Applied Machine Learning (AML)

Class Starting at 4:10pm



THE UNIVERSITY *of* EDINBURGH
informatics

Scan to check-in



Oisin Mac Aodha • Siddharth N.

Applied Machine Learning

Week 10: Representing Data and Exploratory Data Analysis

This slides will be made available on the project website after the class.
This session will be recorded.

Overview

1. Outline your tasks this for week
2. Discussion of Week 9’s topics

Exam

- Friday, 11th December 2026, from 9:30am to 11:30am.
- The exam will be on campus and will be **closed book**.
- Split over multiple locations by surname; please check personalised timetable to confirm location.
- Some topics from previous AML exams not be on the exam this year. See discussion in Q&A session in Week 7 for a full list.

For more information see course webpage:
https://groups.inf.ed.ac.uk/teaching/aml/exam/

Coursework

- Main • Thu 26 Nov, 12pm
  - Submit via Learn: Assessment → AML Coursework (Main)
  - DO NOT include names, or change template!
- Supplement • Tue 01 Dec, 5pm
  - Submit via Learn: Assessment → AML Coursework (Supplement)
  - MUST be submitted, but not evaluated
  - Source for both project and report
  - See instructions: https://groups.inf.ed.ac.uk/teaching/aml/submission-instructions/

Week 10: Your tasks for this week

1) Attend Tutorial 4 - solutions will be online later this week
2) Watch videos for next Q&A - **Optimisation** and **Generalisation**
3) Ask questions on EdStem if stuck
4) Continue working on the coursework; Due 12pm Thu 26th Nov

AML Course Feedback

- Please take a few minutes to fill this out!
- All are responses anonymised.
- Fill in by Fri 15th Jan to enter a draw to win one of two £25 vouchers.

For any comments particular to strike action, please use Q10 to elaborate.



Reality of Applied Machine Learning

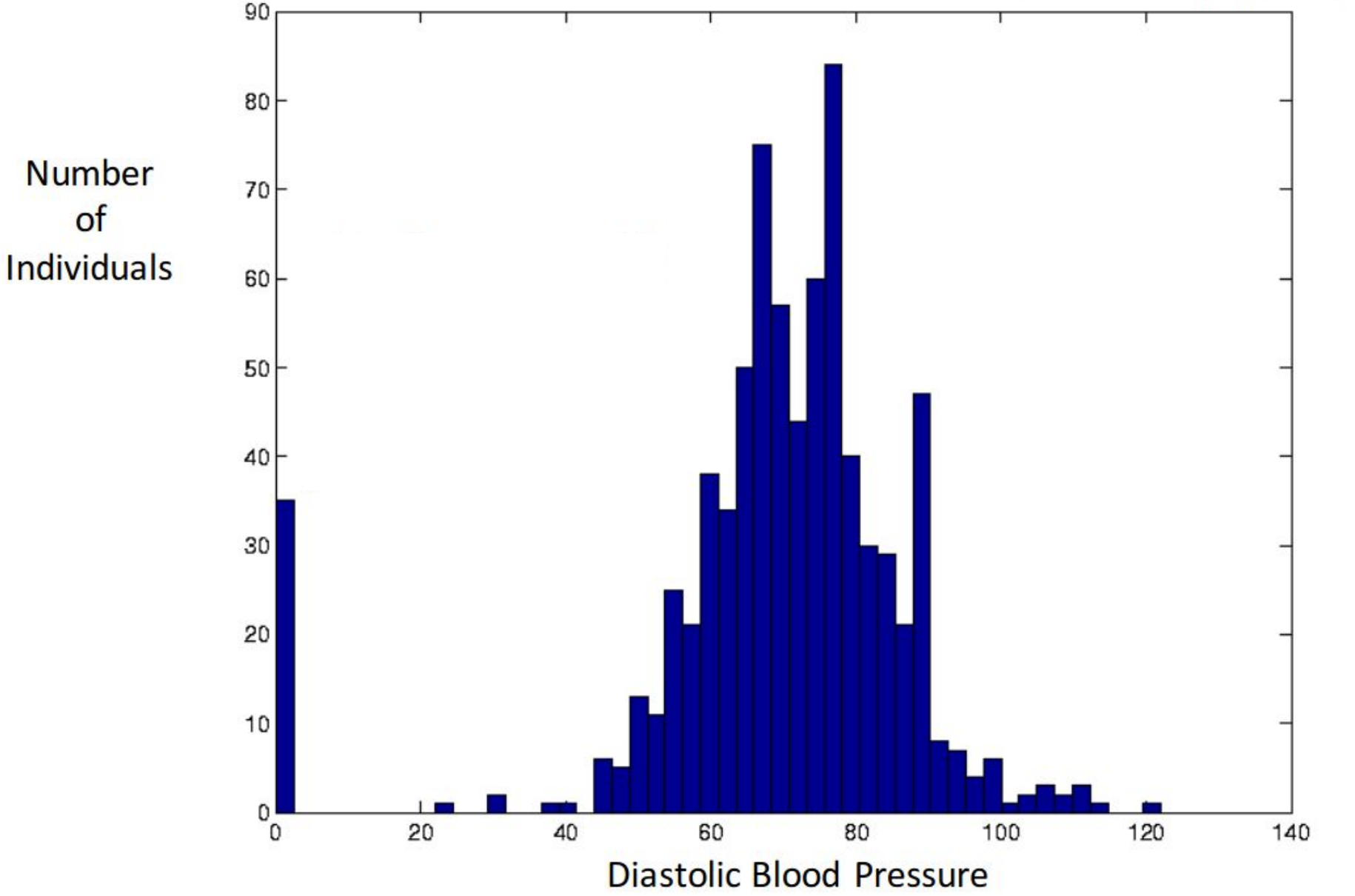
- Majority of time is not spent designing new ML algorithms
- It will be spent:
  - understanding the input data
  - defining the ML task
  - cleaning the data
  - designing features/attributes
  - visualising the data
  - comparing models (evaluating performance)
  - repeat

Visualisation

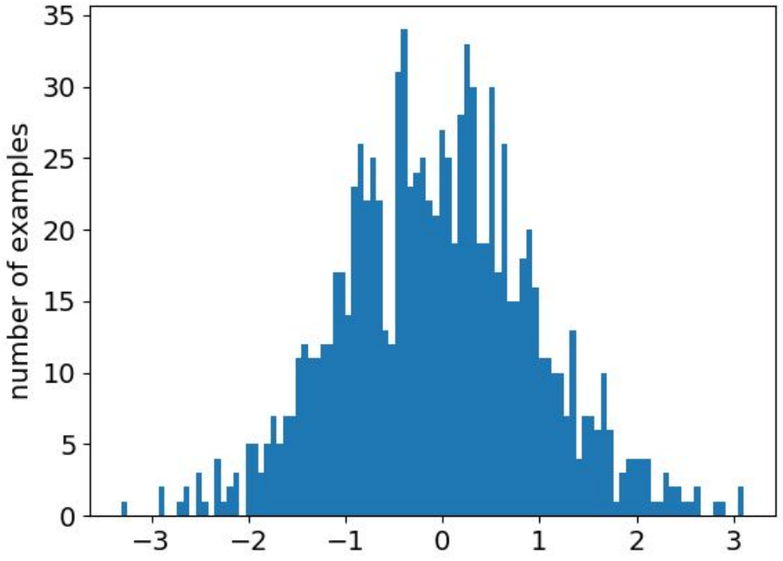
1) Crucial part of exploratory data analysis

    Identify problems/anomalies in your data

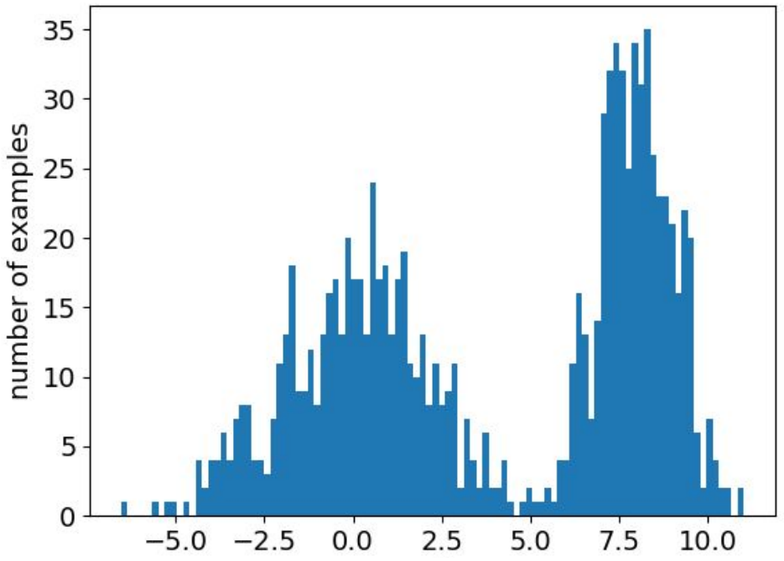
Pima Indians Diabetes Data,
From UC Irvine Machine Learning Repository



