Applied Statistics - Notes - v0.1.0

260236

March 2025

Preface

Every theory section in these notes has been taken from two sources:

- The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition. [2]
- An Introduction to Statistical Learning: with Applications in Python. [3]
- Applied Multivariate Statistical Analysis. [4]
- Course slides. [5]

About:

GitHub repository

These notes are an unofficial resource and shouldn't replace the course material or any other book on applied statistics. It is not made for commercial purposes. I've made the following notes to help me improve my knowledge and maybe it can be helpful for everyone.

As I have highlighted, a student should choose the teacher's material or a book on the topic. These notes can only be a helpful material.

Contents

1 Business Data Analytics

1.1 Multivariate Descriptive Statistics

When we move from **analyzing** a single variable (univariate analysis) to **multiple variables at once**, we enter the realm of **Multivariate (MV) analysis**. A natural question arises: *Is multivariate analysis just a replication of univariate analysis across several variables?*

The answer is no, multivariate analysis introduces new and fundamental questions that cannot be answered by simply analyzing variables individually. The **core focus** shifts to **understanding how these variables interact with each other**. Specifically, we are concerned with the **dependence** and **correlation between variables**.

Covariance: Measuring Joint Variability

To capture how two variables vary together, we use **Covariance**. The **Sample Covariance** between variables x_j and x_k is calculated as:

$$\text{cov}_{jk} = s_{jk} = \frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k) \tag{1}$$

- $s_{jk} = 0 \Rightarrow$ implies that there is **no linear relationship** between the two variables.
- $s_{jk} > 0 \Rightarrow$ suggests that **as one variable increases**, the **other tends to increase**.
- $s_{jk} < 0 \Rightarrow$ one **variable** tends to **decrease when the other increases**.

Covariance Is Not Standardized

The **value of covariance is not standardized**, it **depends on the units of measurement**, which makes comparisons difficult. For example:

- Suppose we're measuring
  - Height in centimeters
  - Weight in kilograms
- The covariance between height and weight will be expressed in *centimeter-kilogram* units.

Now imagine we convert height to meters. The covariance value changes, because now you're multiplying meters × kilograms instead of centimeters × kilograms. Even though the **relationship between height and weight hasn't changed**, the **numerical value of covariance does change due to this unit change**. **Because of unit dependency, it's hard to compare covariances between different variable pairs**. Finally, it is **hard to interpret the magnitude of covariance in any absolute sense** (e.g., is 50 a large covariance or small? It depends on the units!).

Correlation: Standardized Covariance

To standardize covariance and **measure the strength of a linear relationship** on a scale between -1 and 1, we use the **Correlation** coefficient, defined as:

$$\text{cor}_{jk} = r_{jk} = \frac{s_{jk}}{\sqrt{s_{jj}}\sqrt{s_{kk}}} = \frac{\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)}{\sqrt{\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2}\sqrt{\sum_{i=1}^{n}(x_{ik} - \bar{x}_k)^2}} \tag{2}$$

This formula divides the covariance by the product of the standard deviations of the two variables, giving a **unitless value**:

- $r = 0$: No linear correlation
- $r > 0$: Positive correlation (**both variables increase or decrease together**)
- $r < 0$: Negative correlation (**one increases while the other decreases**)
- $|r| = 1$: Perfect correlation (**exact linear dependence**)

Thus, correlation not only **reveals the direction of the relationship** but also its **strength**.

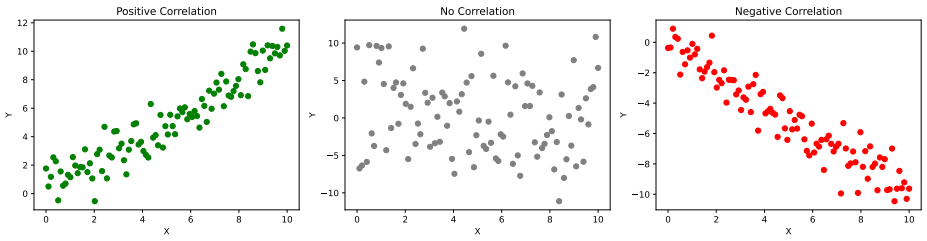

Figure 1: Direction of correlation: positive (left), none (center), negative (right).

