Scott Brackett

Sbrack5@wgu.edu

D212 Data Mining II – Task 2 Western Governor’s University

Table of Contents

[A. Part I: Research Question 2](#_Toc146217827)

[Define PCA Research Question 2](#_Toc146217828)

[Define PCA Goal 2](#_Toc146217829)

[B. Part II: Method Justification 3](#_Toc146217830)

[Explain PCA analysis and expected outcomes 3](#_Toc146217831)

[Summarize one assumption of PCA 5](#_Toc146217832)

[C. Part III: Data Preparation 5](#_Toc146217833)

[Identify continuous dataset variables to answer research question 5](#_Toc146217834)

[Standardize continuous dataset variables 10](#_Toc146217835)

[D. Part IV: Analysis 10](#_Toc146217836)

[Determine matrix of all principal components 10](#_Toc146217837)

[Identify the total number of principal components 12](#_Toc146217838)

[Identify variance of each principal component 13](#_Toc146217839)

[Identify total variance captured by the chosen principal components 13](#_Toc146217840)

[Summarize results of analysis 13](#_Toc146217841)

[E. Part V: Attachments 13](#_Toc146217842)

[Code citations 13](#_Toc146217843)

[Bibliography 14](#_Toc146217844)

# Part I: Research Question

## Define PCA Research Question

\*\* Propose **one** question relevant to a real-world organizational situation that you will answer by using principal component analysis (PCA).

To aid in strategic planning for our hospitals, the data analysis team sought to reduce their high dimensional data set down to a more manageable number of variables without losing a significant amount of data. To that end, we asked the question “Can I reduce the dimensionality of this data using PCA?”

## Define PCA Goal

\*\* Define **one** goal of the data analysis. Ensure that your goal is reasonable within the scope of the scenario and is represented in the available data.

Our primary goal was to reduce the dimensionality of the data, and obtain Principal Component variables that explain the variance of the dataset. If we were able to accomplish this goal, we would be in a good position to pass those components on to the next team to begin exploring trends in the data.

# Part II: Method Justification

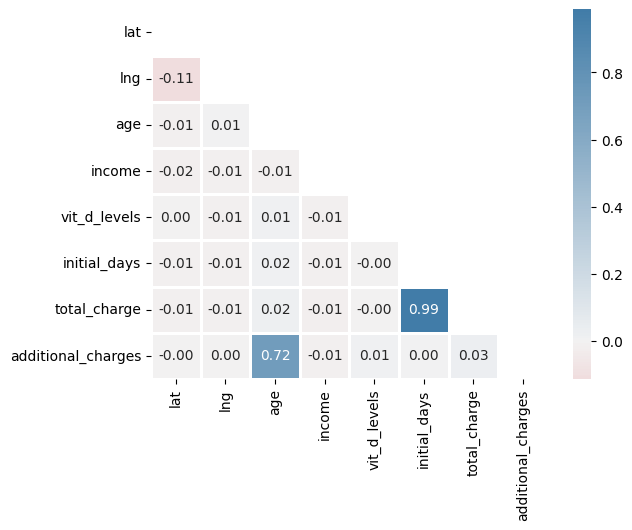
## Explain PCA analysis and expected outcomes

\*\*Explain how PCA analyzes the selected data set. Include expected outcomes.

Principal Component analysis is used to help reduce the dimensionality of datasets with many features (Keboola, 2022). Datasets with a large number of features may suffer “the curse of dimensionality” which results in models trained on that data being over-fitted and having low accuracy for new data that the model was not trained on.

Excluding the unique identification columns, there are 8 columns that have continuous variables and are therefore compatible with the PCA (Boeye, 2021).

By creating a Heat Map of the correlation between the variables, we can see that total\_charge and initial\_days have a correlation of .99. Additionally, we can see that additional\_charges and age have a correlation of .72 (see figure below).



