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.**

***The submission lists all principal components of the data set.***

Principal component 1,2 appears to be linear putting together the main predictable variables that entails the enormous variance within the data set.(it narrows the total of squared ). The variance inform us on how the data was assigned separately among the 8 principal components. Therefore the first 8 components have 90.34% of data. #PCA1 = 21.13%, PCA2 = 12.70% , PCA2= 11.81%, PCA3 = 11.38%, PCA4 = 11.23%, PCA5 = 10.63%, PCA6 =10.52%, PCA7 = 09.43% analytically.

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

***The description of how the principle components of the data set were identified is accurate and complete.***

PCA Projection during the process of PCA, the data sets produced 8 principle components columns =['Population', 'Children', 'Age', 'Income', 'Outage\_sec\_perweek', 'Tenure', 'MonthlyCharge', 'Bandwidth\_GB\_Year']). Because they have values that are > 1 and that is the reason why we keep them based on the principal component analysis.

- According to Carolina Bento (2020), Principle component analysis is a mechanism of acquiring extensive variables (in form of components) from a gigantic set of variables accessible.

- A Scree plot was applied in the analysis of the variance because this is an effective model because we have 8 components making 8 PCAs = 90.34% of the variance in the data set.

- During this analysis the eigenvalues visualized below reflect the values > or = 1.

