National Institute of Diabetes and Digestive and Kidney Diseases

Dimensionality Reduction in Python

Emma Brown, Booz Allen Hamilton

December 22, 2023

Speaker Introduction

Emma Brown is a Health Data Scientist at Booz Allen Hamilton, where she focuses on education of machine learning and data science applications to aspects of healthcare spanning medical image processing to population health.

She has supported the NIH Office of Data Science Strategy in creating a cloud-based data repository and analytics platform for public health researchers.

Prior to joining Booz Allen Hamilton, Emma conducted research on clinical neuroimaging of TBI and PTSD in post-9/11 service members and veterans to characterize deployment-related pathologies (Translational Research Center for TBI and Stress Disorders, VA Boston Healthcare System).

Data Centric Challenge – Submission Requirements

1. **A single "raw" dataset** A single "Raw" dataset resulting from data aggregation and harmonization of all study data files (from TEDDY or TrialNet – as per level of participation), but has not otherwise been altered. This dataset must be represented as a single rectangular file (i.e., tabular, spreadsheet, or matrix) in .csv file format
2. **An "AI-ready" version** of the raw dataset that has been enhanced for AI-readiness. This dataset must be represented as a single rectangular file (i.e., tabular, spreadsheet, or matrix) in .csv file format
3. **The code script,** in Python or R, used to generate the raw and AI-ready files submitted to your private GitHub repository
4. **A human-readable data dictionary/codebook** documenting the AI-ready dataset (Excel format preferred, with the following information included at a minimum: variable name, variable label/description, variable type, measurement unit as applicable (e.g., pounds, kilograms), and corresponding code lists as needed (e.g., 0 = No, 1 = Yes).
5. **Challenge Solution Submission Form**, describing the 1) AI-ready dataset, 2) methods for preparing the AI-ready dataset, and 3) potential use cases for the prepared dataset as it relates to T1D, or other disease areas of interest to NIDDK.

Submission Instructions are posted on Challenge.gov under the How to Enter tab for Phase 2: Data Enhancement