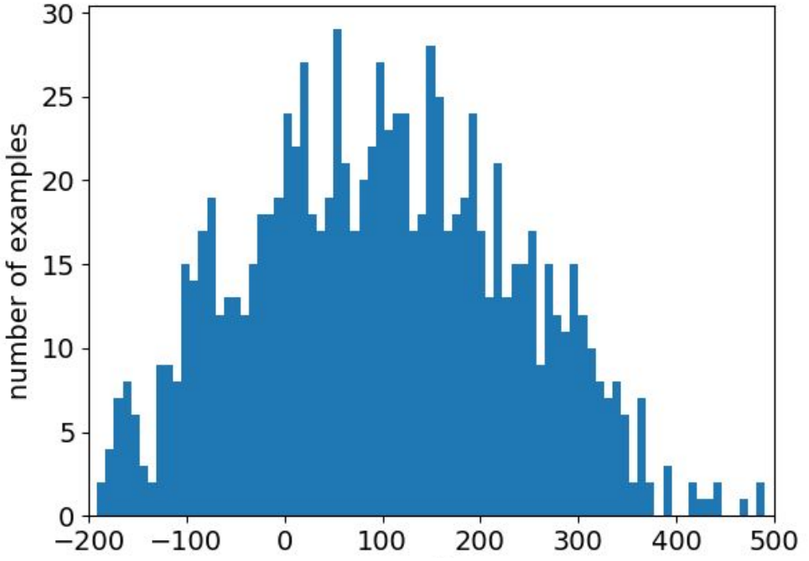
Slide Credit Padhraic Smyth


number of examples
35
30
25
20
15
10
5
0
−3
−2
−1
0
1
2
3

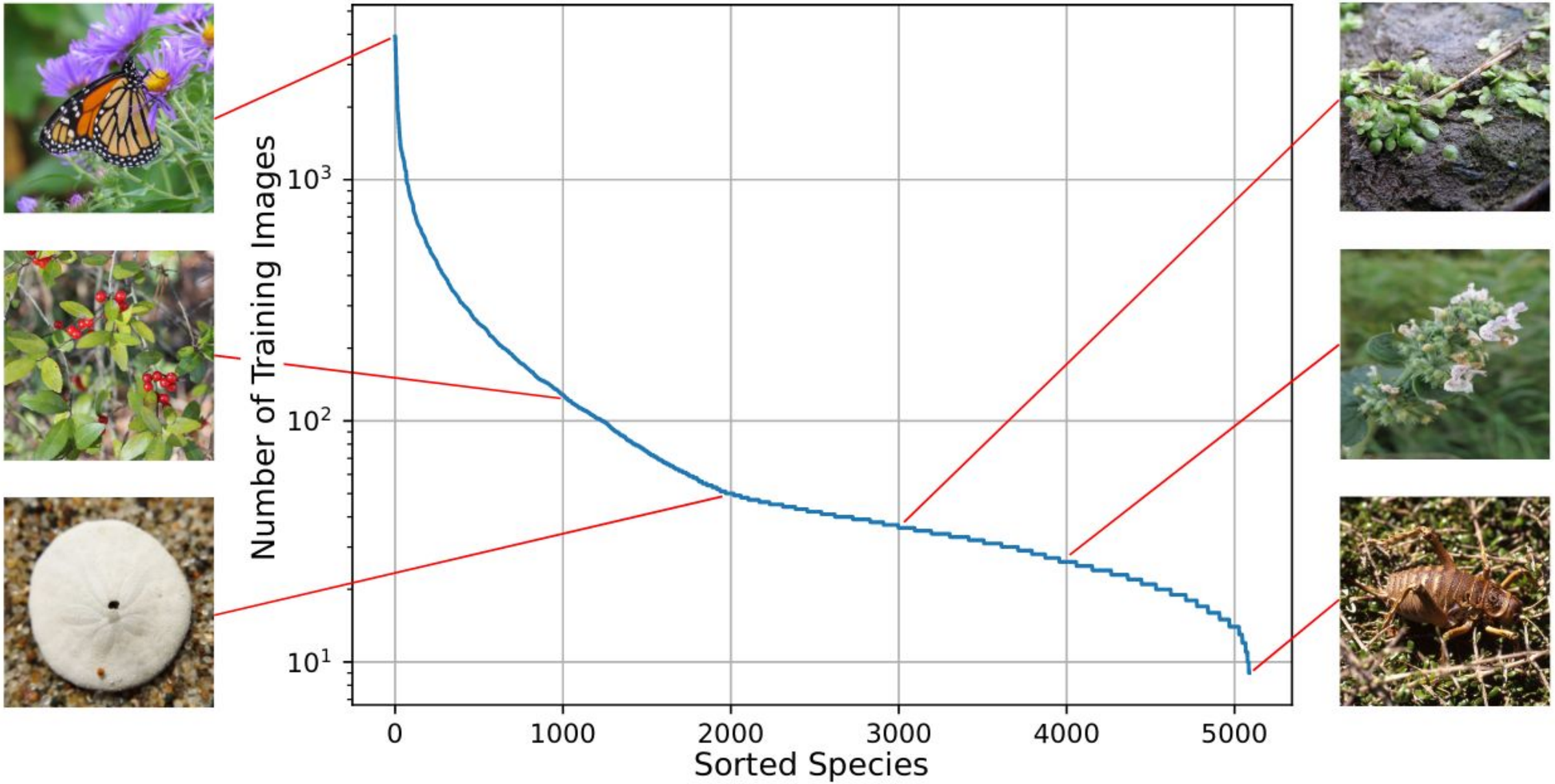
x_1


number of examples
35
30
25
20
15
10
5
0
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