Describing MV Data: Vector and Matrices

When analyzing multivariate data:

- We compute the **vector of Sample Means**:

$$\bar{\mathbf{X}} = [\bar{X}_1, \bar{X}_2, \ldots, \bar{X}_p] \tag{3}$$

- And the **Variance-Covariance matrix S**, which **summarizes the covariances between all pairs of variables**:

$$S = \begin{bmatrix} s_{11} & \cdots & s_{1p} \\ \vdots & \ddots & \vdots \\ s_{p1} & \cdots & s_{pp} \end{bmatrix} \tag{4}$$

- Alternatively, we can use the **Correlation matrix R**, where **all diagonal elements are 1** (because each variable is perfectly correlated with itself) and **off-diagonal elements are correlation coefficients**:

$$R = \begin{bmatrix} 1 & \cdots & r_{1p} \\ \vdots & \ddots & \vdots \\ r_{p1} & \cdots & 1 \end{bmatrix} \tag{5}$$

Scatterplots - Visualizing Variable Pairs

One of the most intuitive and widely used tools in multivariate analysis is the **2D Scatterplot**. Each plot shows how two variables relate to each other:

- ✔ Clusters or linear trends can indicate correlation or dependence.
- ✔ Scatterplots are **ideal for spotting positive, negative, or no correlation** visually.

However, scatterplots have a **limitation**: they **only** allow us to **analyze two variables at a time**. When dealing with many variables, the **number of possible pairings becomes large**, making it **difficult to read or interpret** the scatterplots individually. This is where quantitative measures (like correlation matrices) and higher-dimensional graphics come into play.

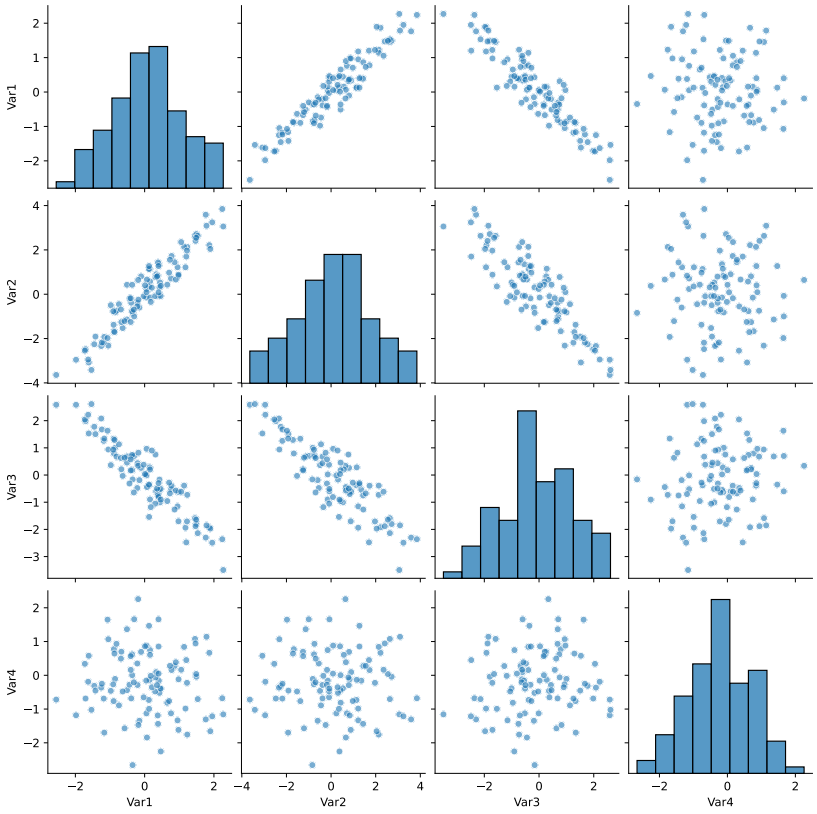

Figure 2: Scatterplot matrix of four variables. This scatterplot matrix displays all pairwise relationships among four variables:

- Diagonal plots (top-left to bottom-right): Histograms showing the distribution of each variable.
- Off-diagonal plots: 2D scatterplots illustrating the relationship between each pair of variables.

Rotated Plots in 3D - Capturing Complexity

When dealing with **three variables**, we can **extend scatterplots into 3D space** using **Rotated plots**. These visualizations allow us to:

- ✔ **Explore interdependencies** among three variables at once.
- ✔ **Gain spatial insight** into how data points spread in three-dimensional space.
- ✔ Observe **complex patterns that are invisible in 2D**.

Yet again, as we move **beyond three variables**, visualizing becomes **impractical**, our brains cannot easily comprehend 4D or higher dimensions. Hence, dimensionality reduction techniques like PCA are often used alongside visualizations to make high-dimensional data "digestible".

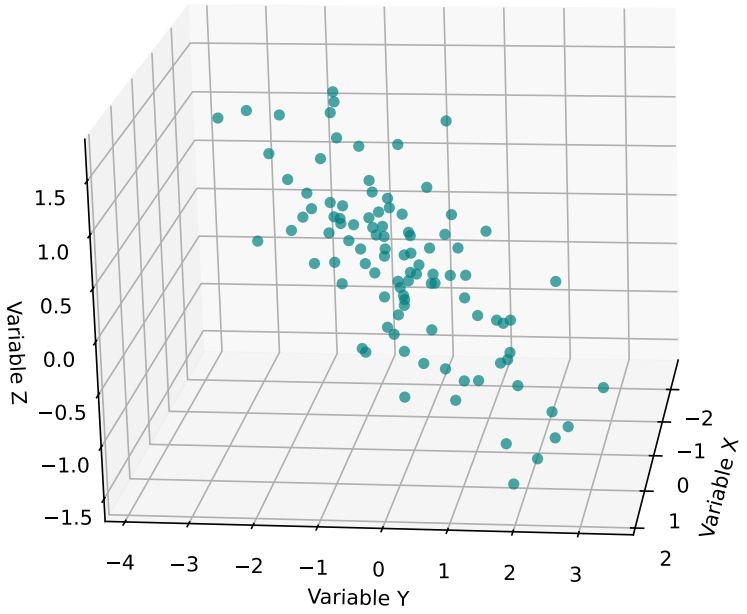

Figure 3: A simple rotated plots in 3D.

Star Plots - Shape-Based Comparison

Star plots offer a creative way to **represent multivariate data**:

- **Each variable is represented as a ray** (spoke) starting from a central point.
- The **length of each ray** corresponds to the **value** of that variable.
- When **rays are connected**, they form a "star-like shape" **unique** to each observation.

This method is **excellent for comparing patterns** between observations:

✔ **Similar shapes** suggest **similar data profiles**.

✔ **Differences** in shape can **quickly highlight outliers** or clusters.

However, star plots have limitations:

✘ They **do not quantify correlation**.

✘ The **direction** and **magnitude** of relationships between variables are **not explicit**.

✘ They are better for **visual pattern recognition** than for precise statistical analysis.