If we use all of these variables for PCA, I would expect the outcome to be similar in terms of the amount of information that explains variance if we were to drop one variable from each of these strongly correlating pairs.

Regardless of initial correlation, I would also expect that we will be able to create principal components that would allow us to choose less than 8 columns from the resulting dataset and still explain a large portion of the variance in the data.

## Summarize one assumption of PCA

It is assumed that the data is scaled before performing PCA (Keboola, 2022). Because PCA compares the number ranges across variables, it is very important that the variables are represented in the same scale. It is standard practice to use a scaling function to convert all variables to scaled values between -1 and 1 so that they are comparable (Boeye, 2021).

# Part III: Data Preparation

## Identify continuous dataset variables to answer research question

To determine which variables would be appropriate for PCA, I manually created a csv file containing lookup data that defines each variable. I also used this same sheet to rename the variables to a more consistent naming convention.

Below is a table version of all the variables in the data:

|  |  |  |  |
| --- | --- | --- | --- |
| Variable Name | Standardized Name | Data Type | Sub Type |
| CaseOrder | case\_order | Qualitative | Ordinal |
| Customer\_id | customer\_id | Qualitative | Nominal |
| Interaction | interaction | Qualitative | Nominal |
| UID | uid | Qualitative | Nominal |
| City | city | Qualitative | Nominal |
| State | state | Qualitative | Nominal |
| County | county | Qualitative | Nominal |
| Zip | zip | Qualitative | Nominal |
| Lat | lat | Quantitative | Continuous |
| Lng | lng | Quantitative | Continuous |
| Population | population | Quantitative | Discrete |
| Area | area | Qualitative | Nominal |
| TimeZone | tiemzone | Qualitative | Nominal |
| Job | job | Qualitative | Nominal |
| Children | children | Quantitative | Discrete |
| Age | age | Quantitative | Continuous |
| Education | education | Qualitative | Nominal |
| Employment | employment | Qualitative | Nominal |
| Income | income | Quantitative | Continuous |
| Marital | martial | Qualitative | Nominal |
| Gender | gender | Qualitative | Nominal |
| ReAdmis | readmitted\_in\_30\_days | Qualitative | Nominal |
| VitD\_levels | vit\_d\_levels | Quantitative | Continuous |
| Doc\_visits | doc\_visits | Quantitative | Discrete |
| Full\_meals\_eaten | full\_meals\_eaten | Quantitative | Discrete |
| vitD\_supp | vit\_d\_supp | Quantitative | Discrete |
| Soft\_drink | soft\_drink | Qualitative | Nominal |
| Initial\_admin | initial\_admin | Qualitative | Nominal |
| HighBlood | high\_blood | Qualitative | Nominal |
| Stroke | stroke | Qualitative | Nominal |
| Complication\_risk | complication\_risk | Qualitative | Ordinal |
| Overweight | overweight | Qualitative | Nominal |
| Arthritis | arthritis | Qualitative | Nominal |
| Diabetes | diabetes | Qualitative | Nominal |
| Hyperlipidemia | hyperlipidemia | Qualitative | Nominal |
| BackPain | back\_pain | Qualitative | Nominal |
| Anxiety | anxiety | Qualitative | Nominal |
| Allergic\_rhinitis | allergic\_rhinitis | Qualitative | Nominal |
| Reflux\_esophagitis | reflux\_esophagitis | Qualitative | Nominal |
| Asthma | asthma | Qualitative | Nominal |
| Services | services | Qualitative | Nominal |
| Initial\_days | initial\_days | Quantitative | Continuous |
| TotalCharge | total\_charge | Quantitative | Continuous |
| Additional\_charges | additional\_charges | Quantitative | Continuous |
| Item1 | survey\_timely\_admission | Qualitative | Ordinal |
| Item2 | survey\_timely\_treatment | Qualitative | Ordinal |
| Item3 | survey\_timely\_visits | Qualitative | Ordinal |
| Item4 | survey\_reliability | Qualitative | Ordinal |
| Item5 | survey\_options | Qualitative | Ordinal |
| Item6 | survey\_hours\_of\_treatment | Qualitative | Ordinal |
| Item7 | survey\_courteous\_staff | Qualitative | Ordinal |
| Item8 | survey\_doctor\_active\_listens | Qualitative | Ordinal |

