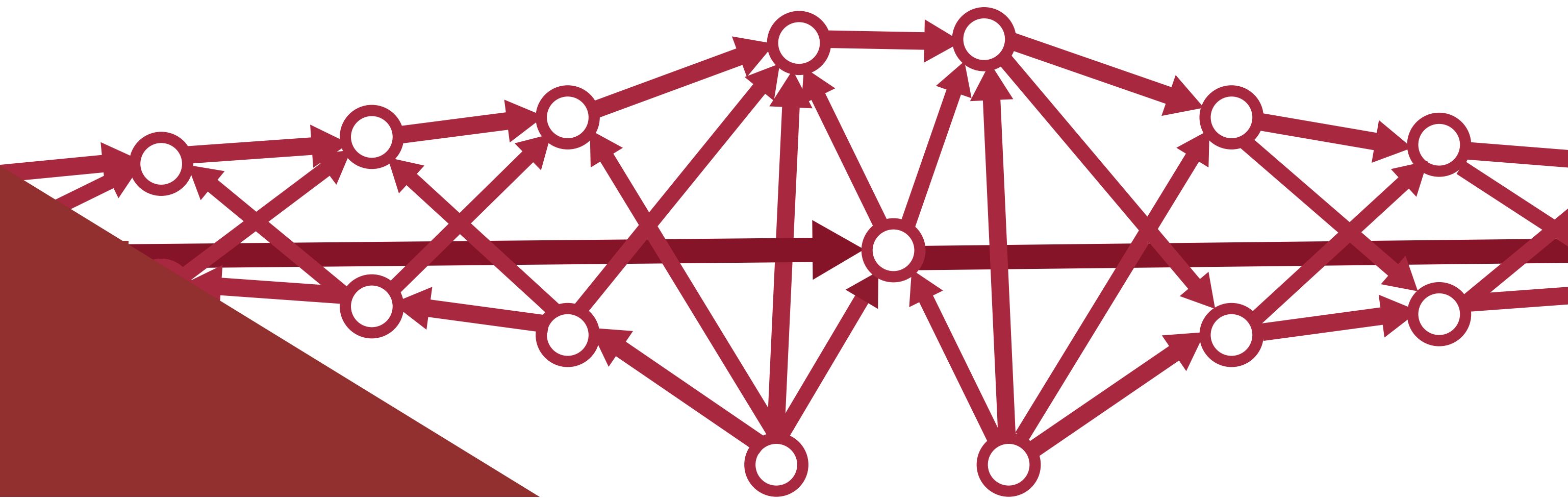


Machine Learning & Data Analysis



Week 4: Visualising data

Elliot J. Crowley, 12/10/26



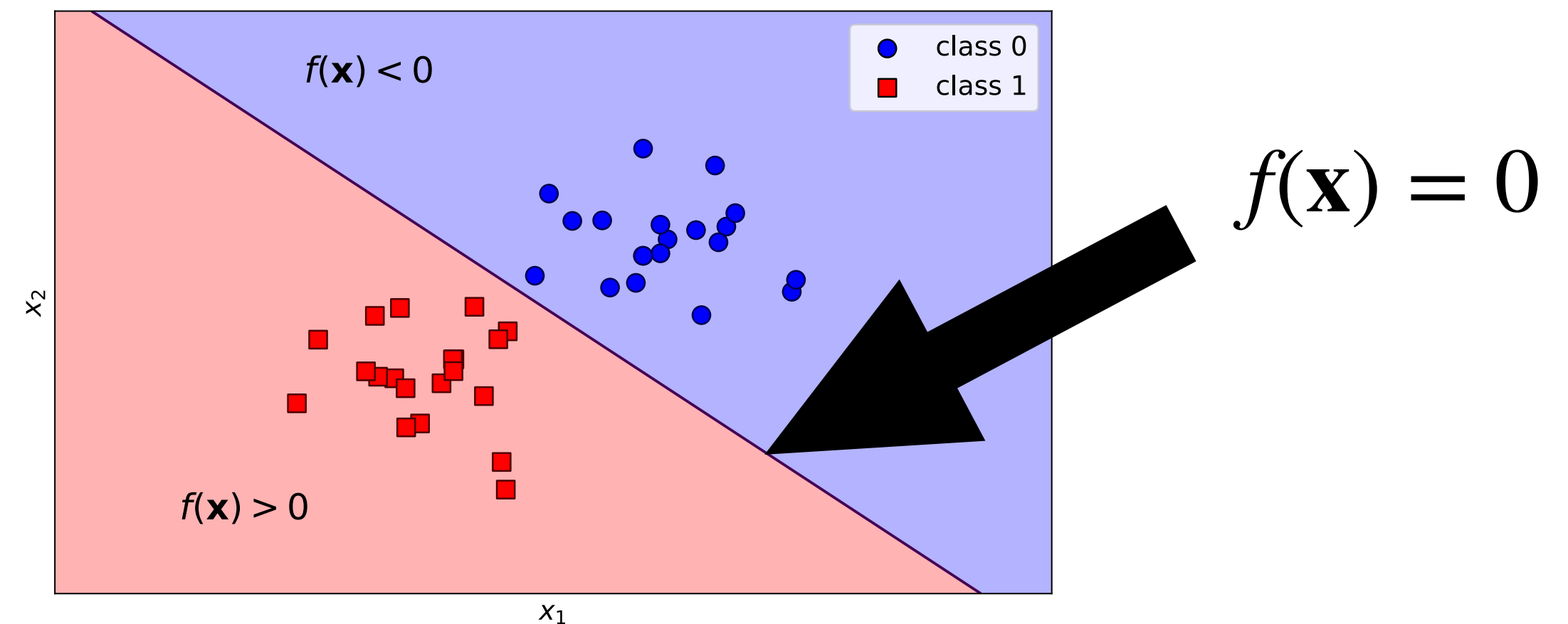
THE UNIVERSITY
of EDINBURGH

Recap

- We learnt that a linear classifier consisted of a linear model + threshold function
- We saw that this gave rise to a (linear) decision boundary

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$

$$\hat{y} = \begin{cases} 1 & \text{if } f(\mathbf{x}) > 0 \\ 0 & \text{if } f(\mathbf{x}) < 0 \end{cases}$$



- We looked at different loss functions for fitting the model parameters

Why “Machine Learning and Data Analysis”?

- Analysing your data is **essential**
- Machine learning algorithms are only as good as the data they receive
- The way you represent your data for ML is extremely important
- If you can't spot a pattern in the data, there may not be one!

Look before you leap (leap = do ML)

- Do the values for your features make sense?

Covid: Man offered vaccine after error lists him as 6.2cm tall

🕒 18 February 2021



Coronavirus pandemic



LIAM THORP

Liam Thorp was wrongly classed as morbidly obese according to his height and weight

A man in his 30s with no underlying health conditions was offered a Covid vaccine after an NHS error mistakenly listed him as just 6.2cm in height.

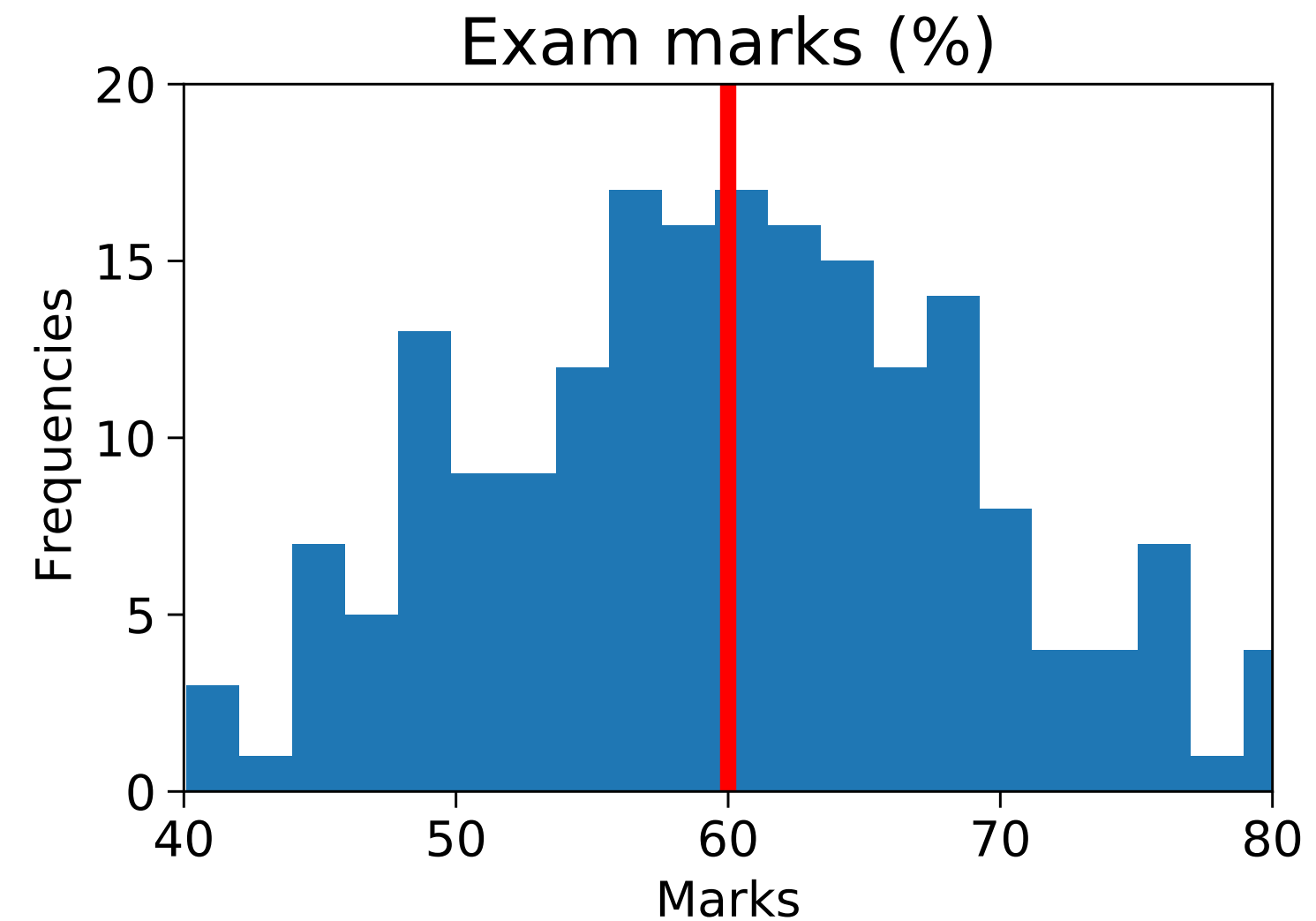
Source: <https://www.bbc.co.uk/news/uk-england-merseyside-56111209>

Look at feature means

- For feature X we have N measurements $\{x^{(n)}\}_{n=1}^N$

$$\mu_x = \frac{1}{N} \sum_{n=1}^N x^{(n)}$$

	Mark (%)
0	60
1	40
2	45
...	...



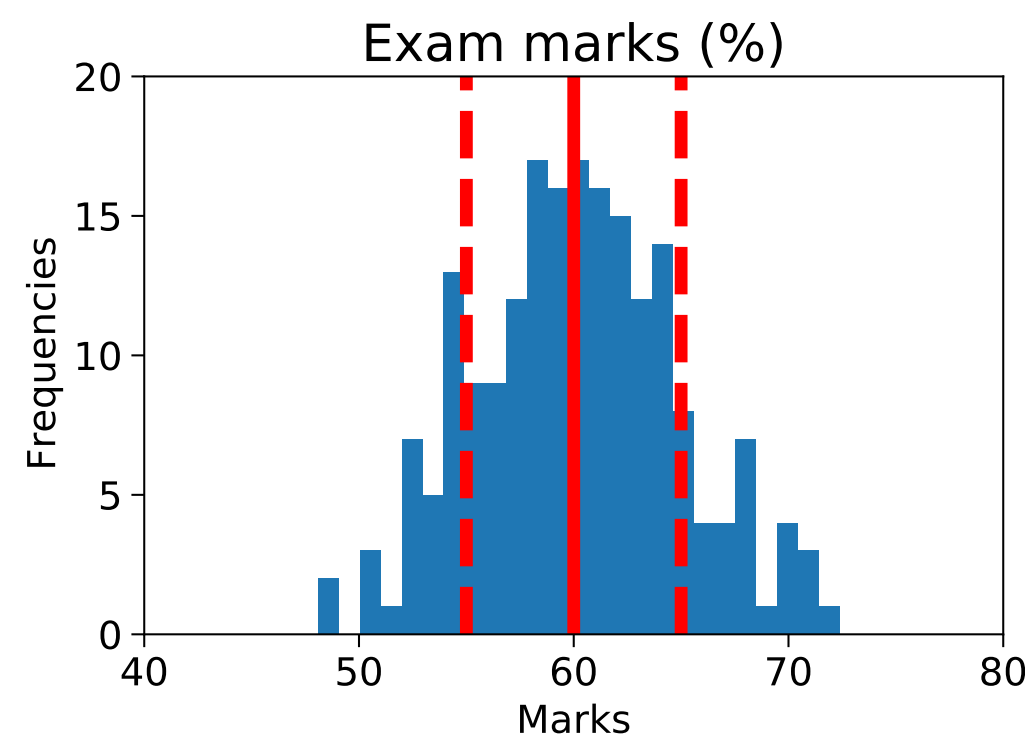
Look at feature variances/SDs

- Let's denote variance as σ^2 and standard deviation (SD) as σ
- For feature X we have N measurements $\{x^{(n)}\}_{n=1}^N$

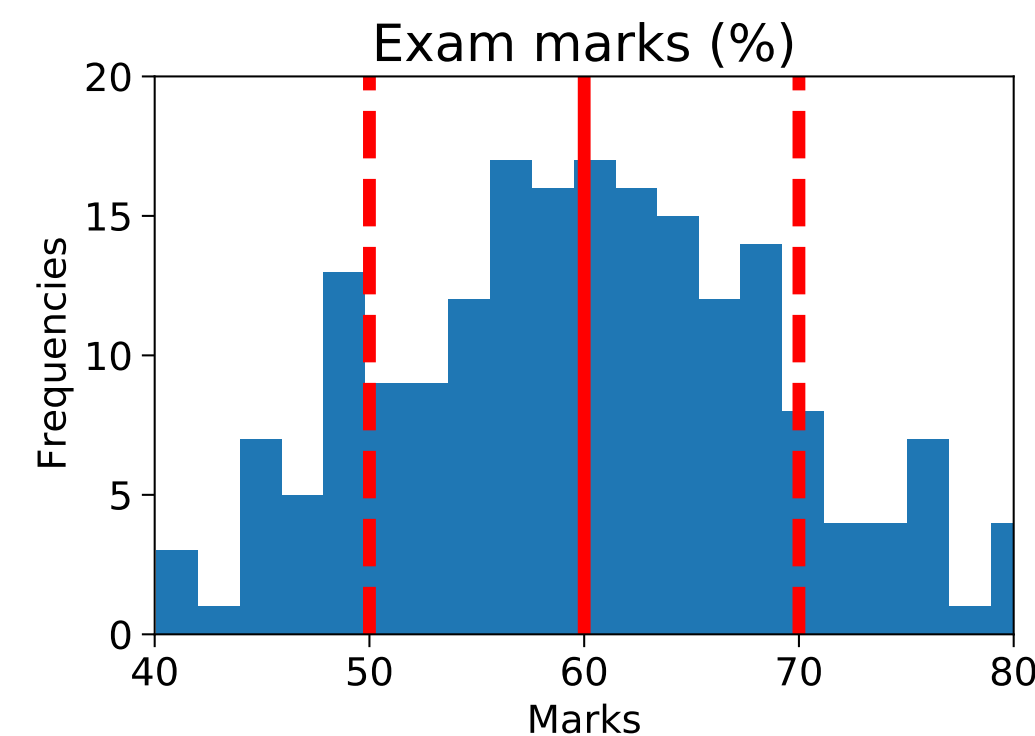
$$\sigma_x^2 = \frac{1}{N} \sum_{n=1}^N (x^{(n)} - \mu_x)^2$$

- SD measures the extent to which measurements deviate from the mean

$$\sigma = 5$$



$$\sigma = 10$$

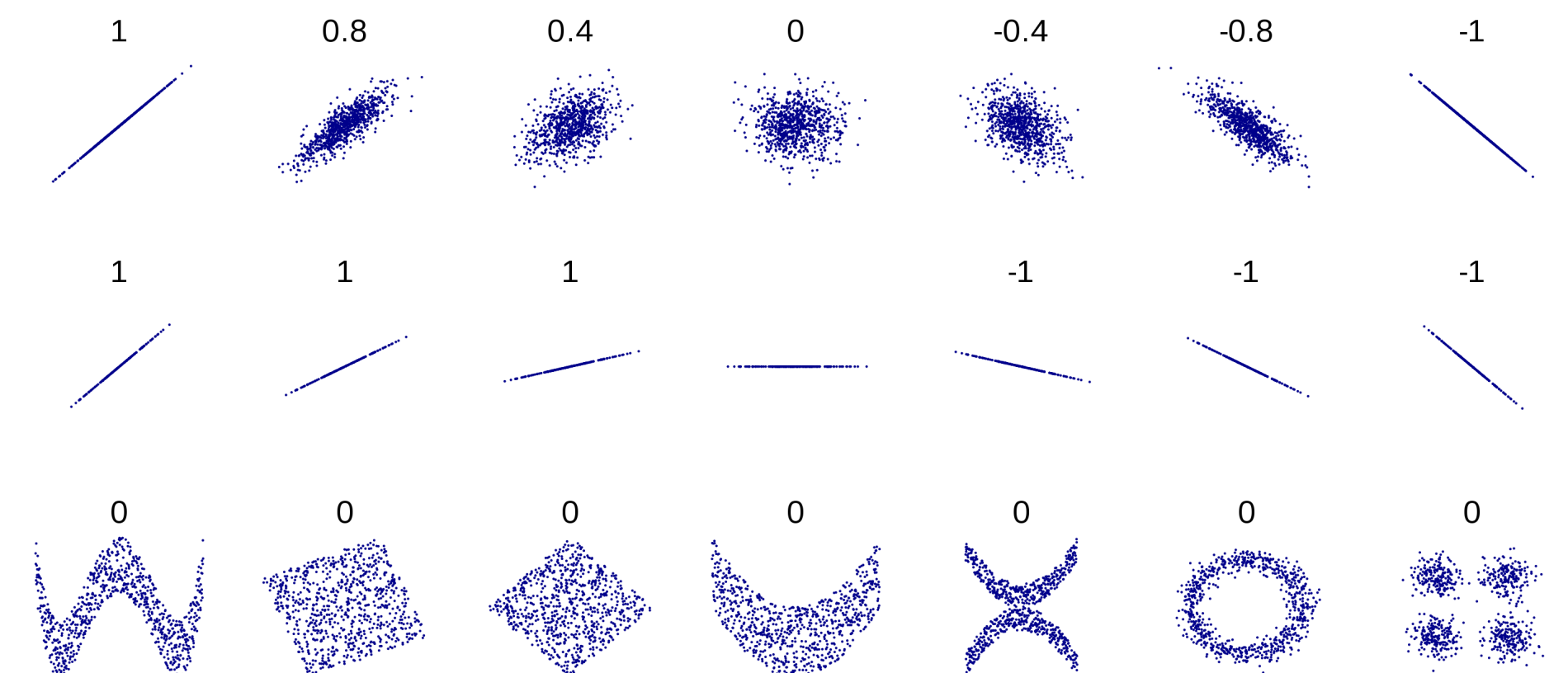


Look at correlations between features

- We have two features X and Y each with N (paired) measurements
- Let's compute the means and SDs of each: $\mu_x, \mu_y, \sigma_x, \sigma_y$
- The covariance $\sigma_{x,y}$ and **Pearson correlation coefficient** $\rho_{x,y}$ are given by:

$$\sigma_{x,y} = \frac{1}{N} \sum_{n=1}^N (x^{(n)} - \mu_x)(y^{(n)} - \mu_y)$$

$$\rho_{x,y} = \frac{\sigma_{x,y}}{\sigma_x \sigma_y}$$



Key summary statistics

Mean, std, and percentiles

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
count	150.00	150.00	150.00	150.00
mean	5.84	3.06	3.76	1.20
std	0.83	0.44	1.77	0.76
min	4.30	2.00	1.00	0.10
25%	5.10	2.80	1.60	0.30
50%	5.80	3.00	4.35	1.30
75%	6.40	3.30	5.10	1.80
max	7.90	4.40	6.90	2.50

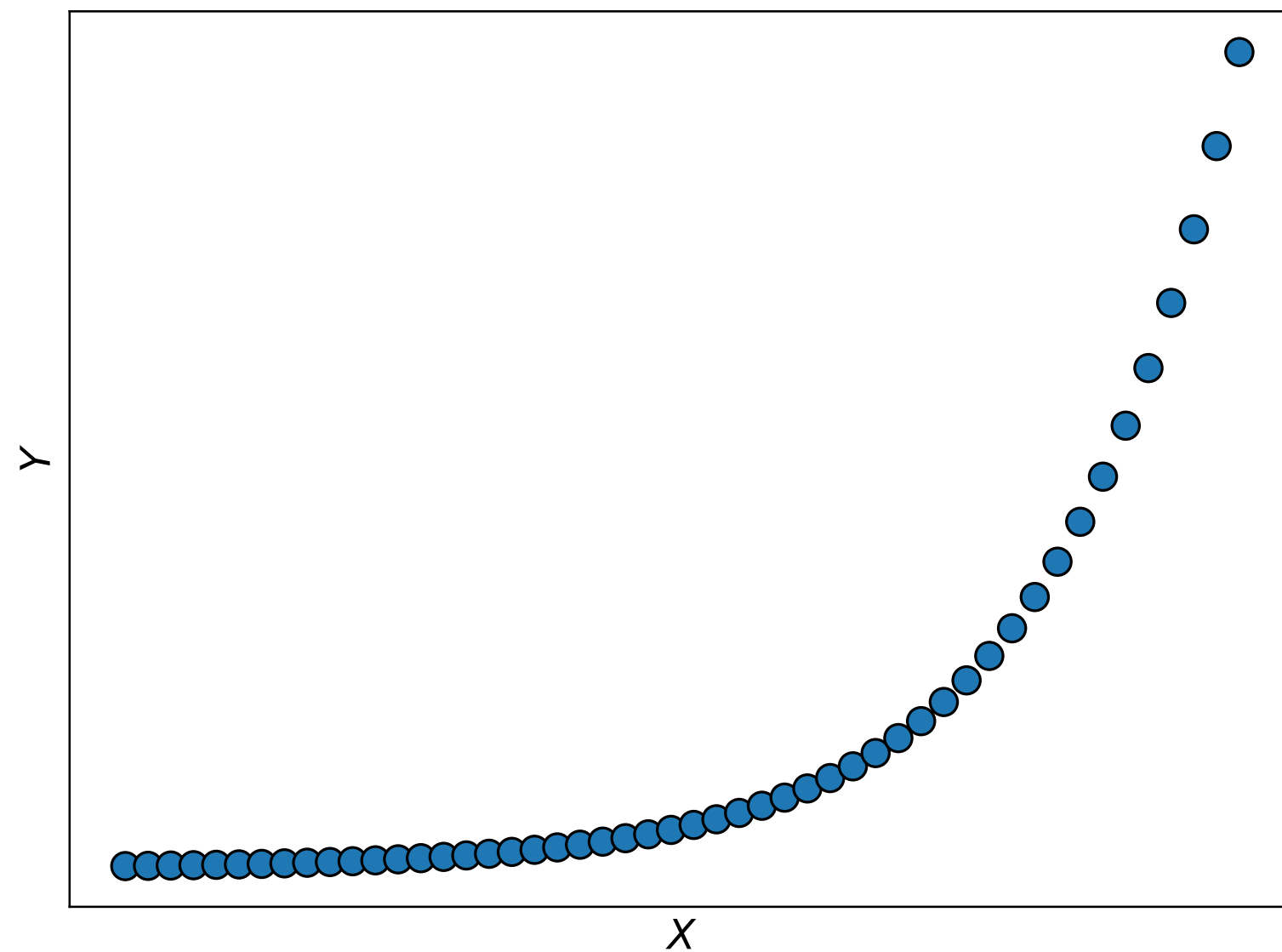
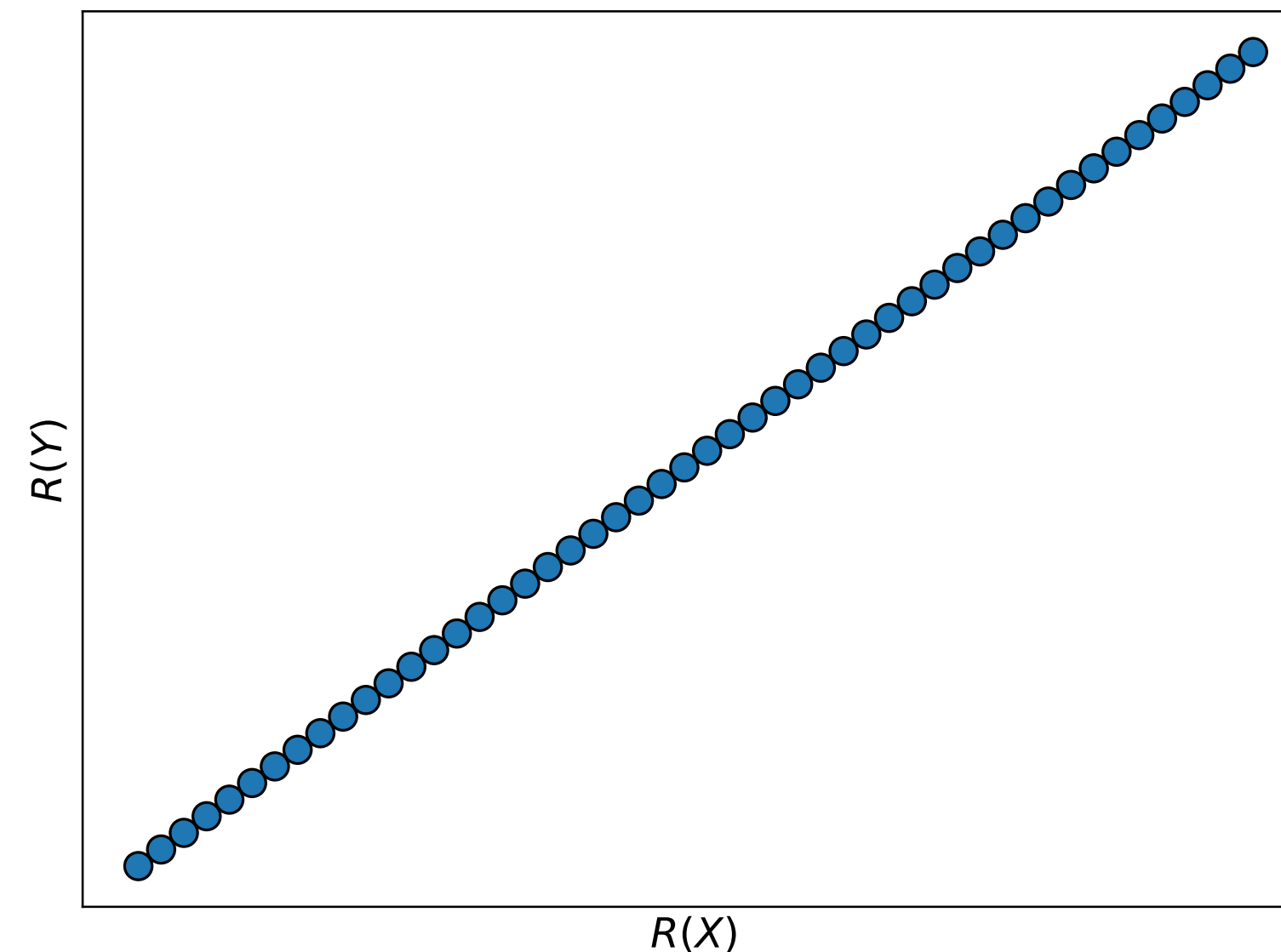
(Pearson) Correlations