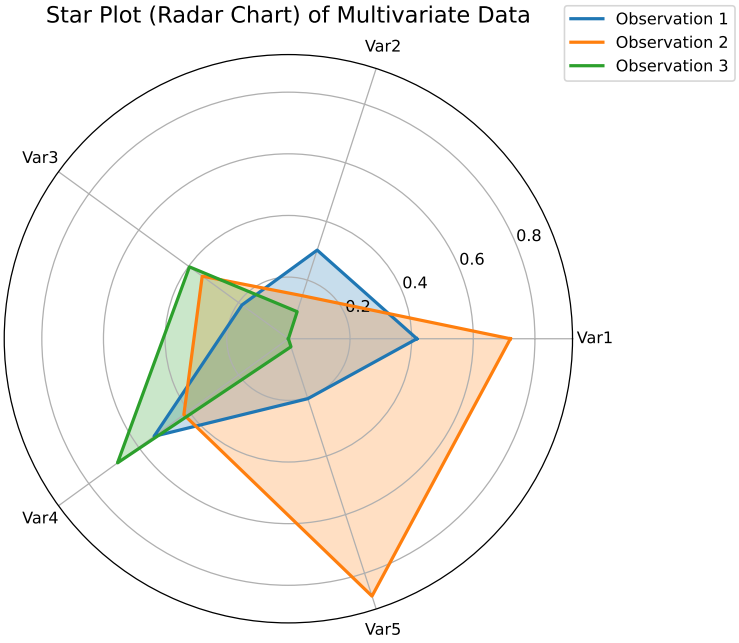

Figure 4: Star Plot (Radar Chart) of Multivariate Data.

Chernoff Faces - Human-Centric Visualization

Chernoff faces [1] are an **innovative visualization method** where **multivariate data is represented as a human face**:

- Each **variable controls a facial feature** (e.g., mouth curvature, eye size, nose length).
- **People are naturally attuned to recognizing faces** and subtle differences in expressions.
- Hence, Chernoff faces **leverage human perception** for quickly comparing **multivariate observations**.

Despite being engaging, Chernoff faces also have **drawbacks**:

- ✗ They **do not provide numerical precision**.
- ✗ The **mapping of variables to facial features** can be **arbitrary**.
- ✗ They work best as a **qualitative summary tool** rather than for deep statistical inference.

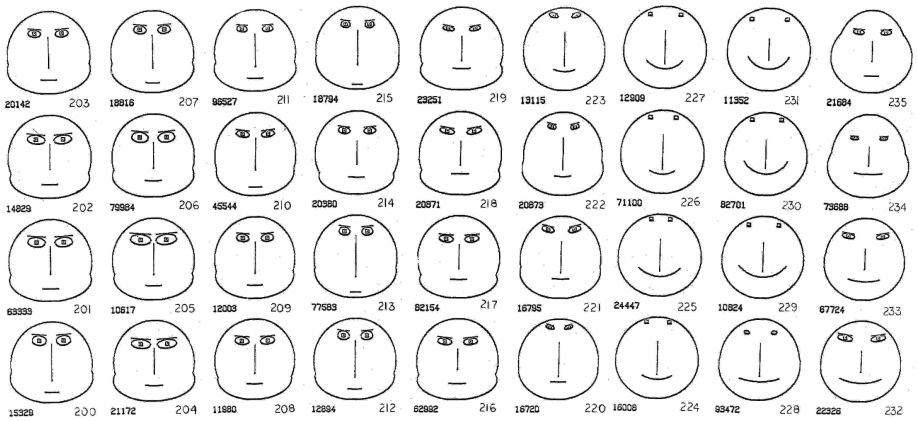

Figure 5: Some Chernoff faces. [1]

Graphic Type	Strengths	Limitations
Scatterplots	Clear view of pairwise relationships	Hard to scale beyond 2 variables
3D Plots	Visualizes 3-variable interaction	Limited to 3 dimensions, requires rotation
Star Plots	Quick shape-based comparison across variables	No quantification, poor at showing correlations
Chernoff Faces	Leverages facial perception for comparison	Subjective, lacks precision

Table 1: When and why to use graphics.

