Outline

1. **Motivations for Dimension Reduction**
2. **Principal Component Analysis**
   - PCA goals and method
   - PCA example using R

curse of dimensionality

The curse of dimensionality

- a term coined by Bellman in 1961
- refers to difficulties associated with multivariate data analysis as dimensionality (number of variables) increases

curse of dimensionality

- Real data usually have thousands, or millions of dimensions
- Huge number of dimensions causes problems
- Data becomes very sparse, some algorithms become meaningless (e.g. density based clustering)
- The complexity of several algorithms depends on the dimensionality and they become infeasible.

curse of dimensionality

Many implications of the curse of dimensionality...

- Data becomes very sparse
  - Exponential growth in the quantity of data required to maintain a sampling density

curse of dimensionality

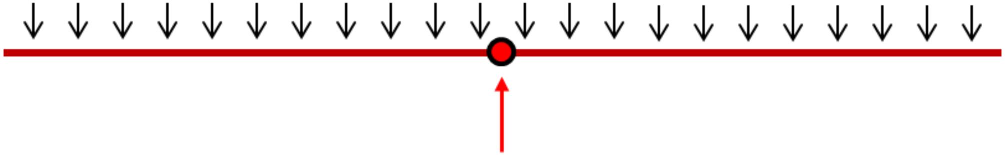
Suppose there are data points uniformly distributed in a d-dimensional unit hypercube.

Question: If we want to construct a hypercube neighborhood of point x_0 which captures 10% of the observations, what is the edge length, L, of this cube?

curse of dimensionality

Suppose there are data points uniformly distributed in a d-dimensional unit hypercube, where $d = 1$.

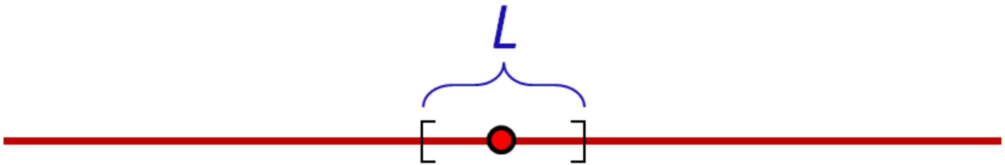
Question: If we want to construct a hypercube neighborhood of point x_0 which captures 10% of the observations, what is the edge length, L, of this cube?



curse of dimensionality

Suppose there are data points uniformly distributed in a d-dimensional unit hypercube, where $d = 1$.

Question: To construct a hypercube neighborhood of point x_0 which captures 10% of the observations, the edge length, L, of this cube: $L = 0.1$

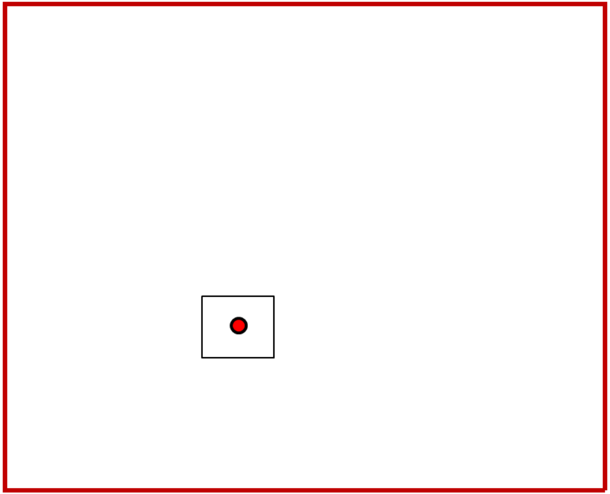


curse of dimensionality

Suppose there are data points uniformly distributed in a d-dimensional unit hypercube, where $d = 2$.

What is the edge length of this cube that captures 10% of the observations?

$L = 0.316$

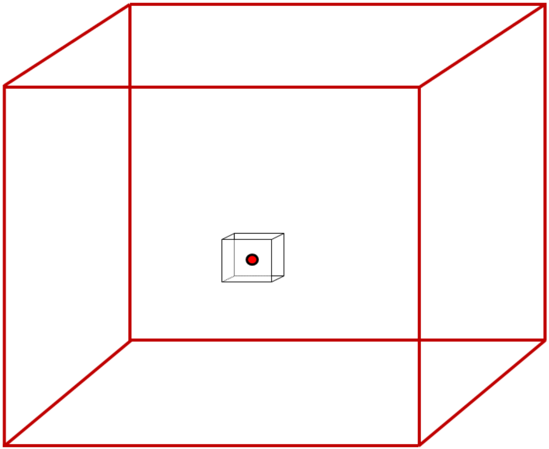


curse of dimensionality

Suppose there are data points uniformly distributed in a d-dimensional unit hypercube, where $d = 3$.

What is the edge length of this cube that captures 10% of the observations?

$$L = \sqrt[3]{0.1} = 0.464$$



curse of dimensionality

Suppose there are data points uniformly distributed in a d-dimensional unit hypercube.

To construct a hypercube neighborhood of point x_0 which captures 10% of the observations, the edge length of the cube is a function of the dimensions: $L = \sqrt[d]{0.1}$

- For $d = 10$, $L = \sqrt[10]{0.1} \approx 0.79$
- For $d = 100$, $L = \sqrt[100]{0.1} \approx 0.98$

curse of dimensionality

Many implications of the curse of dimensionality...