x_2


number of examples
30
25
20
15
10
5
0
−200
−100
0
100
200
300
400
500

x_3



https://arxiv.org/abs/1707.06642

Higher Dimensions are Challenging

Visualization is essential but not scalable as the number of dimensions increase

Scatter plot matrix

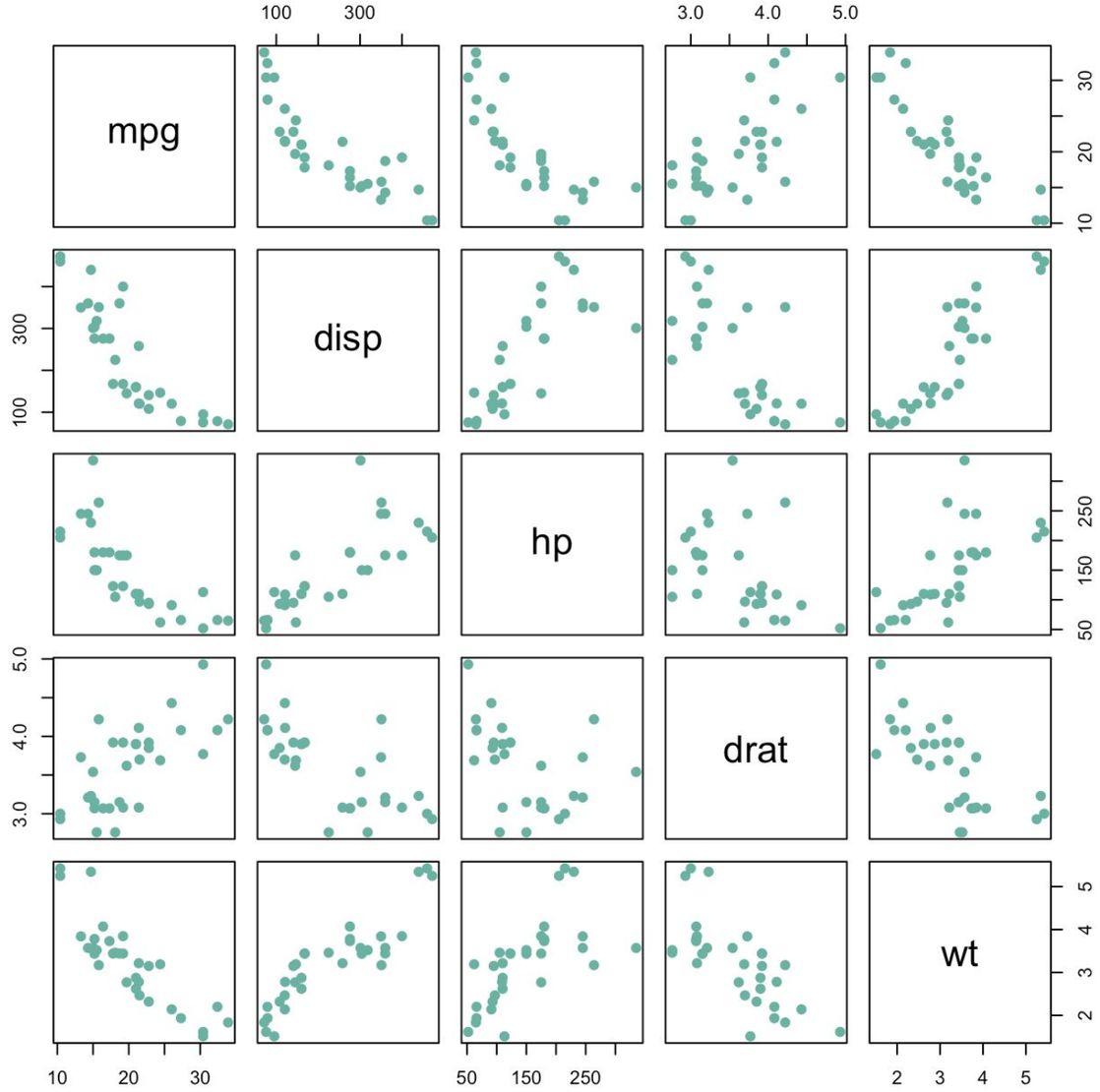


Image credit https://www.r-graph-gallery.com/98-basic-scatterplot-matrix.html

Visualisation

1) Crucial part of exploratory data analysis

    Identify problems/anomalies in your data

1) Visualisation to present results

    Convey message / summarise results

Common Visualization Mistakes

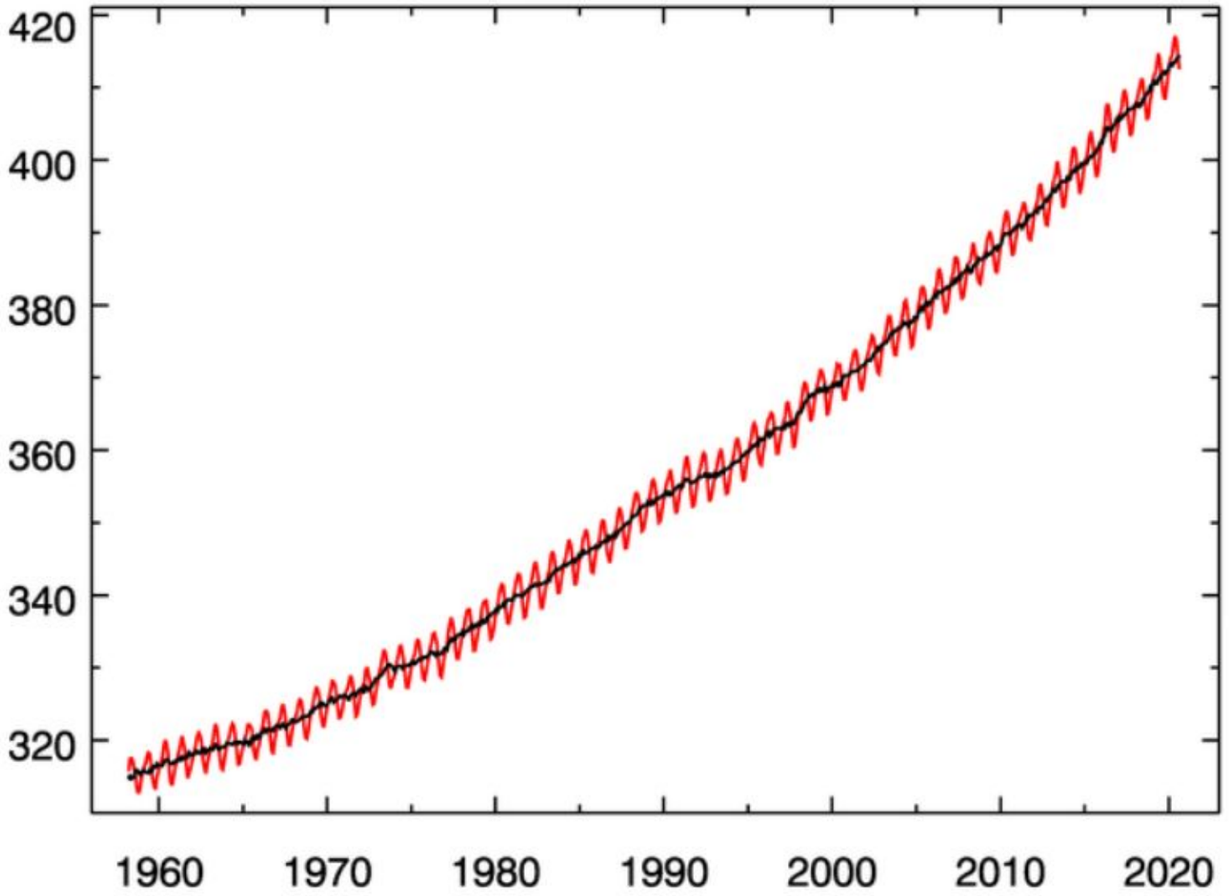


Image credit https://www.esrl.noaa.gov/gmd/ccgg/trends/mlo.html

Common Visualization Mistakes

Label your axes!