Frequently Asked Questions are also posted to Challenge.gov under the FAQs tab

National Institute of
Diabetes and Digestive
and Kidney Diseases

Agenda

Dimensionality Reduction Overview

Advantages and Limitations

Principal Component Analysis

Python Demo

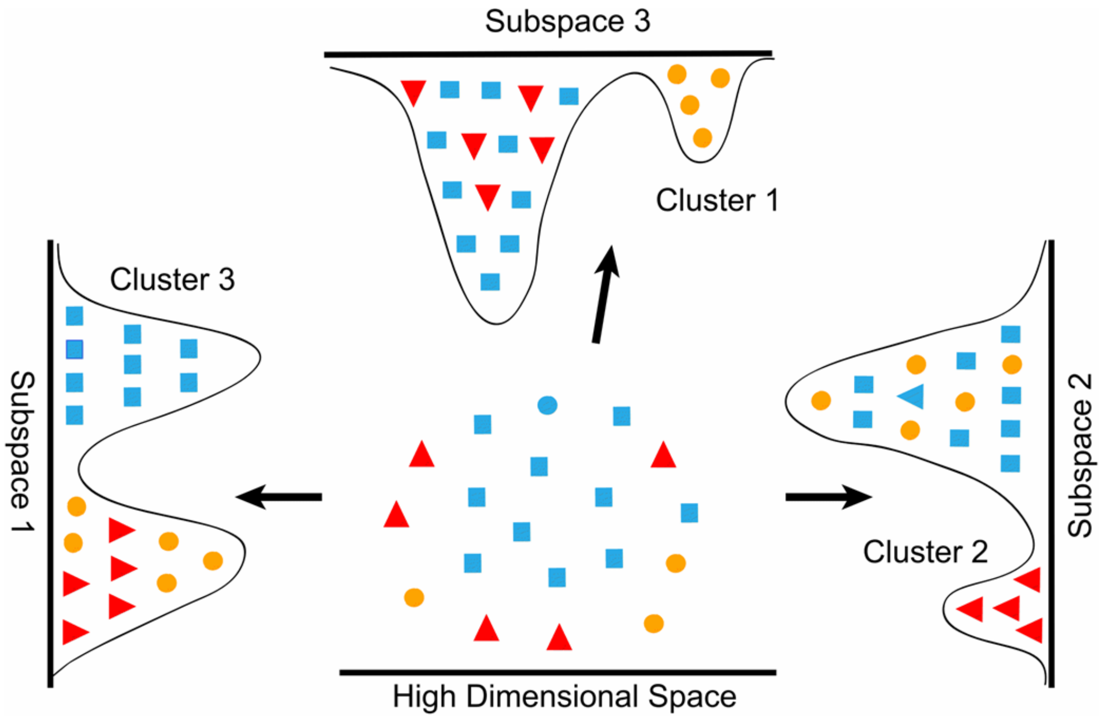
Resources

Dimensionality Reduction Overview

What is Dimensionality Reduction?

A technique to reduce the number of features in a dataset while retaining important information

- The Curse of Dimensionality: Use more information to improve model accuracy, but more features leads to more dimensions and configurations
- Overfitting: Including too many features in a model will decrease generalizability
- Dimensionality reduction can reduce model complexity and elucidate trends
- Two classes of dimensionality reduction: Feature selection and Feature extraction



Feature Selection vs Feature Extraction

Feature Selection

- Selects a subset of features that are most relevant thereby reducing dimensionality
- **Some methods include:**
  - *Filter methods:* Ranks features based on statistical property-based relevance
    - Example: R^2, χ^2
  - *Wrapper methods:* Use model performance as criteria for feature selection
    - Example: Recursive feature elimination
  - *Embedded methods:* Combine feature selection with the model training
    - Example: Least Absolute Shrinkage Selection Operator (LASSO)

Feature Extraction

- Creates a new set of features by combining or transforming original features to capture the essence of the original data in a lower-dimensional space
- **Some methods include:**
  - *Principal component analysis (PCA):* Transforms data into a new coordinate system
  - *Linear discriminant analysis (LDA):* Finds a linear combination of features to characterize objects
  - *T-distributed stochastic neighbor embedding (t-SNE):* Visualizes datapoints in a two or three-dimensional map

Dimensionality Reduction Advantages

Data compression	Reduce taxing computational resources
Data preprocessing	Dimensionality reduction can be applied before a Machine Learning (ML) model
Removes data redundancy	Multiple highly correlated features can be removed
Improve data visualization	More interpretable trends in lower dimensional data
Prevents overfitting	Fewer dimensions reduces the likelihood of overfitting, and the model will generalize more easily to new data
Feature engineering	Limiting a dataset to important features can be useful for ML models
Reduce complexity	Improve performance of ML models by reducing complexity and noise

Dimensionality Reduction in Everyday Life

Image compression	Reducing the size of an image file while preserving essential features
Medical Imaging	Reduce dimensionality of MRI/CT scans; extracting relevant features (e.g., identifying tumors)
Market Research	Understanding correlations and patterns among consumer behaviors and preferences
Outbreak Detection	Epidemiological data can be used to track the spread of infectious diseases
Social Sciences	Analyzing survey responses to identify underlying factors
Supply Chain Management	Optimize inventory management to ensure supplies are available while minimizing waste