- Data becomes very sparse
  - Exponential growth in the quantity of data required to maintain a sampling density
- Humans have an amazing capacity to discern patterns in 1, 2 or 3D; but this is drastically limited for 4+ dimensions

dimension reduction

Our dimension reduction problem can be stated as:

Given an attribute space $\mathbf{x} \in \mathbb{R}^m$ find a mapping $\mathbf{y} = f(\mathbf{x}) : \mathbb{R}^m \to \mathbb{R}^q$ with $q < m$ such that the transformed feature vector $\mathbf{y} \in \mathbb{R}^q$ preserves (most of) the information or structure in $\mathbb{R}^m$.

dimension reduction

In general, an optimal mapping $\mathbf{y} = f(\mathbf{x})$ to construct/extract the most useful "features" will be non-linear...

- no systematic way to generate non-linear transforms
- selection of a subset of transforms is problem dependent and often relies on human input
- however, we can use various linear transforms: $\mathbf{y} = W\mathbf{x}$

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \xrightarrow{\text{linear feature extraction}} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_q \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{q1} & w_{q2} & & w_{qm} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

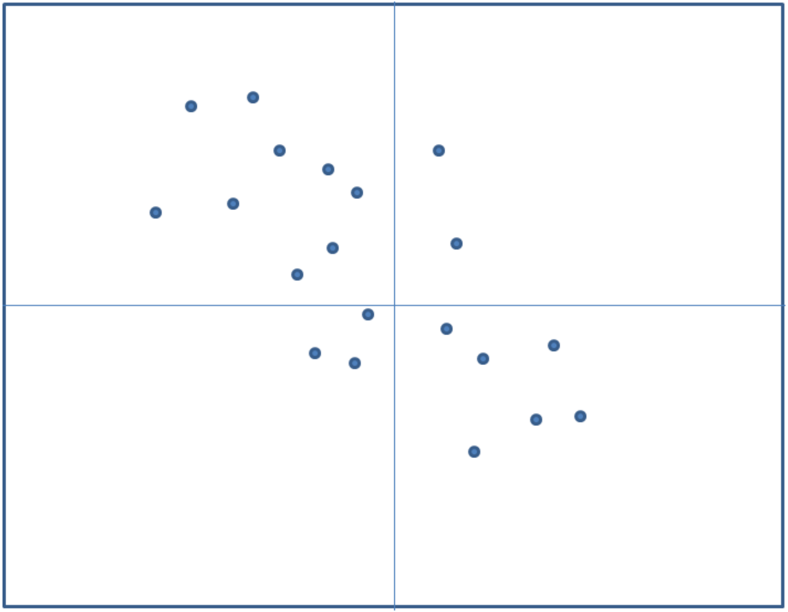
The selection mapping $\mathbf{y} = f(\mathbf{x})$ is guided by an objective function that we seek to maximize (or minimize).

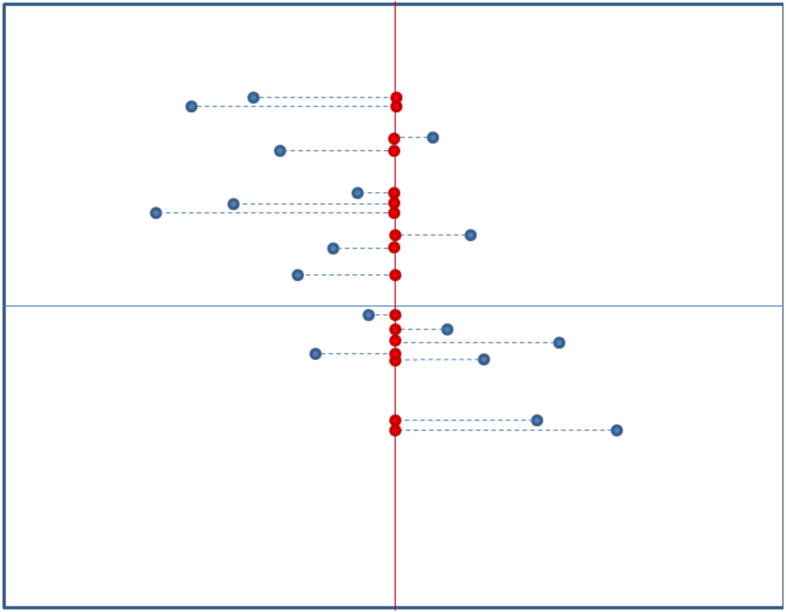
Depending on the criteria measured by the objective function, the techniques are grouped into two categories:

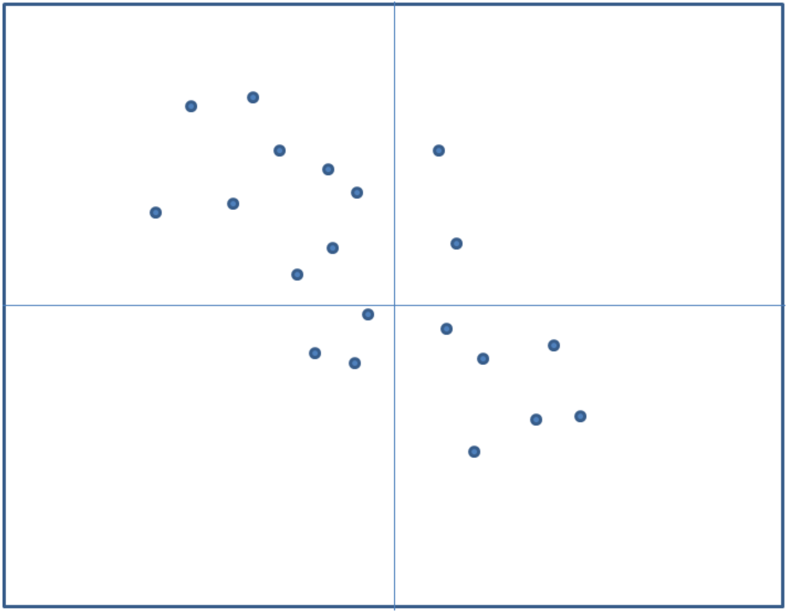
1. representation: represent data accurately in a lower-dimensional space
2. classification: enhance the class-discriminatory information in the lower-dimensional space