By using code to filter out any variables that had not been identified as having a sub type of “continuous” in the “var\_definitions.csv” file, I was able to easily create a separate dataframe with only those values. I performed the column renaming and the continuous column extraction in one step using the following code:

# fetch column mapping from var definition file

new\_col\_df = df\_var\_definitions.copy(deep=True)

var\_dict = new\_col\_df.set\_index('Variable Name')['Standardized Name'].to\_dict()

# rename columns using the mapping defined above

df\_init.rename(columns = var\_dict, inplace=True)

# create two lists of columns: one of all continuous variables and everything else

continuous\_columns = []

unmapped\_columns = []

# test and add columns to appropriate list based on stripped column name from var list

for column in df\_init.columns:

curr\_type = df\_var\_definitions.loc[df\_var\_definitions['Standardized Name'].str.strip() == column,'Sub Type']

try:

if curr\_type.size > 0:

if curr\_type.values[0] == 'Continuous':

continuous\_columns.append(column)

else:

unmapped\_columns.append(column)

except:

print('something went wrong')

# display type and number of continuous colunns found

df\_init[continuous\_columns].info()

Which resulted in the following output:

----------------------------------------

8 of 50 columns mapped successfully

---------------------------------------

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

RangeIndex: 10000 entries, 0 to 9999

Data columns (total 8 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 lat 10000 non-null float64

1 lng 10000 non-null float64

2 age 10000 non-null int64

3 income 10000 non-null float64

4 vit\_d\_levels 10000 non-null float64

5 initial\_days 10000 non-null float64

6 total\_charge 10000 non-null float64

7 additional\_charges 10000 non-null float64

dtypes: float64(7), int64(1)

memory usage: 625.1 KB

A table representation of the filtered variables is below:

|  |  |  |  |
| --- | --- | --- | --- |
| Variable Name | Standardized Name | Data Type | Sub Type |
| Lat | lat | Quantitative | Continuous |
| Lng | lng | Quantitative | Continuous |
| Age | age | Quantitative | Continuous |
| Income | income | Quantitative | Continuous |
| VitD\_levels | vit\_d\_levels | Quantitative | Continuous |
| Initial\_days | initial\_days | Quantitative | Continuous |
| TotalCharge | total\_charge | Quantitative | Continuous |
| Additional\_charges | additional\_charges | Quantitative | Continuous |

## Standardize continuous dataset variables

A cleaned and standardized version of the continuous dataset variables can be found in the root directory of the project in a file named “cleaned\_scaled\_medical\_continuous.csv”

# Part IV: Analysis

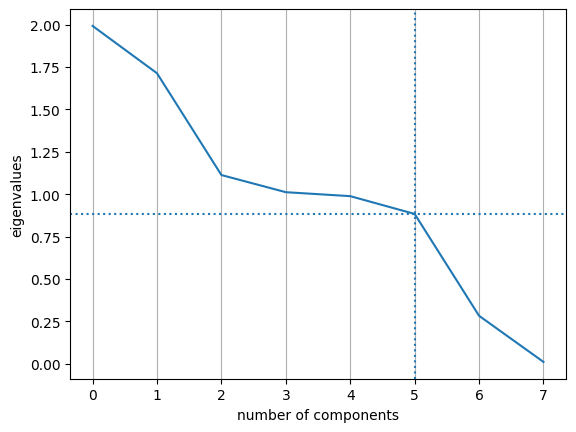
## Determine matrix of all principal components