1.2 Dimensionality Reduction

⚠ The Challenge: Data in High Dimensions

In many real-world problems, we **collect multiple variables for each observation**. For example, in a medical study, a patient might be described by age, weight, blood pressure, and dozens of test results. This leads to **high-dimensional data**, where **each observation is a point in a complex, multi-dimensional space** (formally, a Euclidean space of dimension p).

The problem? As the **number of variables** (p) **increases**, the **data becomes harder to visualize, interpret, and model**:

- Some **variables** might be **redundant** or **highly correlated**.
- **Computations** become more **expensive**.
- **Patterns become obscured** by the complexity.

🏁 Goal of Dimensionality Reduction

We want to **summarize the data using fewer variables**, say k derived variables (with $k < p$), that still **retain most of the information**. This process is a balancing act:

- **Clarity**: fewer variables make data easier to understand and visualize.
- **Risk of oversimplification**: reducing dimensions too much can cause loss of important information.

The key concept here is that in statistics, **information is variability**. The more variability we retain from the original data, the more information we preserve.

Example 1: Blood Cells

Imagine we measure thickness and diameter for a set of red blood cells:

- Each cell = one observation with two variables.
- We can represent this as a table (numbers) or as a 2D scatterplot.

Now we ask ourselves: *Can we describe these cells using only one feature instead of two?*
If we choose only diameter or only thickness, we'll lose detail:

- Some cells will appear more similar than they really are.
- We miss variability that distinguishes them.

So, we seek a better single feature, one that captures the most variation possible from both thickness and diameter combined.

The Statistical Insight: Maximize Variability

Rather than randomly picking a feature, we **analyze the directions along which the data varies the most**.

1. First, we find the **direction of maximum spread**.
2. Then, the **second most spread direction**, orthogonal to the first.

This is the essence of **Principal Component Analysis (PCA)**, a **dimensionality reduction technique that finds the best directions** (linear combinations of variables) **to project the data**, while **maximizing retained variability**.

Dimensionality Reduction in Practice

Let's formalize the idea:

- We start with a **data matrix** X of shape $n \times p$ (n observations, p variables).
- The goal is to **obtain a new matrix** M of shape $n \times k$, with $k < p$, that **captures most of the variability**.
- The **difference** between X and M is **residual variation**, the **information lost**.

In summary, **Dimensionality Reduction** is about simplifying complexity: **transforming a large set of variables into a smaller**, more interpretable set **without losing the essence of the data**. It's central to data exploration, preprocessing, and modeling, especially when working with high-dimensional datasets.

1.3 Principal Component Analysis

Why PCA?

Imagine we have a data set with **many variables**, such as measurements of people, products, or cities. Some **variables might be closely related** (like height and weight), **while others might carry similar information**. This creates two challenges:

1. It becomes **hard to analyze and interpret** the data.
2. **Redundancy** can lead to inefficiency and confusion.

Principal Component Analysis (PCA) helps solve this by providing a **simplified version of the dataset**, where we focus only on the **essential information**.

The Main Idea

PCA works by **creating new variables**, called **principal components**, that are:

- **Combinations** of the original variables.
- **Ordered** so that the **first component captures the most variability** in the data.
- Each **subsequent component captures the next most variability**, but only from what's left over, and each **new component is uncorrelated with the previous** ones.

Imagine this like finding better "angles" or "perspectives" from which to view our data, ones that maximize how much we can see (i.e., variability) with as few perspectives as possible.

Variability is Information

In statistics, **Variability**, **how much values change from one observation to the next**, is considered **information**. The more variability a component captures, the **more useful it is** in understanding the data. So, PCA's job is to **find the directions in which the data varies the most**, and to use those directions to summarize the dataset.

How PCA Changes the Dataset

Let's say we start with p variables (like height, weight, age, income...). **PCA gives us p principal components**, but the *magic* is that **we usually only need the first few** to understand most of what's going on.

So instead of working with the full, original dataset, we now have a **simpler version**:

- We still have the **same number of observations**.
- But each observation is now described by **fewer, more informative variables** (components).

