**Part 1**

A1.

Research question: How can principal component analysis (PCA) be employed by a telecommunications company to identify key patterns or structures within customer data that contribute to variability to enable dimensionality reduction?

A2.

One goal of data analysis is to utilize principal component analysis (PCA) to identify underlying patterns and structures within the telecommunications company’s customer data that explain variability. By transforming high-dimensional data into principal components, PCA aims to identify characteristics driving variability. Reduction in the dataset’s dimensionality simplifies a complex dataset while preserving the most important features, leading to a clearer understanding of the underlying patterns and structures within the dataset.

**Part 2**

B1.

Principal component analysis (PCA) is a statistical technique used to reduce the dimensionality of a dataset while preserving its essential features. For this dataset, PCA aims to pinpoint the most significant variables or characteristics that drive variance in telecommunications customers’ behaviors and preferences.

To initiate analysis, exploratory data analysis is conducted to ensure there are no missing values and only relevant variables are selected and kept into the dataset. Before, performing PCA, it is essential to standardize the variables to ensure that they are on the same scale. This step ensures that all variables have a mean of 0 and a standard deviation of 1. This standardization step prepares the data for comparison and analysis. PCA then examines how variables in the dataset are related to each other. By understanding these relationships, PCA can later adjust the data to reduce redundancy and focus on the most essential information. Next, variance is broken down into vectors that represent the different directions in which data varies the most, and values that indicate the importance or significant of these directions. These vectors and values help PCA find the most meaningful directions or patterns in the data, discarding features with low variance. The vectors are ranked based on their corresponding values, with higher values indicating the principal components that capture the most variance in the data. Typically, only the vectors with the highest values, or principal components, are retained for further analysis.

PCA streamlines the dataset’s dimensionality by transforming the original variables into a new set of variables, which are linear combinations of the original ones. This transformation involves rotating and aligning the samples to new coordinate axes defined by the principal components. Each principal component represents a distinct pattern or characteristic within the data.

It is anticipated that PCA will identify the principal components that explain most of the variance in the dataset, which represent underlying patterns. By preserving only, the most significant principal components, PCA simplifies data interpretation and visualization.

B2.

One assumption of PCA is that the variables are linearly related to each other. This means that PCA works best when the relationships between variables can be represented by straight lines. If the relationships are non-linear then, PCA might not accurately capture the underlying structure of the data.

In text citations:

(“Visualizing the PCA transformation”, n.d.)

(“Dimension reduction with PCA”, n.d.)

(“Principal component analysis”, n.d.)

**Part 3**

C1.

Continuous data set variables:

selected\_columns = ['Population','Income', 'Email', 'Tenure', 'Children', 'Age', 'Outage\_sec\_perweek', 'Contacts', 'Yearly\_equip\_failure', 'MonthlyCharge','Bandwidth\_GB\_Year', 'Item1', 'Item2', 'Item3', 'Item4', 'Item5', 'Item6', 'Item7', 'Item8']

C2.

Please see code below and attached to view the standardized continuous dataset.

# Standardizing the data

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

scaled\_churn\_PCA\_data = scaler.fit\_transform(churn\_clean\_PCA)

scaled\_churn\_PCA\_df = pd.DataFrame(scaled\_churn\_PCA\_data, columns=churn\_clean\_PCA.columns)

print(scaled\_churn\_PCA\_df)

In text citations:

(“PCA applications”, n.d.)

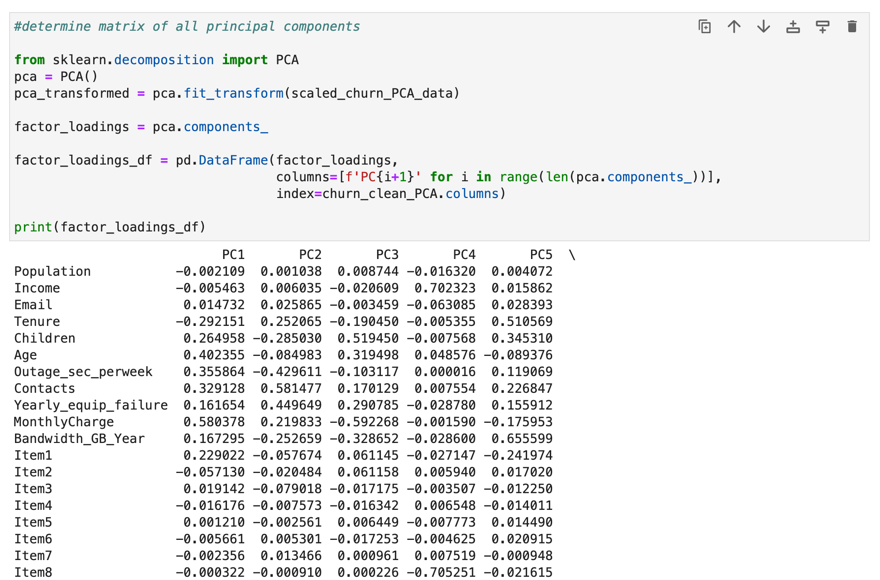
(“Dimension reduction with PCA”, n.d.)