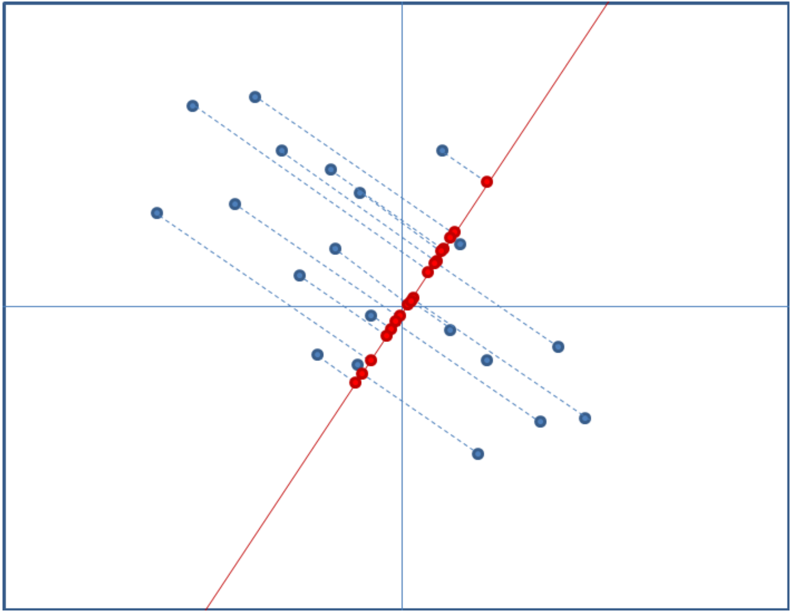
Two techniques are commonly used:

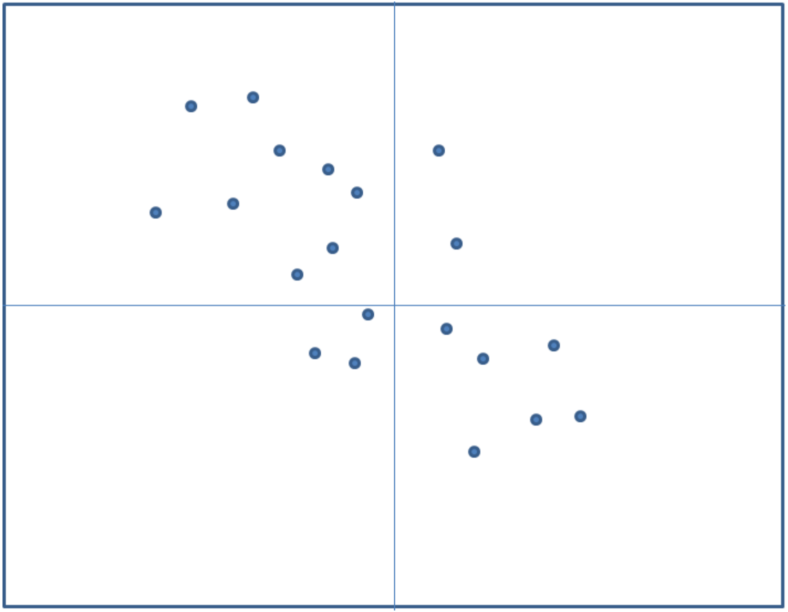
- Principal Components Analysis
  - representation method that preserves variance
- Linear Discriminant Analysis
  - classification method that enhances separability