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
sepal length (cm)	1.00	-0.12	0.87	0.82
sepal width (cm)	-0.12	1.00	-0.43	-0.37
petal length (cm)	0.87	-0.43	1.00	0.96
petal width (cm)	0.82	-0.37	0.96	1.00

The 50% percentile is the median which is a robust statistic

Spearman's rank correlation coefficient r_s

- This is the Pearson correlation coefficient of the **ranks** of two features
- e.g. if $X = \{1, 100, 1000, 10000\}$, $R(X) = \{1, 2, 3, 4\}$
- It measures how monotonic the relationship between the features is

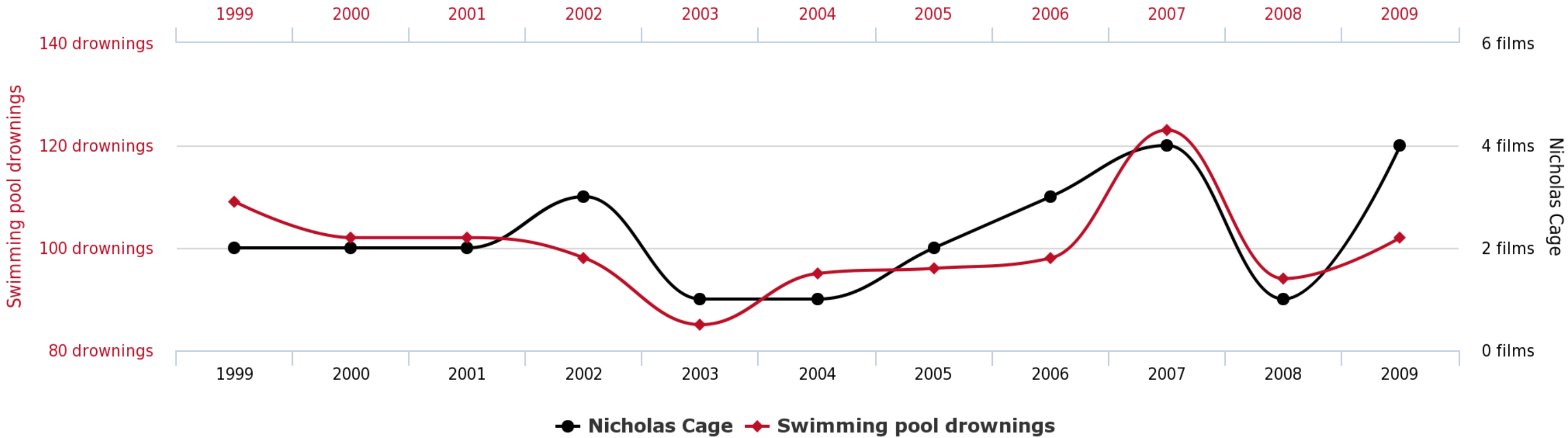


The Pearson correlation here is 0.82
The Spearman correlation is 1

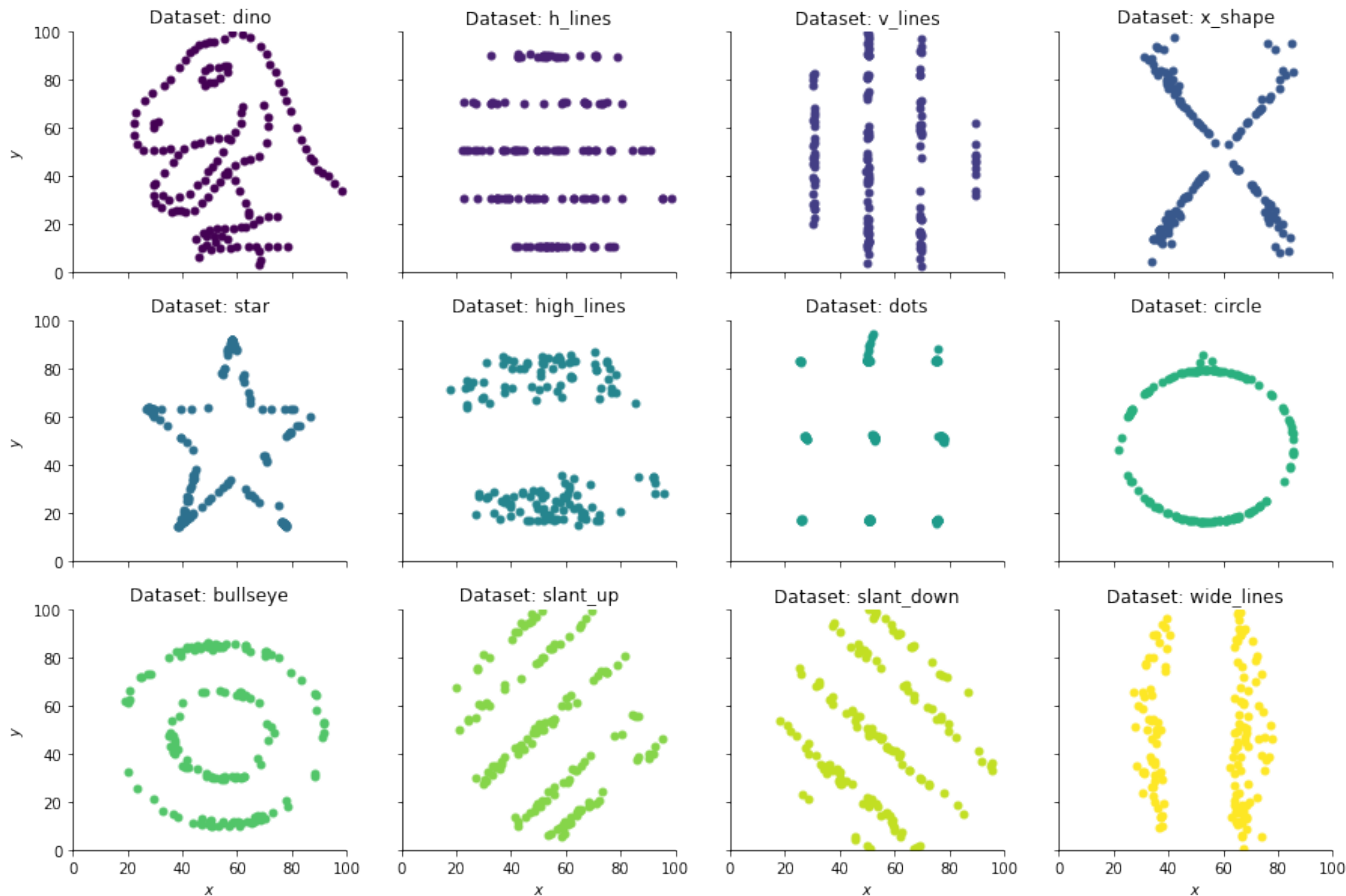
Correlation does not imply causation



Number of people who drowned by falling into a pool
correlates with
Films Nicolas Cage appeared in



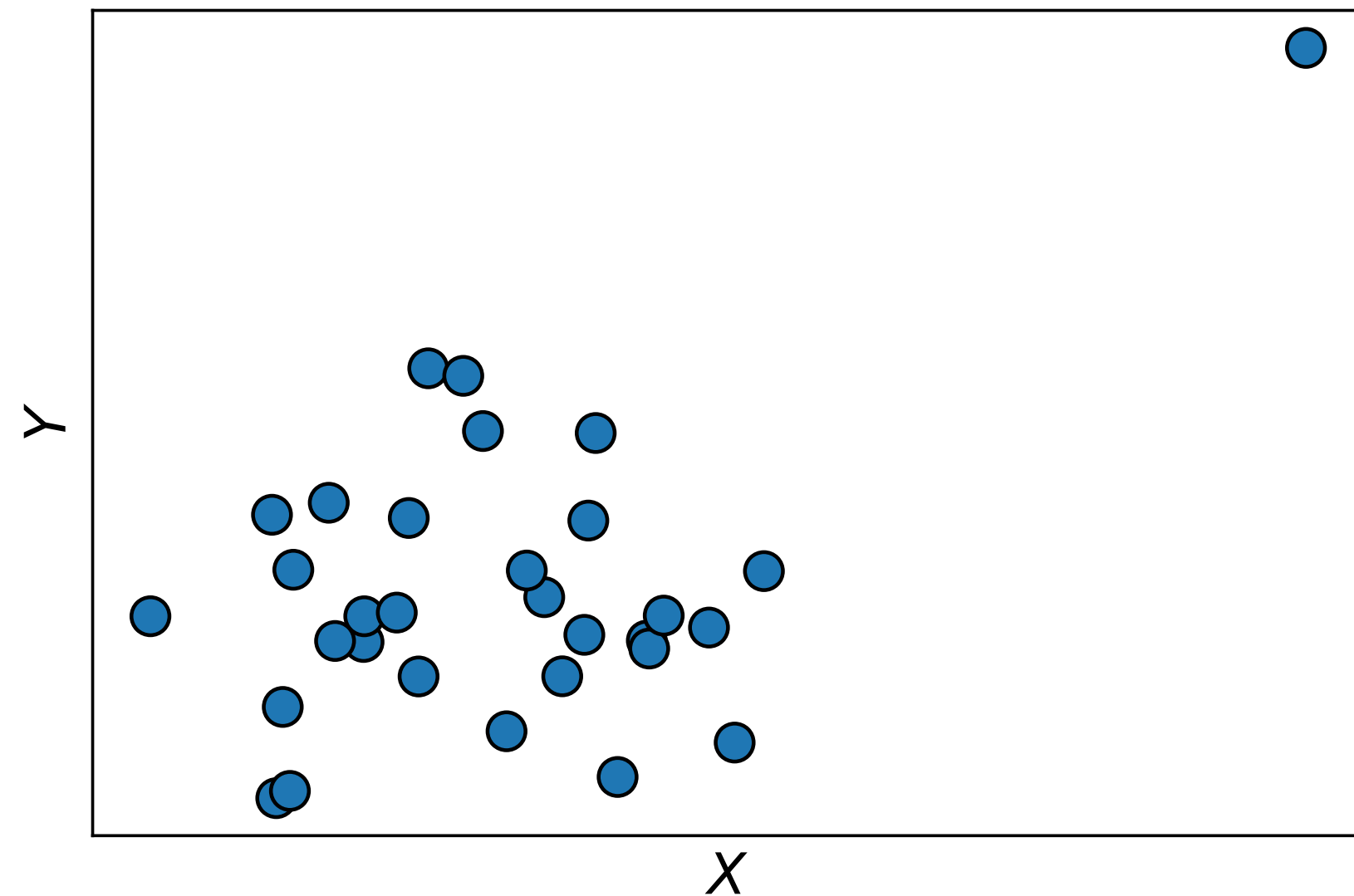
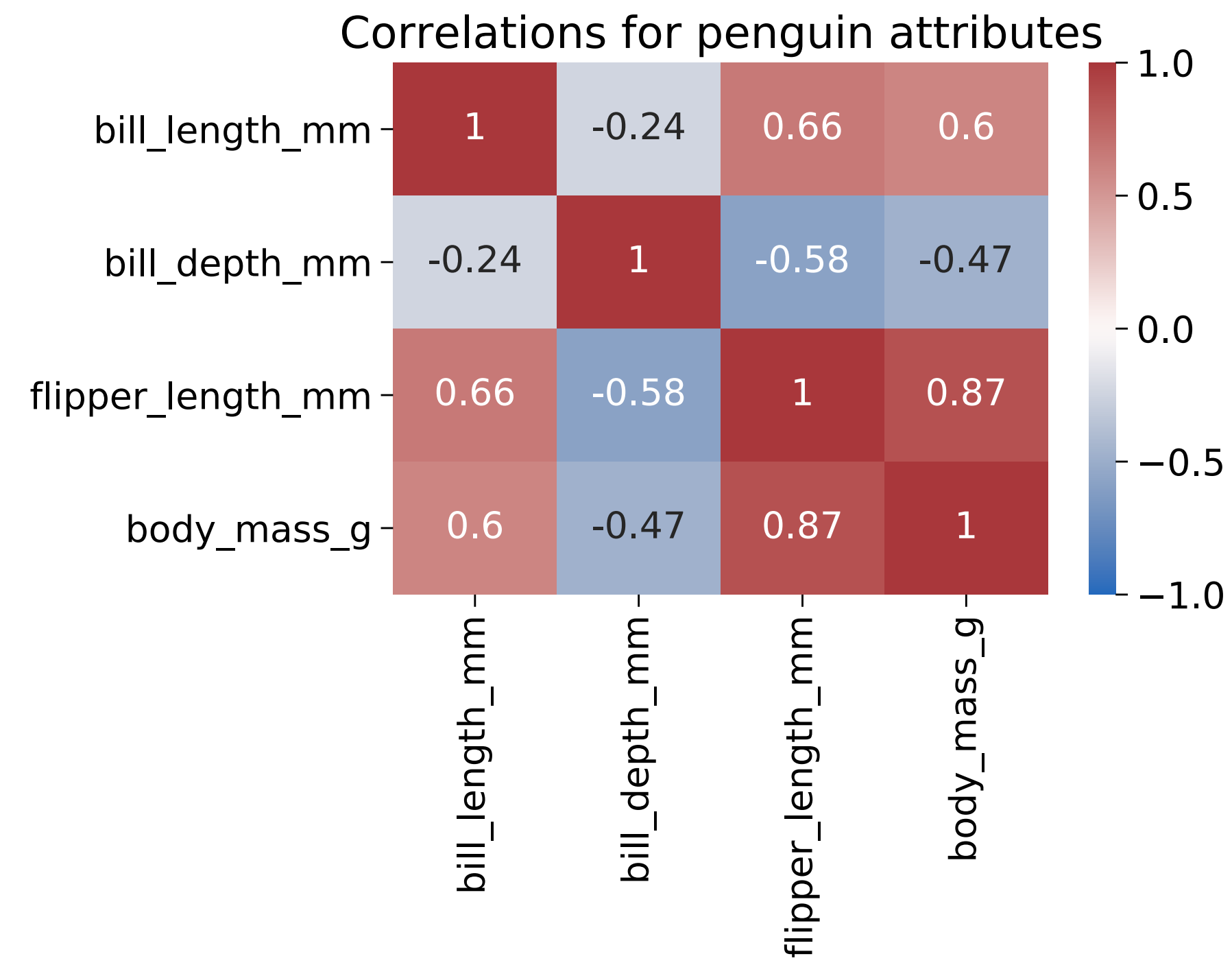
The limitations of summary statistics



All 12 of these datasets have the same $\mu_x, \mu_y, \sigma_x, \sigma_y, \rho_{x,y}$

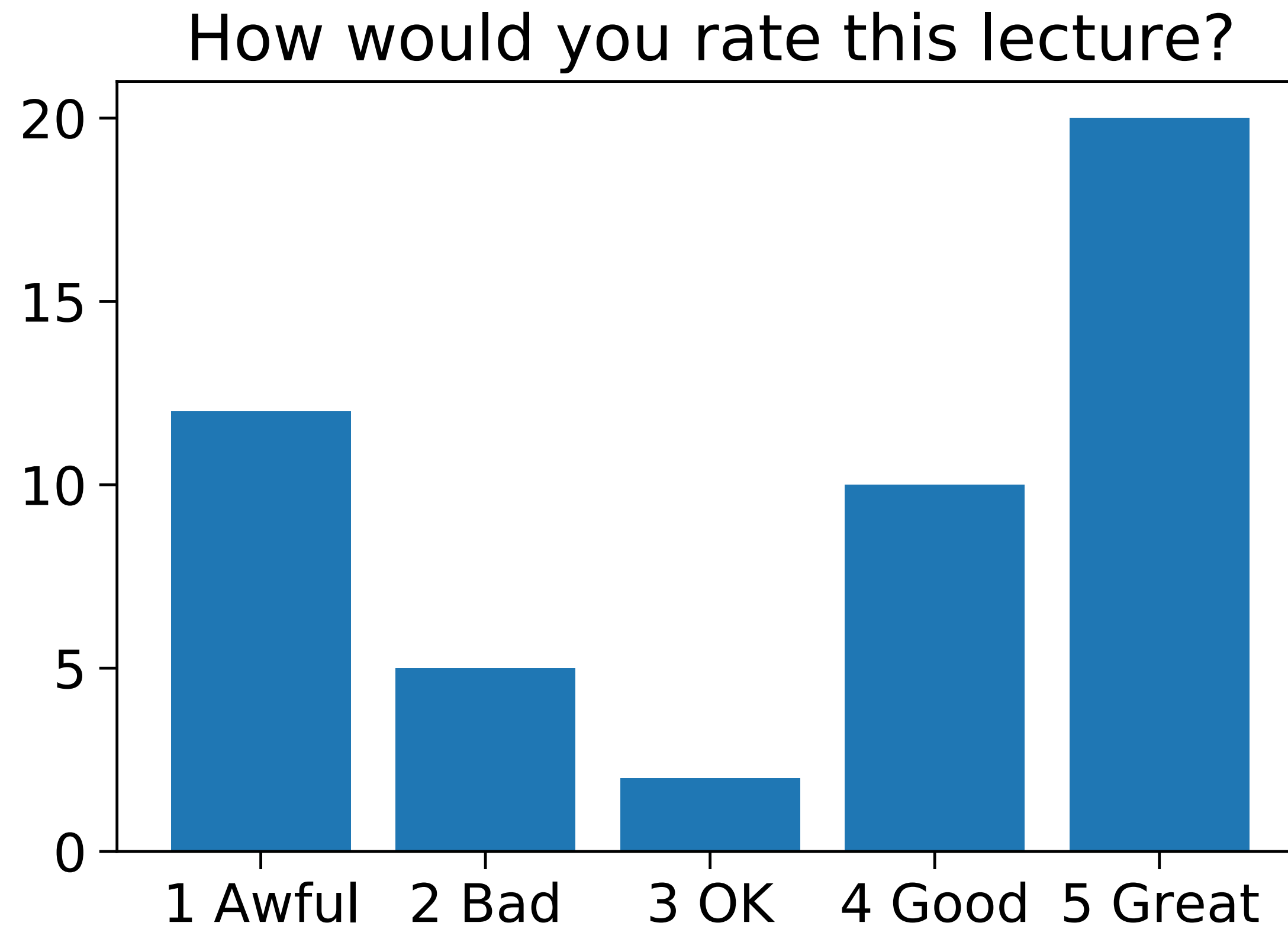
Exploratory data analysis

- This lets us find patterns, spot outliers/errors, identify important features...
- It helps us decide which machine learning methods to use (if any!)
- **We want to know if the data makes sense and if it is meaningful**



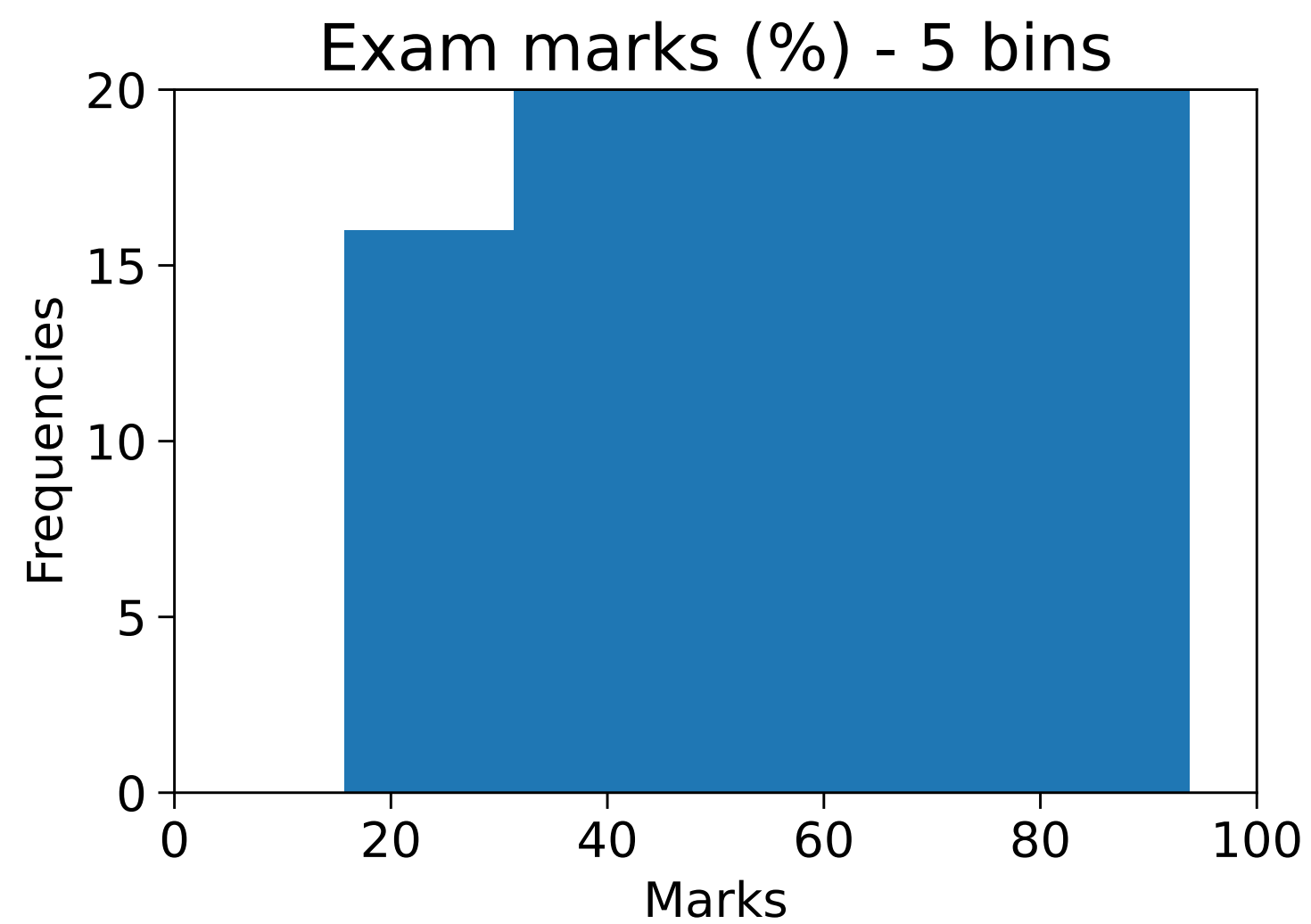
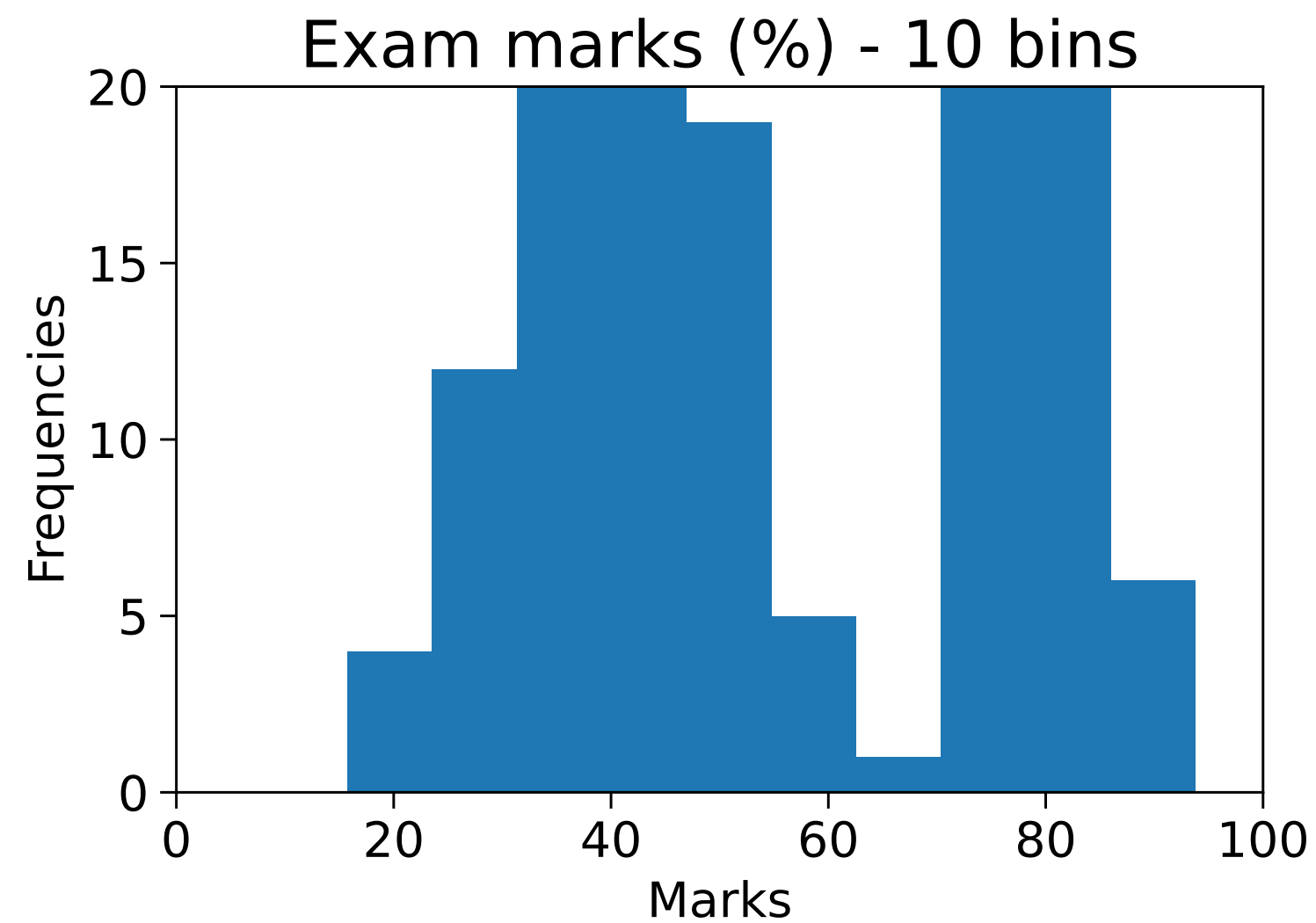
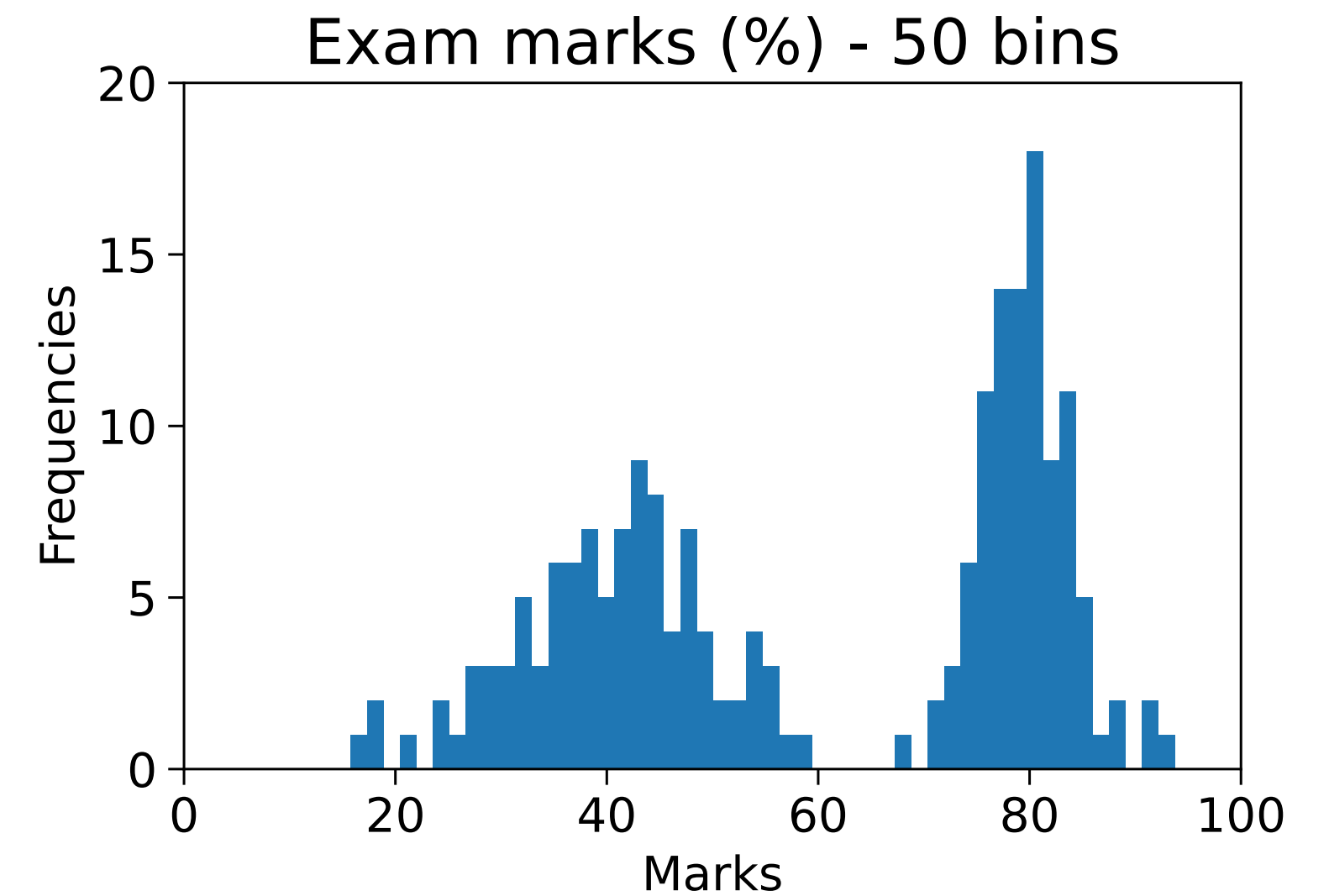
Bar plots