**Part 4**

D1.

The matrix of all the principal components:

A table of numbers and text

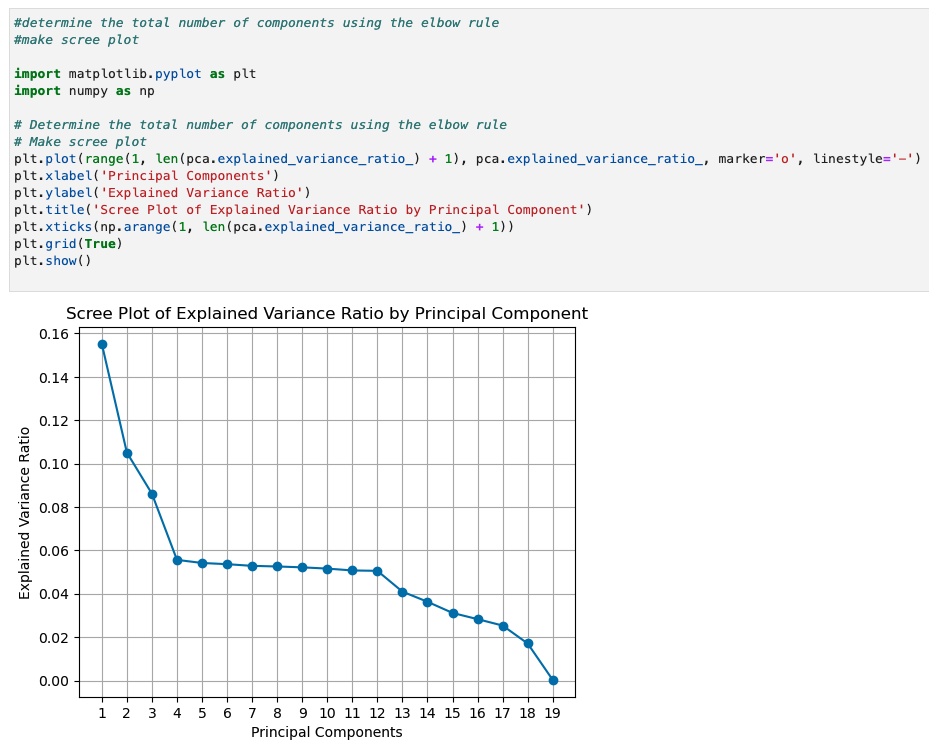
Description automatically generated

A table of numbers and text

Description automatically generated

D2.

Using the elbow rule, it appears that 4 principal components should be retained.



D3.

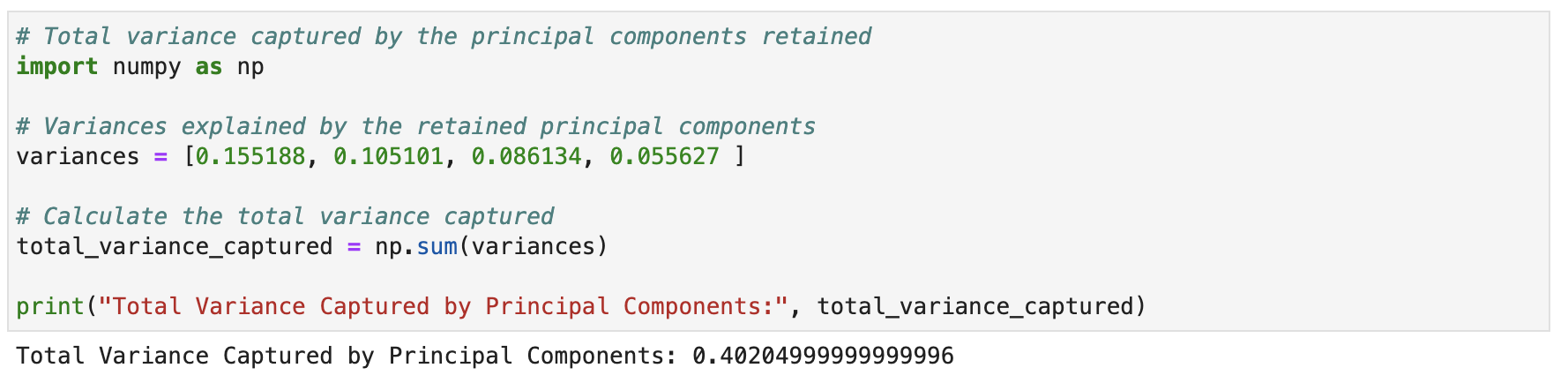
The variance of each of the principal components:

A screenshot of a computer code

Description automatically generated

D4.