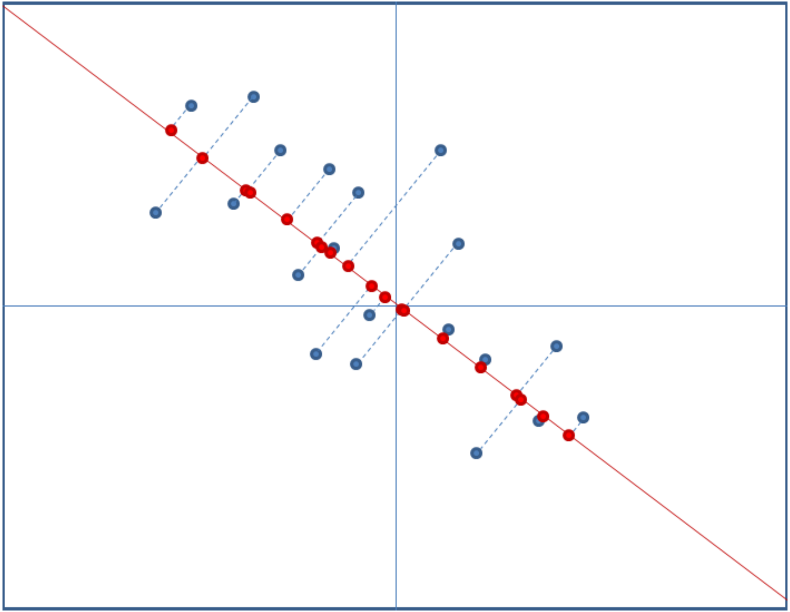


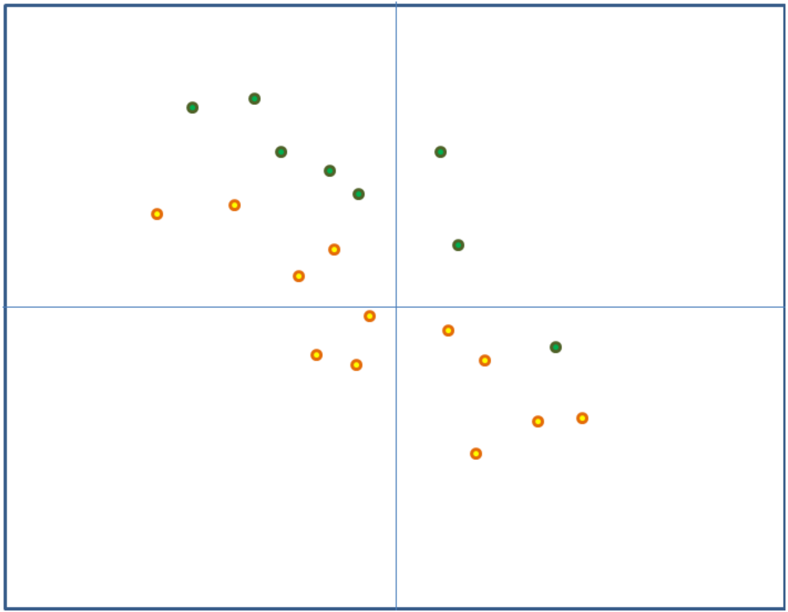


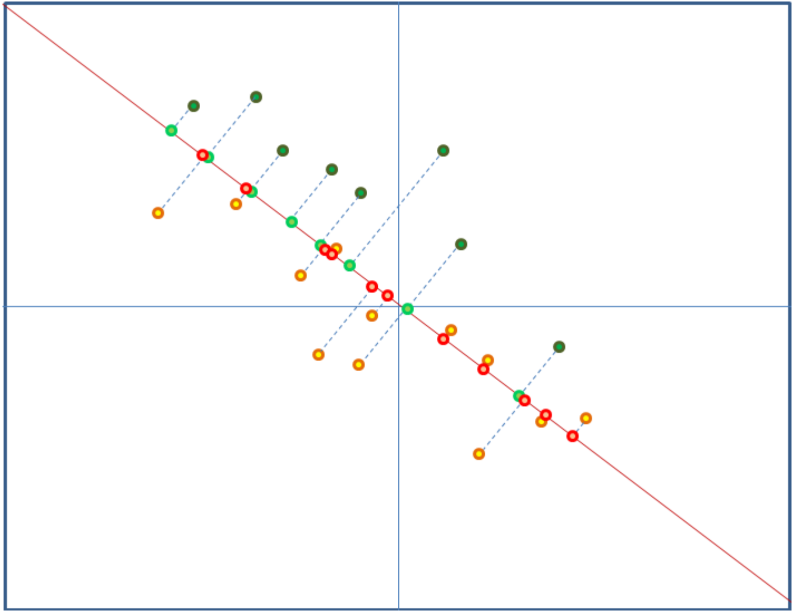


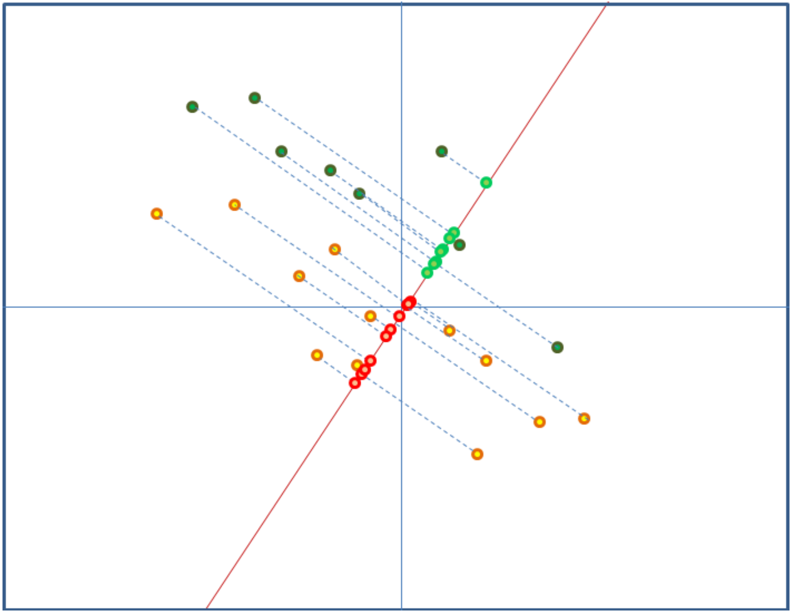












Outline

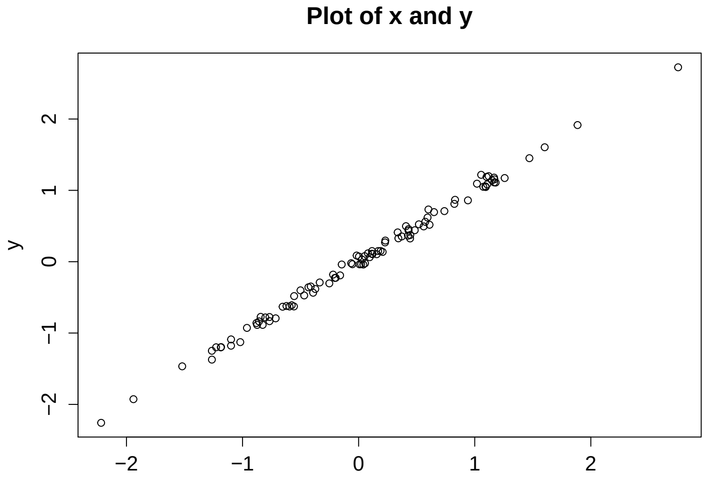
principal component analysis

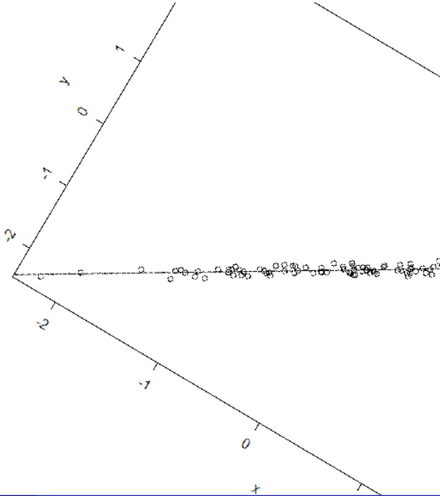
- Principal component analysis (PCA) is the oldest technique in multivariate analysis
- first introduced by Pearson in 1901
- involves a mathematical procedure that transforms a number of (possibly) correlated variables into a (smaller) number of *uncorrelated* variables called principal components.
- the first PC accounts for as much variability in the data as possible, and each succeeding PC accounts for as much of the remaining variability as possible.

principal component analysis

- mathematically, PCA relies on the fact that many of the attributes are correlated, sometimes highly correlated
- it results in a *rotation* of the coordinate system in such a way that the axes show a maximum of variation (covariance) along their directions.
- this description can be mathematically condensed to a so-called eigenvalue problem.

simple example





variance

PCA uses the variance in the data as the structure preservation criterion.

PCA tries to preserve as much of the original variance of the data when projected to a lower-dimensional space.

(Sample) variance for a numerical attribute:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

covariance matrix

Variance formula with vectors (using outer product), e.g. for 2D:

$$\mathbf{S} = \frac{1}{n-1}\sum_{i=1}^{n}\left(\begin{pmatrix} x_i \\ y_i \end{pmatrix} - \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix}\right)\left(\begin{pmatrix} x_i \\ y_i \end{pmatrix} - \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix}\right)^{\top} = \begin{pmatrix} s_x^2 & s_{xy} \\ s_{xy} & s_y^2 \end{pmatrix}$$

where