- Good for visualising categorical variables
- If the variable is ordinal then make sure that the columns are in order



Histograms

- Sorts measurements for numerical variables into equal sized bins
- The number of bins (and/or bin width) may need tweaking depending on use



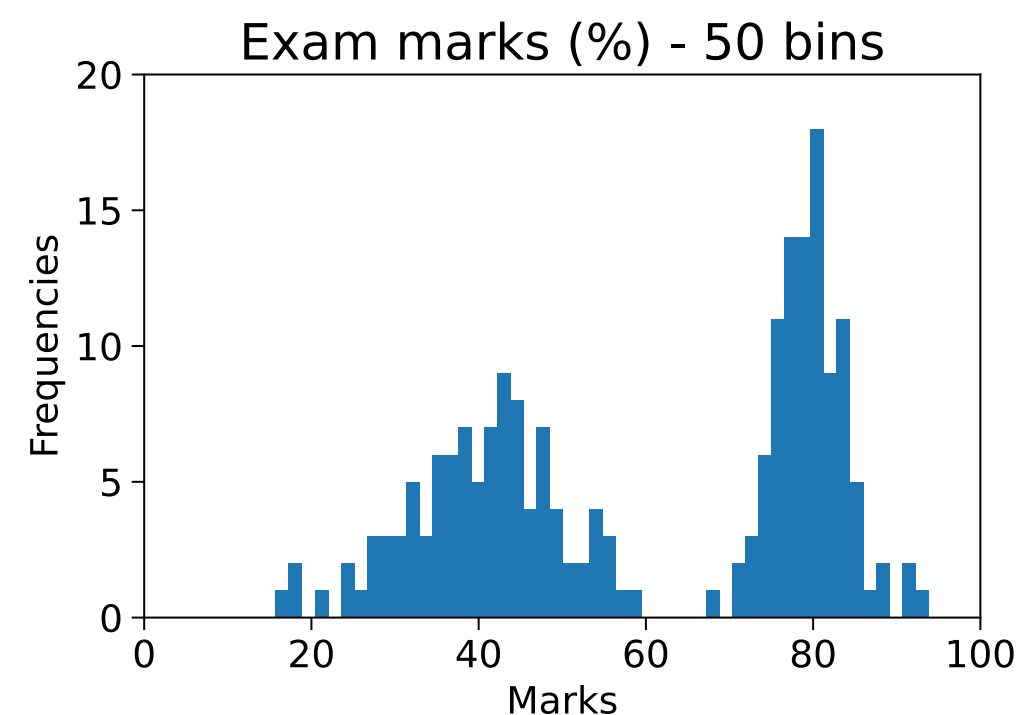
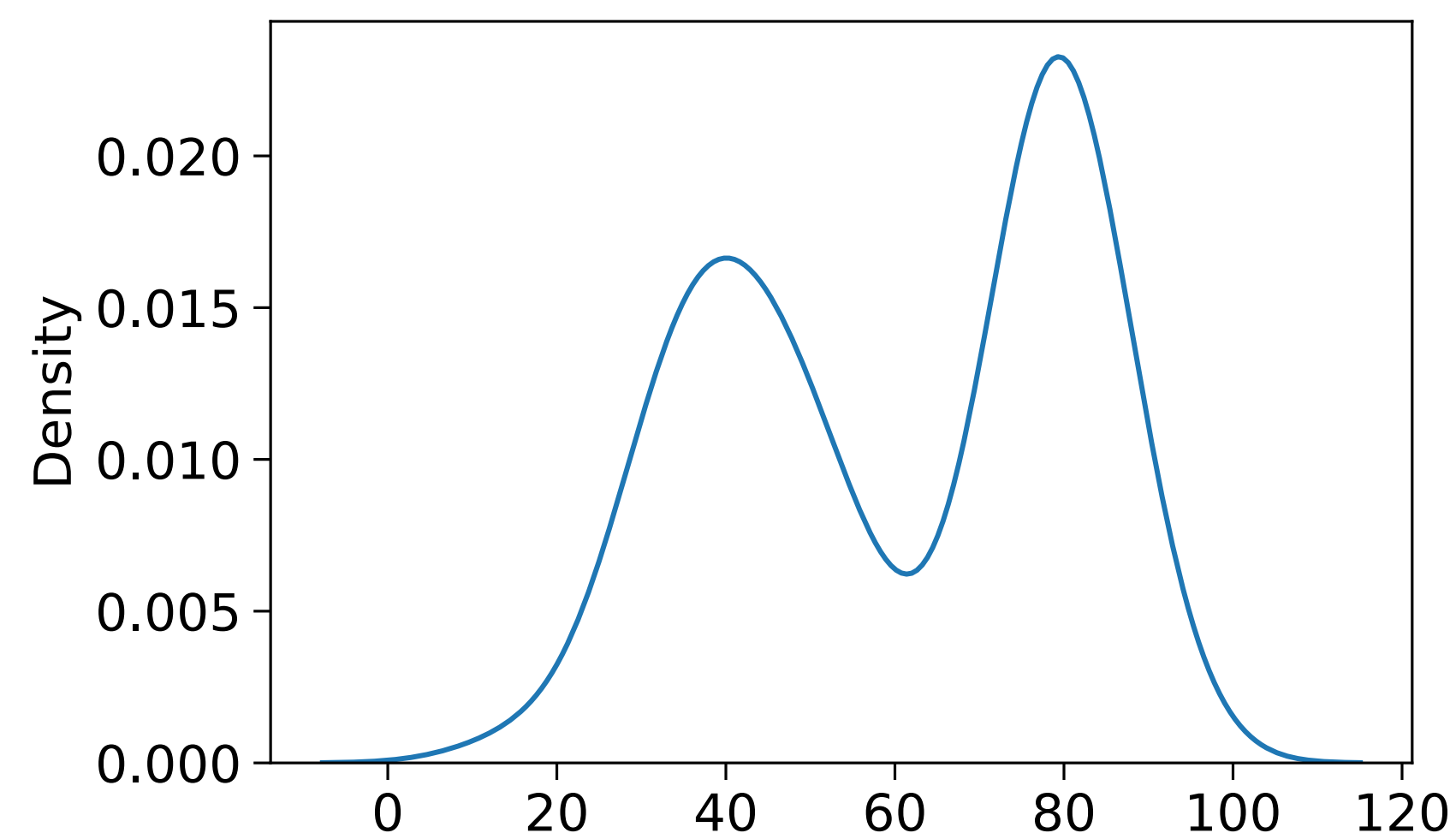
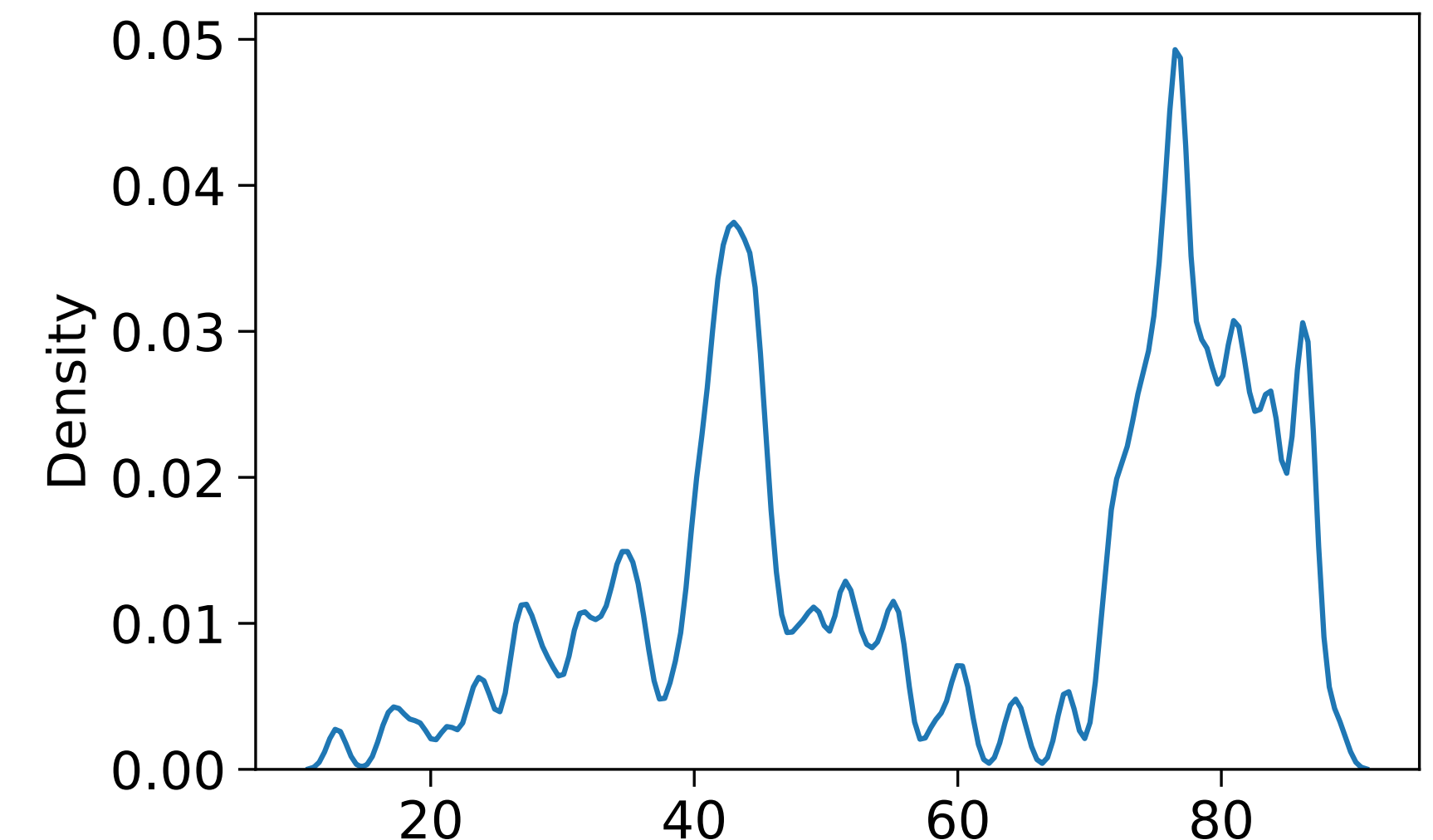
Kernel density estimate (KDE) plot

- A histogram can be seen as a discrete pdf. We can display a (plausible) continuous pdf using a KDE plot

$$p(x) = \frac{1}{N} \sum_{n=1}^N k(x, x^{(n)})$$

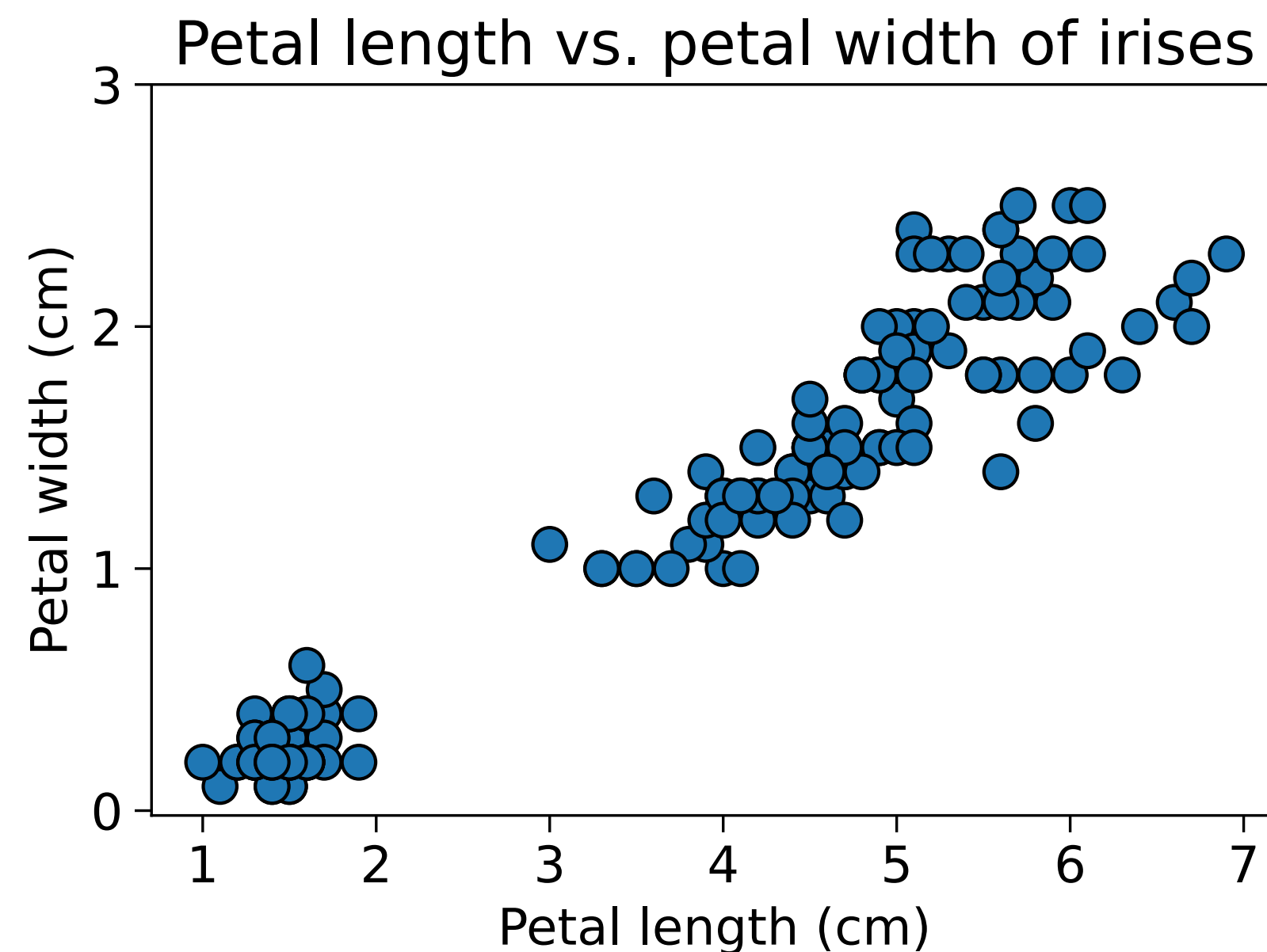
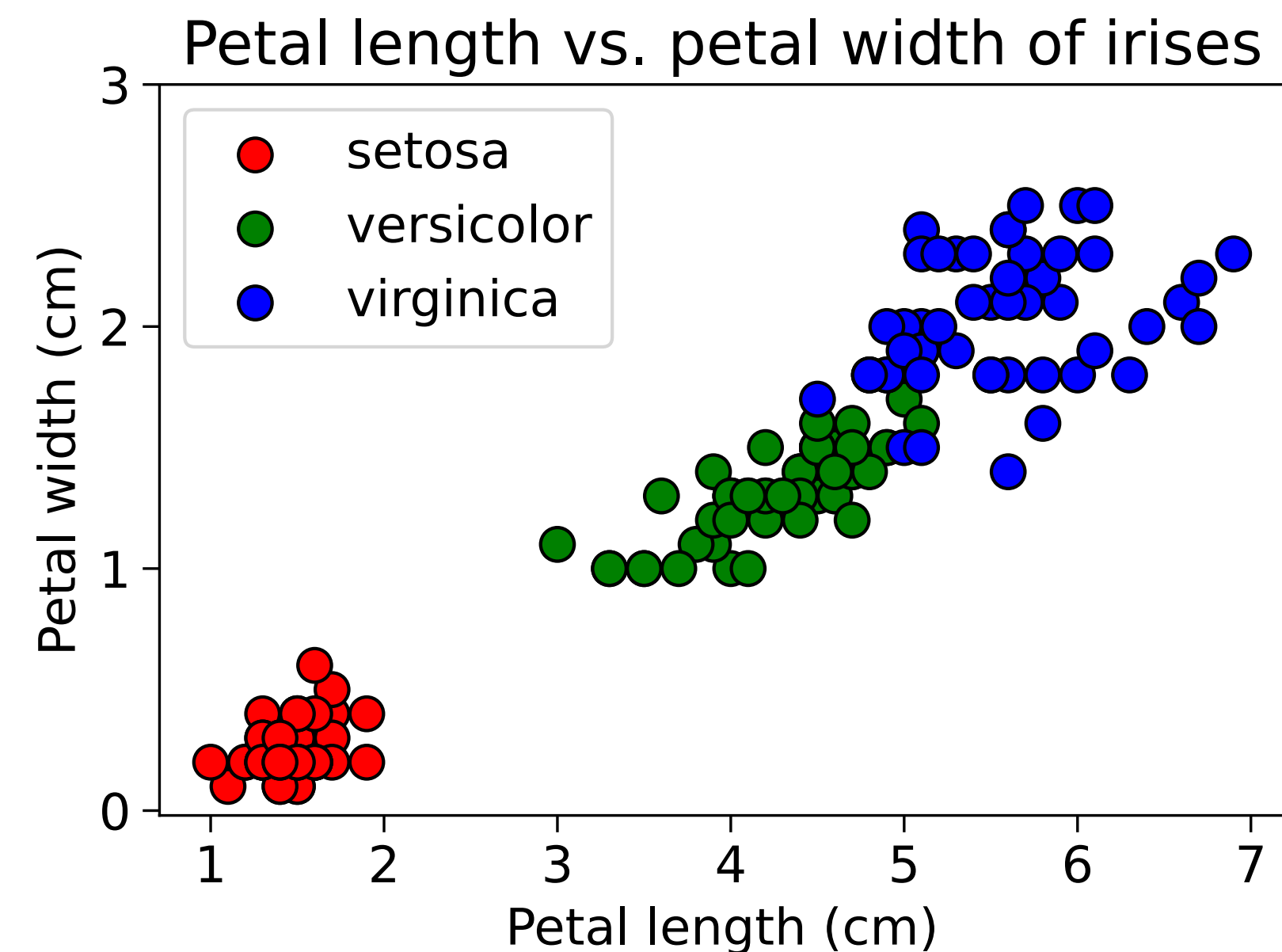
See Murphy 16.3 for more details

- k is a kernel function with a tuneable bandwidth parameter. Different bandwidths give different distributions



Scatter plots in 2D

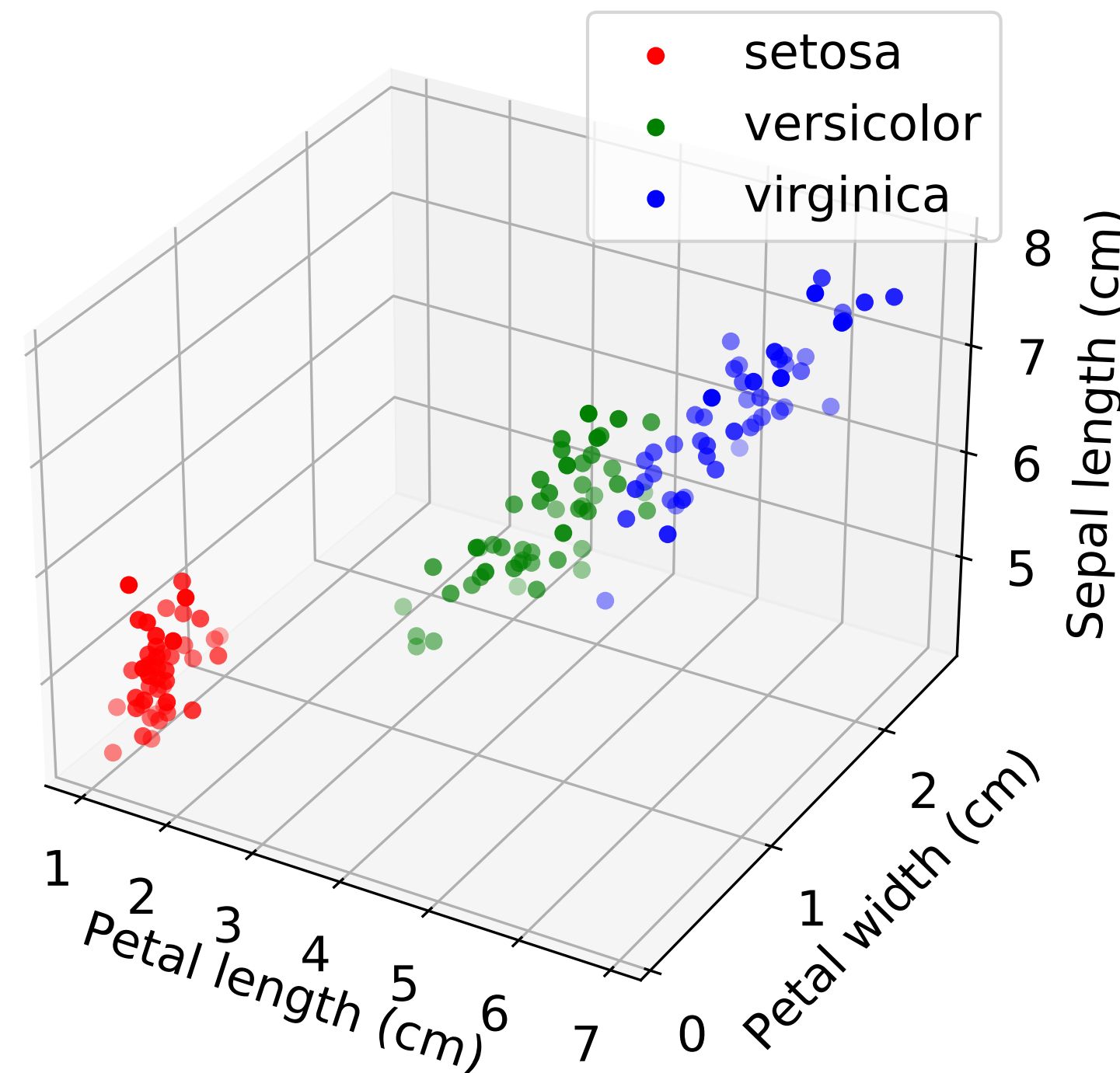
- Each point corresponds to a data point
- The x , y values for that point are measurements of two variables
- We can also distinguish points by category by using different colours/shapes



Scatter plots in 3D

- We can have x , y , z values to show three measurements per point
- But beware: we can't see the space properly as it's only a 2D projection :(