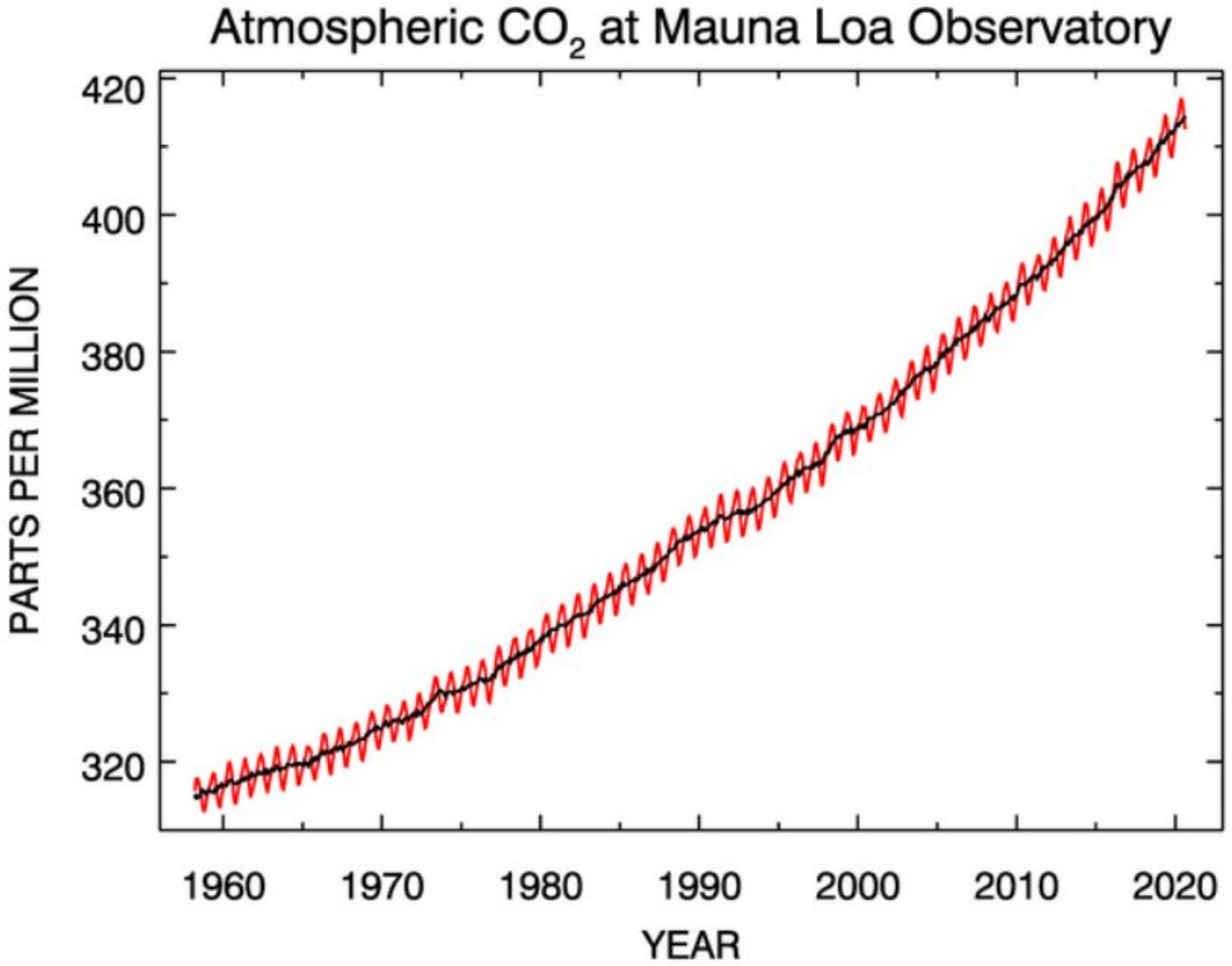
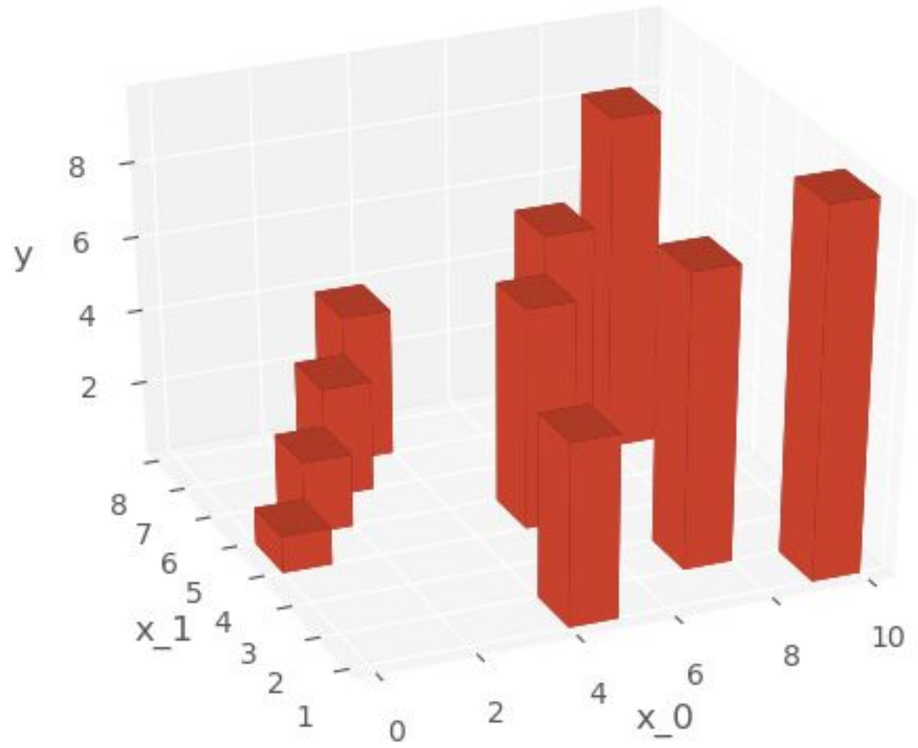


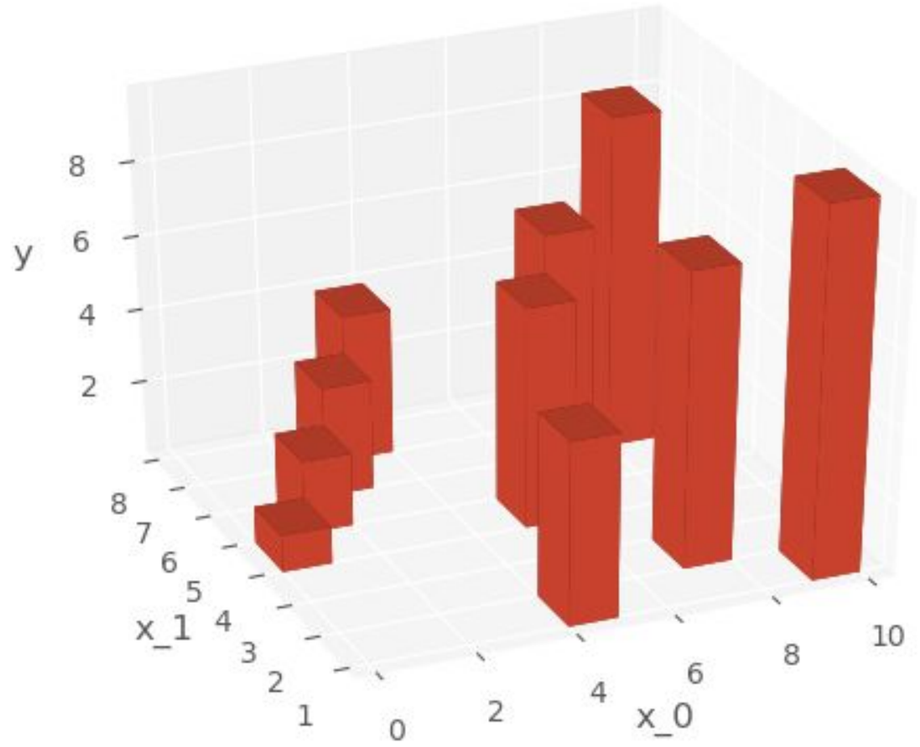
Image credit https://www.esrl.noaa.gov/gmd/ccgg/trends/mlo.html

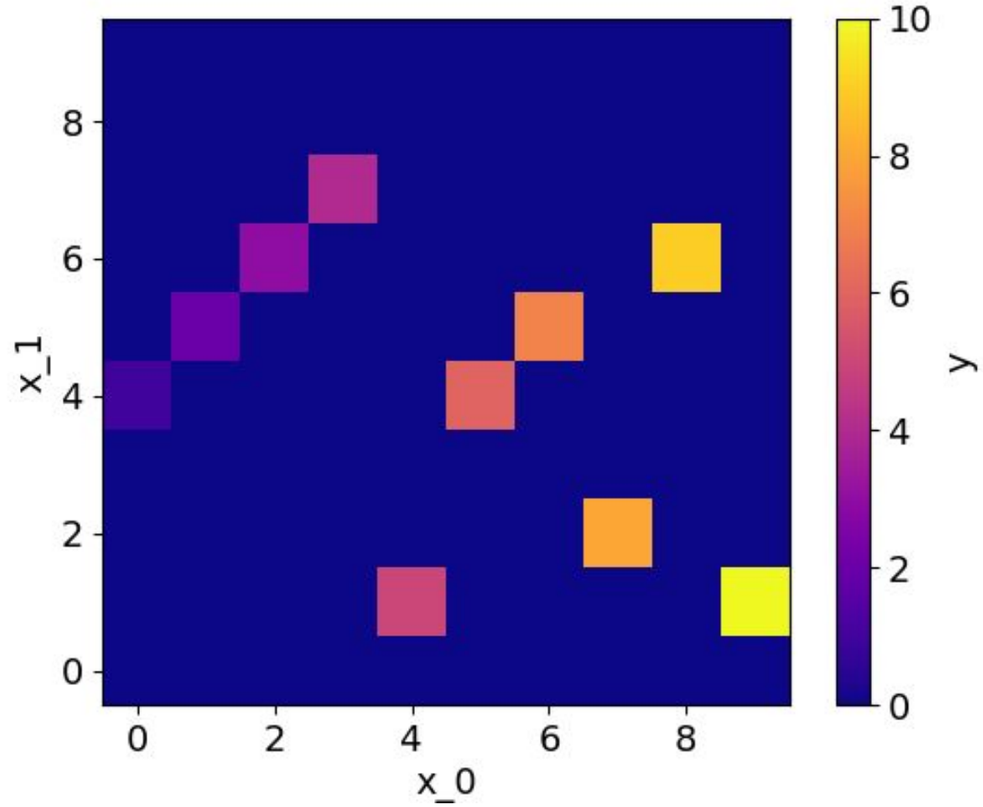
Common Visualization Mistakes



Common Visualization Mistakes

Don’t overcomplicate things e.g. 3D is rarely a good idea - use colour





Common Visualization Mistakes

Keep it “clean”