- The loadings in this principle components analysis connotes the correlation among the variables which are the rows alongside the factors referred to as columns.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # Principal Components Analysis  import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  import seaborn as sb  from sklearn.decomposition import PCA  from sklearn.preprocessing import StandardScaler  from collections import Counter  from scipy import stats  %matplotlib inline  # Principal Component Analysis (PCA) - Decomposition  df = pd.read\_csv("icu\_csv\_file.csv", index\_col=1)  df2 = df[['Population', 'Children', 'Age', 'Income', 'Outage\_sec\_perweek', 'Tenure', 'MonthlyCharge', 'Bandwidth\_GB\_Year']]  df2.head()   |  | **Population** | **Children** | **Age** | **Income** | **Outage\_sec\_perweek** | **Tenure** | **MonthlyCharge** | **Bandwidth\_GB\_Year** | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Unnamed: 0.1** |  |  |  |  |  |  |  |  | | **1** | 38 | 1.0 | 68.0 | 28561.990000 | 6.972566 | 6.795513 | 171.449762 | 904.536110 | | **2** | 10446 | 1.0 | 27.0 | 21704.770000 | 12.014541 | 1.156681 | 242.948015 | 800.982766 | | **4** | 13863 | 1.0 | 48.0 | 18925.230000 | 15.206193 | 17.087227 | 120.249493 | 2164.579412 | | **5** | 11352 | 0.0 | 83.0 | 40074.190000 | 8.960316 | 1.670972 | 150.761216 | 271.493436 | | **6** | 17701 | 3.0 | 83.0 | 39936.762226 | 7.814859 | 7.000994 | 184.401558 | 1039.357983 |   df2\_normalized=(df2-df2.mean())/df2.std()  df2\_normalized.head()   |  | **Population** | **Children** | **Age** | **Income** | **Outage\_sec\_perweek** | **Tenure** | **MonthlyCharge** | **Bandwidth\_GB\_Year** | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Unnamed: 0.1** |  |  |  |  |  |  |  |  | | **1** | -0.673892 | -0.183588 | 0.822555 | -0.296454 | -0.638332 | -1.096612 | -0.076194 | -1.192259 | | **2** | 0.035912 | -0.183588 | -1.449165 | -0.779943 | 0.084388 | -1.320330 | 1.588086 | -1.242123 | | **4** | 0.268945 | -0.183588 | -0.285601 | -0.975924 | 0.541881 | -0.688294 | -1.267994 | -0.585514 | | **5** | 0.097700 | -1.268954 | 1.653672 | 0.515250 | -0.353407 | -1.299925 | -0.557766 | -1.497087 | | **6** | 0.530689 | 1.987143 | 1.653672 | 0.505561 | -0.517597 | -1.088460 | 0.225287 | -1.127339 |   df212\_pca = PCA(n\_components=df2\_normalized.shape[1])  df212\_pca.fit(df2\_normalized)  Output: PCA(n\_components=8)  df212\_pca\_df = pd.DataFrame(df212\_pca.transform(df2\_normalized),  columns=['Population','Children', 'Age', 'Income', 'Outage\_sec\_perweek', 'Tenure', 'MonthlyCharge', 'Bandwidth\_GB\_Year'])  # Creation of Scree Plot to view explaied variance for components  plt.figure(figsize=(10, 10))  plt.title("Scree Plot", fontweight="bold", color='r')  plt.plot(df212\_pca.explained\_variance\_ratio\_)  plt.xlabel('number of components')  plt.ylabel('explained variance')  plt.show()    # The Creation of Eigen Values for components  cov\_matrix = np.dot(df2\_normalized.T, df2\_normalized) / df2.shape[0]  eigen\_values = [np.dot(eigenvector.T, np.dot(cov\_matrix, eigenvector)) for eigenvector  in df212\_pca.components\_]  plt.figure(figsize=(10, 10))  plt.title("Eigen Values for Components", fontweight="bold", color='r')  plt.plot(eigen\_values)  plt.xlabel('number of components')  plt.ylabel('eigen value')  plt.show()    # Create loading of the values for components  columns=['Population', 'Children', 'Age', 'Income', 'Outage\_sec\_perweek', 'Tenure', 'MonthlyCharge', 'Bandwidth\_GB\_Year']  df212\_loadings = pd.DataFrame(df212\_pca.components\_.T,  columns=['PC1','PC2','PC3','PC4','PC5','PC6','PC7','PC8'], index=df2.columns)  df212\_loadings  columns**=**['Population', 'Children', 'Age', 'Income', 'Outage\_sec\_perweek', 'Tenure', 'MonthlyCharge', 'Bandwidth\_GB\_Year']  df212\_loadings **=** pd.DataFrame(df212\_pca.components\_.T,  columns**=**['PC1','PC2','PC3','PC4','PC5','PC6','PC7','PC8'], index**=**df2.columns)  df212\_loadings  Out[603]:   |  | **PC1** | **PC2** | **PC3** | **PC4** | **PC5** | **PC6** | **PC7** | **PC8** | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Population** | 0.002700 | -0.043558 | 0.395852 | 0.654739 | 0.401535 | 0.501453 | 0.004505 | 0.002792 | | **Children** | -0.018533 | 0.025965 | 0.536714 | -0.473968 | -0.434420 | 0.545441 | -0.003060 | -0.006132 | | **Age** | -0.002602 | -0.059468 | -0.392183 | 0.490729 | -0.722350 | 0.240724 | 0.148152 | 0.012143 | | **Income** | 0.039118 | -0.031777 | -0.629313 | -0.308772 | 0.340167 | 0.624674 | -0.011925 | 0.003427 | | **Outage\_sec\_perweek** | 0.041912 | 0.705365 | 0.009226 | -0.015226 | 0.073651 | 0.008280 | 0.703382 | 0.012129 | | **Tenure** | 0.703624 | -0.070936 | 0.021512 | 0.000981 | -0.014997 | -0.012662 | 0.042827 | -0.705123 | | **MonthlyCharge** | 0.058814 | 0.699698 | -0.063172 | 0.101305 | -0.081258 | 0.049658 | -0.693285 | -0.054759 | | **Bandwidth\_GB\_Year** | 0.705556 | -0.027094 | 0.029293 | -0.004544 | -0.017117 | -0.013342 | -0.025591 | 0.706719 |   data\_frame = pd.DataFrame(data\_frame, columns=['Population', 'Children', 'Age',  'Income', 'Outage\_sec\_perweek', 'Tenure', 'MonthlyCharge', 'Bandwidth\_GB\_Year'])  # Generate the correlation of the matrix for the components and a heatmap  cmatrix = data\_frame.corr()  plt.figure(figsize=(20, 10))  sb.heatmap(cmatrix, annot=True)  plt.show()    data\_frame = pd.DataFrame(data\_frame, columns=['Population', 'Children', 'Age',  'Income', 'Outage\_sec\_perweek', 'Tenure', 'MonthlyCharge', 'Bandwidth\_GB\_Year'])  df212\_loadings = pd.DataFrame(df212\_pca.components\_.T,  columns=['PC1','PC2','PC3','PC4','PC5','PC6','PC7','PC8'], index=df2.columns)  df212\_loadings   |  | **PC1** | **PC2** | **PC3** | **PC4** | **PC5** | **PC6** | **PC7** | **PC8** | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Population** | 0.002700 | -0.043558 | 0.395852 | 0.654739 | 0.401535 | 0.501453 | 0.004505 | 0.002792 | | **Children** | -0.018533 | 0.025965 | 0.536714 | -0.473968 | -0.434420 | 0.545441 | -0.003060 | -0.006132 | | **Age** | -0.002602 | -0.059468 | -0.392183 | 0.490729 | -0.722350 | 0.240724 | 0.148152 | 0.012143 | | **Income** | 0.039118 | -0.031777 | -0.629313 | -0.308772 | 0.340167 | 0.624674 | -0.011925 | 0.003427 | | **Outage\_sec\_perweek** | 0.041912 | 0.705365 | 0.009226 | -0.015226 | 0.073651 | 0.008280 | 0.703382 | 0.012129 | | **Tenure** | 0.703624 | -0.070936 | 0.021512 | 0.000981 | -0.014997 | -0.012662 | 0.042827 | -0.705123 | | **MonthlyCharge** | 0.058814 | 0.699698 | -0.063172 | 0.101305 | -0.081258 | 0.049658 | -0.693285 | -0.054759 | | **Bandwidth\_GB\_Year** | 0.705556 | -0.027094 | 0.029293 | -0.004544 | -0.017117 | -0.013342 | -0.025591 | 0.706719 |   # PCA projection  from sklearn.decomposition import PCA  pca = PCA(n\_components=8)  principalComponents = pca.fit\_transform(x)  principalDf = pd.DataFrame(data = principalComponents  , columns = ['PC1','PC2','PC3','PC4','PC5','PC6','PC7','PC8'])  principalDf.head()   |  | **PC1** | **PC2** | **PC3** | **PC4** | **PC5** | **PC6** | **PC7** | **PC8** | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **0** | -1.564672 | -0.143830 | -0.621166 | -1.455324 | -0.186409 | -0.723662 | 3.312535 | -1.035914 | | **1** | -1.711797 | 1.507400 | -0.144431 | 0.228661 | -1.054190 | -1.208228 | 0.133763 | -1.323574 | | **2** | -0.980731 | -0.663848 | 1.870836 | 0.765091 | -0.522566 | 0.122098 | 0.732410 | 0.994448 | | **3** | -1.997056 | -0.961632 | 1.934390 | -0.826545 | 1.697417 | 0.758413 | -0.230237 | 0.346571 | | **4** | -1.623467 | -0.188370 | -0.965085 | 0.881464 | 1.379075 | 2.226450 | -0.400568 | -0.154231 |   pca.explained\_variance\_ratio\_  array([0.21137699, 0.12707084, 0.11819191, 0.11382165, 0.11239538,  0.10639158, 0.10521735, 0.09431203])  8 PCAs = 90.34% , However, The rest of the PCA are contain the rest of the variance in the dataset. |