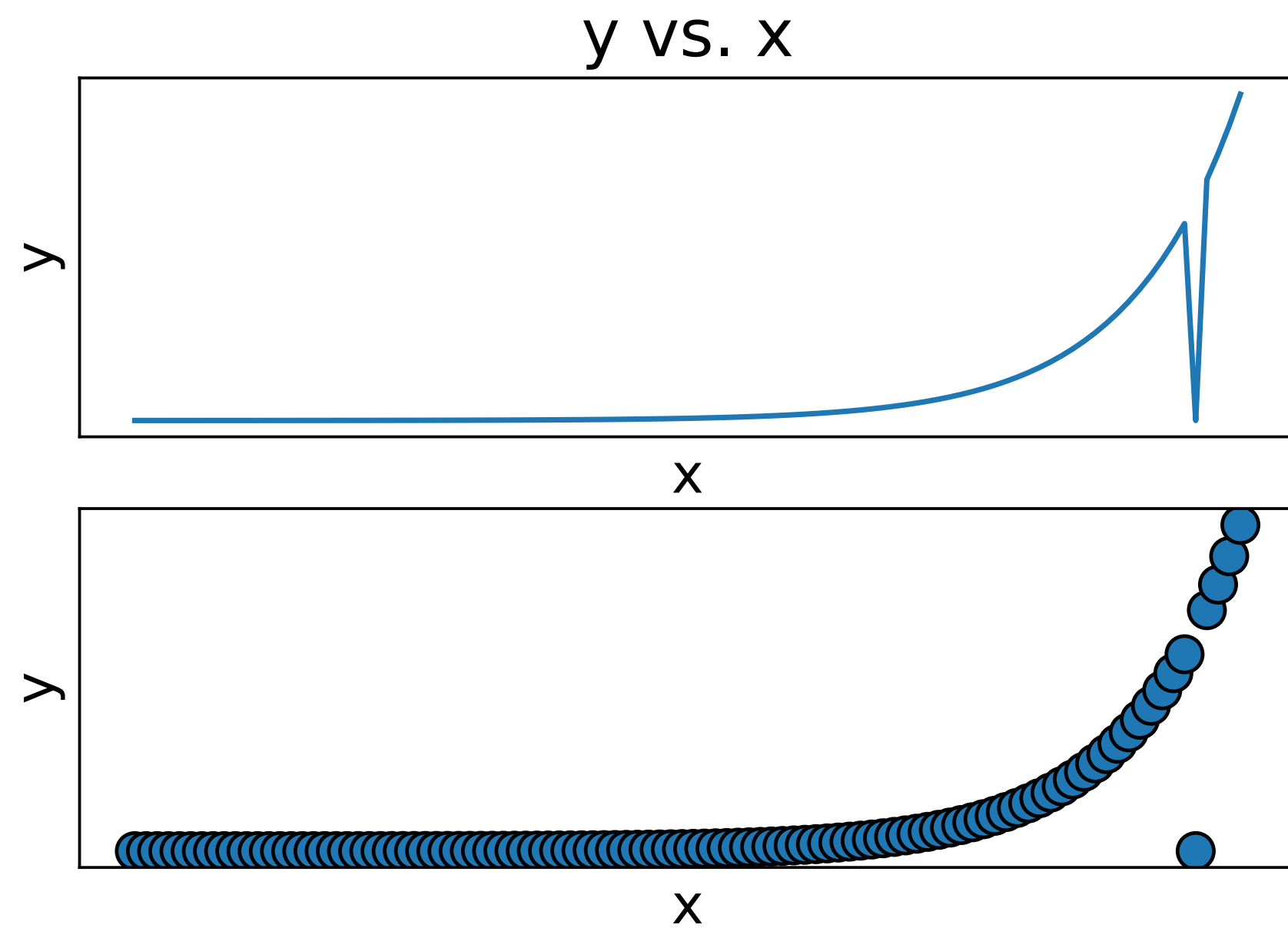
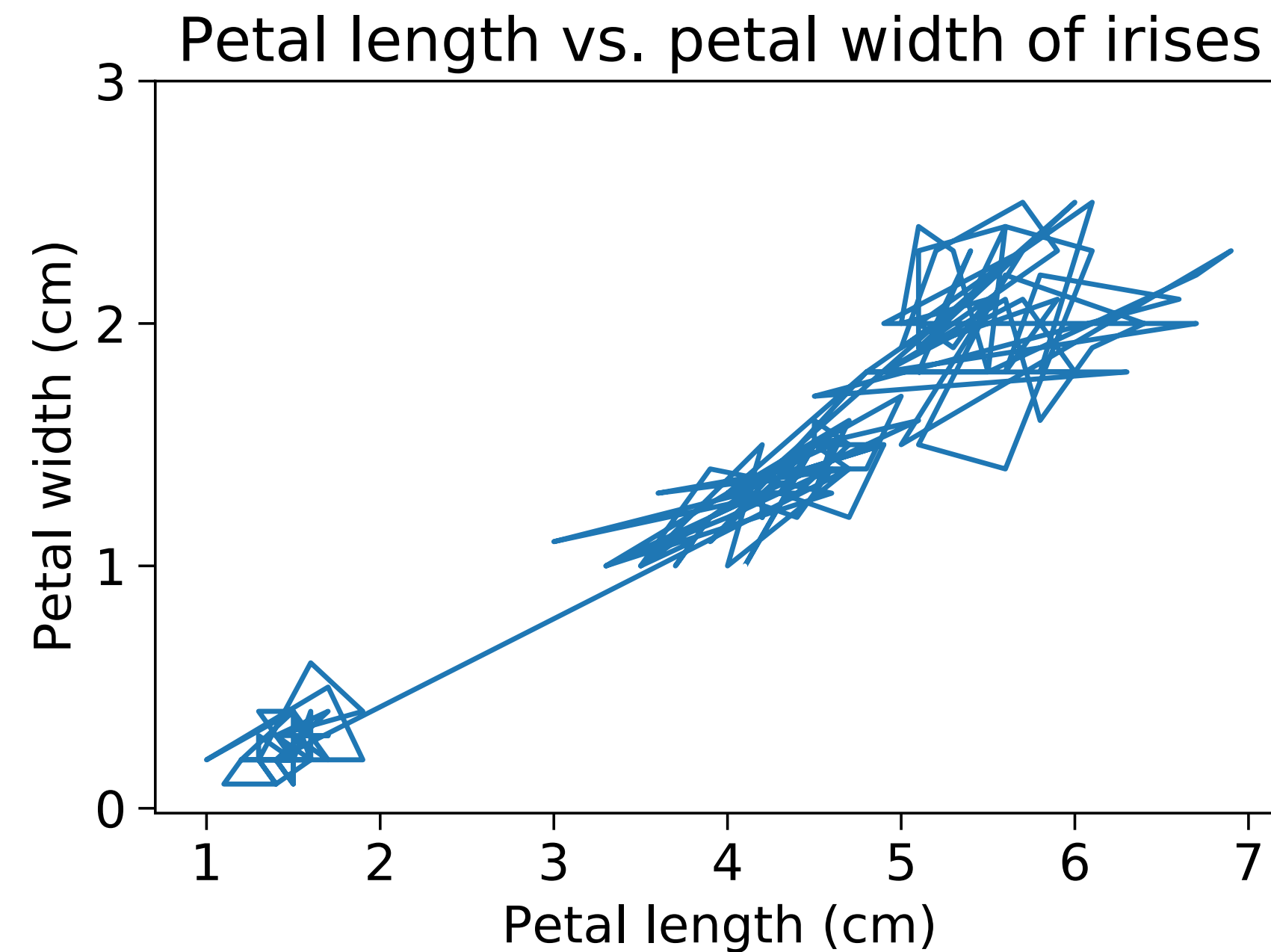
Sepal Length vs. Petal length vs. petal width of irises



I tend to avoid 3D plots where possible

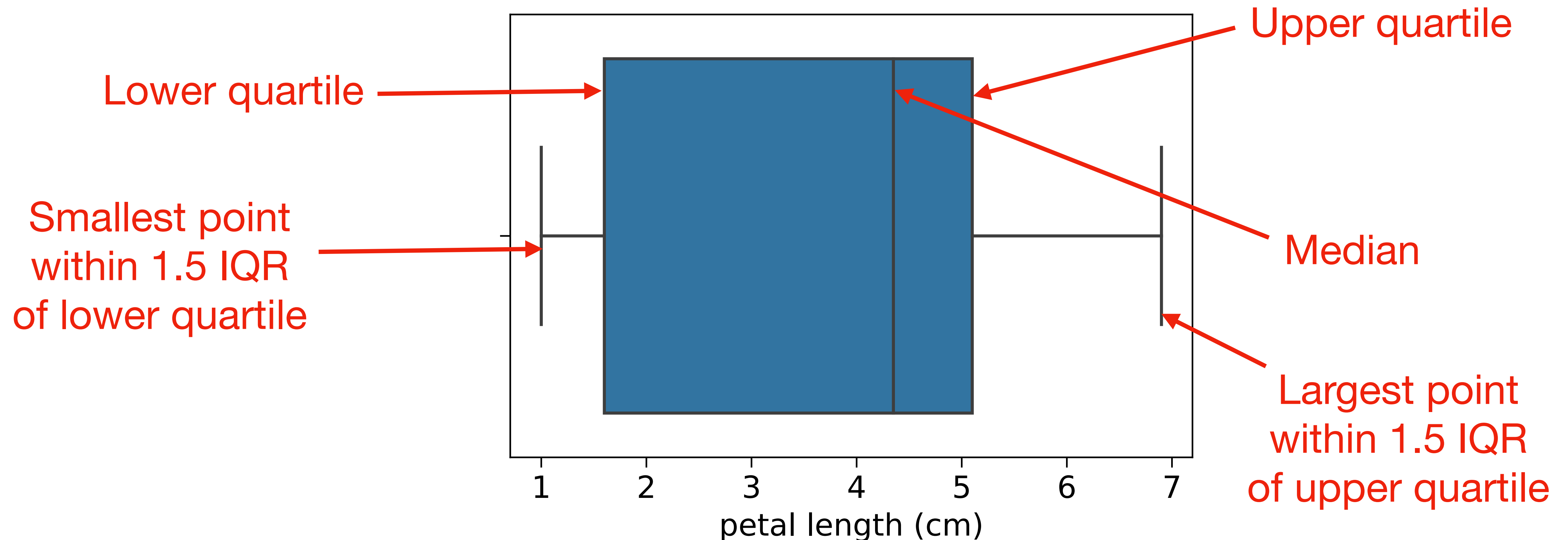
Line plots

- Can be useful for interpolation
- But can depict a functional relationship that doesn't exist if used carelessly



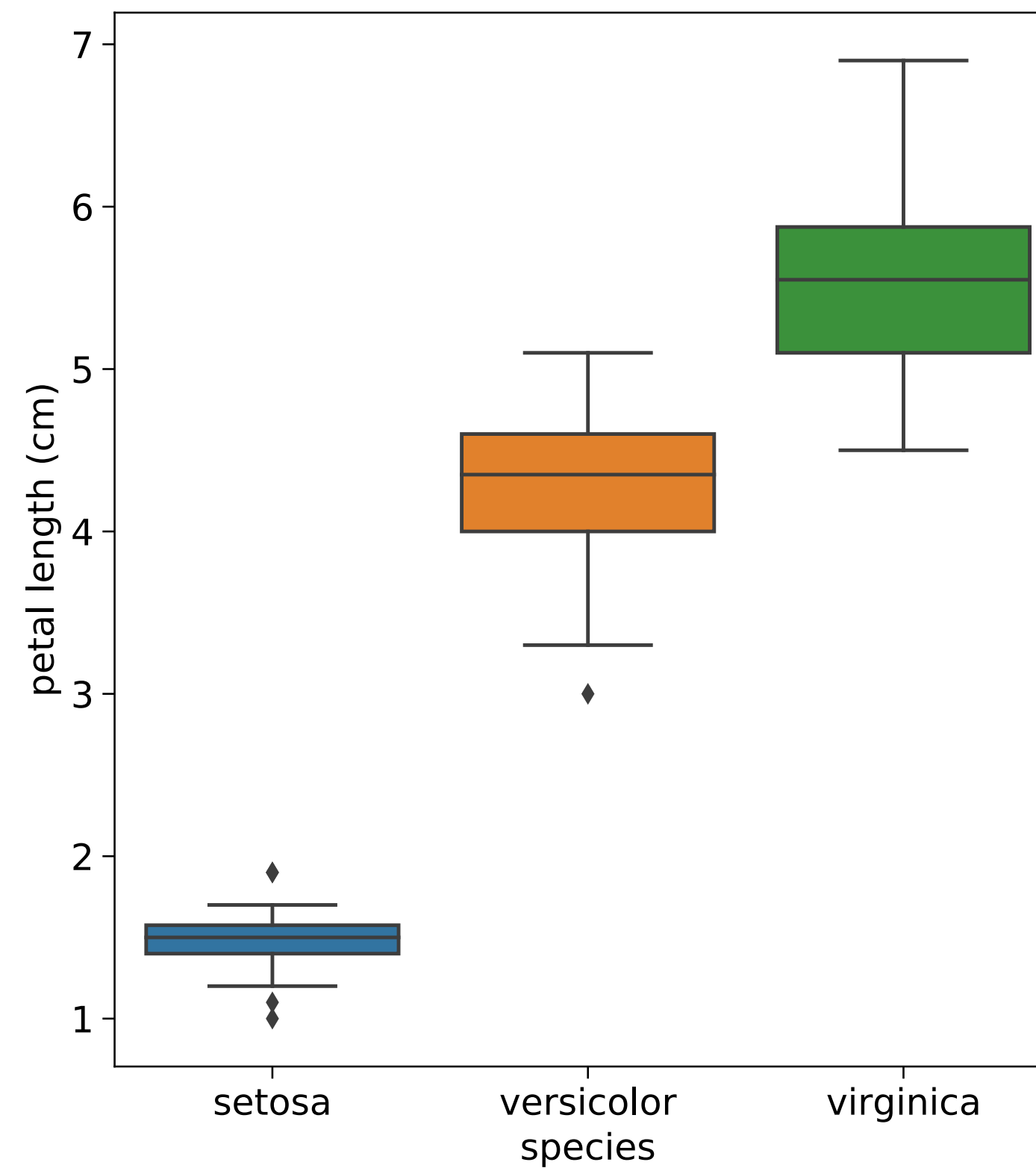
Box plots

- Shows 5 key statistics of a variable, each being an actual measurement
- Interquartile range (IQR) = upper quartile - lower quartile



Box plots

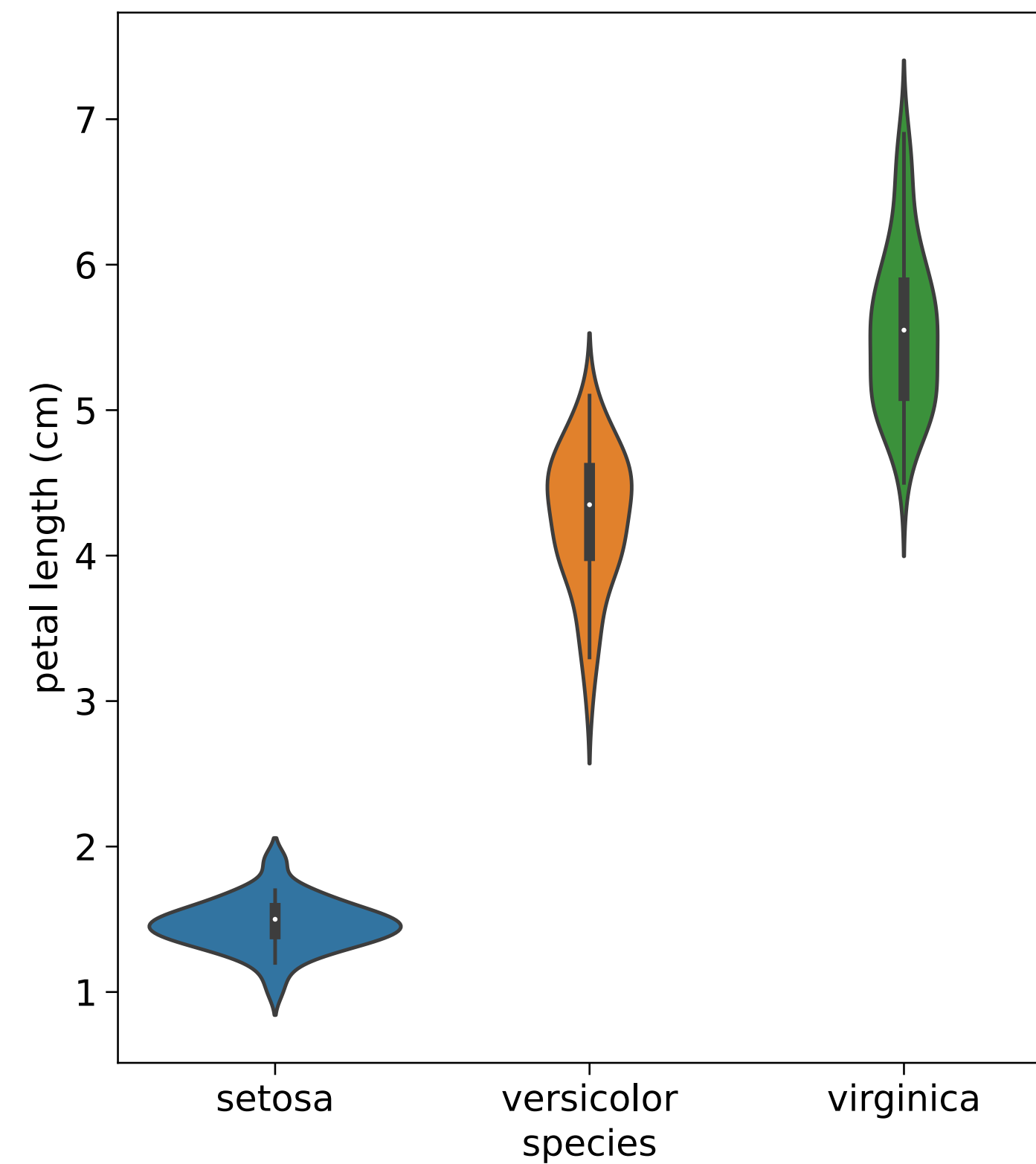
- We can view these statistics split by category
- Any points outside of the *whiskers* are plotted



Plot can be
horizontal or
vertical

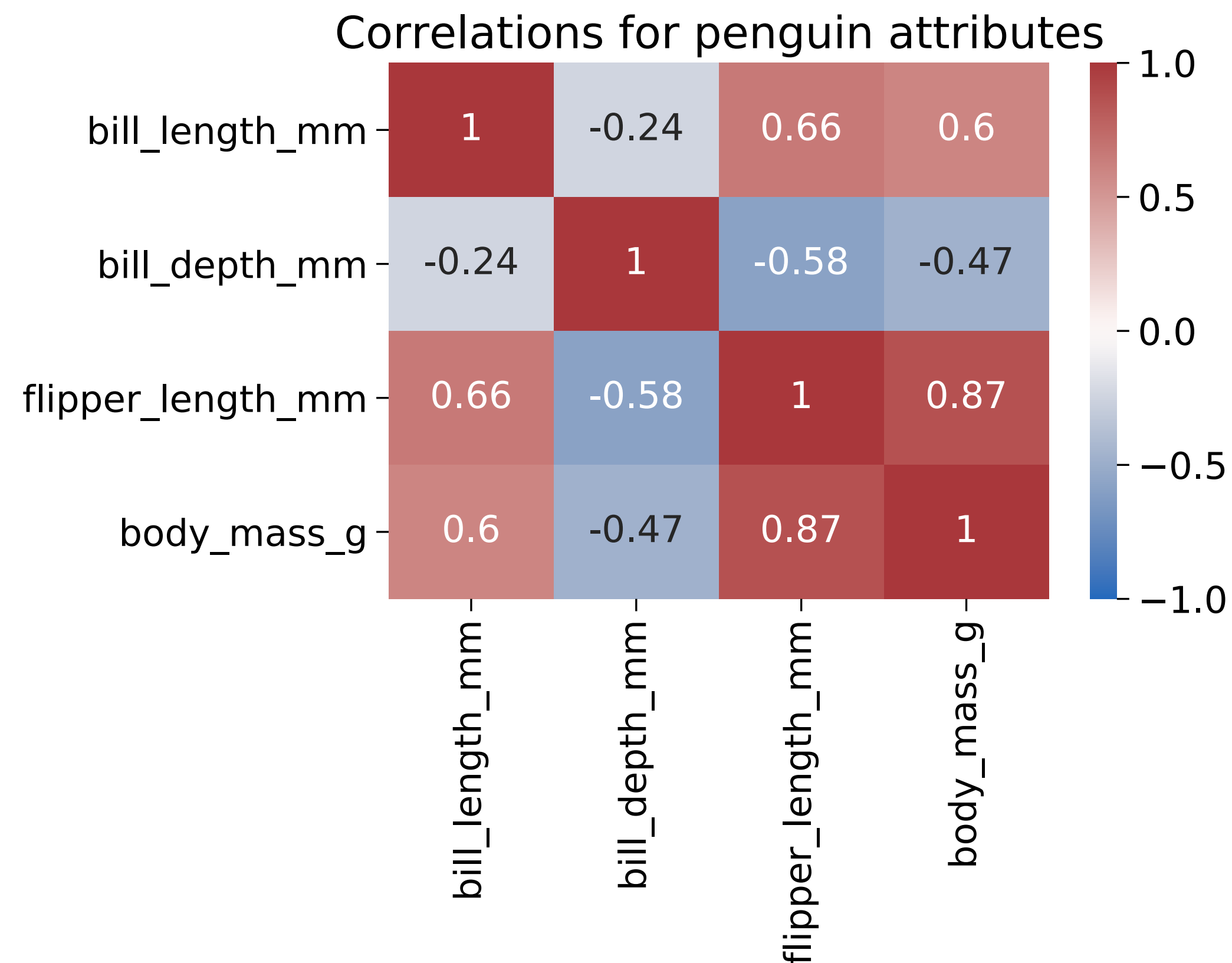
Violin plots

- These combine box plots and KDE plots



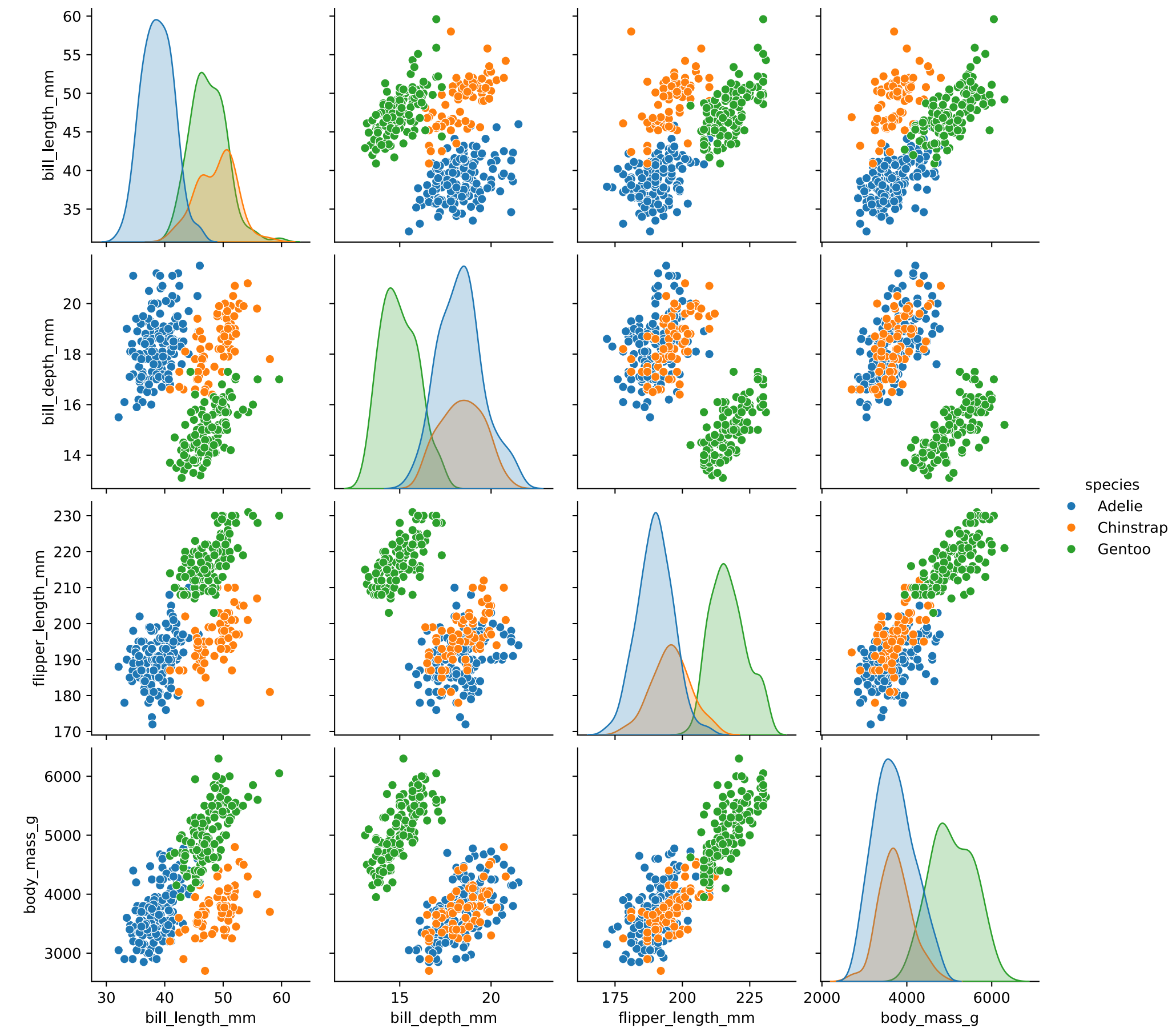
Heat maps

- A matrix of colours representing different magnitudes of some quantity
- Here we have Pearson correlation of different attributes of penguins



Pair plots

- Scatter plots for variable pairs off-diagonal
- KDE on-diagonal
- These are for penguins, coloured by species

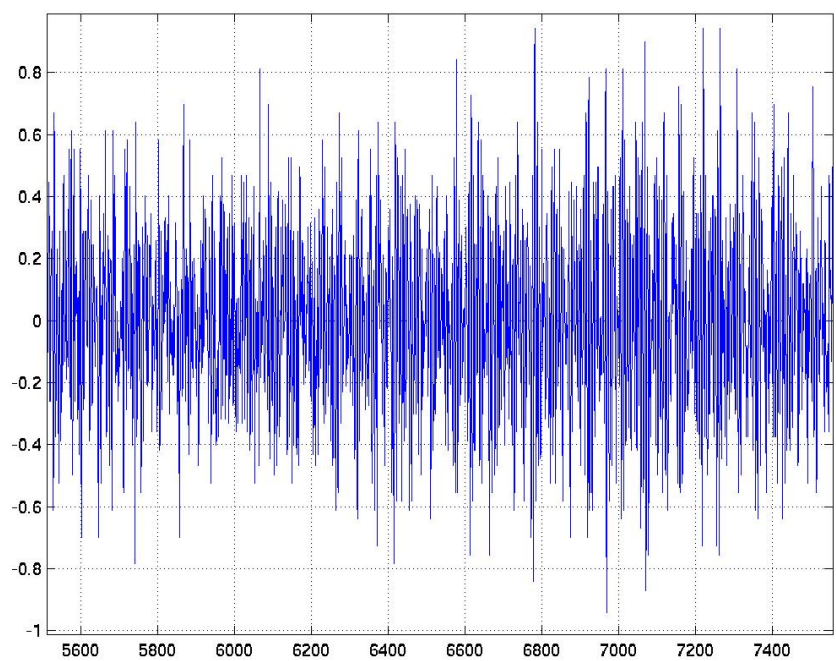


Dimensionality reduction

- Most data is high dimensional
- This makes it hard to visualise patterns across a whole dataset using standard plotting tools

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8	5
...	...	...	...	...	...	...	...	...	...	...	...	...
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	5
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0	6

Tables with lots of columns



Time series with thousands of points



Images with millions of pixels

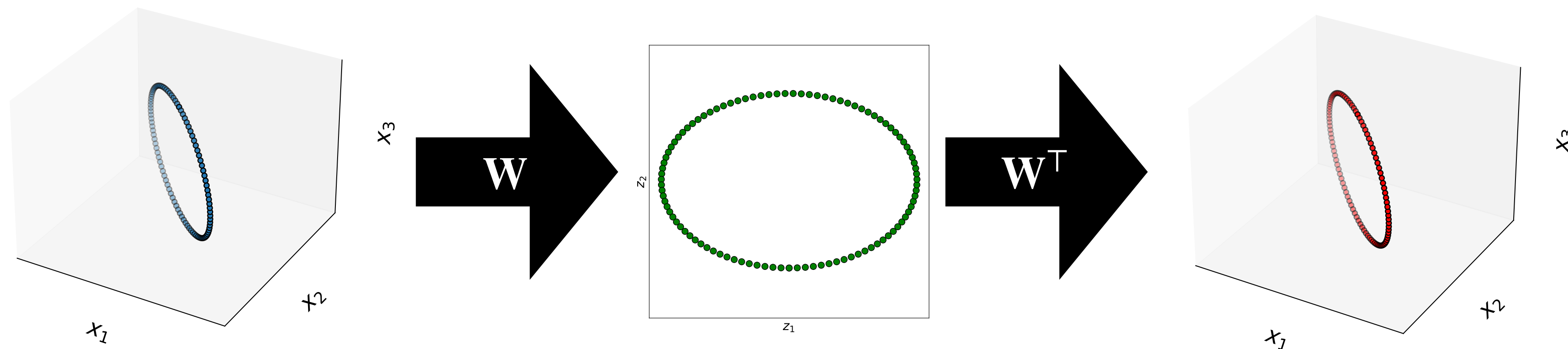
Principal Component Analysis (PCA)

Dimensionality reduction using a matrix

- We have a dataset $\{\mathbf{x}^{(n)}\}_{n=1}^N$ where $\mathbf{x} \in \mathbb{R}^D$ which can be represented using a dataset matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$
- We'll use $\mathbf{Z} = \mathbf{X}\mathbf{W}$ with $\mathbf{W} \in \mathbb{R}^{D \times d}$ to transform $\{\mathbf{x}^{(n)}\}_{n=1}^N$ (the rows of $\mathbf{X}$) into $\{\mathbf{z}^{(n)}\}_{n=1}^N$ (the rows of $\mathbf{Z}$) where $\mathbf{x} \in \mathbb{R}^D$ and $\mathbf{z} \in \mathbb{R}^d$ ($d < D$)
- We can then trivially look at a scatter plot of $\{\mathbf{z}^{(n)}\}_{n=1}^N$ (if e.g. $d = 2$) to see patterns
- But how do we know what the best transform is?