$$s_{xy} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i y_i - \bar{x}\bar{y}) \qquad \text{(covariance of } x \text{ and } y\text{)}$$

principal component analysis

The data points are first mean-centered, i.e. centered around the origin by subtracting the mean values.

Goal: Find a projection in the form of a linear mapping $\mathbb{R}^m \to \mathbb{R}^q$ (for visualization choose $q = 2$ or $q = 3$):

$$\mathbf{y} = W \cdot (\mathbf{x} - \bar{\mathbf{x}})$$

where W is a $m \times q$ matrix such that the variance of the projected data $\mathbf{y_i} = W \cdot (\mathbf{x_i} - \bar{\mathbf{x}})$ is as large as possible.

principal component analysis

Problem: Without restrictions the entries in W can be chosen arbitrary large, the data would be projected and *stretched*, leading to an arbitrary large variance of the projected data.

Solution: Introduce constraints such that the matrix W is only a projection.

Constraints: Each column $\mathbf{v_i}$ of the matrix

$$W = (\mathbf{v_1}, \ldots, \mathbf{v_m})$$

must be normalized, i.e. $\| \mathbf{v_i} \| = 1$.

principal component analysis

Solution of the constrained optimization problem in PCA:

$$W = (\mathbf{v_1}, \ldots, \mathbf{v_m})$$

where the principal components $\mathbf{v_1}, \ldots, \mathbf{v_m}$ are the *normalized eigenvectors* of the covariance matrix of the data

$$\mathbf{S} = \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{x_i} - \bar{\mathbf{x}})(\mathbf{x_i} - \bar{\mathbf{x}})^\top$$

PCA for dimension reduction

Let $\lambda_1 \geq \ldots \geq \lambda_m$ be the eigenvalues of the covariance matrix.

When we project the data to the first q principal components $v_1, \ldots, v_q$ corresponding to the eigenvalues $\lambda_1, \ldots, \lambda_q$, this projection will preserve a fraction of

$$\frac{\lambda_1 + \ldots + \lambda_q}{\lambda_1 + \ldots + \lambda_m}$$

of the variance of the original data.

normalization

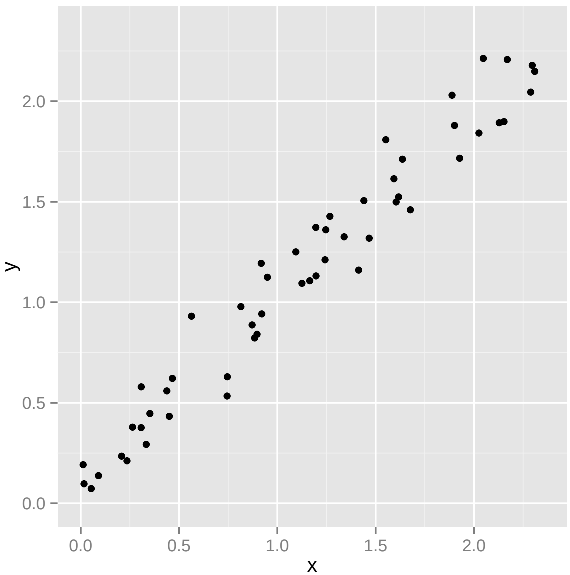
Important Note!