Dimensionality Reduction Limitations

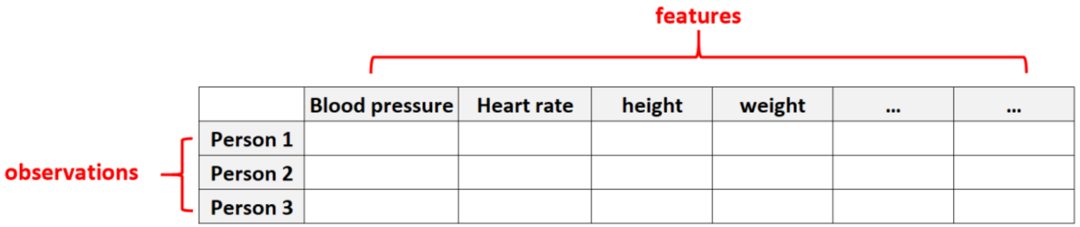
Data loss	Discarding some features can result in less interpretable data
Non-linearity	Some methods for dimensional reduction are not ideal for complex, non-linear data
Interpretability	Reduced dimensions may not be easily interpretable, and it may be difficult to understand the relationship between original features and reduced dimensions
Overfitting	Some methods may lead to overfitting, especially when the number of components is chosen based on the training data
Sensitivity to outliers	Techniques are sensitive to outliers which can lead to a biased representation
Computational complexity	Some techniques are computationally intensive, especially when dealing with large non-linear datasets
Algorithm selection	Choosing the best dimensionality reduction technique can be challenging, it is important to understand the dataset's features

Principal Component Analysis (PCA)

Principal Component Analysis

An unsupervised feature extraction method that creates a new set of features while preserving the essence of the original dataset

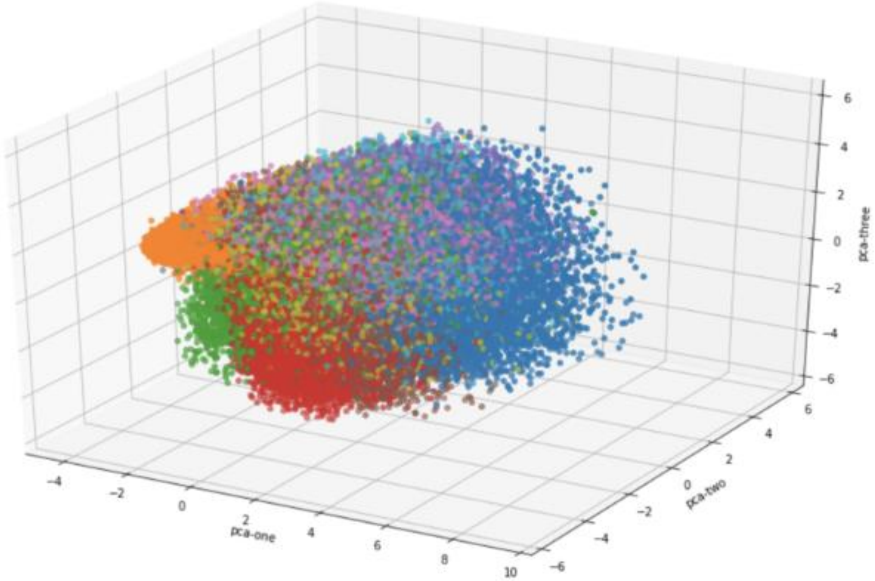
- Linear relationships are three-dimensional with unseen features impacting relationships and outcomes
- Healthcare datasets often have more features than observations
  - Example: What determines the length of a patient’s stay in the hospital?
    - Severity/complexity of condition
    - Age
    - BMI
    - And more!
- PCA will create a low-dimensional representation of a dataset
- PCA should only apply to *continuous* variables



features

observations

	Blood pressure	Heart rate	height	weight	…	…
Person 1						
Person 2						
Person 3						