Minimising reconstruction loss

- Treat the matrix $\mathbf{W} \in \mathbb{R}^{D \times d}$ as an encoder. We apply it to get to a low dimensional space $\mathbf{Z} = \mathbf{XW}$ where $\mathbf{Z} \in \mathbb{R}^{N \times d}$
- We can then apply its transpose to *decode* $\hat{\mathbf{X}} = \mathbf{ZW}^\top$ where $\hat{\mathbf{X}} \in \mathbb{R}^{N \times D}$
- The rows of $\hat{\mathbf{X}}$: $\{\hat{\mathbf{x}}^{(n)}\}_{n=1}^N$ are reconstructions of the data points $\{\mathbf{x}^{(n)}\}_{n=1}^N$
- We should minimise the distance between points and their reconstructions so that $\mathbf{Z}$ is a faithful low dimensional representation of the dataset



Minimising reconstruction loss

- We should minimise the (average square) distance between points and their reconstructions $\frac{1}{N} \sum_n \|\mathbf{x}^{(n)} - \hat{\mathbf{x}}^{(n)}\|^2 = \frac{1}{N} \sum_n \|\mathbf{x}^{(n)} - \mathbf{W}\mathbf{W}^\top \mathbf{x}^{(n)}\|^2$
- We constrain the columns of $\mathbf{W}$ to be orthonormal: $\mathbf{W}^\top \mathbf{W} = \mathbf{I}$
- Overall we want to solve minimise $\frac{1}{N} \sum_n \|\mathbf{x}^{(n)} - \mathbf{W}\mathbf{W}^\top \mathbf{x}^{(n)}\|^2$ s.t. $\mathbf{W}^\top \mathbf{W} = \mathbf{I}$
- We use **principal components** to solve this

$\mathbf{Z}\mathbf{W}^\top = \mathbf{X}\mathbf{W}\mathbf{W}^\top$ for the dataset matrix means
 $\mathbf{W}\mathbf{z} = \mathbf{W}\mathbf{W}^\top \mathbf{x}$ for each column vector data point

Computing principal components

To compute principal components for a standardised dataset $\mathbf{X} \in \mathbb{R}^{N \times D}$:

1. Construct the covariance matrix $\mathbf{\Sigma} = \frac{1}{N} \mathbf{X}^\top \mathbf{X}$
2. Eigendecompose $\mathbf{\Sigma}$ to get eigenvalue, eigenvector pairs
3. Sort pairs by decreasing eigenvalue and denote as $\{\lambda_d\}_{d=1}^D, \{\mathbf{w}_d\}_{d=1}^D$

These vectors are
the principal components

Principal Component Analysis (PCA)

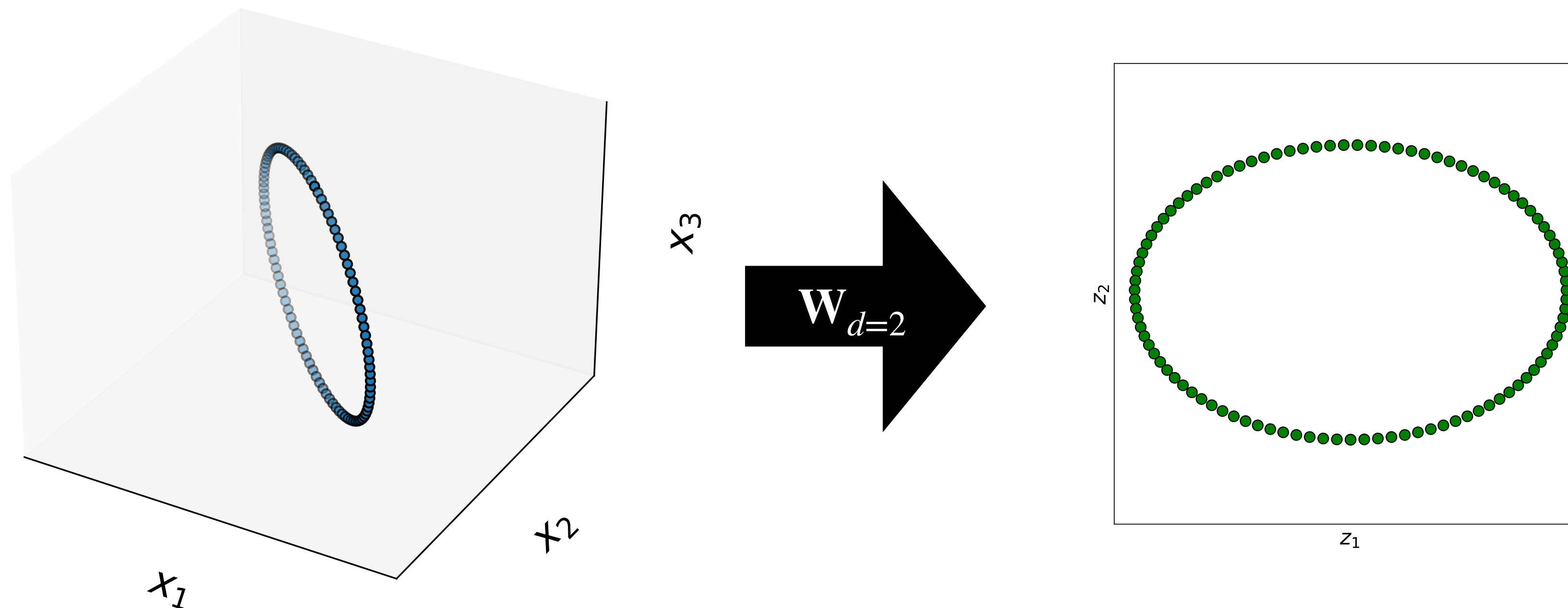
- For a **standardised** dataset $\mathbf{X} \in \mathbb{R}^{N \times D}$, PCA returns a matrix $\mathbf{W}_{PC} \in \mathbb{R}^{D \times D}$
- The columns of $\mathbf{W}_{PC}$: $\{\mathbf{w}_d\}_{d=1}^D$ are the **principal components** of the data
- The matrix that solves minimise $\frac{1}{N} \sum_n \|\mathbf{x}^{(n)} - \mathbf{W}\mathbf{W}^\top \mathbf{x}^{(n)}\|^2$ s.t. $\mathbf{W}^\top \mathbf{W} = \mathbf{I}$
for $\mathbf{W} \in \mathbb{R}^{D \times d}$ is $\mathbf{W} = [\mathbf{w}_1 \quad \mathbf{w}_2 \quad \dots \quad \mathbf{w}_d]$
- i.e. it's the matrix whose columns are the first d principal components

See Murphy 20.1.2 for the proof

$\mathbf{X}\mathbf{W}\mathbf{W}^\top$ for the dataset matrix translates to
 $\mathbf{W}\mathbf{W}^\top \mathbf{x}$ for each column vector data point

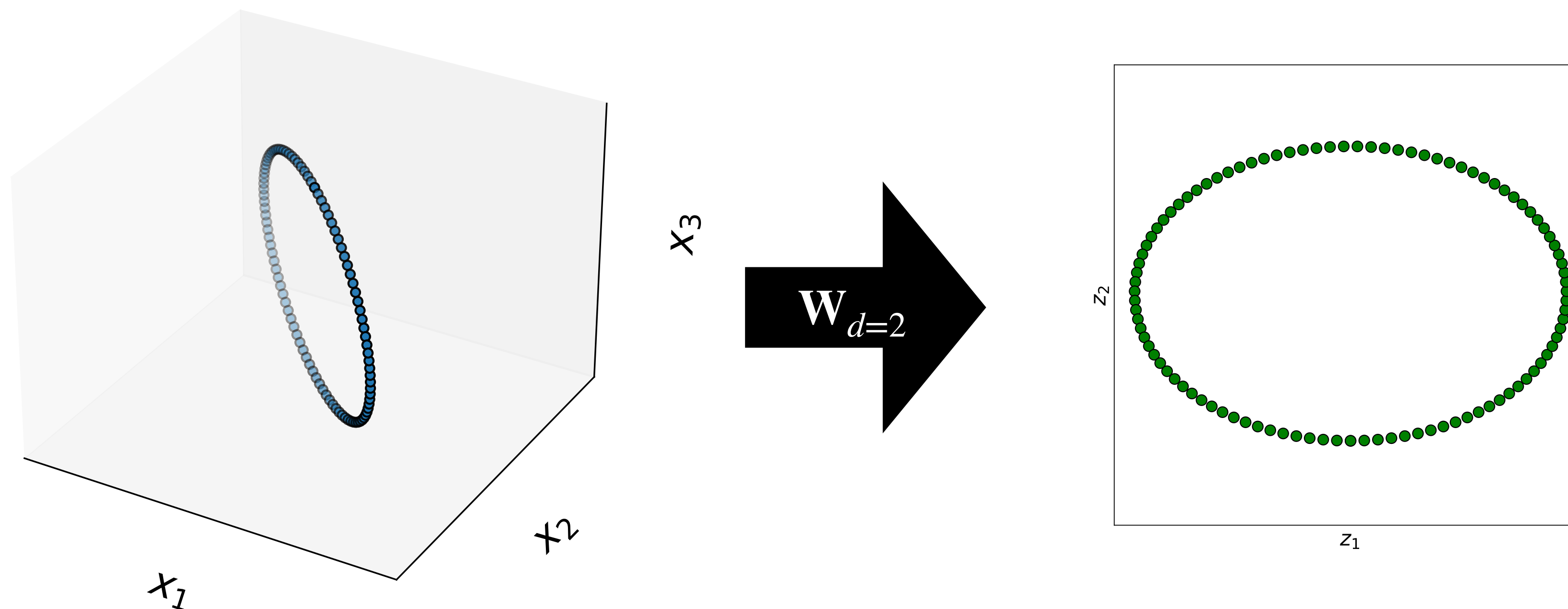
PCA for dimensionality reduction

- PCA gives us $\mathbf{W} \in \mathbb{R}^{D \times D}$ where $\mathbf{W} = [\mathbf{w}_1 \quad \mathbf{w}_2 \quad \dots \quad \mathbf{w}_D]$
- To reduce to $d < D$ dimensions we can just keep the first d columns
- e.g. $\mathbf{W}_{d=2} = [\mathbf{w}_1 \quad \mathbf{w}_2]$ would take our data to 2D using $\mathbf{Z} = \mathbf{X}\mathbf{W}_{d=2}$



Minimising reconstruction error maximises variance

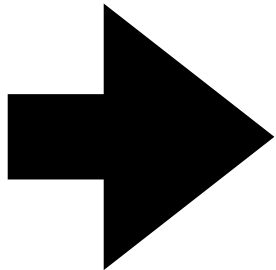
- PCA gives us the (linear) **direction of maximum variance** in z_1
- It gives us the (orthogonal) next largest direction of maximum variance in z_2
- And so on. This is neat, but to me, less intuitive than reconstruction error



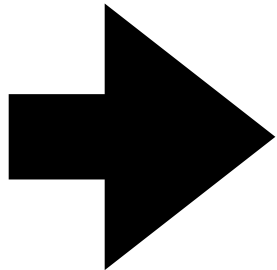
PCA for dimensionality reduction on irises

- The iris dataset contains 150 data points
- Let's take the numeric columns to form a dataset matrix $\mathbf{X} \in \mathbb{R}^{150 \times 4}$
- Make sure that $\mathbf{X}$ is standardised

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica



$$\begin{bmatrix} 5.1 & 3.5 & 1.4 & 0.2 \\ 4.9 & 3.0 & 1.4 & 0.2 \\ 4.7 & 3.2 & 1.3 & 0.2 \\ 4.6 & 3.1 & 1.5 & 0.2 \\ 5.0 & 3.6 & 1.4 & 0.2 \\ \dots & \dots & \dots & \dots \\ 0.7 & 3.0 & 5.2 & 2.3 \\ 6.3 & 2.5 & 5.0 & 1.9 \\ 6.5 & 3.0 & 5.2 & 2.0 \\ 6.2 & 3.4 & 5.4 & 2.3 \\ 5.9 & 3.0 & 5.1 & 1.8 \end{bmatrix}$$



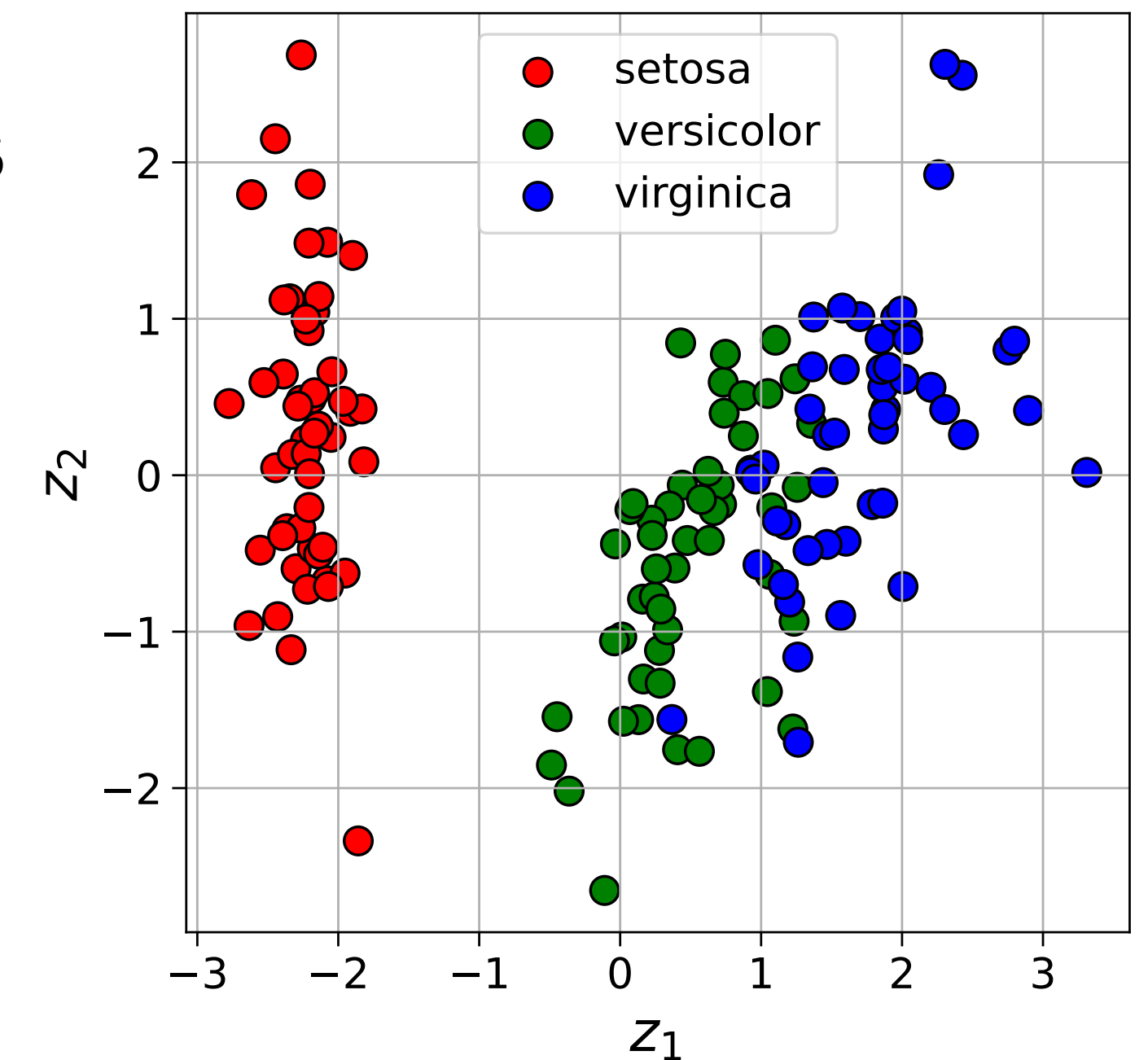
$$\begin{bmatrix} 5.1 & 3.5 & 1.4 & 0.2 \\ 0.9 & 1.0 & -1.3 & -1.3 \\ -1.1 & -0.1 & -1.3 & -1.3 \\ -1.4 & 0.3 & -1.4 & -1.3 \\ -1.5 & 0.1 & -1.3 & -1.3 \\ -1.0 & 1.2 & -1.3 & -1.3 \\ \dots & \dots & \dots & \dots \\ .0 & -0.1 & 0.8 & 1.4 \\ 0.6 & -1.3 & 0.7 & 0.9 \\ 0.8 & -0.1 & 0.8 & 1.1 \\ 0.4 & 0.8 & 0.9 & 1.4 \\ 0.1 & -0.1 & 0.8 & 0.8 \end{bmatrix}$$

PCA for dimensionality reduction on irises

- Use PCA to form $\mathbf{W}_{PC} \in \mathbb{R}^{4 \times 4}$
- Now use $\mathbf{Z} = \mathbf{X} [\mathbf{w}_1 \quad \mathbf{w}_2]$ to project down to 2D
- Different species are distinguishable just by looking at z_1
- These new features (z_1, z_2) are just linear combinations of the original features (x_1, x_2, x_3, x_4)

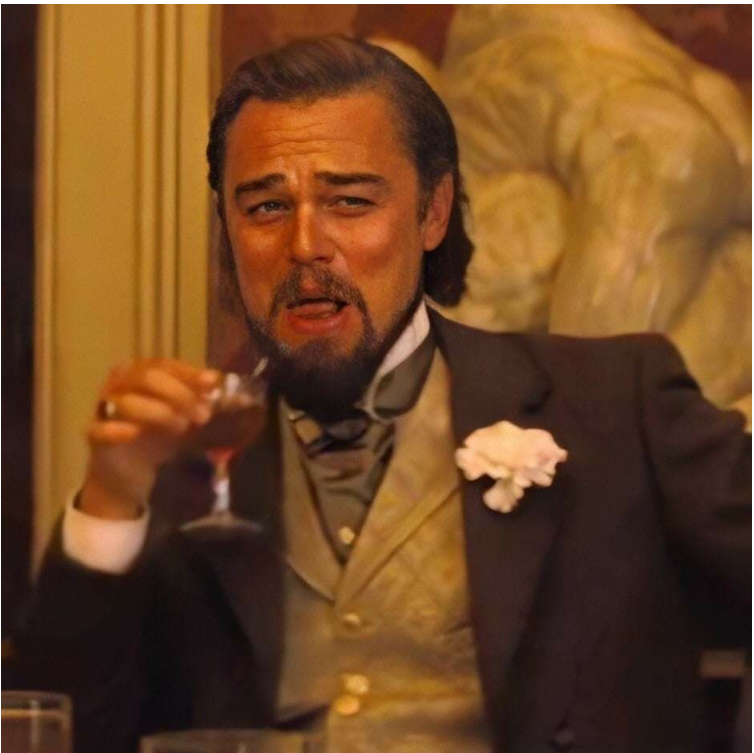
$$z_1 = -0.52x_1 - 0.27x_2 - 0.58x_3 + 0.56x_4$$

$$z_2 = -0.38x_1 + 0.92x_2 + 0.02x_3 + 0.07x_4$$



PCA for dimensionality reduction on wine

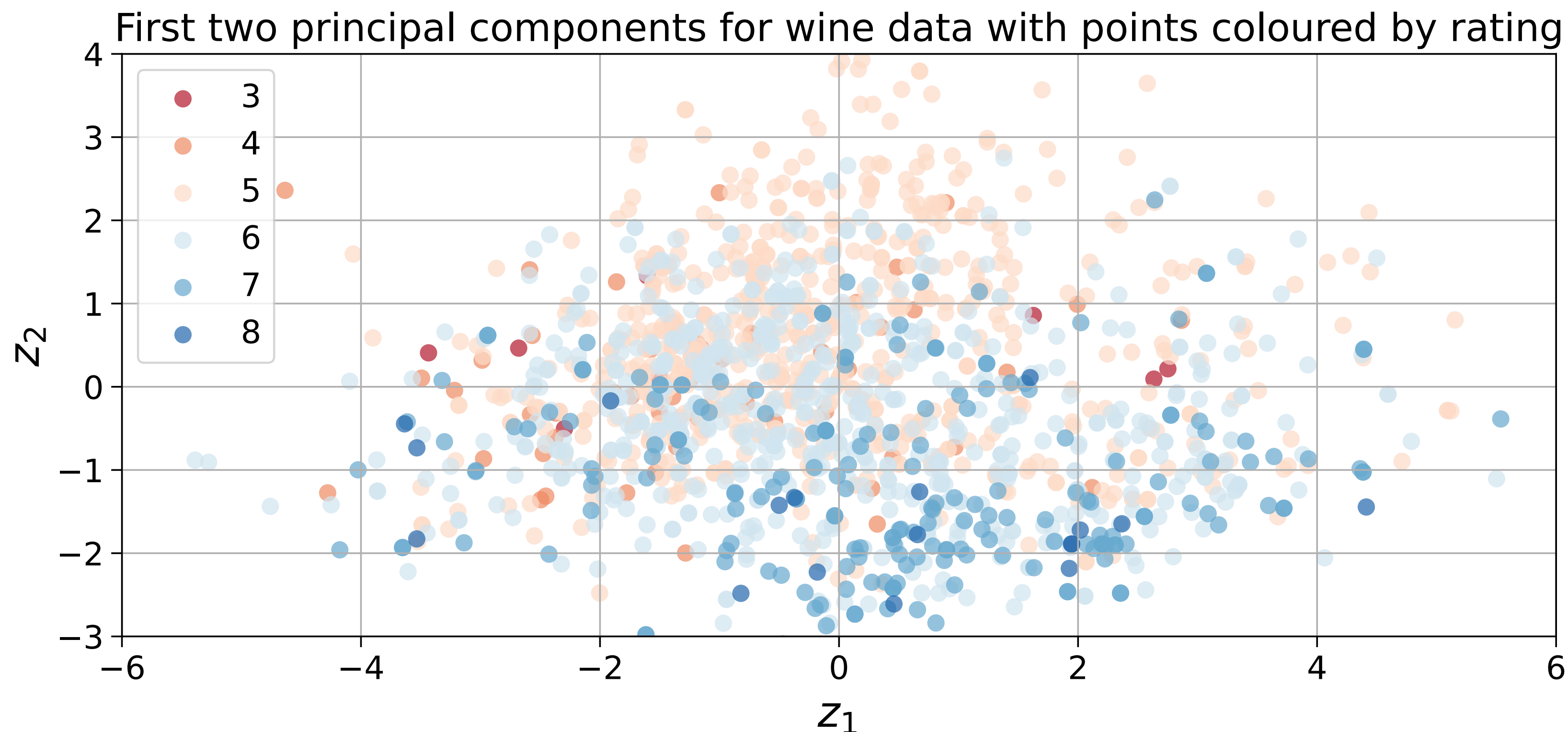
- We have a red wine dataset $\mathbf{X} \in \mathbb{R}^{1599 \times 11}$
- Each wine has also been scored by an expert between 0 and 10
- We can look at a few examples but it's hard to get the full picture



	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	3.51	0.56	9.4	5
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	3.20	0.68	9.8	5
...	...	...	...	...	...	...	...	...	...	...	...	...
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	3.57	0.71	10.2	5
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	3.39	0.66	11.0	6