Usually, the data should be z-score standardized $x \mapsto \frac{x-\bar{x}}{s}$ to ensure that all attributes contribute equally to the overall variance.

simple PCA example

PCA on simple 2D data:

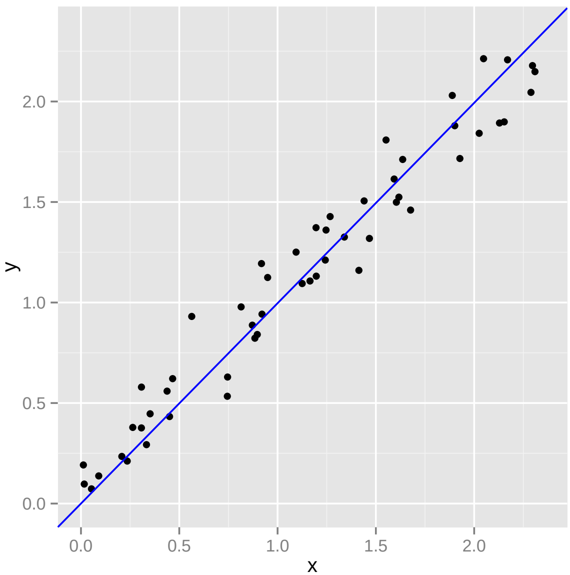
- $Y = X + \epsilon$
- expect PC1 to be a diagonal axis $(\cos(\frac{\pi}{4}), \sin(\frac{\pi}{4}))$



simple PCA example

Rotation matrix:

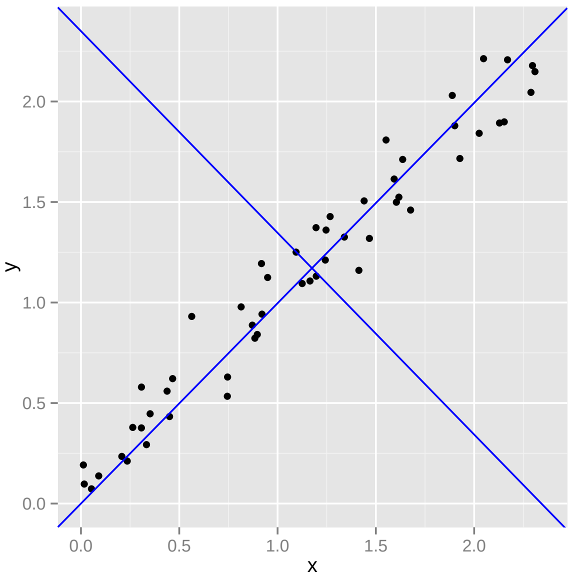
	PC1	PC2
x	0.7059041	-0.7083074
y	0.7083074	0.7059041



simple PCA example

Rotation matrix:

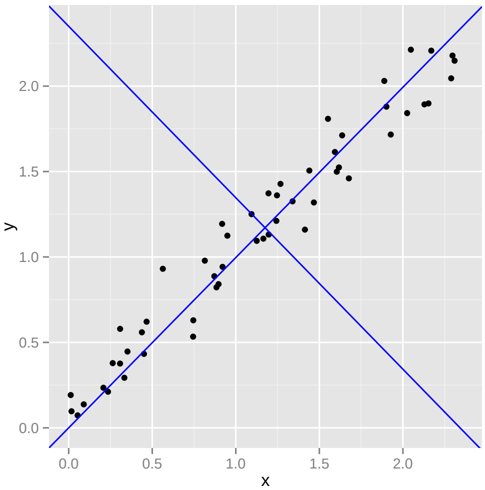
	PC1	PC2
x	0.7059041	-0.7083074
y	0.7083074	0.7059041



simple PCA example

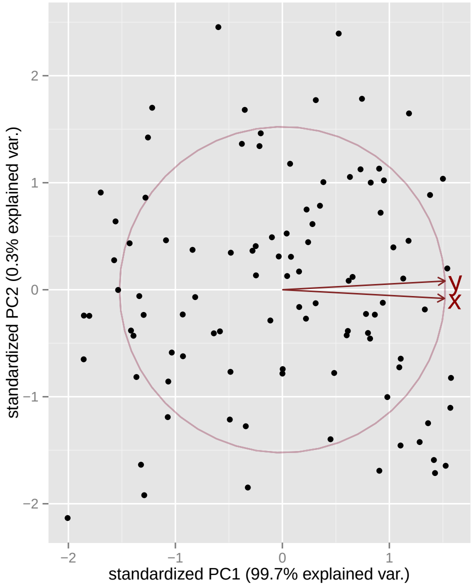
Importance of components:

	PC1	PC2
Standard deviation	2.0077	0.10779
Proportion of Variance	0.9971	0.00287
Cumulative Proportion	0.9971	1.00000



simple PCA example

- biplots are PCA visualizations
- PC1 is on x-axis
- PC2 is on y-axis
- relationship between variables and PC's depicted as vectors



PCA example using R

The `state.x77` data set is available by in the `datasets` package in R; it's a compilation of data about the US states put together from the 1977 Statistical Abstract of the United States

The 8 variables are:

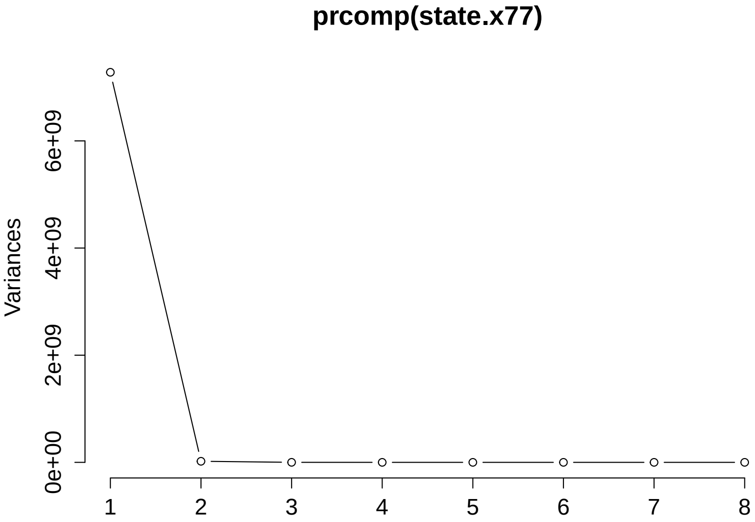
Population	in thousands
Income	dollars per capita
Illiteracy	percent of the population
Life Exp	years of life expectancy at birth
Murder	murders and non-negligent manslaughters per 100k people
HS Grad	percent of adults who were high-school graduates
Frost	mean days per year with low temperatures below freezing
Area	in square miles