This is called a **low-dimensional representation** of the data.

Scores and Loadings - The Ingredients and the Result

In this new system:

- The **Scores** tell us where **each observation lies along the new axes** (the principal components).
- The **Loadings** tell us **how the original variables contribute to each component**.

Think of it like cooking:

- **Original variables** = ingredients
- **Loadings** = recipe instructions
- **Principal components** = final dishes
- **Scores** = ratings of the dishes for each person (observation)

Matrix Representation

Let's denote the data matrix as X, with n rows (observations) and p columns (variables). PCA is essentially about **transforming this matrix X into a new matrix**, where:

- The data is now described by **principal components** instead of the original variables.
- The **goal is to rotate and simplify** the data in a way that **emphasizes the most important directions** (maximum variability).

Two key matrices in PCA:

- **Loadings Matrix (V)**
  - This matrix contains the **weights** used to build the principal components.

  - Each **column** in V corresponds to a **principal component**.
  - Each **value** in V shows **how much each original variable contributes** to the component.

  We can think of V as the recipe book: it tells us how to combine original variables to create the new components.

- **Scores Matrix (U)**
  - This matrix contains the **projections of the data** onto the principal components.
  - Each **row** in U represents an **observation in the new PCA space**.
  - These are called scores, they tell us **where each observation lands along the new axes** (PC1, PC2...).

  So U is the result: it shows **how our data looks in the new, simplified system**.

PCA can be **computed using Singular Value Decomposition (SVD)**:

$$X = U \cdot S \cdot V^T \tag{6}$$

PCA **factorizes the data matrix** X. Here's what each matrix means:

- X: Original data $(n \times p)$
- U: Scores matrix $(n \times p)$, orthogonal (columns are independent)
- S: Diagonal matrix of **singular values** (related to the variance explained)
- V^T: Transposed loadings matrix $(p \times p)$, also orthogonal.

But for understanding, focus on this **simpler form**:

$$\textbf{PCA result} = \text{Scores} = X \cdot V$$

We **multiply the data** X **by the loadings matrix** V to obtain the **scores**.

However, both U and V are **orthogonal matrices**, meaning:

- Their columns are perpendicular (**no redundancy**).
- Principal **components are uncorrelated**, each new component captures new, non-overlapping information.

This is mathematically elegant and practically useful because it **removes multicollinearity** and makes downstream **analysis simpler and more robust**.

How Much Information Do We Keep?

Each principal component has a **percentage of variance explained**, this tells us how much of the original data's information it retains. Often, **the first 1 or 2 components explain so much that we can ignore the rest**.

In conclusion, **PCA helps us focus on what matters** in our data. It's like cleaning our glasses: everything becomes sharper, simpler, and more meaningful. We go from being overwhelmed by numbers to seeing clear patterns. It's a tool used everywhere, from finance to biology, marketing to engineering, whenever people need to make sense of complex data.

1.4 PCA Reference System

Why Do We Talk About Projections in PCA?

When we say that PCA projects data, we mean it **transforms the data points by placing them onto new axes**, the principal components, that better represent the structure of variability.

But why are projections so important? Because PCA has **two goals**:

1. **Maximize variance**: We want the **data to spread out as much as possible along the new axis**. This spread means we're capturing differences between observations.
2. **Minimize residuals**: We want to **minimize the error** when we approximate each point using the new axes. This is done by projecting each point perpendicularly onto the new axes, just like the shortest path between a point and a line.

This is why we "keep talking about projections": they allow us to **retain the most information** with **the least distortion**.

Generalizing to More Dimensions

In real datasets, we often have **more than two variables**. PCA scales to p dimensions, and the idea of projection still applies:

- PC1: The **direction in p-dimensional space** where data varies most.
- PC2: The next direction, **perpendicular to the first**, that captures remaining variability.
- This continues until we have p principal components, each orthogonal to the others.