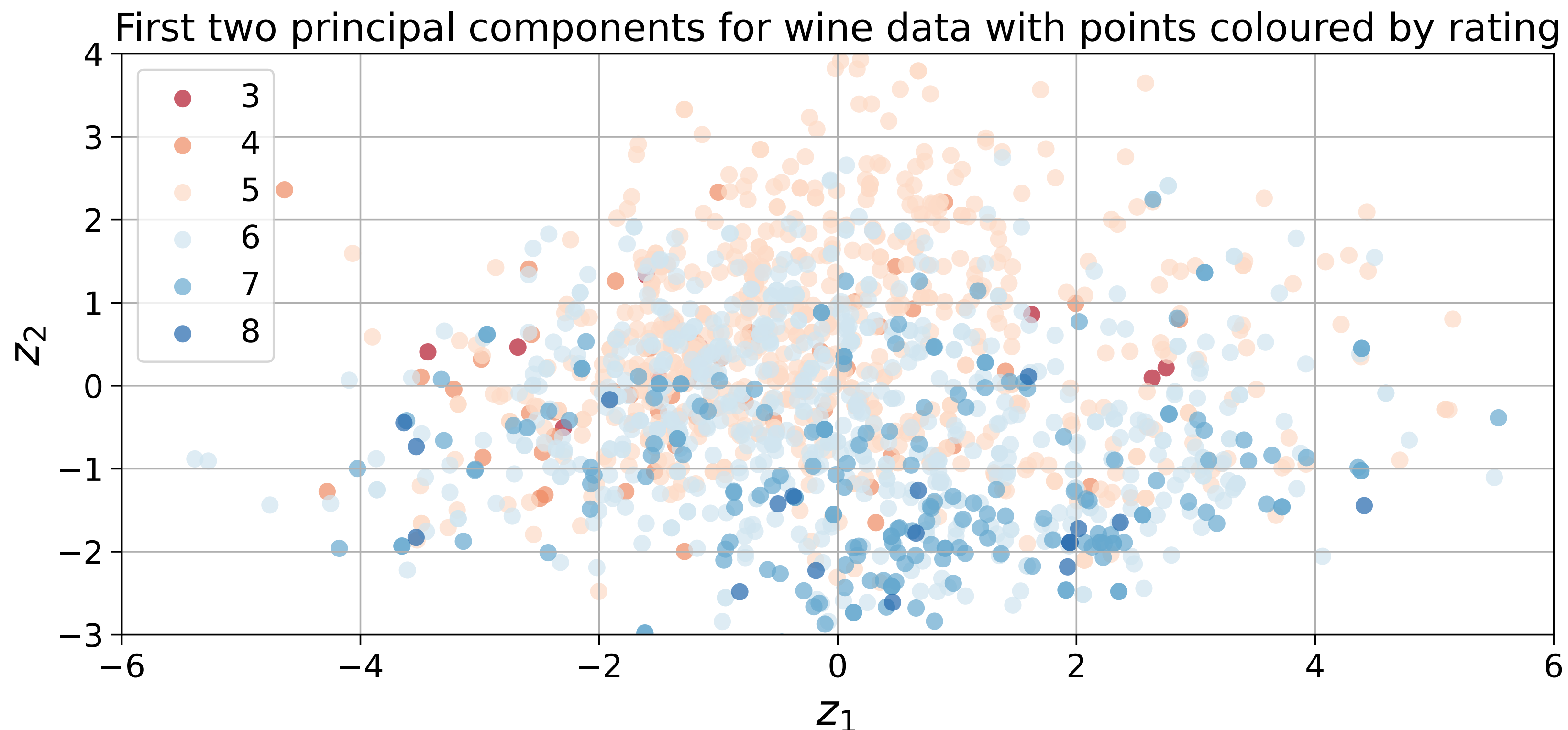
PCA for dimensionality reduction on wine

- Let's standardise our data, and then use PCA to form $\mathbf{W}_{PC} \in \mathbb{R}^{11 \times 11}$
- Now use $\mathbf{Z} = \mathbf{X} [\mathbf{w}_1 \quad \mathbf{w}_2]$ to project down to 2D



PCA for dimensionality reduction on wine

- We can see in this space that good wines tend to be near the bottom
- What makes a good wine? A negative z_2 of course!



Good wine recipe: make z_2 negative

- The new features are just linear combinations of the original features

$$z_2 = -0.11x_1 + 0.27x_2 - 0.15x_3 + 0.27x_4 + 0.15x_5 + 0.51x_6 + 0.57x_7 + 0.23x_8 + 0.01x_9 - 0.04x_{10} - 0.39x_{11}$$

- In a lot of cases the new features aren't very intuitive
- PCA is best used for exploratory data analysis

Importance of components

- Performing PCA gives us eigenvalue, eigenvector pairs $\{\lambda_d\}_{d=1}^D, \{\mathbf{w}_d\}_{d=1}^D$
- The eigenvectors are our principal components
- The eigenvalues are an importance weighting for each component

The first principal component explains $\frac{\lambda_1}{\sum_{d=1}^D \lambda_d}$ of the variance of the data

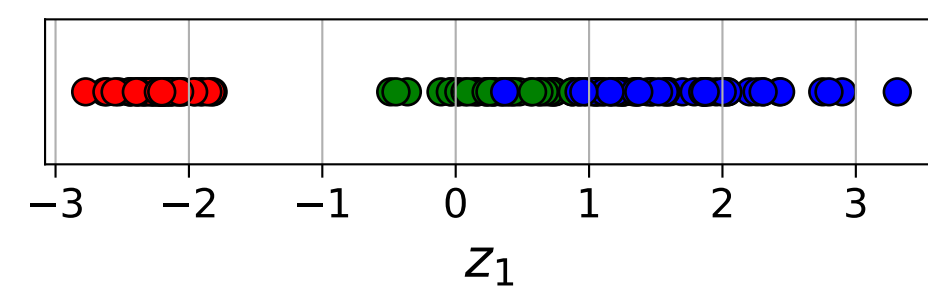
Importance of components

The first principal component explains $\frac{\lambda_1}{\sum_{d=1}^D \lambda_d}$ of the variance

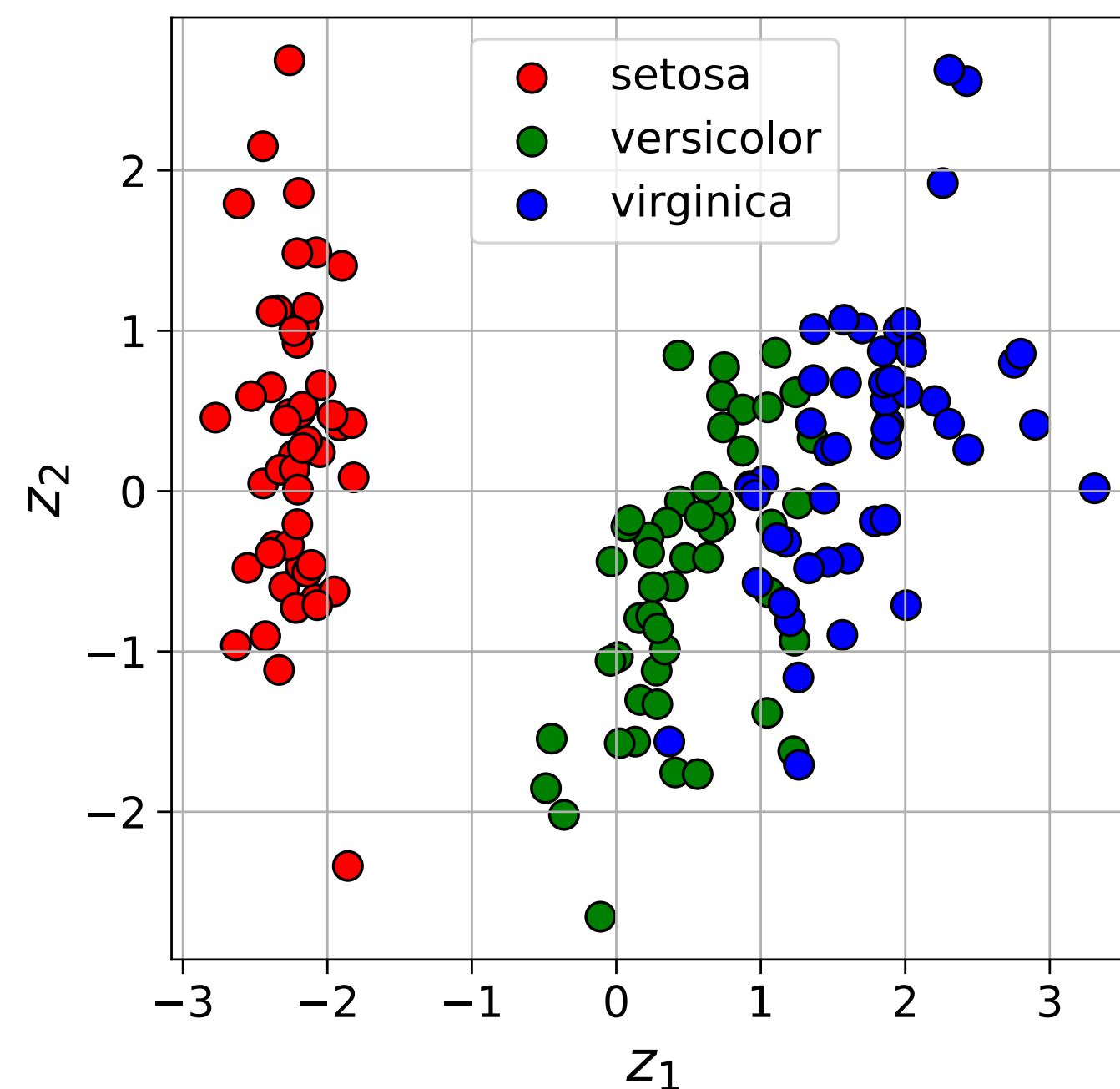
It follows that the first M principal components account for $\frac{\sum_{m=1}^M \lambda_m}{\sum_{d=1}^D \lambda_d}$

Be careful throwing away dimensions if not enough variance is explained

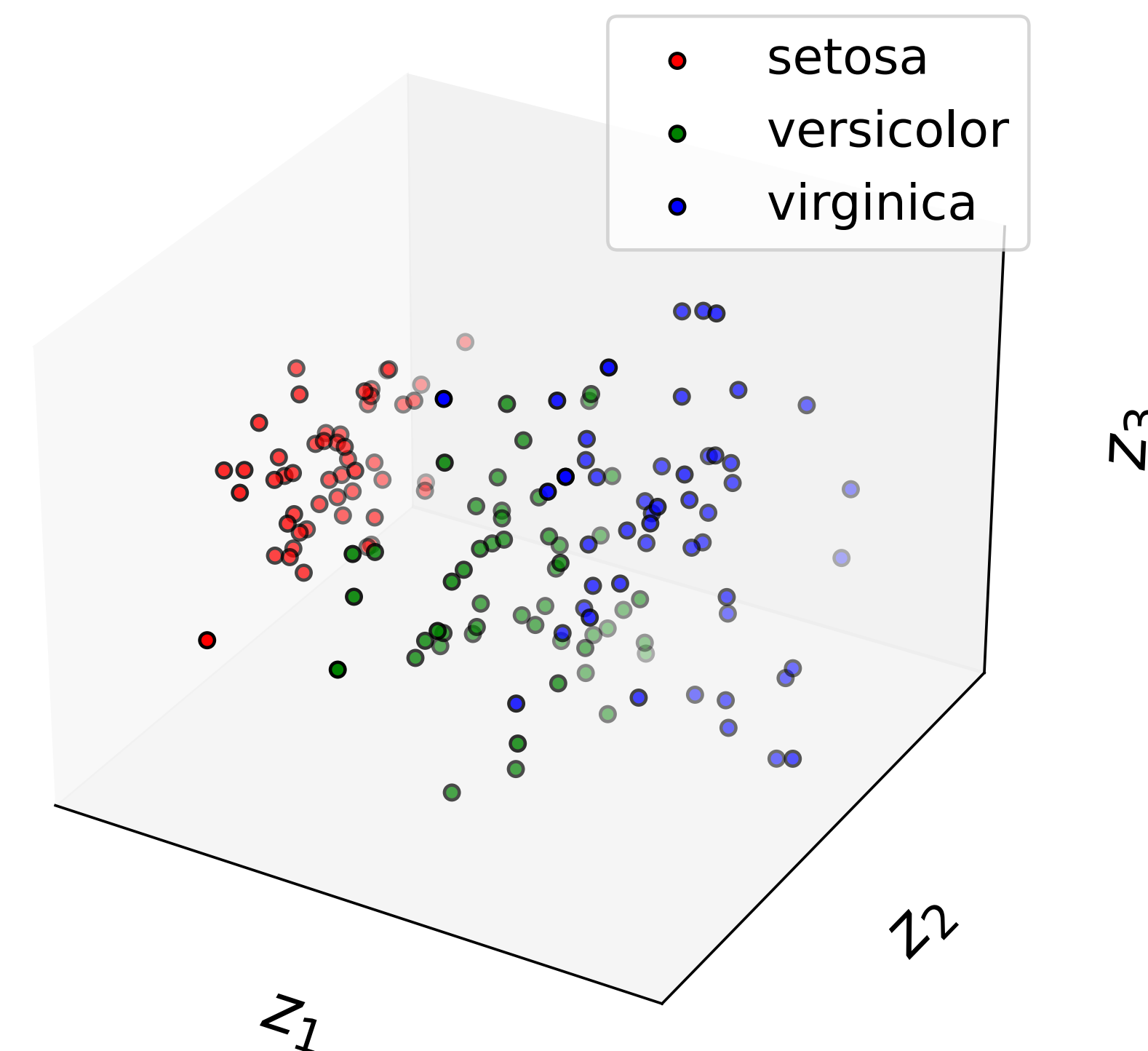
Explaining variance of irises



1D: 73%

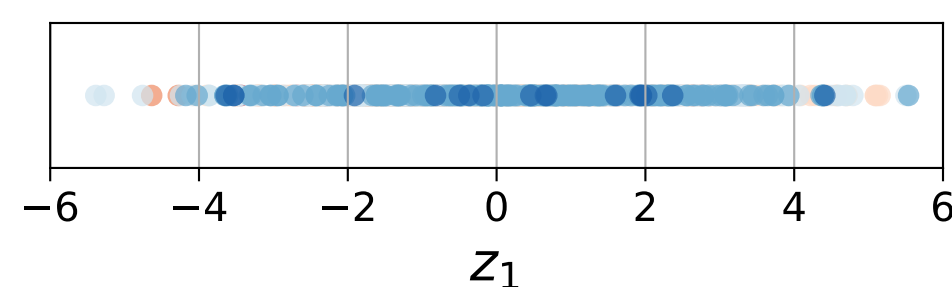


2D: 96%

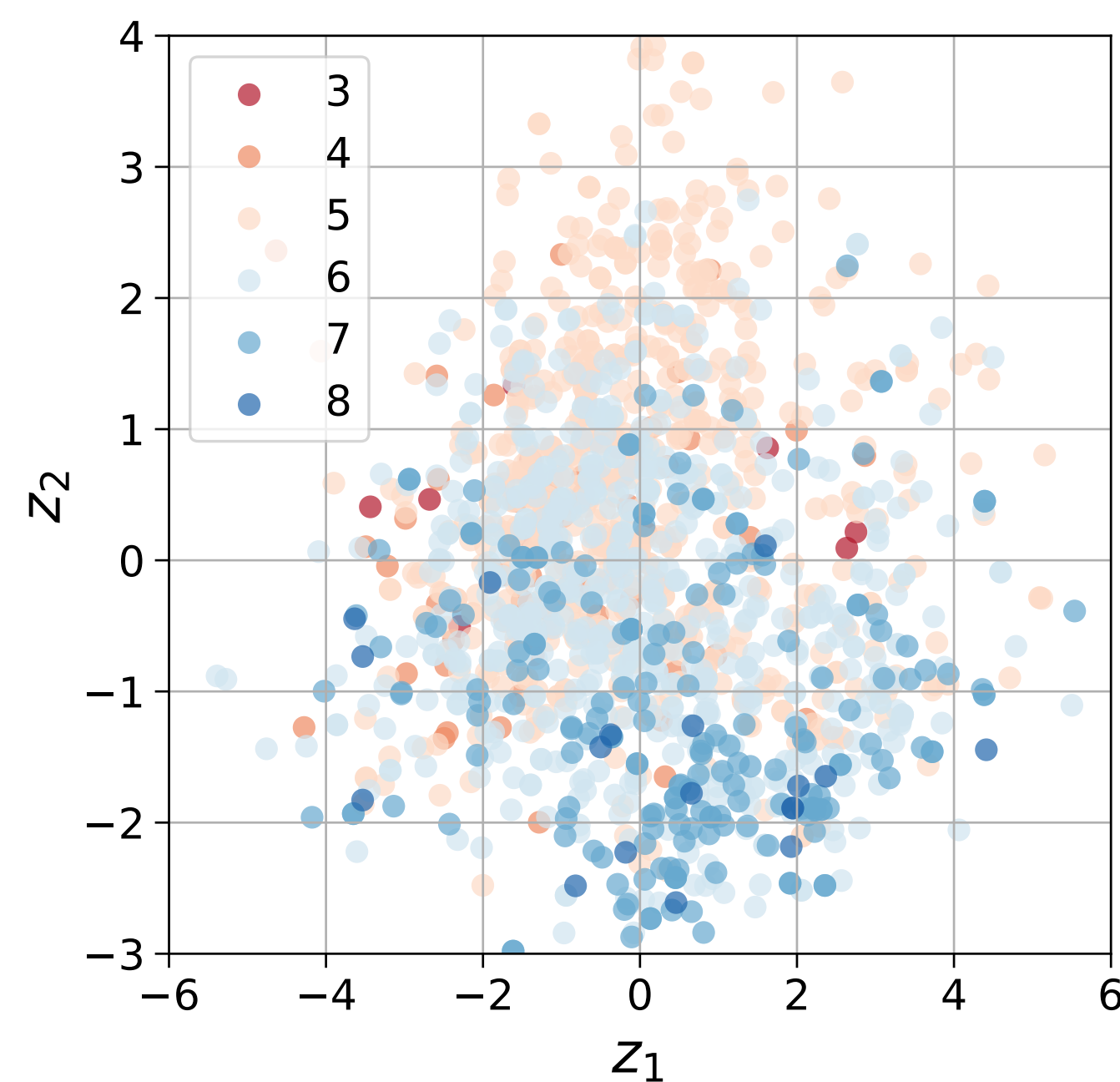


3D: 99%

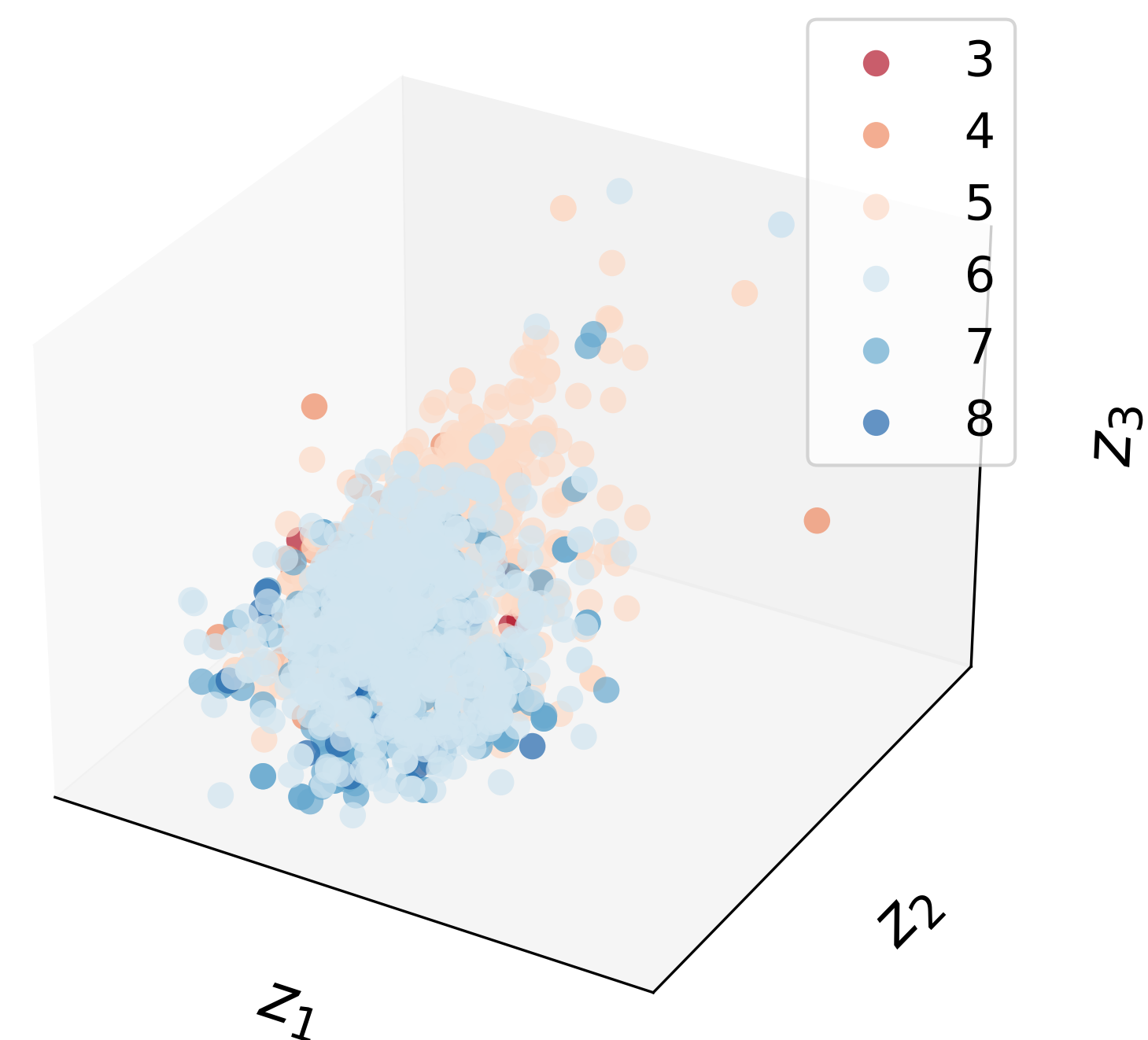
Explaining variance of wine



1D: 28%



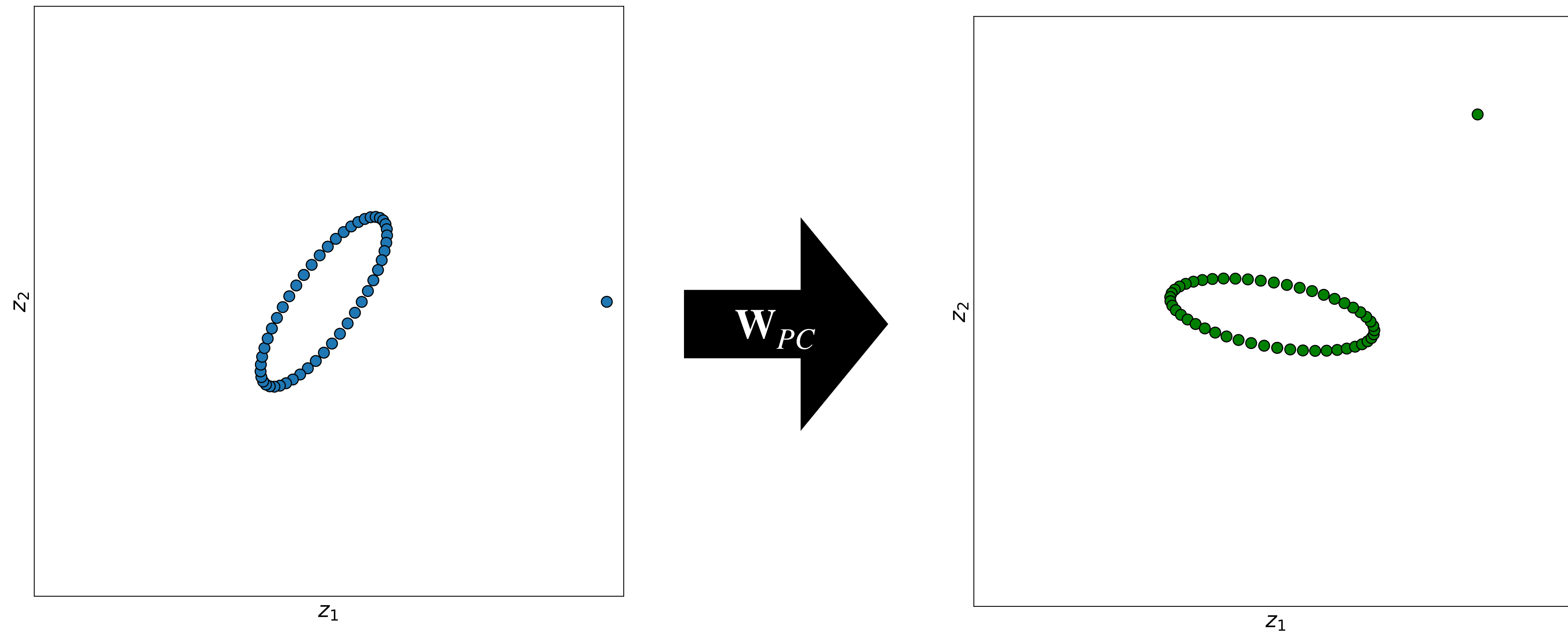
2D: 45%



3D: 60%

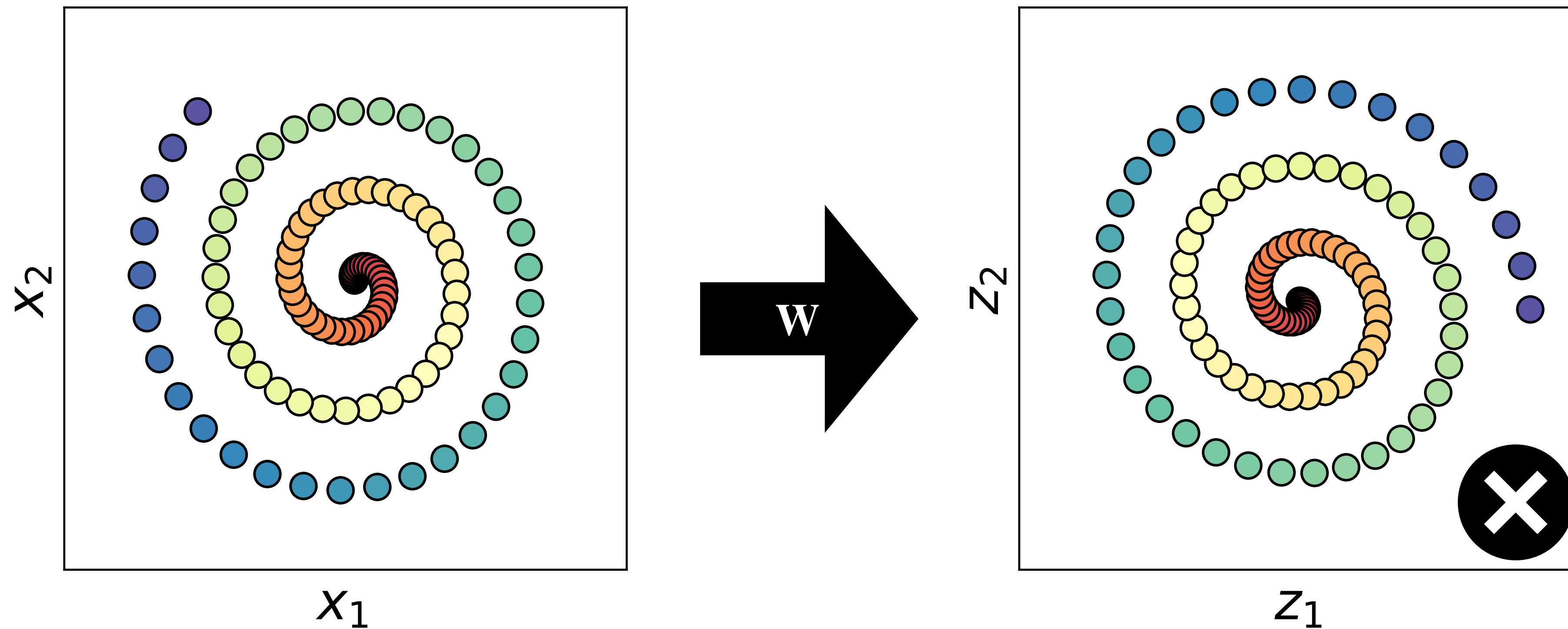
Limitations: PCA is susceptible to outliers

Outliers can change the direction of maximum variance



Limitations: PCA is linear

If the true direction of maximum variance isn't a line, PCA can't find it



t-distributed Stochastic neighbour embedding (t-SNE)

Start with Stochastic neighbour embeddings (SNE)

- Consider a high-dimensional dataset $\{\mathbf{x}^{(n)}\}_{n=1}^N$ where $\mathbf{x} \in \mathbb{R}^D$
- We want to pair all our points s.t. pairings are preserved in a low dimensional embedding space we can visualise (that is almost always 2D)
- Let's define the probability that $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ are a pair as:

σ is a hyperparameter that is implicitly configured through another hyperparameter called “perplexity”

$$p_{i,j} = \frac{\exp(-\frac{1}{2\sigma^2} \|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|^2)}{\sum_{k \neq l} \exp(-\frac{1}{2\sigma^2} \|\mathbf{x}^{(k)} - \mathbf{x}^{(l)}\|^2)}$$

This probability must be zero if $i = j$. No self-pairing allowed!