PCA Step One: Clean and Standardize Data

Standardize the continuous variables so that they contribute equally

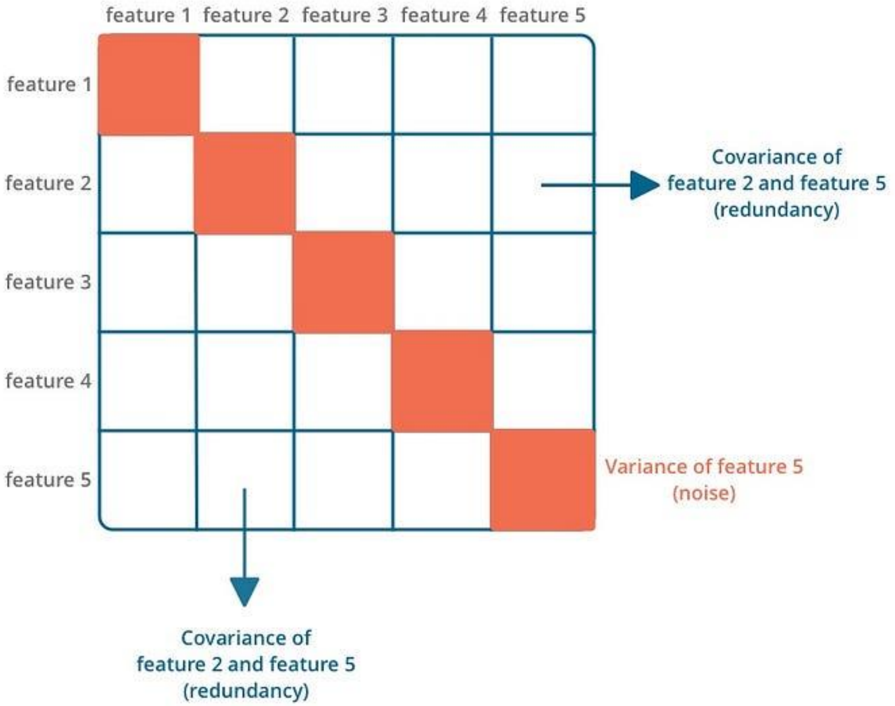
- PCA is sensitive to variance
- Large differences between the ranges of variables will impact results
- Identify potential outliers in the dataset
- Normalize by subtracting the mean and dividing by the standard deviation for each value of each variable to get a *z*-score
- Example: Height in inches, weight in pounds

$$\boxed{normalized\ value = \frac{value - feature\ mean}{feature\ standard\ deviation}}$$

PCA Step Two: Compute Covariance Matrix

Create a table that summarizes the relationships between all possible feature pairs

- A symmetric matrix of covariances
- Highly correlated pairs are redundant
- Covariance = joint variability of two variables
- Correlation = extent to which variables increase/decrease in together
- Positive covariance = moving in the same direction (*correlated*)
- Negative covariance = moving in opposite direction (*inversely correlated*)



Covariance of feature 2 and feature 5 (redundancy)

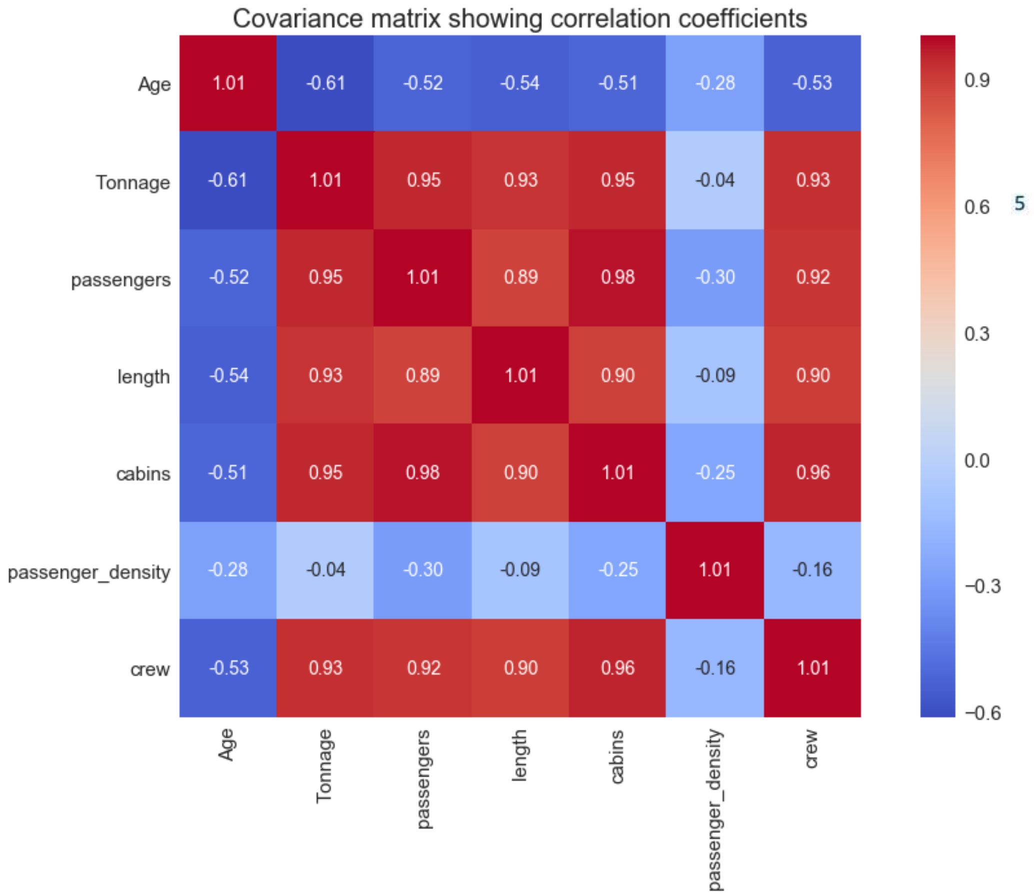
Variance of feature 5 (noise)