Total variance captured by the principal components:



D5.

The data analysis results provide valuable insights into the structure and variability of the dataset. The matrix of all principal components offers a transformed representation of the data, where each row corresponds to an observation and each column represents a principal component. In PCA, these components are linear combinations of the original variables, aiming to capture the maximum variance in the data. Each element in the matrix represents the loading of the corresponding variable onto the respective principal component. This matrix sheds light on which variables contribute most to each principal component, providing insights into the underlying structure of the data. It also facilitates the transformation of the original dataset into a new set of variables, the principal components, that capture most of the variability present in the data.

To determine the number of principal components to retain, I applied the elbow rule. This method observes the explained variance curve, or scree plot, which displays the variance explained by each principal component. The goal is to identify a point on the plot where there is a significant change in slope, known as the elbow. This point indicates the number of principal components where additional components contribute less to the overall variance explained. By observing the scree plot I identified the elbow point at approximately 4 principal components. This suggests that the first few components capture a substantial amount of variance in the data while additional components contribute less. Although there is another slight peak around the 12th principal component, I chose to retain 4 principal components. These 4 principal components explain a higher portion of the total variance and adding more components beyond the elbow point may introduce unnecessary complexity to the analysis without significantly increasing the variance explained. By keeping 4 principal components, I capture a significant amount of the variance in the data while reducing dimensionality.

The analysis identified four principal components (PC1, PC2, PC3, and PC4) and their respective explained variances: PC1 explain 15%, PC2 explains 10%, PC3 explains 8%, and PC4 explains 5% of the total variance in the dataset. These values represent the proportion of variability captured by each principal component, providing insight into the significance of each component in explaining the dataset’s variability.

After dimensionality reduction using PCA, the retained components explain a total variance of 0.402 units with 4 components. In comparison, the original dataset has a total variance of 19.0 units explained by 19 components. By comparing these values, it suggests that a relatively small proportion of the original variability has been retained through dimensionality reduction with 4 principal components. Although, this proportion may be considered low, this analysis can still enhance interpretability, making it a favorable choice for deriving meaningful insights from the data.

In text citations:

(“Principal component analysis”, n.d.)

(“Lesson7: How to Perform PCA in Python”, n.d.)

**Part 6**

D. Citations for code:

DataCamp. (n.d.). Principal component analysis [Video file]. Retrieved from https://campus.datacamp.com/courses/dimensionality-reduction-in-python/feature-extraction?ex=5

DataCamp. (n.d.). Dimension reduction with PCA [Video file]. Retrieved from https://campus.datacamp.com/courses/unsupervised-learning-in-python/decorrelating-your-data-and-dimension-reduction?ex=9

DataCamp. (n.d.). PCA applications [Video file]. Retrieved from https://campus.datacamp.com/courses/dimensionality-reduction-in-python/feature-extraction?ex=9

Lesson7: How to Perform PCA in Python. Retrived from https://cgp-oex.wgu.edu/courses/course- v1:WGUx+OEX0026+v02/courseware/1f468770545f494fa657b4dc0ed3762f/2b5f23c5dad64357b35272 8993788677/6?activate\_block\_id=block- v1%3AWGUx%2BOEX0026%2Bv02%2Btype%40vertical%2Bblock%403f0422d47d8b4a3eaec25de3bced8 bf8

DataCamp. (n.d.). Visualizing the PCA transformation [Video file]. Retrieved from https://campus.datacamp.com/courses/unsupervised-learning-in-python/decorrelating-your-data-and-dimension-reduction?ex=1

E. Citations for content:

DataCamp. (n.d.). Principal component analysis [Video file]. Retrieved from https://campus.datacamp.com/courses/dimensionality-reduction-in-python/feature-extraction?ex=5

DataCamp. (n.d.). Dimension reduction with PCA [Video file]. Retrieved from https://campus.datacamp.com/courses/unsupervised-learning-in-python/decorrelating-your-data-and-dimension-reduction?ex=9

DataCamp. (n.d.). PCA applications [Video file]. Retrieved from https://campus.datacamp.com/courses/dimensionality-reduction-in-python/feature-extraction?ex=9

Lesson7: How to Perform PCA in Python. Retrived from https://cgp-oex.wgu.edu/courses/course- v1:WGUx+OEX0026+v02/courseware/1f468770545f494fa657b4dc0ed3762f/2b5f23c5dad64357b35272 8993788677/6?activate\_block\_id=block- v1%3AWGUx%2BOEX0026%2Bv02%2Btype%40vertical%2Bblock%403f0422d47d8b4a3eaec25de3bced8 bf8

DataCamp. (n.d.). Visualizing the PCA transformation [Video file]. Retrieved from https://campus.datacamp.com/courses/unsupervised-learning-in-python/decorrelating-your-data-and-dimension-reduction?ex=1