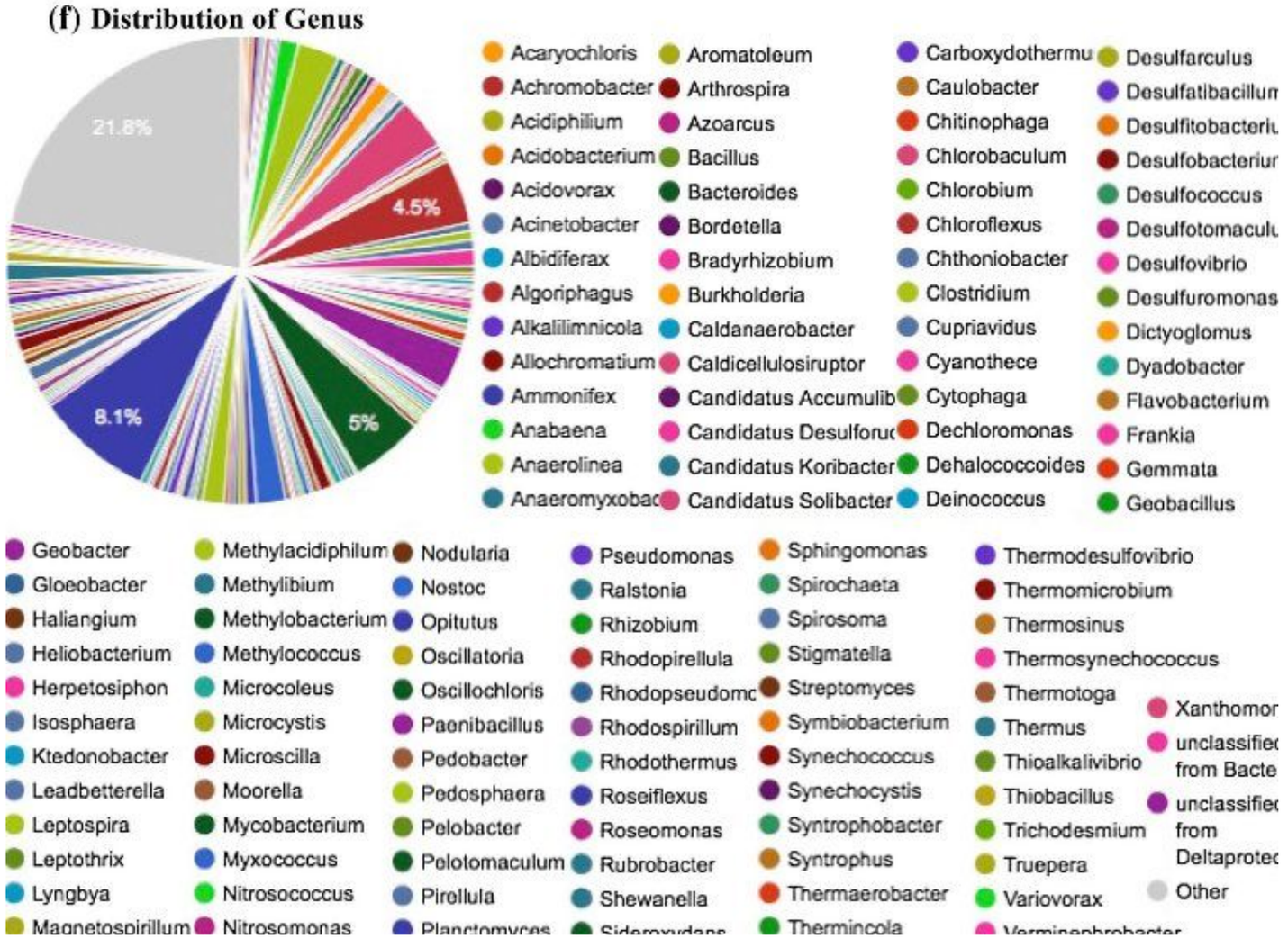
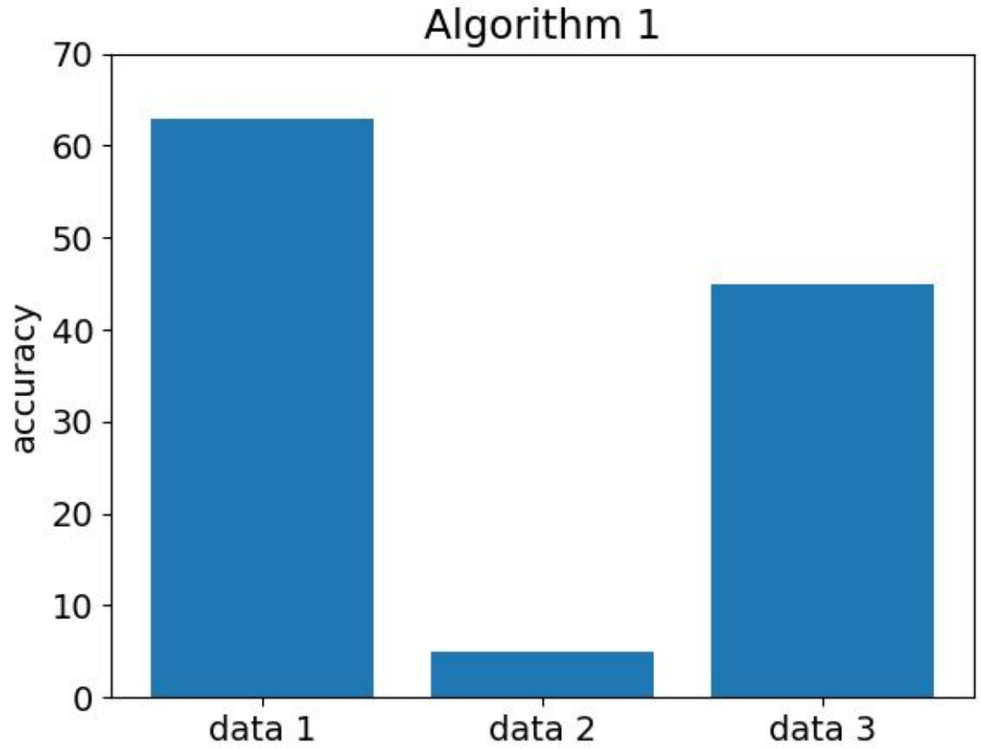
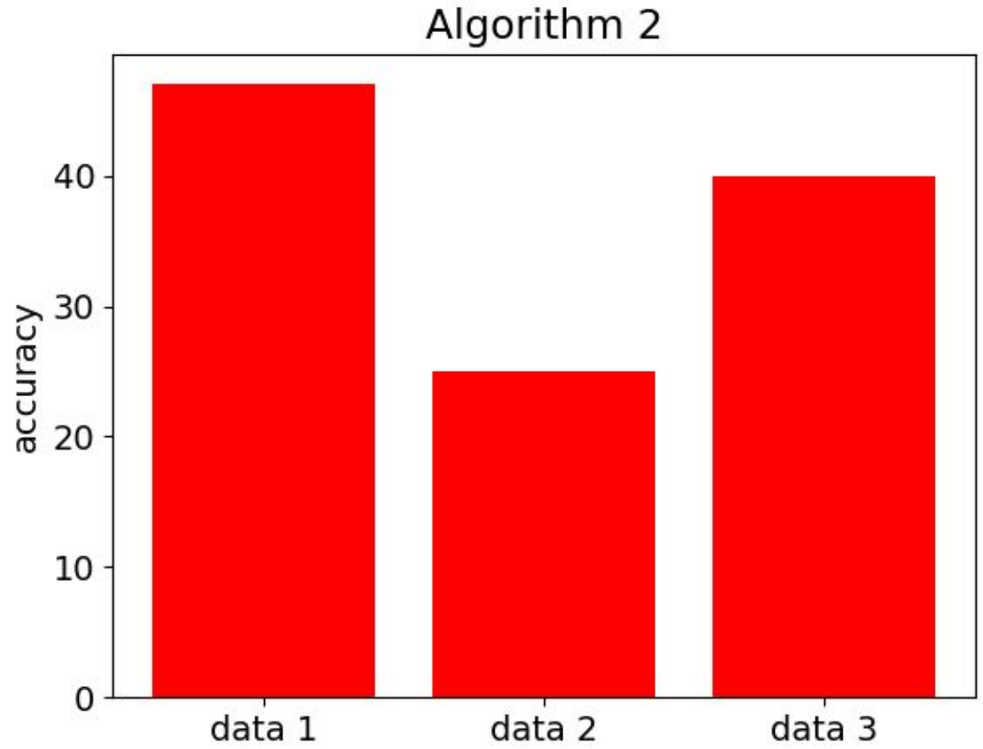