Covariance of feature 2 and feature 5 (redundancy)

PCA Step Two: Compute Covariance Matrix

Create a table that summarizes the relationships between all possible feature pairs

- A symmetric matrix of covariances
- Highly correlated pairs are redundant
- Covariance = joint variability of two variables
- Correlation = extent to which variables increase/decrease in together
- Positive covariance = moving in the same direction (*correlated*)
- Negative covariance = moving in opposite direction (*inversely correlated*)



Covariance matrix showing correlation coefficients

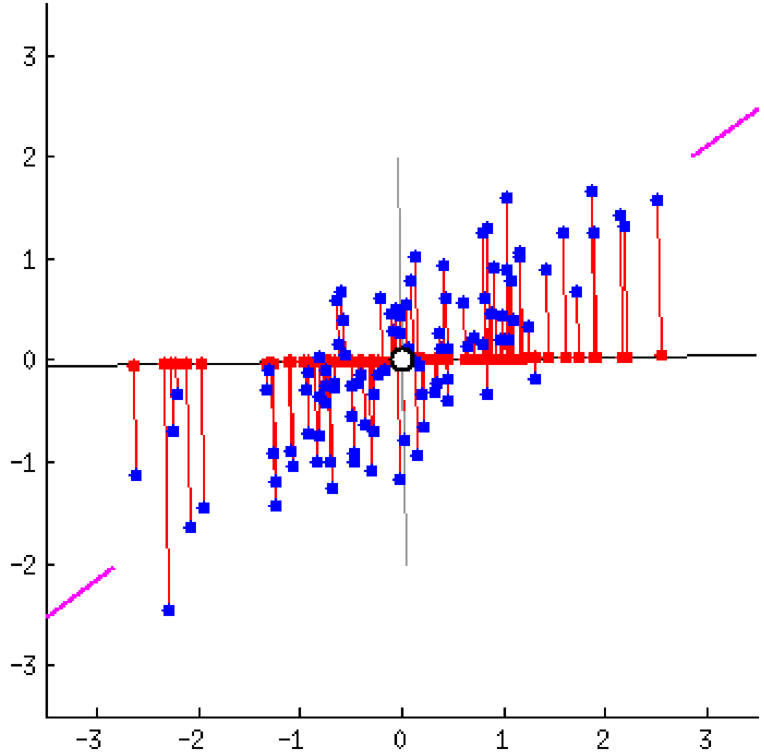
	Age	Tonnage	passengers	length	cabins	passenger_density	crew
Age	1.01	-0.61	-0.52	-0.54	-0.51	-0.28	-0.53
Tonnage	-0.61	1.01	0.95	0.93	0.95	-0.04	0.93
passengers	-0.52	0.95	1.01	0.89	0.98	-0.30	0.92
length	-0.54	0.93	0.89	1.01	0.90	-0.09	0.90
cabins	-0.51	0.95	0.98	0.90	1.01	-0.25	0.96
passenger_density	-0.28	-0.04	-0.30	-0.09	-0.25	1.01	-0.16
crew	-0.53	0.93	0.92	0.90	0.96	-0.16	1.01