PCA example using R

`prcomp` is the preferred command for PCA in R.

```
> prcomp(state.x77)

Rotation:
                     PC1           PC2          PC3
Population  1.182966e-03 -9.996005e-01  0.0278490777 -4.67125
Income      2.616550e-03 -2.796866e-02 -0.9991766328  2.82173
Illiteracy  5.518945e-07 -1.420515e-05  0.0005844687  7.10074
Life Exp   -1.688521e-06  1.928393e-05 -0.0010367078 -3.87596
Murder      9.881522e-06 -2.787128e-04  0.0027764911  2.81609
HS Grad     3.157288e-05  1.882545e-04 -0.0082661337 -2.78454
Frost       3.607163e-05  3.871630e-03 -0.0280421226 -9.98773
Area        9.999959e-01  1.255538e-03  0.0025827049 -3.16884
```

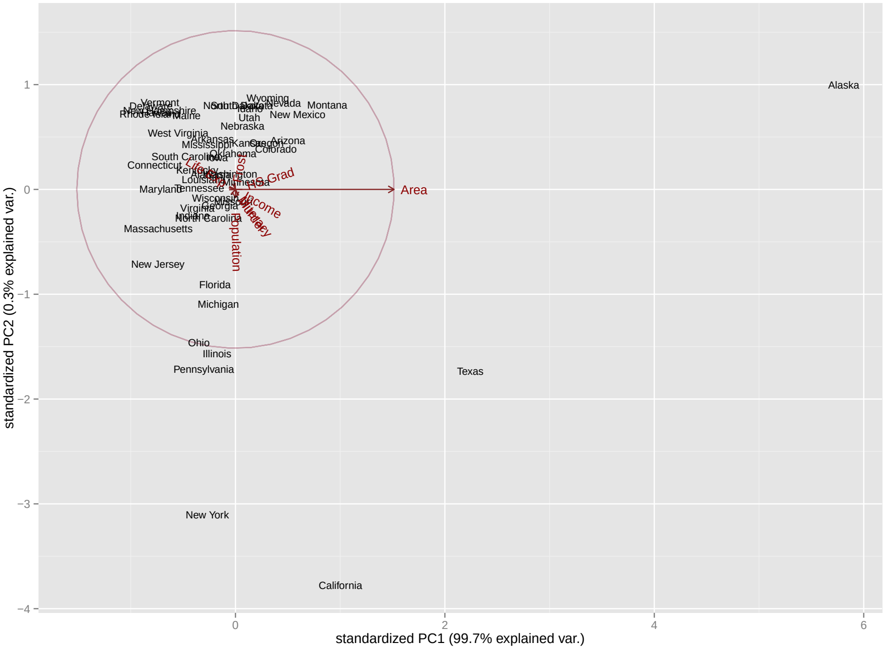
screeplot



PCA example using R

We've chosen units where one variable is immensely larger than the others, so it varies much more...

```
> apply(state.x77,2,sd)

  Population       Income   Illiteracy     Life Exp
4.464491e+03 6.144699e+02 6.095331e-01 1.342394e+00

      Murder      HS Grad        Frost         Area
3.691540e+00 8.076998e+00 5.198085e+01 8.532730e+04
```

