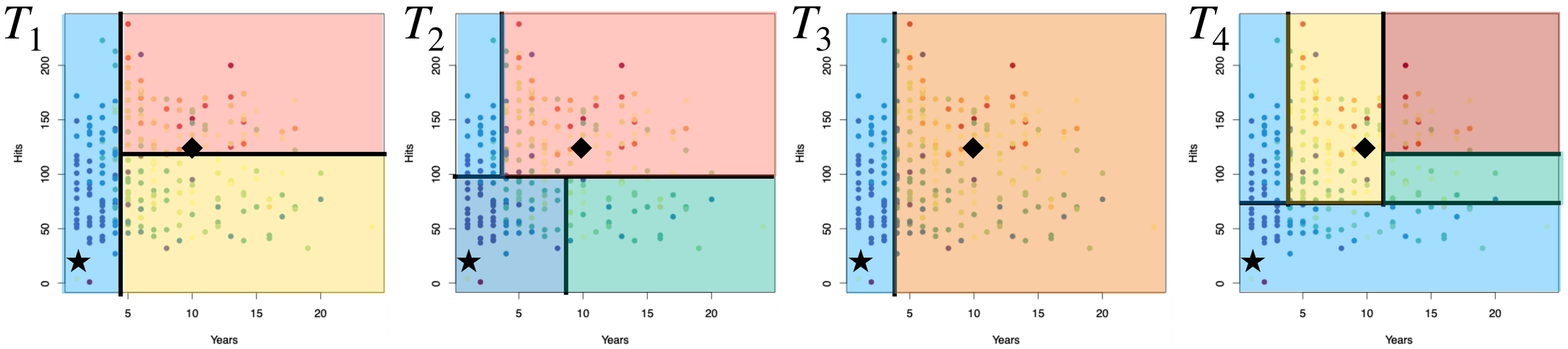
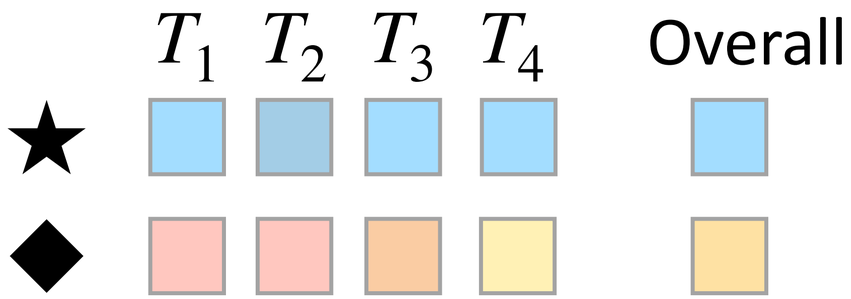
https://github.com/tedinburgh/ads2023