- Points close together are more likely to be paired

A numerical example for $\sigma^2 = 10$ with MNIST

$$p_{i,j} = \frac{\exp(-\frac{1}{2\sigma^2} \|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|^2)}{\sum_{k \neq l} \exp(-\frac{1}{2\sigma^2} \|\mathbf{x}^{(k)} - \mathbf{x}^{(l)}\|^2)}$$

				
	0.00	0.14	0.02	0.03
	0.14	0.00	0.00	0.00
	0.02	0.00	0.00	0.30
	0.03	0.00	0.30	0.00

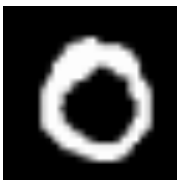
We want pairings to be preserved in low dimensions

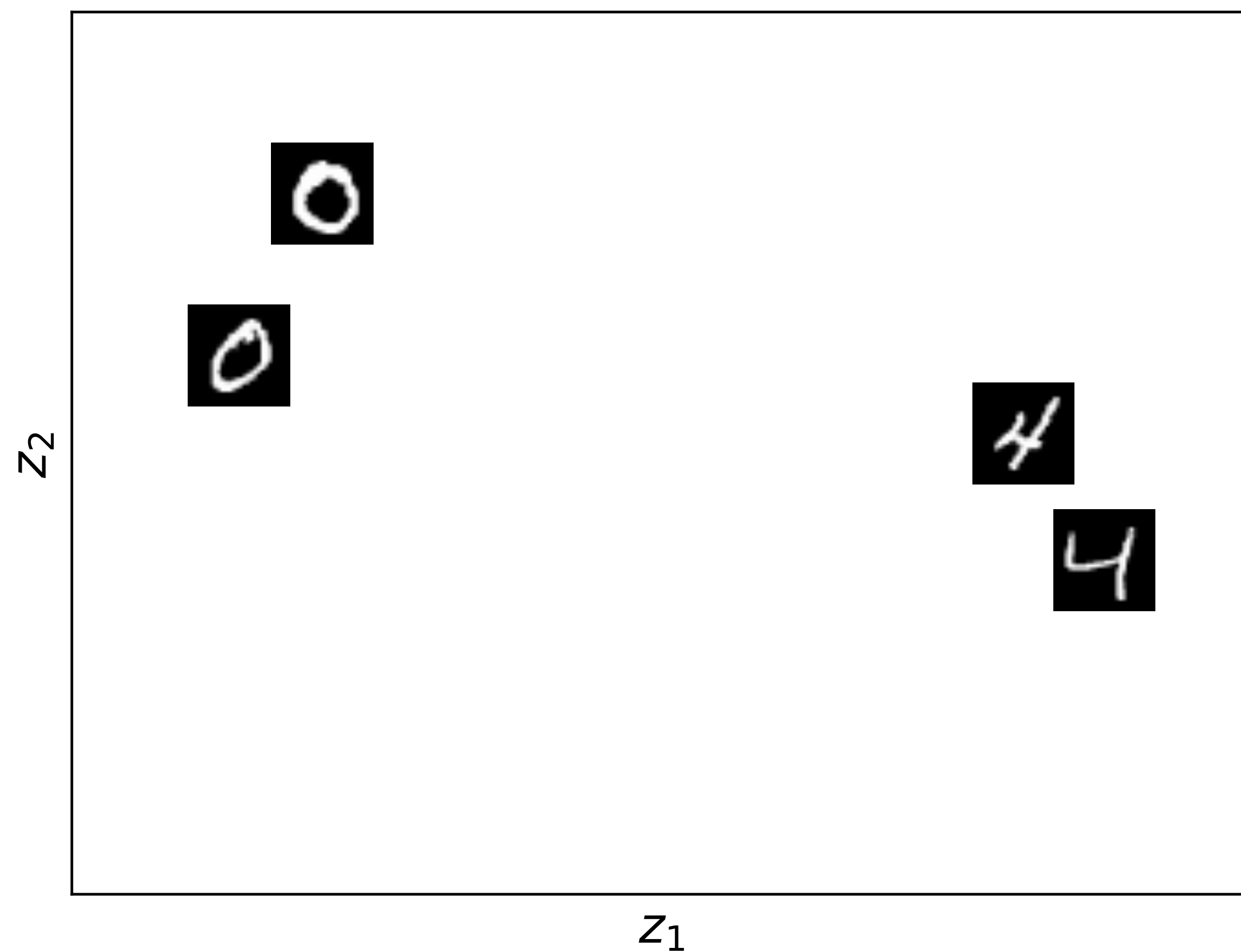
- Consider low dimensional embeddings of our data $\{\mathbf{z}^{(n)}\}_{n=1}^N$ where $\mathbf{z} \in \mathbb{R}^d$
- In this space, we can define the probability that $\mathbf{z}^{(i)}$ and $\mathbf{z}^{(j)}$ are a pair as:

$$q_{i,j} = \frac{\exp(-\|\mathbf{z}^{(i)} - \mathbf{z}^{(j)}\|^2)}{\sum_{k \neq l} \exp(-\|\mathbf{z}^{(k)} - \mathbf{z}^{(l)}\|^2)}$$

- We want all our $q_{i,j}$ to be similar to our $p_{i,j}$
- We need to place our embeddings so that this happens

Retain neighbours in low dimensions

				
	0.00	0.14	0.02	0.03
	0.14	0.00	0.00	0.00
	0.02	0.00	0.00	0.30
	0.03	0.00	0.30	0.00



How do we do this?

- All our $p_{i,j}$ define a discrete probability distribution P
- All our $q_{i,j}$ define a discrete probability distribution Q

Just think of these as bar charts where the bars sum to 1

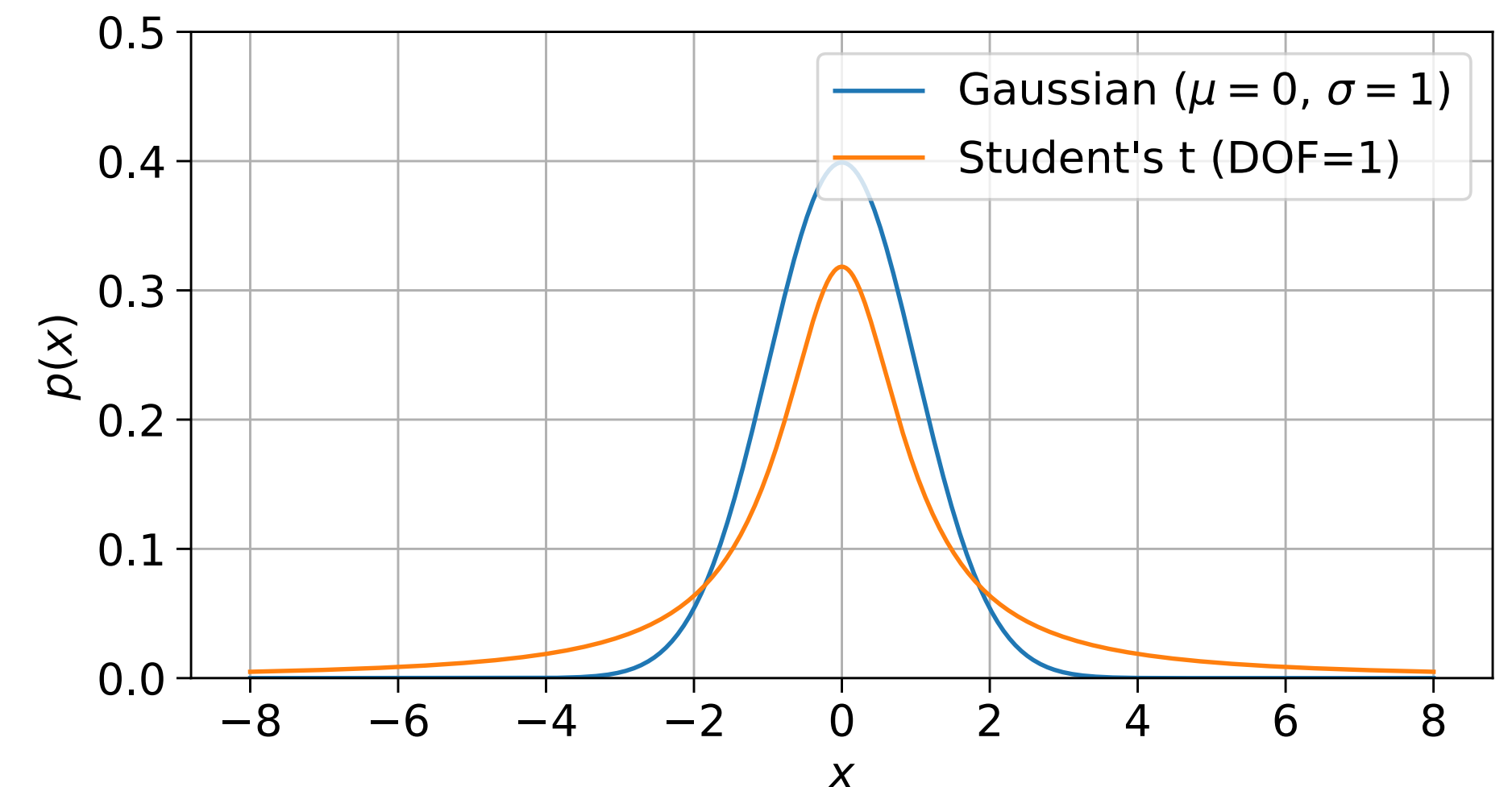
- The Kullback-Leibler (KL) divergence $D_{KL}(P||Q) = \sum_i \sum_j p_{i,j} \log \frac{p_{i,j}}{q_{i,j}}$
measures how different P is from Q

- We can find the embeddings that minimise this quantity. It's another optimisation problem! Solve $\underset{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \dots, \mathbf{z}^{(N)}}{\text{minimise}} D_{KL}(P||Q)$

That was SNE. What is t-SNE?

- The problem with SNE is points that are quite far apart in the original space tend to get pushed together in the embedding space
- In t-SNE, instead of pair probabilities in the embedding space being Gaussian, they are Student's t distributed with 1 DOF (a Cauchy distribution)
- This has long tails, so points that are well separated in the original space will be reasonably well spaced in the embedding space

$$q_{i,j} = \frac{(1 + \|\mathbf{z}^{(i)} - \mathbf{z}^{(j)}\|^2)^{-1}}{\sum_{k \neq l} (1 + \|\mathbf{z}^{(k)} - \mathbf{z}^{(l)}\|^2)^{-1}}$$



The gradient

- To minimise the effect of outliers, $p_{i,j}$ can be redefined as $\frac{p_{i|j} + p_{j|i}}{2N}$ where

$$p_{j|i} = \frac{\exp(-\frac{1}{2\sigma^2} \|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|^2)}{\sum_{k \neq i} \exp(-\frac{1}{2\sigma^2} \|\mathbf{x}^{(i)} - \mathbf{x}^{(k)}\|^2)}$$

This re-definition really isn't important to conceptual understanding. Don't stress.

- We can use gradient descent to solve minimise $D_{KL}(P\|Q)$
 $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N$

$$\nabla_{\mathbf{z}_i} D_{KL} = 4 \sum_j (p_{i,j} - q_{i,j})(\mathbf{z}^{(i)} - \mathbf{z}^{(j)})(1 + \|\mathbf{z}^{(i)} - \mathbf{z}^{(j)}\|^2)^{-1}$$

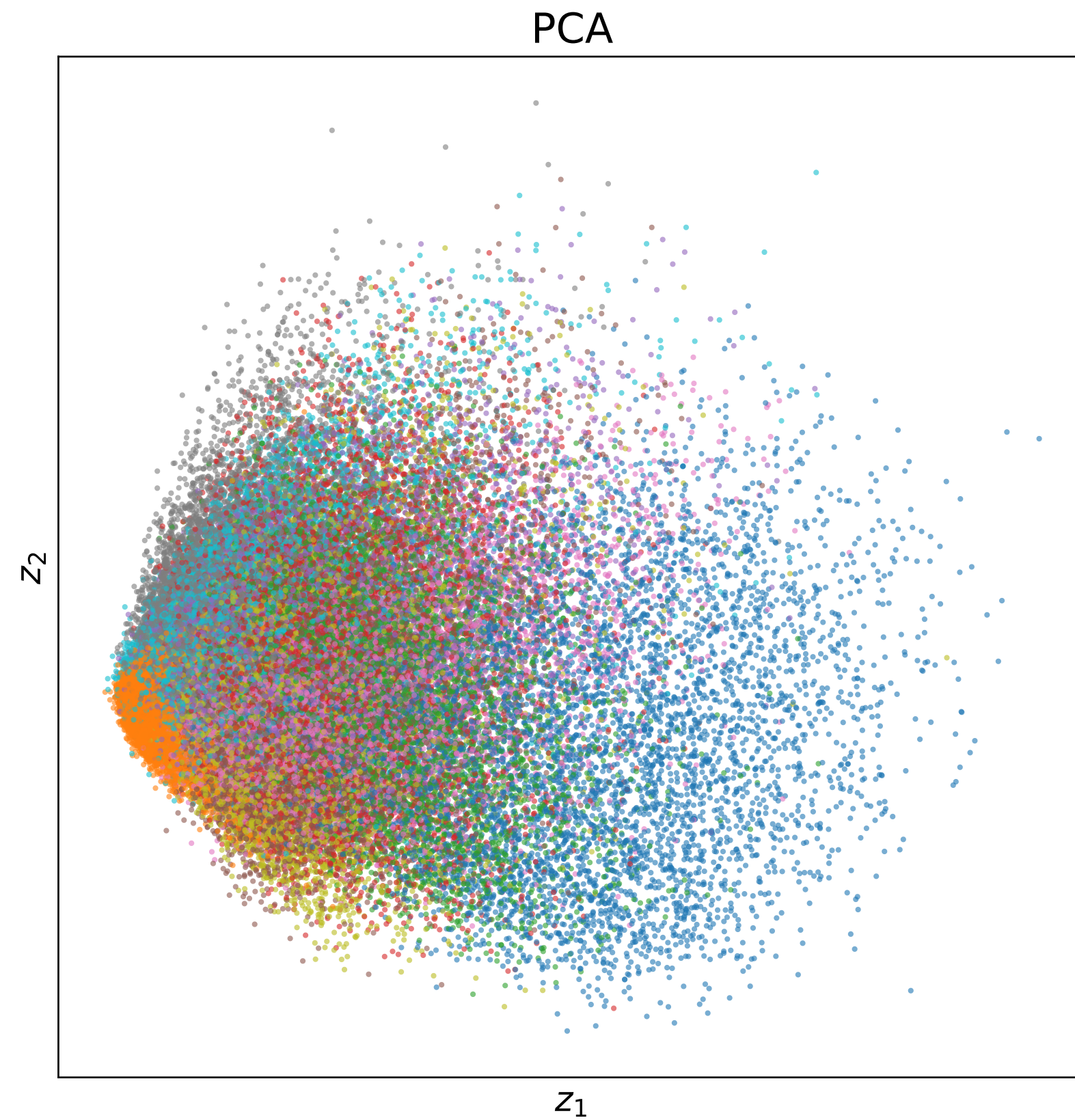
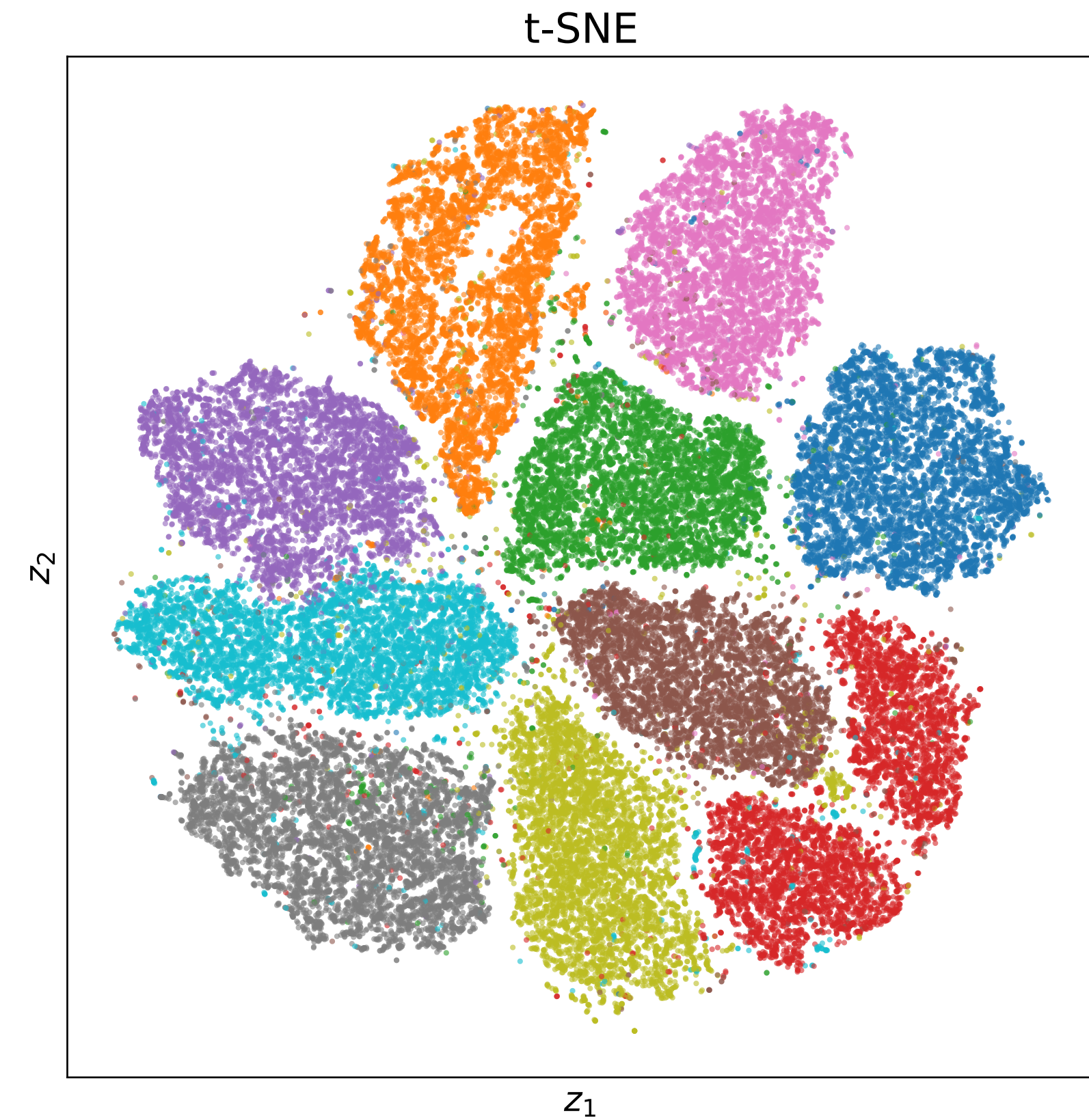
This gradient is actually quite intuitive!

$$\nabla_{\mathbf{z}_i} D_{KL} = 4 \sum_j (p_{i,j} - q_{i,j})(\mathbf{z}^{(i)} - \mathbf{z}^{(j)})(1 + \|\mathbf{z}^{(i)} - \mathbf{z}^{(j)}\|^2)^{-1}$$

- This gradient is a product of three terms
- Can you interpret what they do?

PCA and t-SNE on MNIST

- Different colours are different digits!



Clustering with K-means (optional)

Motivation

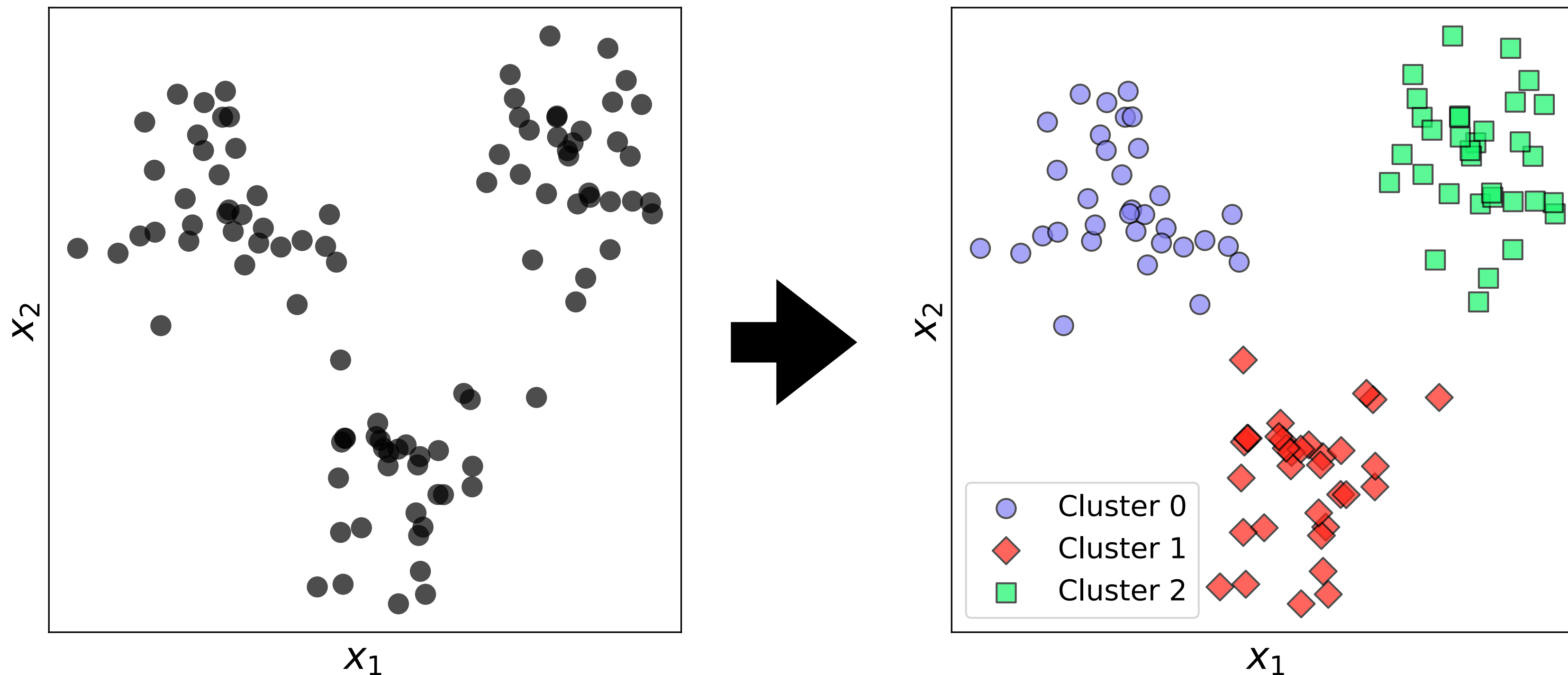
You have a dataset that you want to split into groups

- people with low, medium, high income for marketing
- grouping shoppers to recommend products
- identifying personality types for a dating website



K-means

- We can use K-means to automatically split our dataset into groups
- Other clustering algorithms are available!