Image credit https://www.kdnuggets.com/2017/10/5-common-mistakes-bad-data-visualization.html

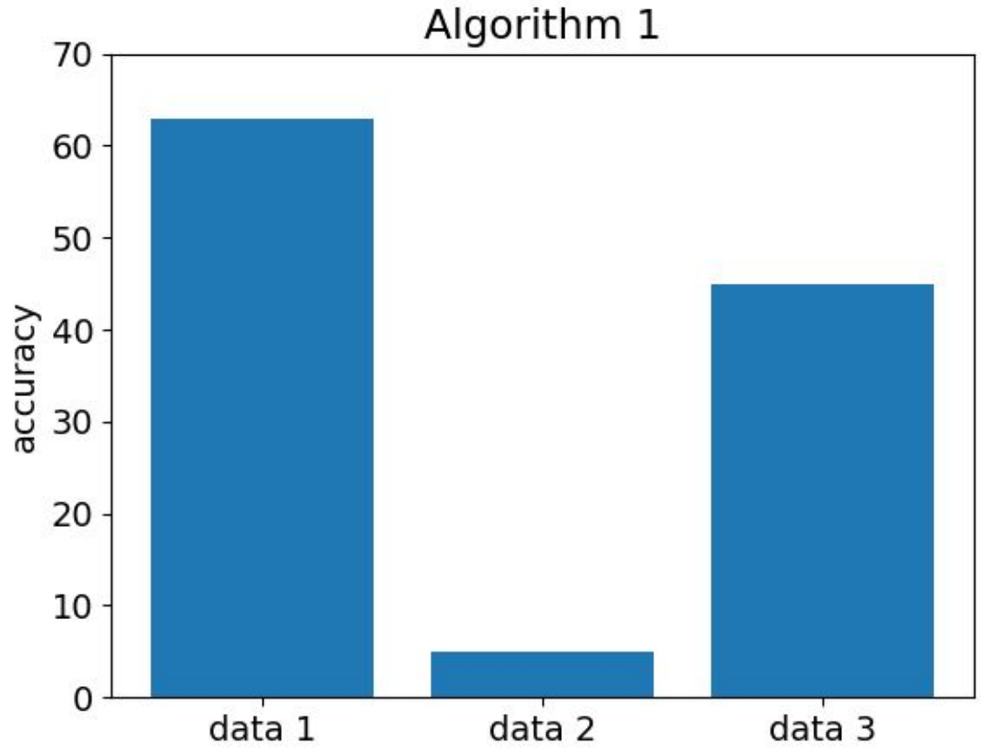
Common Visualization Mistakes

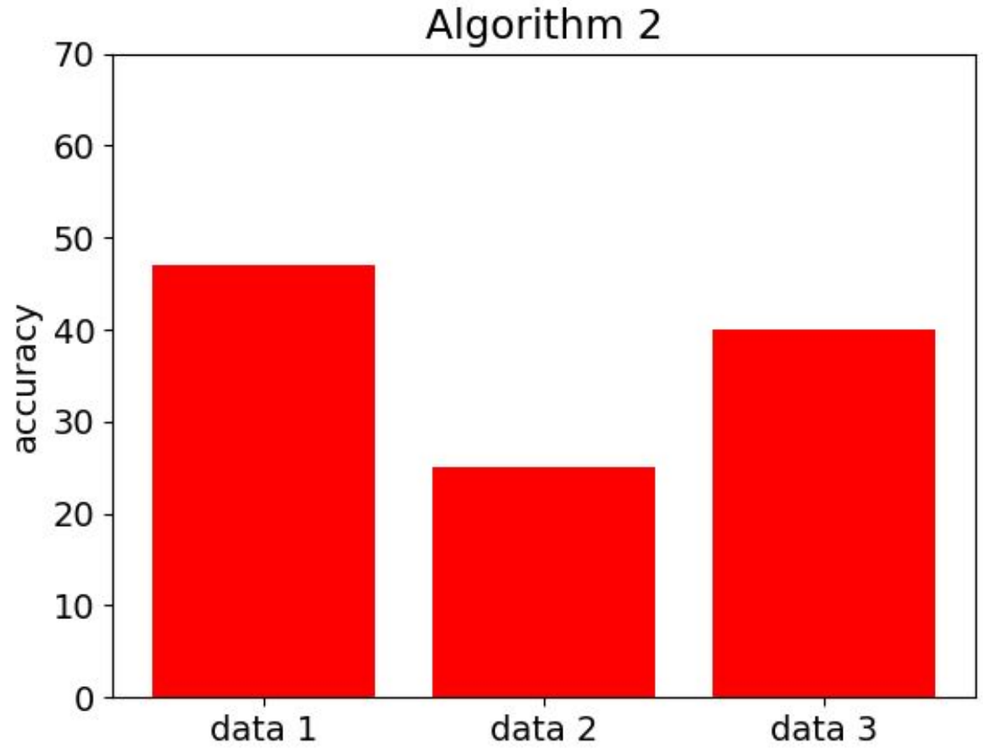




Common Visualization Mistakes

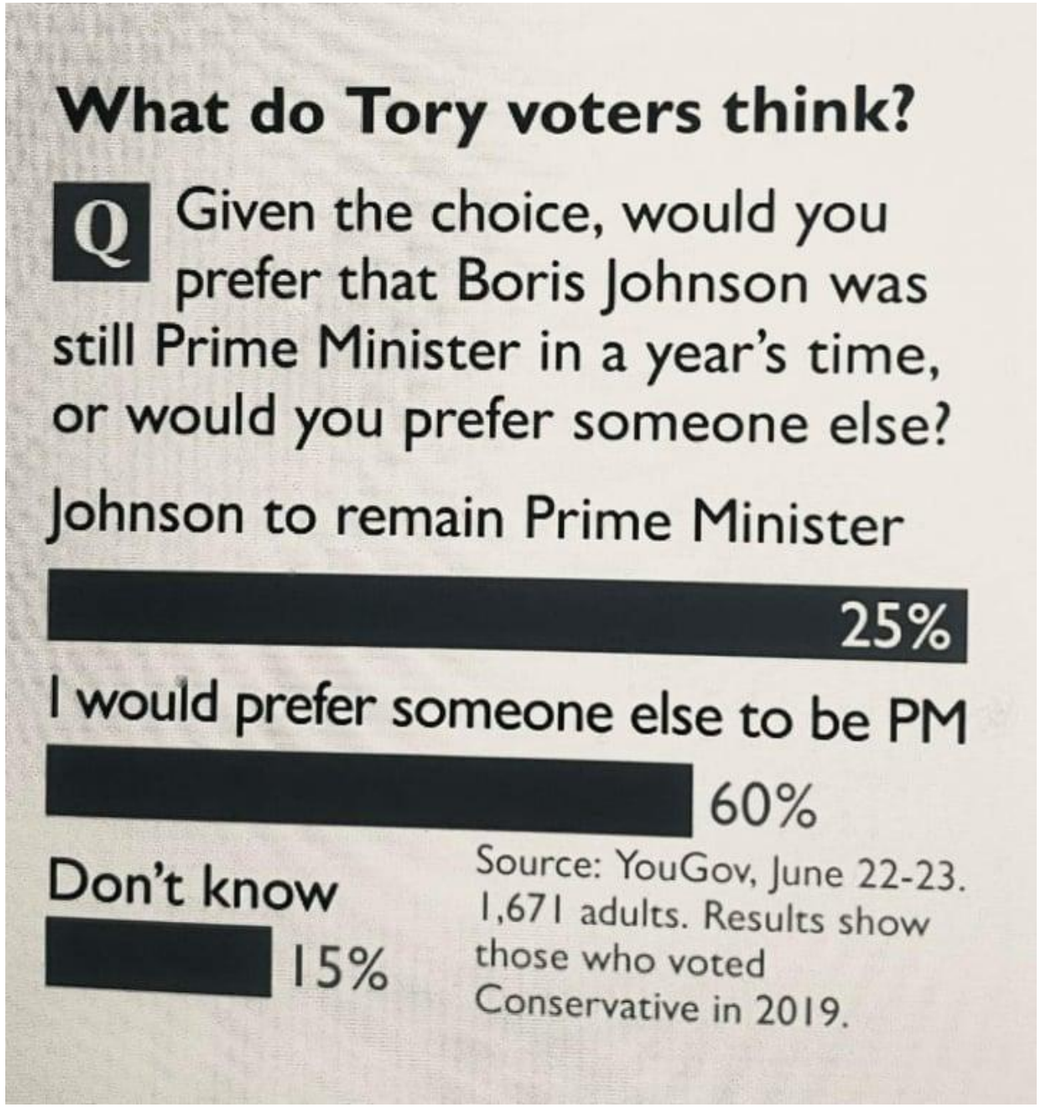
Don’t mislead





Common Visualization Mistakes

Don’t mislead



Extra Resources

GOV.UK:
https://www.gov.uk/government/publications/a-bite-sized-guide-to-visualising-data-a-dstl-biscuit-book

EU Publication Office:
https://data.europa.eu/apps/data-visualisation-guide/

Additional Links

Chartjunk - “decorations in a graphic that have no purpose”

https://www.edwardtufte.com/bboard/q-and-a-fetch-msg?msg_id=00040Z

Examples of bad plots used in media

https://www.kaiserfung.com/kaiser-fung-in-the-press

[old link: https://junkcharts.typepad.com]

Plots to avoid

https://genomicsclass.github.io/book/pages/plots_to_avoid.html

Classifying New Data - Ham Likelihood

- Given an *new* email we would like to be able to classify it
- For example, given the test email: "review us now"
- $\boldsymbol{x}_t = [0, 1, 0, 1, 0, 0]^\intercal$

- Class priors:

  $p(\text{spam}) = 4/6 \quad p(\text{ham}) = 2/6$

- Per-class likelihoods:

$p(x_d\|\text{spam})$	$p(x_d\|\text{ham})$	x_d
2/4	0/2	password
1/4	2/2	review
3/4	1/2	send
3/4	1/2	us
3/4	1/2	your
1/4	1/2	account

$$p(\boldsymbol{x}_t|\text{ham}) = p(0, 1, 0, 1, 0, 0|\text{ham})$$
$$= (1 - \frac{0}{2})(\frac{2}{2})(1 - \frac{1}{2})(\frac{1}{2})(1 - \frac{1}{2})(1 - \frac{1}{2}) = 0.0625$$

PCA: Maximising Variance

Recall Xv projects X onto v

$$\begin{aligned}\mathrm{Var}[Xv] &= \frac{1}{N}(Xv)^\top(Xv) \\ &= \frac{1}{N}v^\top X^\top Xv \\ &= v^\top \frac{X^\top X}{N} v \\ &= v^\top S v\end{aligned}$$

$max\ v^\top S v,\ \text{s.t. } v^\top v = 1$

solved using *Lagrange multipliers* as

$$max\ \underbrace{v^\top S v - \lambda\left(v^\top v - 1\right)}_{\mathcal{L}}$$

computing derivative w.r.t v and setting $= 0$

$$\frac{d\mathcal{L}}{dv} = 2Sv - 2\lambda v = 0$$

$$Sv = \lambda v \quad \square$$

$v \rightarrow$ direction of max variance

$$Sv = \lambda v$$