Below is the loading matrix for all principal components:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PC** | **lat** | **lng** | **age** | **income** | **vit\_d\_levels** | **initial\_days** | **total\_charge** | **additional\_charges** |
| **PC 1** | -0.01308 | -0.010528 | 0.085809 | -0.02052 | -0.001636 | 0.70105 | 0.702184 | 0.086013 |
| **PC 2** | -0.00822 | 0.010934 | 0.701541 | -0.01869 | 0.019146 | -0.091424 | -0.080868 | 0.701455 |
| **PC 3** | 0.707443 | -0.698601 | 0.002471 | -0.08785 | 0.059987 | -0.00029 | -0.001272 | 0.012542 |
| **PC 4** | 0.010501 | -0.137959 | 0.017929 | 0.725834 | -0.673096 | 0.007525 | 0.005167 | 0.023631 |
| **PC 5** | -0.09376 | -0.114966 | -0.000156 | 0.661202 | 0.735289 | 0.008485 | 0.009042 | 0.000546 |
| **PC 6** | -0.70026 | -0.69241 | -0.003099 | -0.16578 | -0.048152 | -0.014659 | -0.013262 | -0.000853 |
| **PC 7** | 0.003858 | -0.006681 | 0.706716 | 0.002308 | -0.001932 | 0.031677 | -0.031469 | -0.706038 |
| **PC 8** | 0.001483 | -0.000357 | 0.026313 | 0.001312 | -0.001552 | -0.706274 | 0.706486 | -0.036819 |

## Identify the total number of principal components

There were 8 total principal components which we plotted against their eigenvalues below. The elbow method appeared to be difficult to determine with this dataset, and did not immediately suggest a clear number of components (see scree plot below).



However, by using the Kaiser Criteria, and only keeping components with an eigenvalue of one or greater, we were able to get a more definitive answer. Using the following code, I filtered for eigenvalues of one or greater:

eigen\_np\_array = np.array(eigenvalues)

eigen\_np\_array[eigen\_np\_array >= 1]

Which produced an array of the eigenvalues of the first four components:

[1.9922474 , 1.71333875, 1.11332099, 1.0121942 ]

Because the elbow plot was difficult to interpret, we chose to use the Kaiser Criteria for our Principal Component selection.

## Identify variance of each principal component

The explained variance of each Principal Component can be found in the table below.

|  | **PC1** | **PC2** | **PC3** | **PC4** | **CP5** | **PC6** | **PC7** | **PC8** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Variance** | 1.992447 | 1.713510 | 1.113432 | 1.012295 | 0.98896 | 0.884006 | 0.283633 | 0.011717 |
| **Variance Ratio** | 0.249056 | 0.214189 | 0.139179 | 0.126537 | 0.12362 | 0.110501 | 0.035454 | 0.001465 |

## Identify total variance captured by the chosen principal components

To obtain the combined variance explained by these principal components, we manually added up the bottom row of the table in the previous section:

# manually add up the first 4 PC's to get a value to compare for

# cumulative explained variance

explained\_variance\_df.transpose()['Variance Ratio'][0:4].sum()

Output: 0.7289605637513796

We also verified this value in code using the cumsum() function to pull that cumulative value from our model, as follows:

# show cumulative variance ratio for principal components

# which confirms that the 4 pc's selected by the Kaiser Criteria

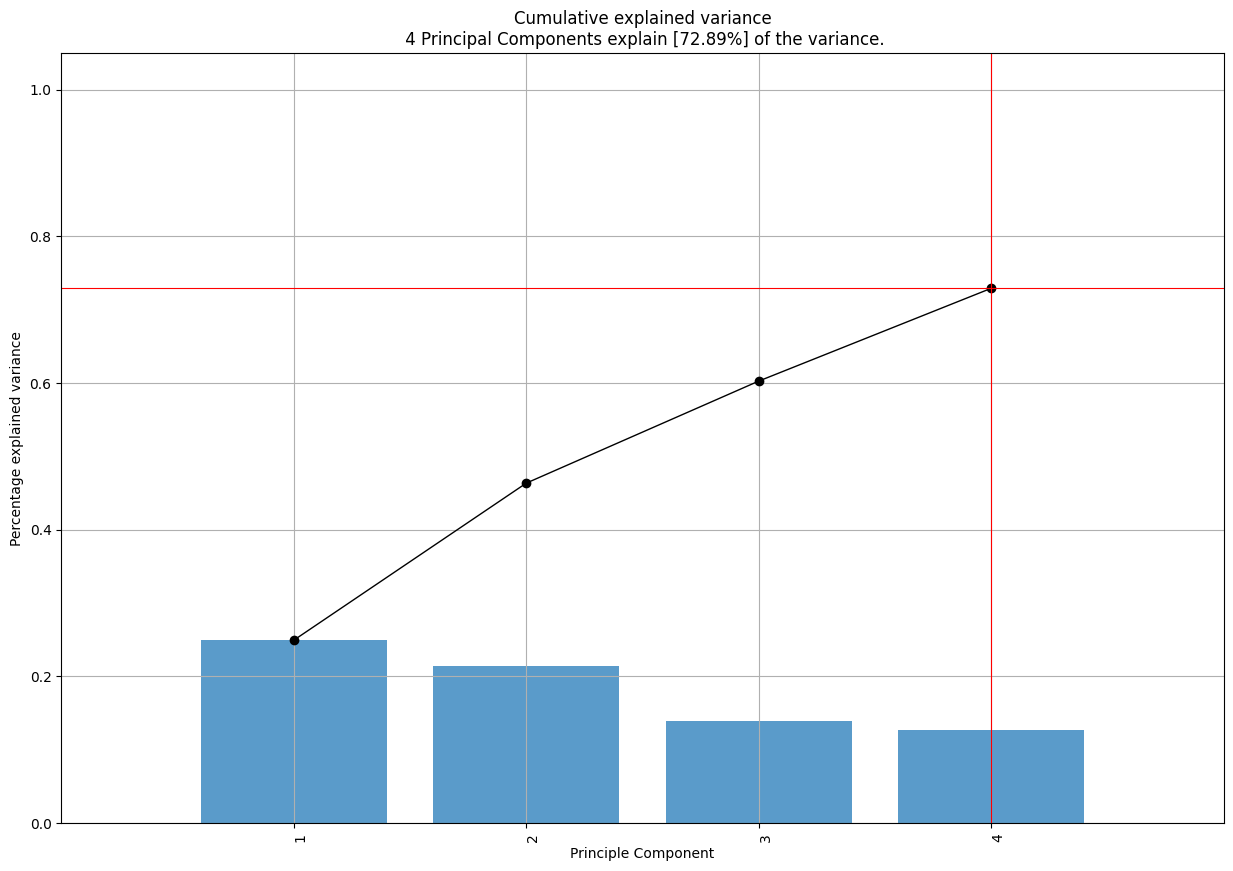
# will account for 72% of explained variance

cumulative\_var\_ratio = pca.explained\_variance\_ratio\_.cumsum()

cumulative\_var\_ratio[3]

Output: 0.7289605637513796

Lastly, using the built-in model.plot() method from the pca python package we plotted the cumulative explained variance, which agrees that the 4 components explain 72.89% of the variance in the dataset (see figure below):



## Summarize results of analysis

# Part V: Attachments

## Code citations

Middleton, K. M. (2022). Getting started with PCA [Slides]. Https://Westerngovernorsuniversity-My.Sharepoint.Com/. <https://westerngovernorsuniversity-my.sharepoint.com/personal/keiona_middleton_wgu_edu/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fkeiona%5Fmiddleton%5Fwgu%5Fedu%2FDocuments%2FDocuments%2FD206%2FUnderstanding%20PCA%2Epdf&parent=%2Fpersonal%2Fkeiona%5Fmiddleton%5Fwgu%5Fedu%2FDocuments%2FDocuments%2FD206&ga=1>

## Bibliography