PCA Step Three: Eigendecomposition

Compress the data into uncorrelated components

$$Y_1 = a_{11}X_1 + a_{12}X_2 + \ldots a_{1p}X_p$$
$$Y_2 = a_{11}X_1 + a_{12}X_2 + \ldots a_{1p}X_p$$
$$\ldots$$
$$Y_p = a_{11}X_1 + a_{12}X_2 + \ldots a_{1p}X_p$$

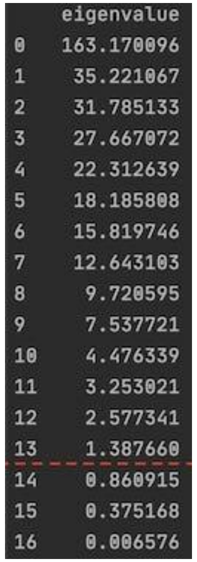
- Principal components: New variables constructed as linear combinations of original variables
  - Uncorrelated
  - Maximize variance
  - Orthogonal
- Eigenvectors: Directions of the maximum variance made up of loading scores from each feature
- Eigenvalues: Coefficients of variance (*importance*) carried by each component
- Rank eigenvectors in order of their eigenvalues



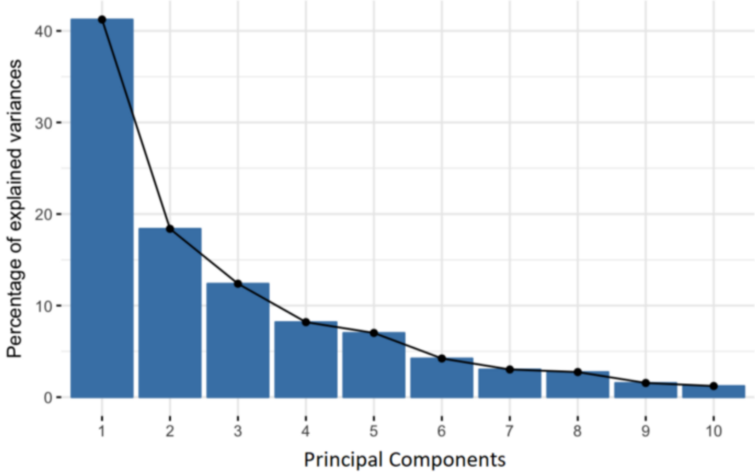
PCA Step Four: Select the Principal Components

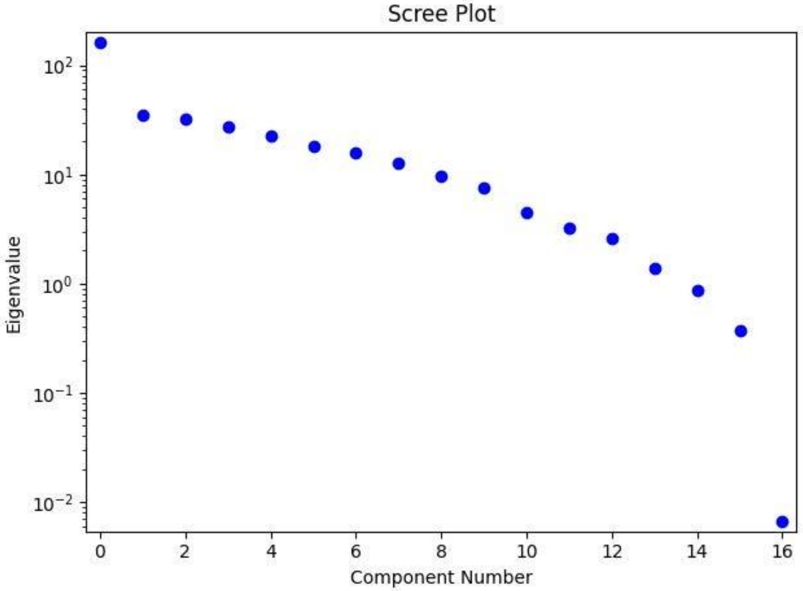
Pick the components that explain the most variance based on one of the following methods