left multiply by $v^\top$

$$\begin{aligned}v^\top S v &= v^\top \lambda v \\ &= \lambda v^\top v \\ &= \lambda \quad \square\end{aligned}$$

$\lambda \rightarrow$ max variance

PCA: Finding Principal Components

More generally, solve for $SV = \Lambda V$ using Eigen decomposition

$$V = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_D], \; \Lambda = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_D \end{bmatrix} \quad \boldsymbol{v}_i \in \mathbb{R}^D, \; V \in \mathbb{R}^{D \times D}, \; \Lambda^{D \times D}$$

Eigenvalues

Solve $|S - \lambda I| = 0$

$$\begin{vmatrix} 2.0 - \lambda & 0.8 \\ 0.8 & 0.6 - \lambda \end{vmatrix} = 0$$

$$\lambda^2 - 2.6\lambda + 0.56 = 0$$

$$\implies \{\lambda_1, \lambda_2\} = \{2.36, 0.23\}$$

Eigenvectors

Find i^{th} eigenvector by solving $S\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i$

$$\begin{bmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{bmatrix} \begin{bmatrix} v_{1,1} \\ v_{1,2} \end{bmatrix} = 2.36 \begin{bmatrix} v_{1,1} \\ v_{1,2} \end{bmatrix} \implies \boldsymbol{v}_1 = \begin{bmatrix} 2.2 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{bmatrix} \begin{bmatrix} v_{2,1} \\ v_{2,2} \end{bmatrix} = 0.23 \begin{bmatrix} v_{2,1} \\ v_{2,2} \end{bmatrix} \implies \boldsymbol{v}_2 = \begin{bmatrix} -0.41 \\ 0.91 \end{bmatrix}$$

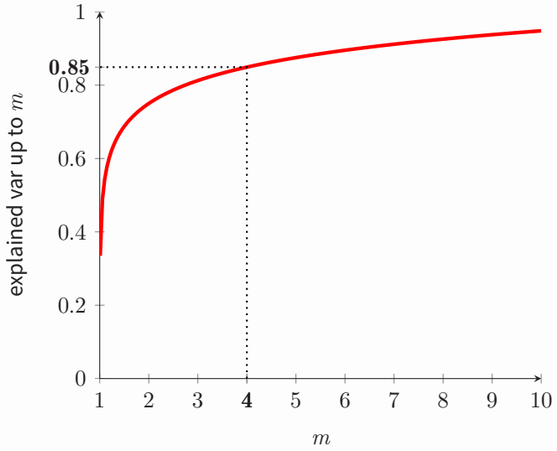
PCA: Picking number of dimensions

Given: eigenvectors $V = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_D]$; **Require:** $M \ll D$

Known: eigenvalue λ_i = variance along $\boldsymbol{v}_i$

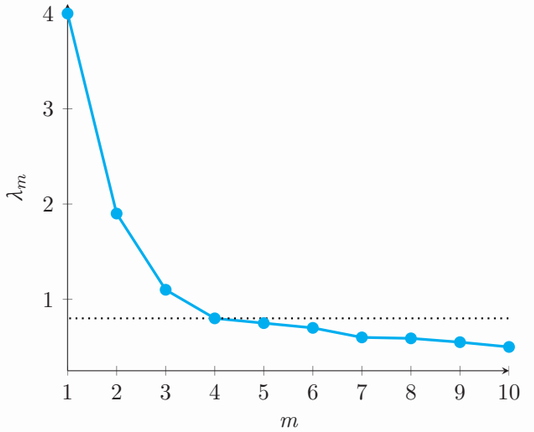
Explained variance

- sort eigenvectors s.t. $\lambda_1 \geq \ldots \geq \lambda_D$
- choose top M eigenvectors that explain "most" variance (typically 85%, 90%, or 95%)



Elbow plot

- plot eigenvalues in descending order $\lambda_1 \geq \ldots \geq \lambda_D$
- choose point at which curve "bends" most (i.e. elbow)



PCA: Dimensionality Reduction

Let $V_M = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_M] \in \mathbb{R}^{D \times M}$ denote the *truncated* eigenvector matrix for $M \ll D$