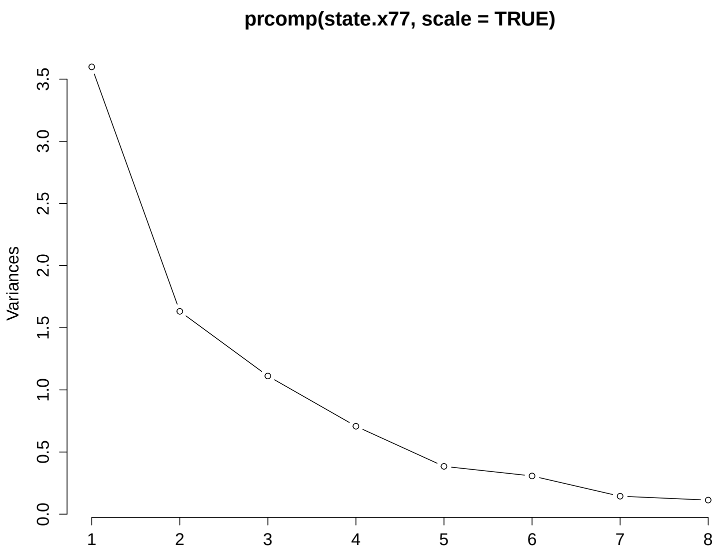


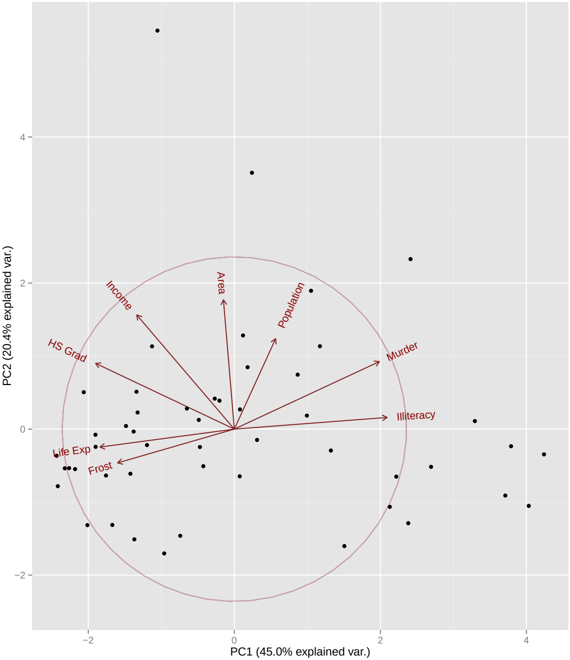
Alaska
Wyoming
Montana
Nevada
New Mexico
Utah
Idaho
Nebraska
Vermont
Delaware
Maine
Rhode Island
West Virginia
Arkansas
Mississippi
Arizona
Oregon
Colorado
Oklahoma
Iowa
South Carolina
Connecticut
Kentucky
Washington
Louisiana
Minnesota
Maryland
Tennessee
Wisconsin
Virginia
Georgia
Indiana
North Carolina
Massachusetts
New Jersey
Florida
Michigan
Ohio
Illinois
Pennsylvania
Texas
New York
California
Area
HS Grad
Income
Population
1
0
−1
−2
−3
−4
2
4
6
standardized PC1 (99.7% explained var.)
standardized PC2 (0.3% explained var.)

Things are a lot better if we z-score standardize the variables. `prcomp` will do this for us.

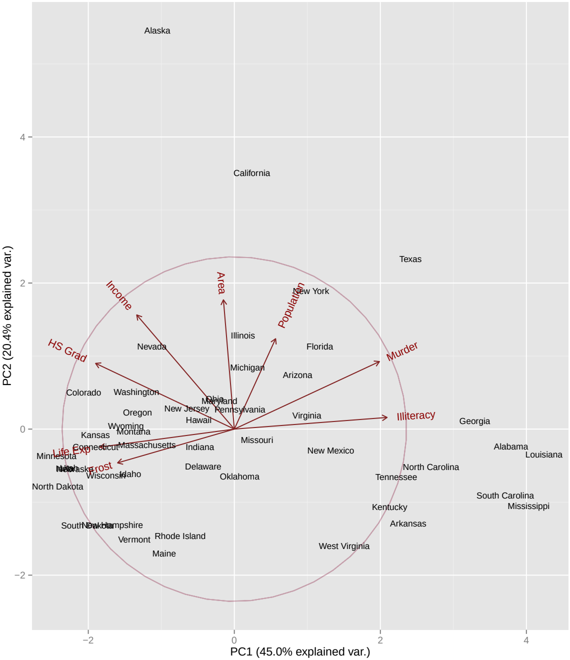
```
> prcomp(state.x77,scale=TRUE)

Rotation:
                   PC1         PC2         PC3         PC4
Population  0.12642809  0.41087417 -0.65632546 -0.40938555  0
Income     -0.29882991  0.51897884 -0.10035919 -0.08844658 -0
Illiteracy  0.46766917  0.05296872  0.07089849  0.35282802  0
Life Exp   -0.41161037 -0.08165611 -0.35993297  0.44256334  0
Murder      0.44425672  0.30694934  0.10846751 -0.16560017 -0
HS Grad    -0.42468442  0.29876662  0.04970850  0.23157412 -0
Frost      -0.35741244 -0.15358409  0.38711447 -0.61865119  0
Area       -0.03338461  0.58762446  0.51038499  0.20112550  0
```

```
> summary(prcomp(state.x77,scale=TRUE))

Importance of components:
                          PC1    PC2    PC3
Standard deviation     1.8971 1.2775 1.0545
Proportion of Variance 0.4499 0.2040 0.1390
Cumulative Proportion  0.4499 0.6539 0.7928

                           PC4     PC5     PC6
Standard deviation     0.84113 0.62019 0.55449
Proportion of Variance 0.08844 0.04808 0.03843
Cumulative Proportion  0.88128 0.92936 0.96780
```


prcomp(state.x77, scale = TRUE)
Variances
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
1
2
3
4
5
6
7
8



PC2 (20.4% explained var.)
PC1 (45.0% explained var.)
Income
Area
Population
HS Grad
Murder
Illiteracy
Life Exp
Frost
4
2
0
−2
−2
0
2
4



Alaska
California
Texas
New York
Illinois
Florida
Nevada
Michigan
Arizona
Colorado
Washington
Ohio
Maryland
New Jersey
Pennsylvania
Oregon
Hawaii
Virginia
Georgia
Wyoming
Montana
Kansas
Connecticut
Massachusetts
Indiana
Missouri
Alabama
Louisiana
Minnesota
New Mexico
North Carolina
Nebraska
Wisconsin
Idaho
Delaware
Oklahoma
Tennessee
North Dakota
South Carolina
Mississippi
Kentucky
Arkansas
New Hampshire
Rhode Island
Vermont
Maine
West Virginia
Income
Area
Population
HS Grad
Murder
Illiteracy
Life Exp
Frost
4
2
0
−2
−2
0
2
4
PC1 (45.0% explained var.)
PC2 (20.4% explained var.)

PCA example using R

PC1 distinguishes between cold states with educated, harmless, long-lived populations, and warm, ill-educated, short-lived, violent states.

The second PC distinguishes big rich educated states from small poor ignorant states, which tend to be a bit warmer, and less murderous.

PCA summary

The central idea of PCA is to reduce the dimensionality of a data set consisting of a large number of interrelated variables, while retaining as much as possible of the variation

This is achieved by transforming to a new set of variables, the principal components, which are ordered so that the first few retain most of the variation.

- visualizing and exploring high-dimensional data
- identifying outliers, clusters, patterns
- understanding the intrinsic dimension of data
- evaluating collinearity of variables