Dimensionality reduction: PCA

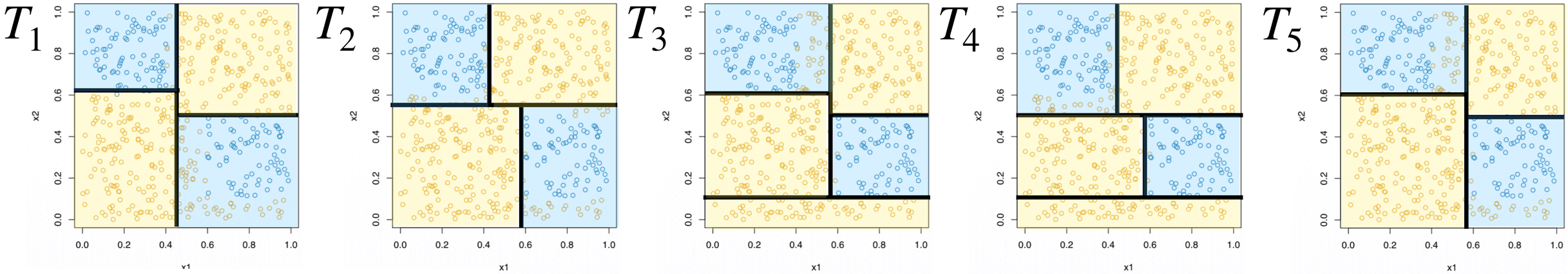
Tom Edinburgh
te269

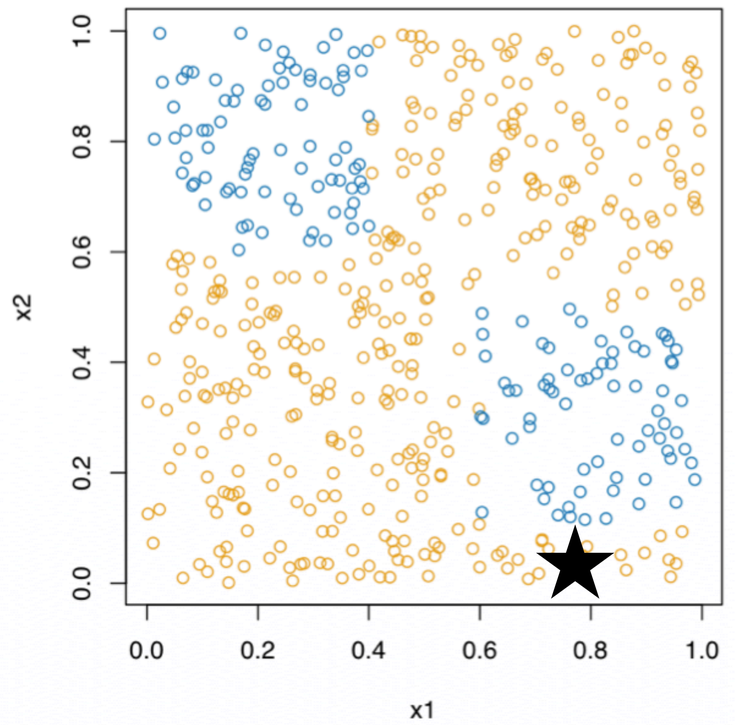
Last time: How to aggregate predictions from bagging