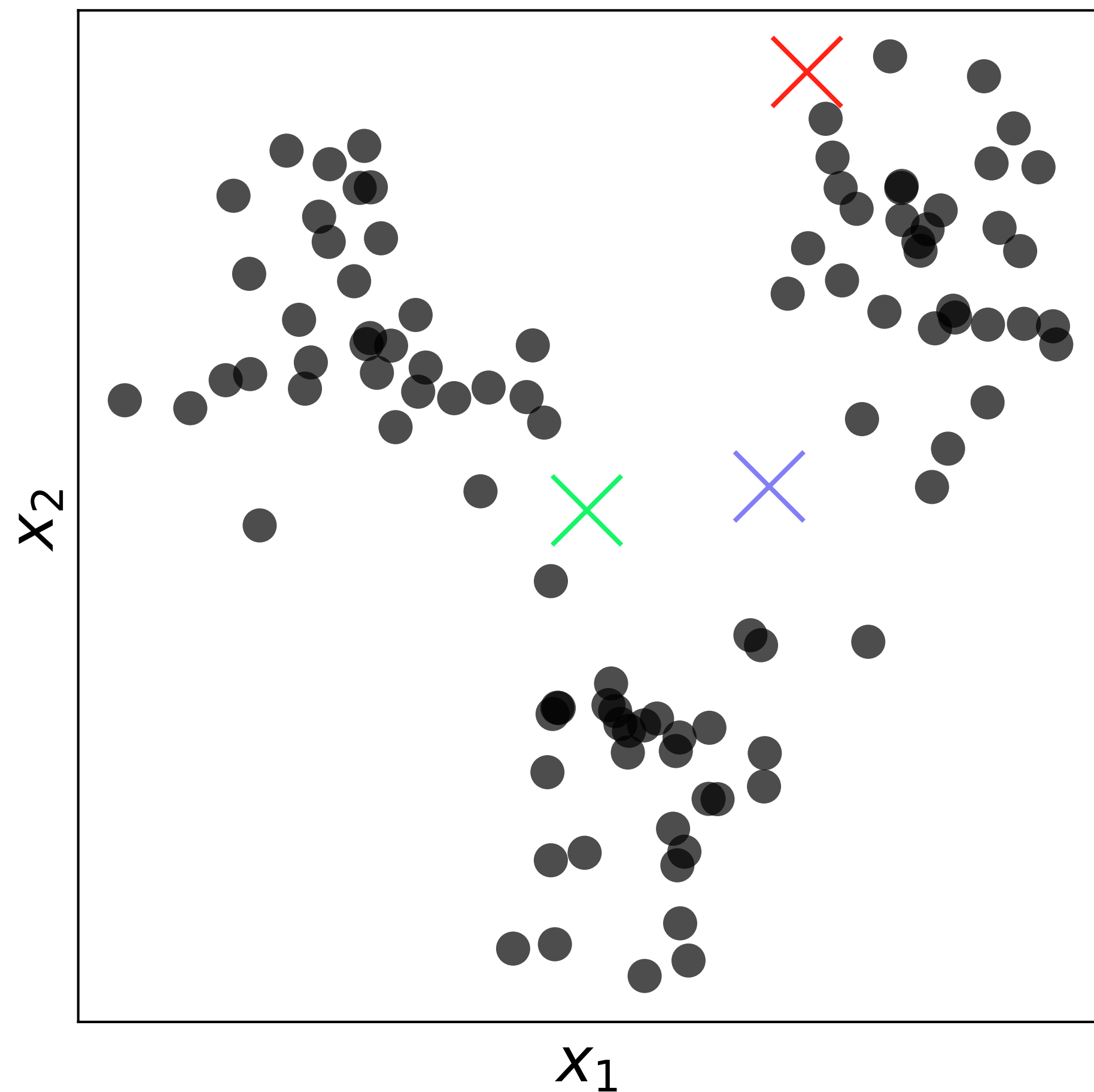


K-means algorithm

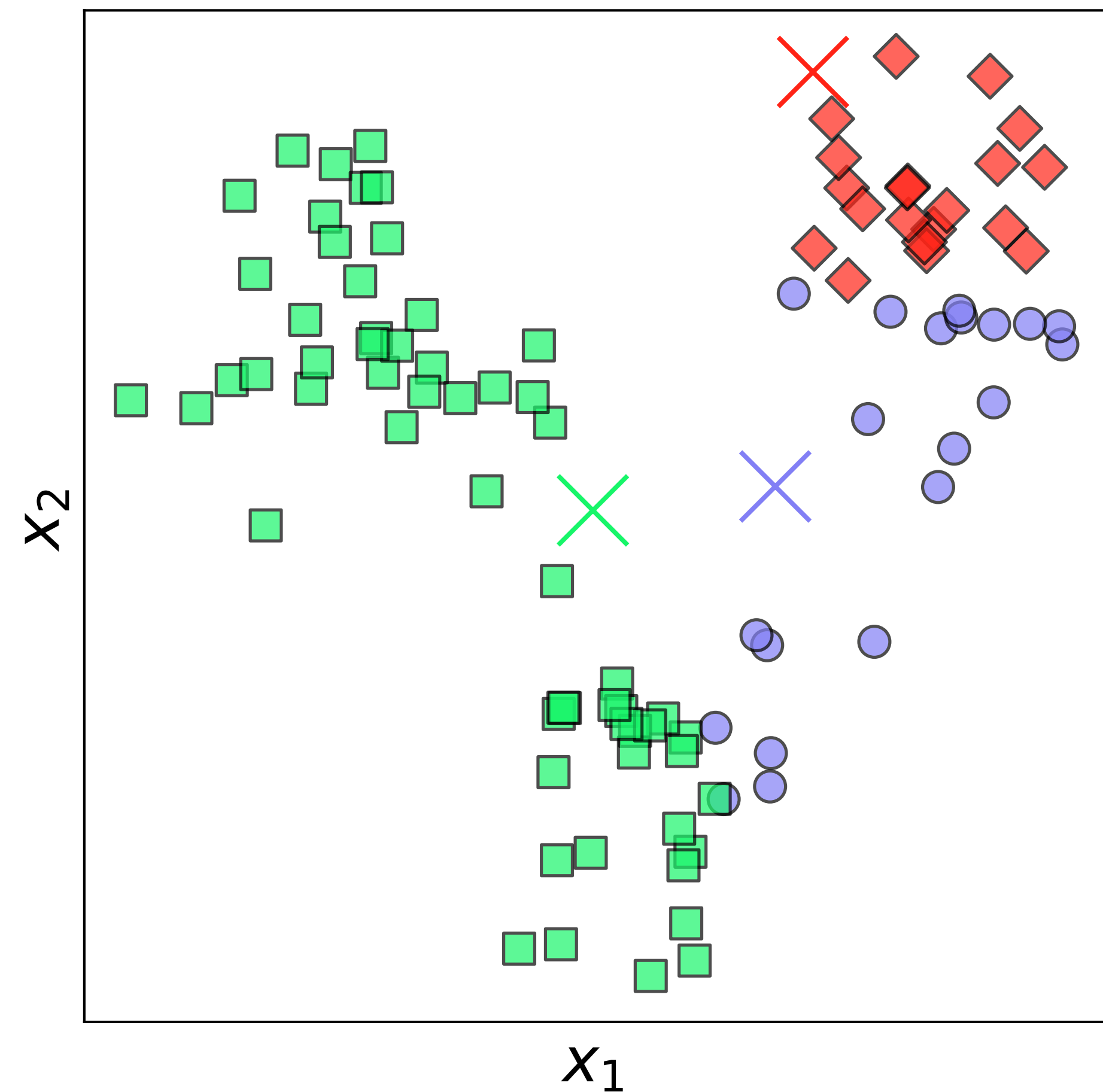
- Select the number of clusters K
- Initialise the cluster centres $\{\mathbf{c}_k\}_{k=1}^K$ at random
- Repeat:
 1. Assign each (ideally standardised) data point to its nearest cluster centre
 2. Update cluster centres as mean of their assigned points
- Until no change

K-means walkthrough with $K = 3$

Initialise the cluster centres $\{\mathbf{c}_k\}_{k=1}^K$ at random

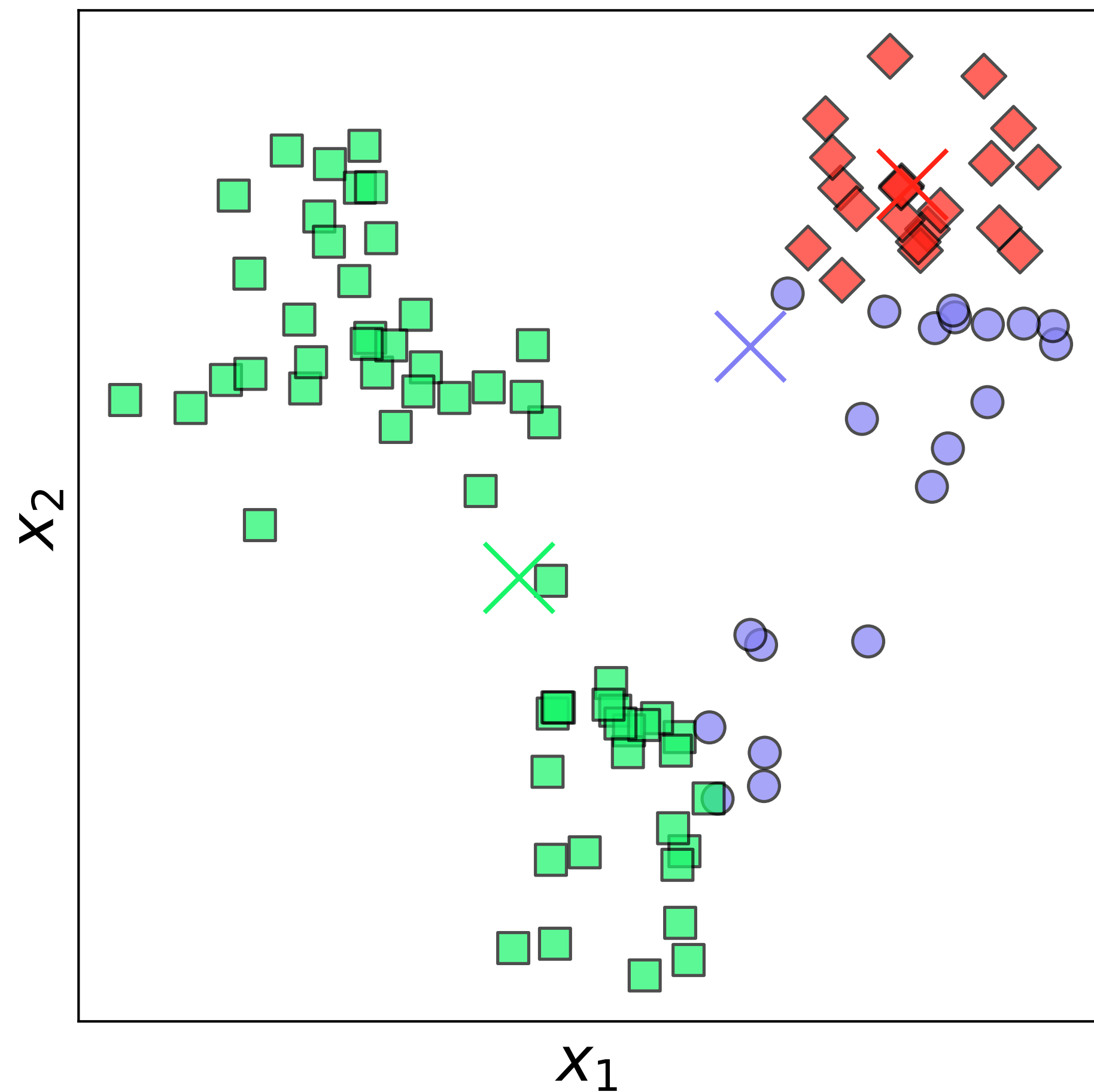
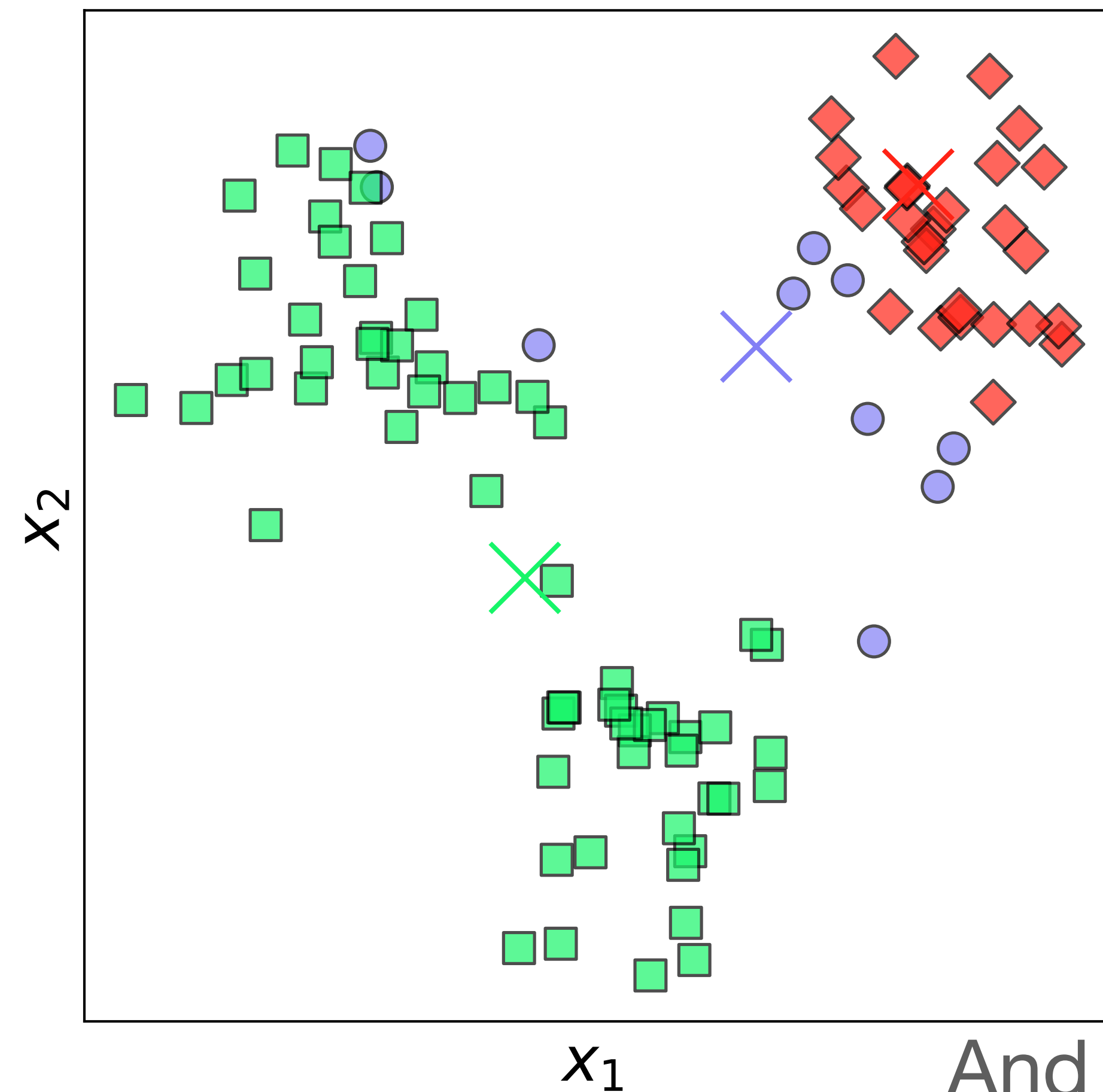


Assign each data point to its nearest cluster centre



K-means walkthrough with $K = 3$

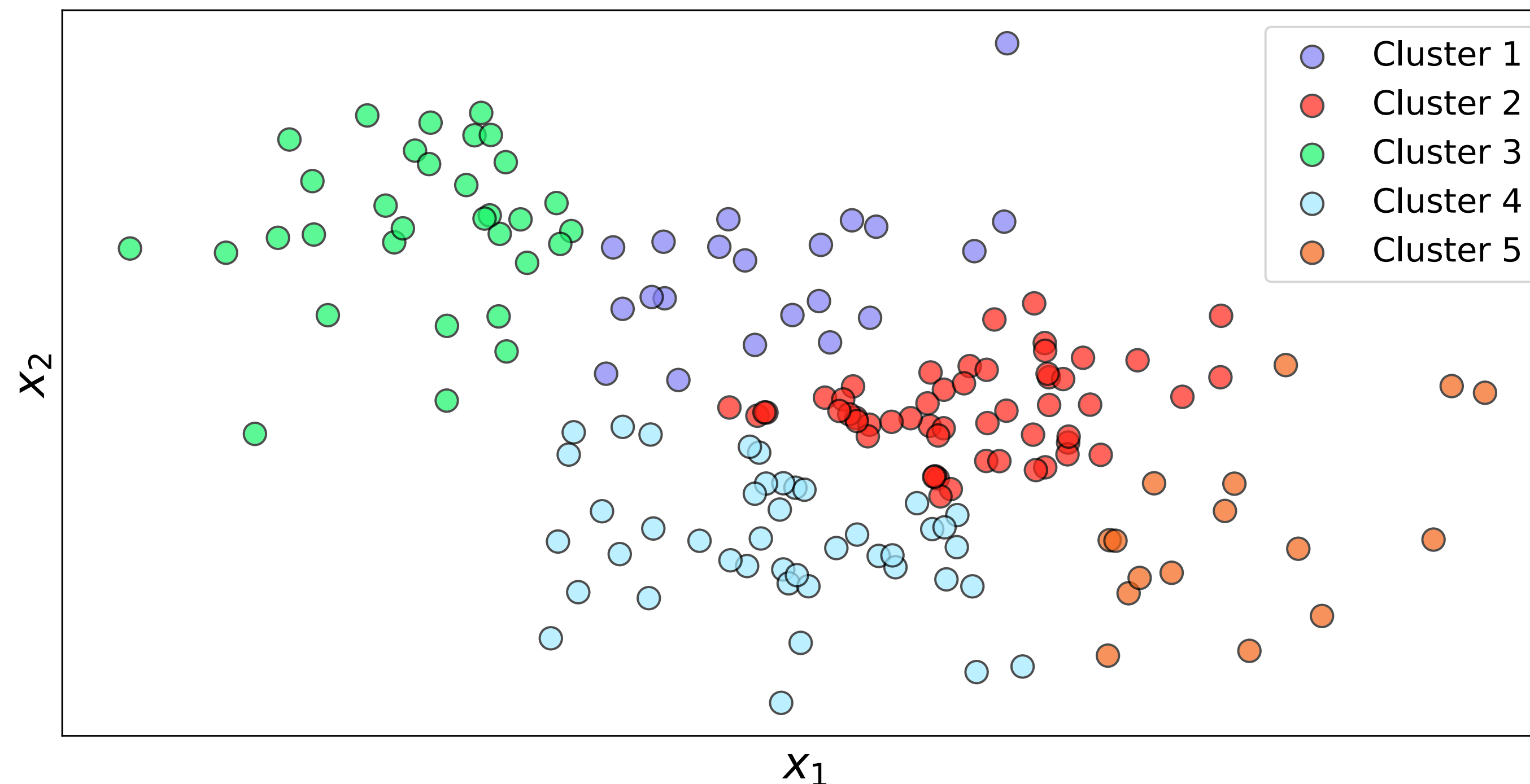
Update cluster centres as mean of their assigned points Assign each data point to its nearest cluster centre



And so on! 60

Warning!

- K-means is very sensitive to where the initial cluster centres are placed
- The number of clusters is user defined
- The clusters **might not be meaningful**



This data is just noise!

Summary

- We have looked at summary statistics
- We have seen many different ways to visualise data
- We have learnt about PCA and how it can be used to perform dimensionality reduction through a linear transformation
- We have learnt about t-SNE and seen that it is non-linear and directly learns embeddings

Questions

Q1

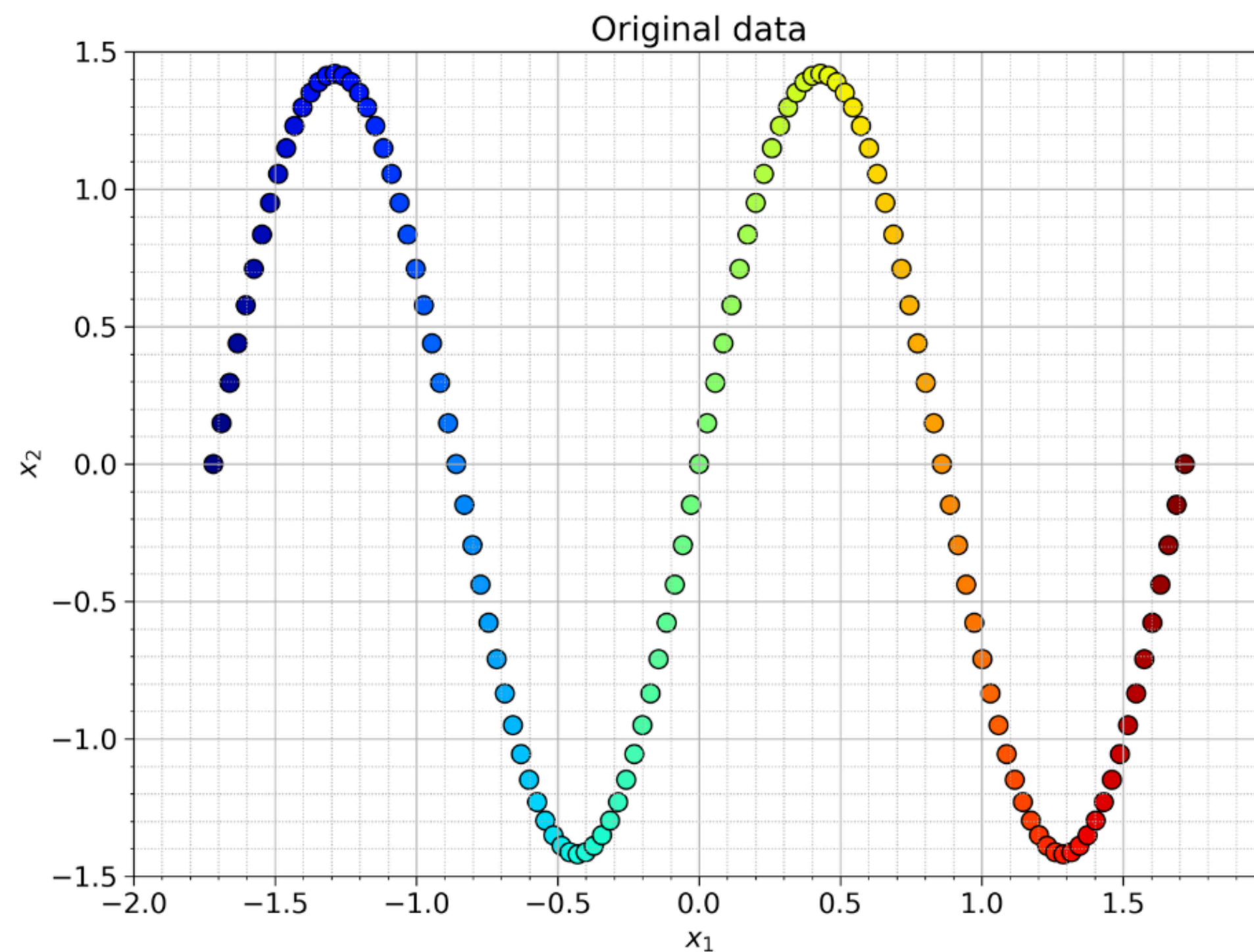
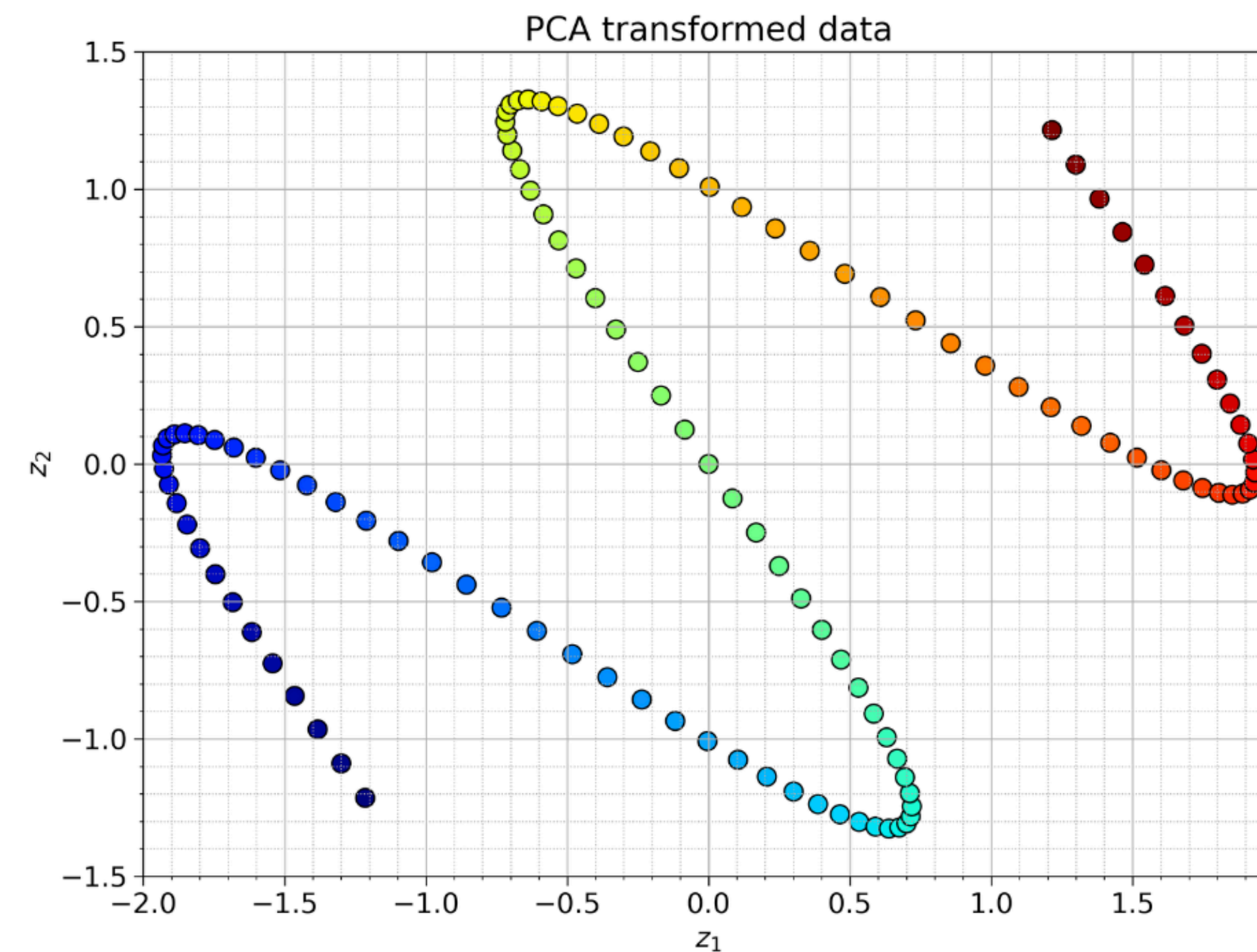
Identify two issues with the data visualisation below.



Q2

The figure shows a 2D dataset before and after PCA (mapping to 2D) is applied.

What is the first principal component of the dataset?



Q3

Caitlyn has a dataset of 10-D features that she wants to visualise. She applies PCA and records the fraction of the variance explained by the first d components as a function of d in the figure.

She asks you what dimensional space she should project her data down to. What would you tell her, and why?