So **PCA rotates the entire dataset into a new reference system** where the data structure is easier to understand.

A New Reference System

After PCA, our data now lives in a **new coordinate system**:

- The **original variables** (e.g., height, weight) are no longer the axes.
- Instead, the axes are **principal components**, which are **combinations of the original variables**.

This new reference system has two main features:

1. **PCA results are rotation invariant**

   What does this means? It means that if we rotate our dataset (e.g., by changing the coordinate system), PCA still finds the same underlying structure, the same principal components (relative to the data).

Why? PCA depends only on the relationships between the data points, specifically: the variances and covariances between variables. These are not affected by rotations of the data in space. Mathematically, PCA extracts eigenvectors of the covariance matrix that are invariant under rotation with respect to relative directions.

2. **Principal Components are independent (uncorrelated)**

What does this means? The principal components (PC1, PC2, ...) are uncorrelated with each other. Knowing the value of one PC tells us nothing about the value of the next.

Why? Each new component is orthogonal to the previous one. And since the correlation is equal to the cosine of the angle between the variables ($\cos(90) = 0$), PAC ensures that the correlation between the PCs is zero.

In high dimensions, **collinearity** often indicates redundant information between variables and can cause problems in the analysis. **PCA solves** this problem by giving us independent, uncorrelated variables.

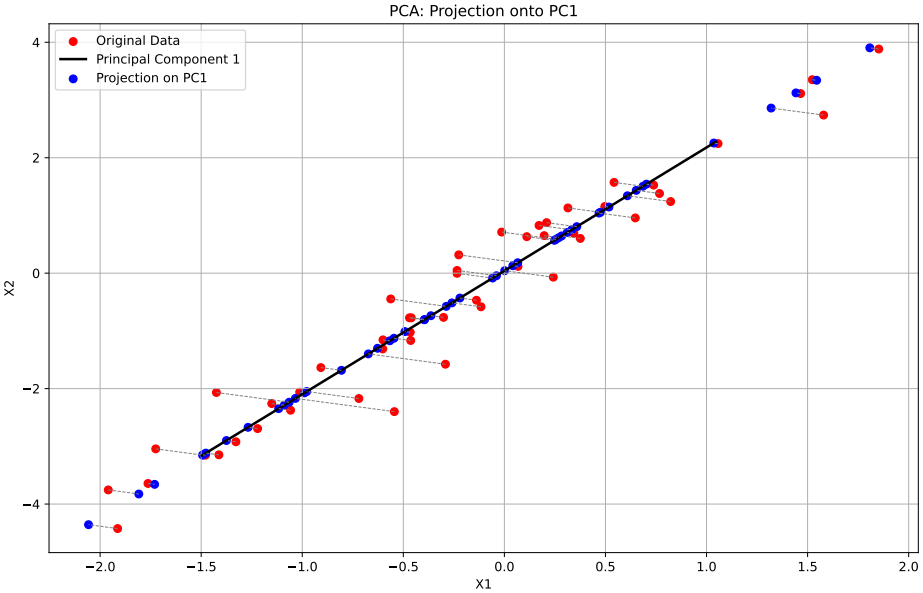

Figure 6: We have a bunch of scattered red dots. We have drawn a black line through the middle of the cloud of dots in the direction where the dots are most scattered. This line is our first principal component (PC1). For each red point, we drop a perpendicular line onto the black line. Where it lands, we place a blue point. This is called projection. Furthermore, the longer the black line, the more red points are taken into account (thus maximizing variance); on the other hand, the blue points must be close to the red points (shorter the distance) to minimize residuals and lose less information.