Reduction

Dimensionality reduction on data $\boldsymbol{x}_i$

$$\boldsymbol{e}_i^\top = \boldsymbol{x}_i^\top V_M \quad \in \mathbb{R}^M$$

More generally, projected data E

$$\begin{aligned} E &= \left[\boldsymbol{e}_1^\top, \ldots, \boldsymbol{e}_N^\top\right] \\ &= \left[\boldsymbol{x}_1^\top V_M, \ldots, \boldsymbol{x}_N^\top V_M\right] \\ &= X V_M \quad \in \mathbb{R}^{N \times M} \end{aligned}$$

Reconstruction

Recover data $\hat{\boldsymbol{x}}_i$ from $\boldsymbol{e}_i$ using $V_M^\top$

$$\hat{\boldsymbol{x}}_i^\top = \boldsymbol{e}_i^\top V_M^\top = \left(\boldsymbol{x}_i^\top V_M\right) V_M^\top \quad \in \mathbb{R}^D$$

More generally, reconstructed data $\hat{X}$

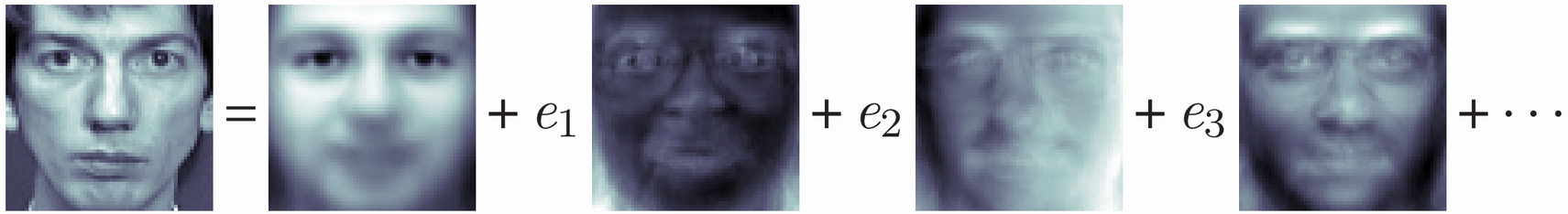
$$\begin{aligned} \hat{X} &= \left[\hat{\boldsymbol{x}}_1^\top, \ldots, \hat{\boldsymbol{x}}_N^\top\right] \\ &= X V_M V_M^\top \quad \in \mathbb{R}^{N \times D} \end{aligned}$$

$V_M V_M^\top \in \mathbb{R}^{D \times D}$ is the data *projection* matrix

PCA Example 2: Eigenfaces

Projection

Projecting face $\boldsymbol{x}_i$ onto $\boldsymbol{e}_i = [e_{i1}, \ldots, e_{iM}]$



Reconstruction

Reconstructing face $\hat{\boldsymbol{x}}_i$ using M components



$M = 10$ $M = 30$ $M = 50$ $M = 70$ $M = 90$

$(90 \ll 4096!)$

The MovieFlix Corporation wants to improve the algorithm it uses to recommend new movies to its customers. It has decidede to release a portion of its data to the public in the hope that researchers will be able to find patterns in users' preferences for the different movies. The data is represented as a matrix X where rows $i = 1, \ldots, n$ correspond to customers, and columns $j = 1, \ldots, c$ correspond to movies, and X_{ij} is a numeric value that reflects the degree to which customer i enjoyed movie j. Assume the data is complete; i.e., we know the rating of every user for every movie.

MovieFlix decides to carry out PCA on X treating each row (i.e. each customer) as a data vector. How will this help them understand the patterns in the preference data?

PCA relationship between X and X^T

Instead of thinking about the data as rows of customer responses, we can think of a movie as the response it elicits from N customers.

PCA relationship between X and X^T

Instead of thinking about the data as rows of customer responses, we can think of a movie as the response it elicits from N customers.

recall $Sv = \lambda v$

$\left(\frac{1}{N} X^T X\right) v = \lambda v \quad \left(\because S = \frac{1}{N} X^T X\right)$

pre-multiply by X

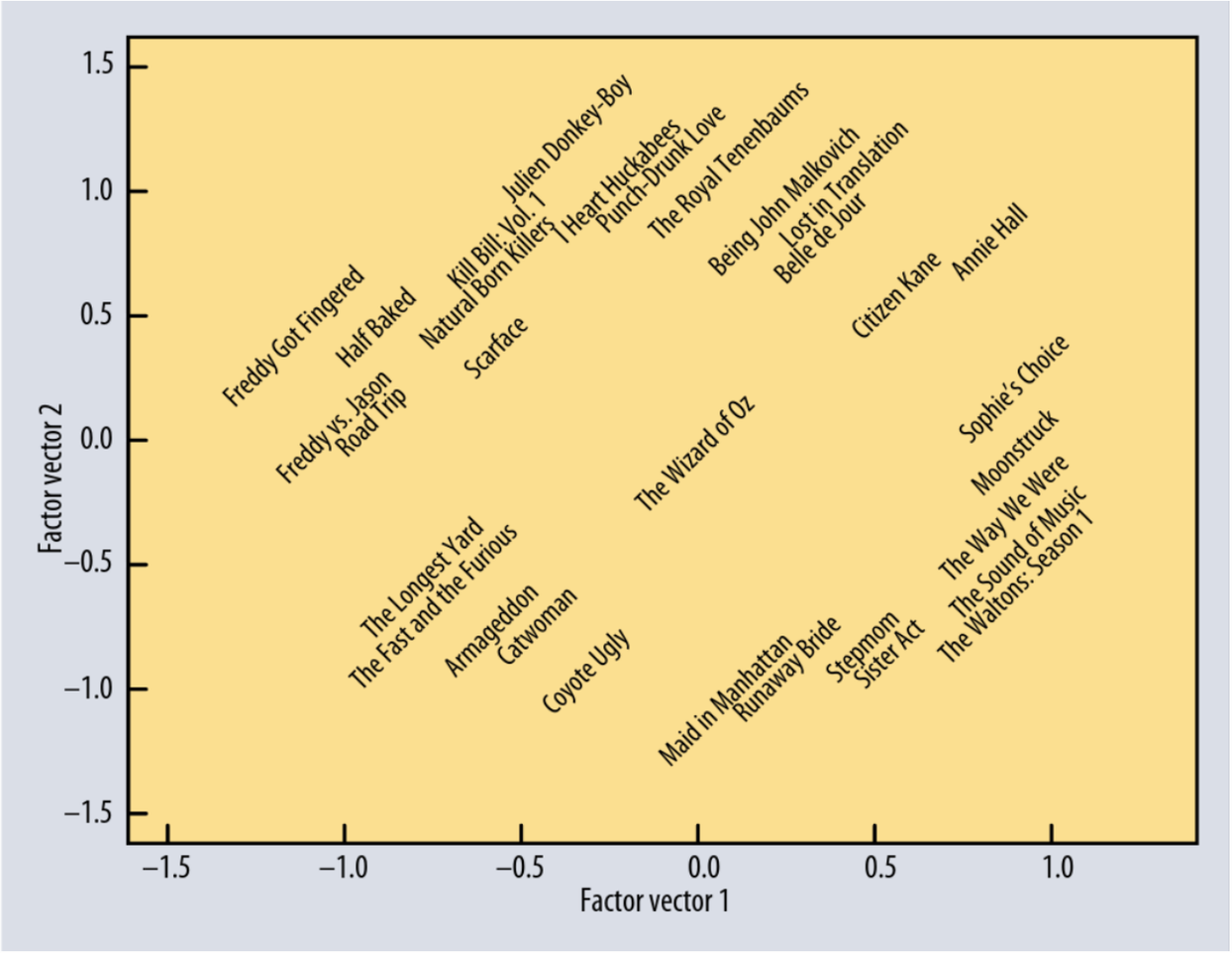
$\left(\frac{1}{N} X X^T\right)(Xv) = \lambda(Xv) \Rightarrow S'v' = \lambda v'$

$S' = \frac{1}{N}(X^T)^T X^T = \frac{1}{N} X X^T$

v' new PC direction

↳ same eigenvalues!

In the famous Netflix challenge, as similar low-rank idea can be used as in PCA, but almost all of the entries in X are missing—not every person has seen every film!



credit: Yehuda Koren, Robert Bell, & Chris Volinsky, Matrix Factorization Techniques for Recommender Systems