Kaiser criterion	Explained variance	Scree plot
Only select PCs that have an eigenvalue greater than 1	Retain enough components for the cumulative variance to capture 80-90%	Plot components by eigenvalue and find the ‘elbow’



```
     eigenvalue
0    163.170096
1     35.221067
2     31.785133
3     27.667072
4     22.312639
5     18.185808
6     15.819746
7     12.643103
8      9.720595
9      7.537721
10     4.476339
11     3.253021
12     2.577341
13     1.387660
-----------------
14     0.860915
15     0.375168
16     0.006576
```



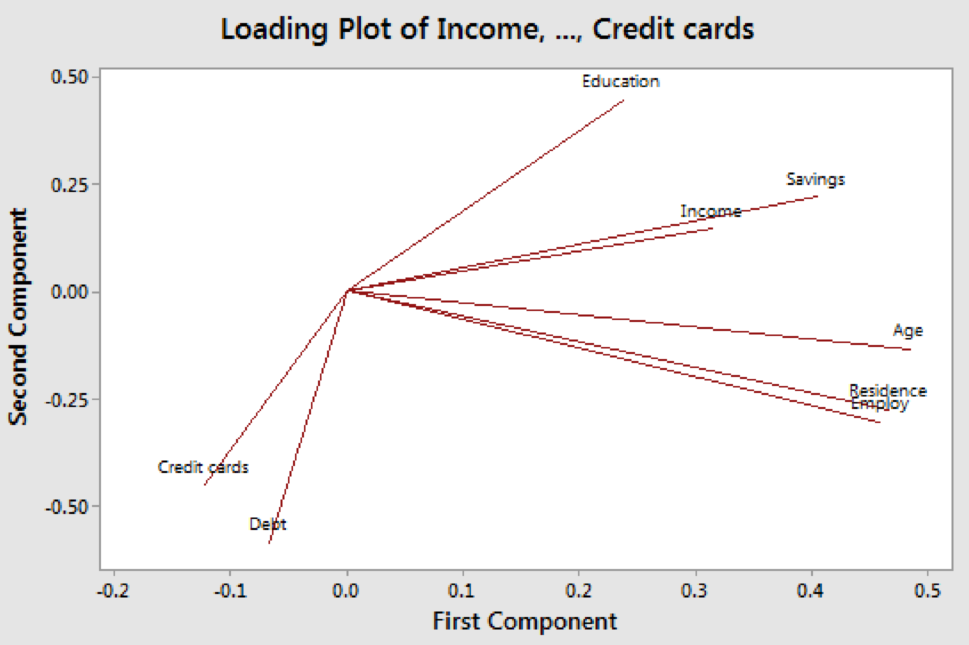


PCA Step Five: Visualize and interpret loadings

Project components to observe underlying trends in the data

- Every datapoint in the original dataset has values along the new data direction
- Each variable has correlations, or *loadings*, with the components
- Loadings can help to identify features underlying the components and bigger trends in the data
- Large loadings indicate strength of relationship to a component
- Sign of loading indicates positive or negative correlation with a component

$$Y_1 = a_{11}X_1 + a_{12}X_2 + \dots a_{1p}X_p$$
$$Y_2 = a_{11}X_1 + a_{12}X_2 + \dots a_{1p}X_p$$
$$\dots$$
$$Y_p = a_{11}X_1 + a_{12}X_2 + \dots a_{1p}X_p$$



Demo in Python

Packages and Functions for PCA

Python	scikit-learn, numpy, pandas, matplotlib
R	prcomp, PCA, FactoMineR, ade4, ExPosition
SAS	PRCOMP, PCA
C++	ALGLIB, libpca

Resources

Resources

Further reading:

- Introduction to Dimensionality Reduction
- Statistical analysis of high-dimensional biomedical data:
  a gentle introduction to analytical goals, common approaches and challenges

Step-wise tutorials:

- Principal Component Analysis (PCA) Explained
- Principal Component Analysis (PCA) in R Tutorial
- PCA Tutorial in Python

Questions