The first and second principal component is highly relevant to keep. After all, they contain most of the important data in this scenario because they have the highest percentage among the variables, followed by the second, third, fourth, fifth, sixth, seventh, and eighth variables which reflect an enormous correlation.

Therefore, the remaining principal components are contained in the remaining variance of the processed data set respectively.

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

***The description of how the organization can benefit from the results of the PCA is logical and accurate.***

The fundamental recognition of PCA is dimensionality (appearance) contraction. It advances algorithm execution by eliminating related appearance nevertheless there is some data shrinkage. However, Principal Component Analysis(PCA) is extensively well-used in minimization of great proportion by creating relevant grouping.

Therefore, We invariably extract tremendous dimensional statistics due to numeric as well as categorical data, and PCA is a helping hand in this scenario. That’s why it is vehement to take systematic consideration when cleaning data. Most importantly, these also concentrate principally according to how PCA is practically, comparatively than mathematical expression, and some relevant points position focal point that requires maximum consideration when performing PCA. diversely, the organization could ascertain erroneous data.

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.

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=bb4f1883-71d9-412f-9671-ae0200532a95>

**G.  Reference the web sources used to acquire segments of third-party code to support the application. Be sure the web sources are reliable.**

***The submission records all web sources used to acquire data or third-party code and all of the web sources are reliable.***

Deepanshu, B. (2020). *How to Import Data in Python.*

RSGB Business Consultant Pvt. Ltd. <https://www.listendata.com/2017/02/import-data-in-python.html#Import-CSV-files>

Carolina B. (2020). *Principal Component Analysis algorithm in Real-Life: Discovering patterns in a real-estate dataset.*

Medium. <https://towardsdatascience.com/principal-component-analysis-algorithm-in-real-life-discovering-patterns-in-a-real-estate-dataset-18134c57ffe7>

Pierre-Louis B. (2020). *Principle Components Analysis(PCA), Fundamentals, Benefits & Insights for Industry.*

Medium. <https://towardsdatascience.com/principal-components-analysis-pca-fundamentals-benefits-insights-for-industry-2f03ad18c4d7>

John S. (2018). *Data Cleaning with python and Pandas: Detecting Missing Values.*

Medium. <https://towardsdatascience.com/data-cleaning-with-python-and-pandas-detecting-missing-values-3e9c6ebcf78b>

Angelica Lo D. (2021). *How to detect outliers with Pytho pyod.*

Medium. <https://towardsdatascience.com/how-to-detect-outliers-with-python-pyod-aa7147359e4b>

Michael G. (2018). *Understanding Boxplots.*

Medium. <https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51>

**H.** **The submission includes in-text citations for sources that are properly quoted, paraphrased, or summarized and a reference list that accurately identifies the author, date, title, and source location as available.**

**References**

Larose, C. D., & Larose, D. T. (2019). Data science using Python and R. ISBN-13: 978-1-119-52684-1.