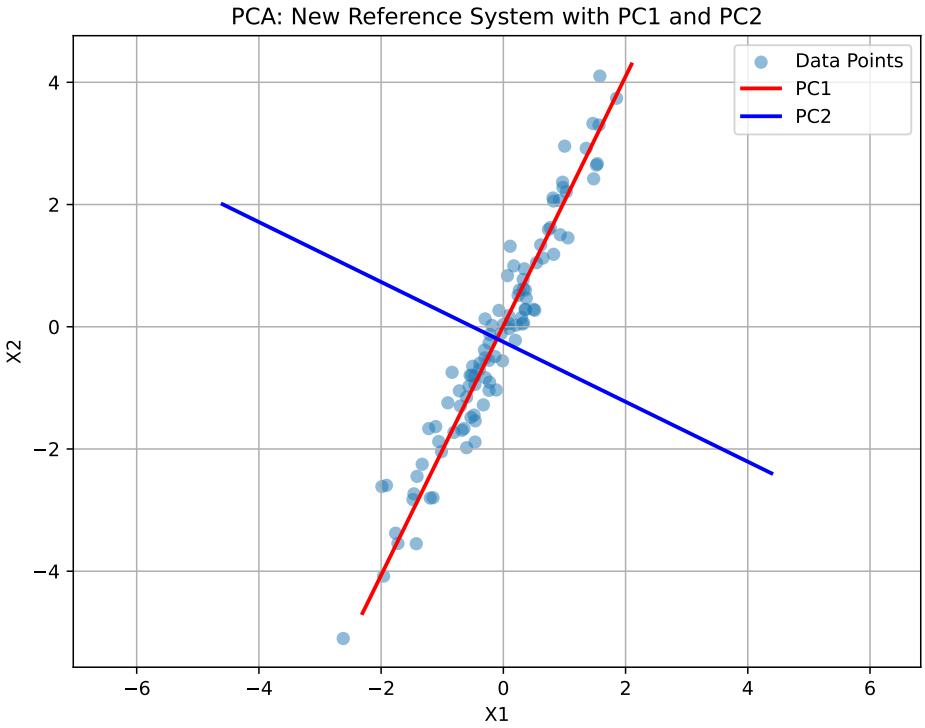

Figure 7: By drawing two lines, PC1 and PC2, we obtain a new reference system. It is a rotated system. The axes are principal components, which are combinations of the original variables.

Principal Components: Linear Combinations

Let's now talk about **how principal components are constructed**. Each principal component is a **linear combination of the original variables**. This means:

$$\text{PC}_1 = \phi_{11} \cdot x_1 + \phi_{21} \cdot x_2 + \cdots + \phi_{p1} \cdot x_p$$

But in general:

$$\text{PC}_j = \phi_{1j} \cdot x_1 + \phi_{2j} \cdot x_2 + \cdots + \phi_{pj} \cdot x_p \tag{7}$$

Where:

- The ϕ-values (phi) are called **Loadings**.
- They are the **weights assigned to each original variable**.
- The **higher the loading**, the **more that variable contributes** to that principal component.

For **each observation** (e.g., a person, product), we now **compute their position in the new PCA space**:

$$z_{ij} = \phi_{1j} \cdot x_{i1} + \phi_{2j} \cdot x_{i2} + \cdots + \phi_{pj} \cdot x_{ip} \tag{8}$$

These values z_{ij} are called **Scores**. They tell us:

- Where each observation lies **along a principal component axis**.
- How much that **component contributes** to describing this observation.

So now, instead of describing each person with p original variables, we describe them with k scores, where $k < p$, but we still capture the essence of their data.

References

[1] Herman Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of the American statistical Association*, 68(342):361–368, 1973.

[2] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Series in Statistics. Springer New York, 2009.

[3] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning: with Applications in Python*. Springer Texts in Statistics. Springer New York, 2013.

[4] R.A. Johnson and D.W. Wichern. *Applied Multivariate Statistical Analysis*. Applied Multivariate Statistical Analysis. Pearson Prentice Hall, 2007.

[5] Beraha Mario. Applied statistics. Slides from the HPC-E master's degree course on Politecnico di Milano, 2024-2025.

Index