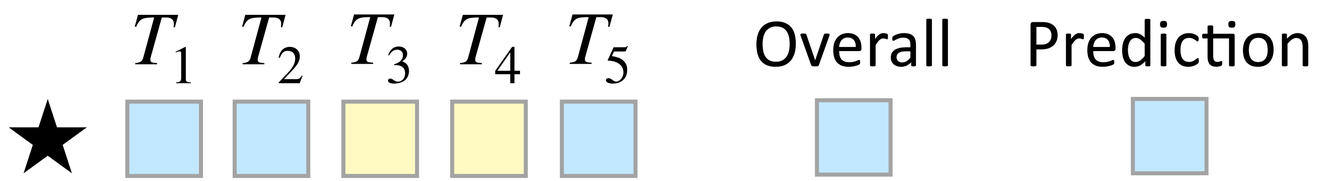


Last time: How to aggregate predictions from bagging

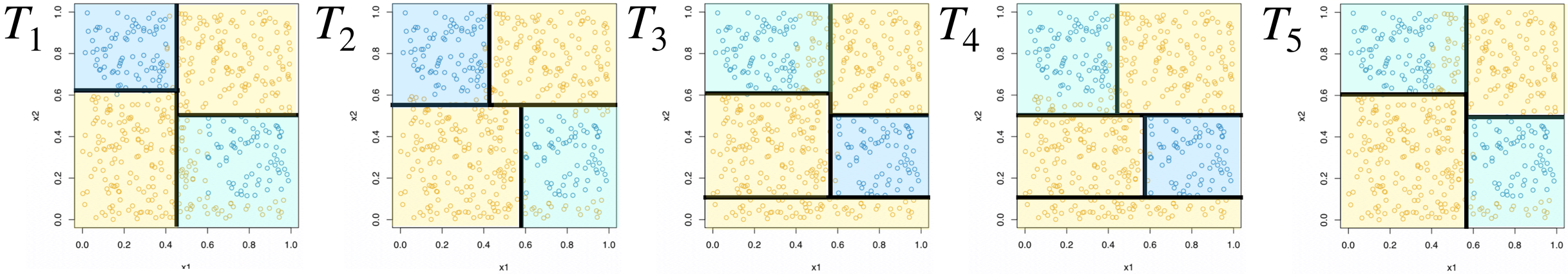


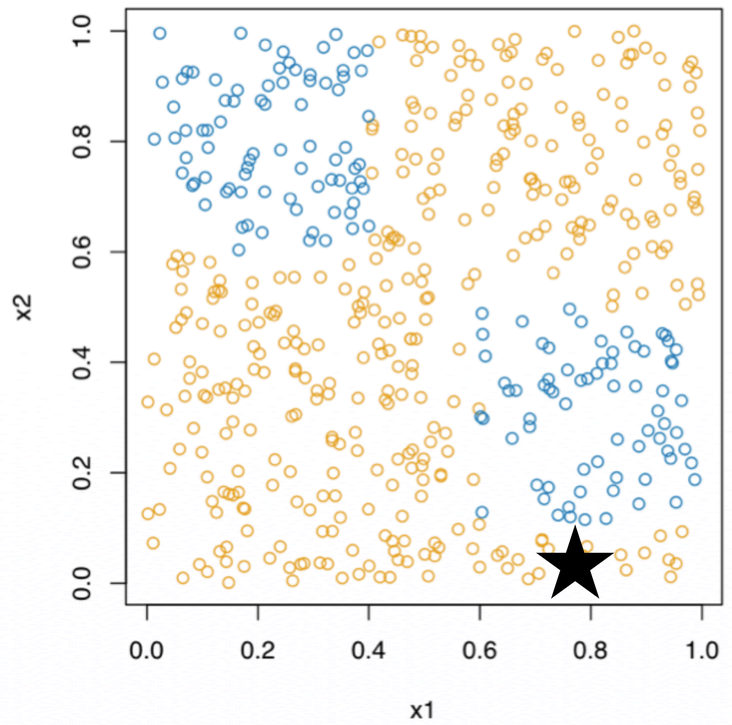


Majority voting



Last time: How to aggregate predictions from bagging





Probability voting

Today: PCA

- Overview
- Projection
- Eigenvectors
- Explained variance
- Information
- Singular value decomposition

Questions: halfway through, at the end, or by email (te269)

Resources

- An Introduction to Statistical Learning with Applications in R/Python (James, Witten, Hastie, Tibshirani, Taylor; 2013/2023)
    - Chapter 12

- Slides adapted from:
    - Prof Stephen Eglen, Cambridge

Format of the data

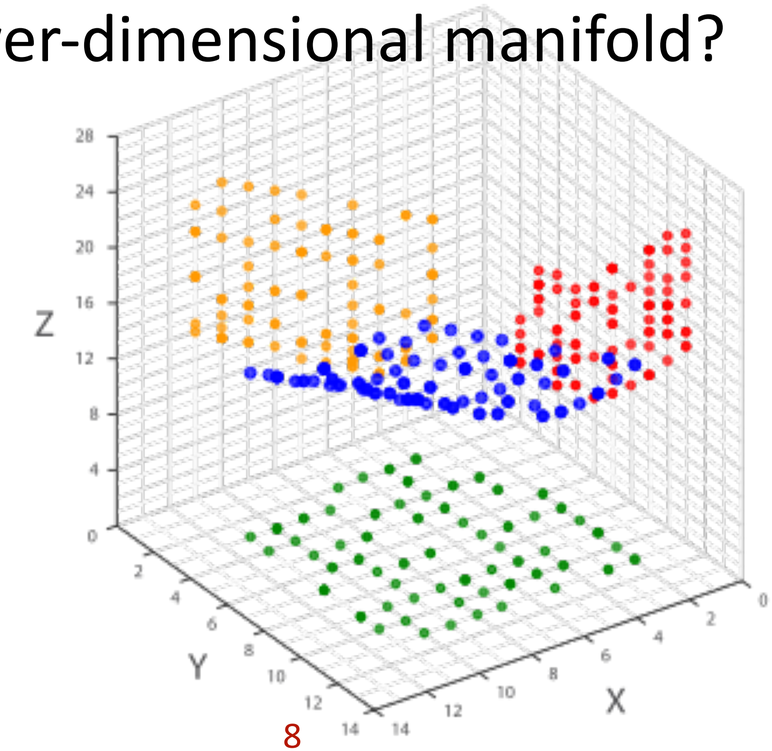
$$X = \begin{pmatrix} X_1 & X_2 & \cdots & X_p \end{pmatrix} = \begin{pmatrix} x_1^T \\ \vdots \\ x_n^T \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}$$

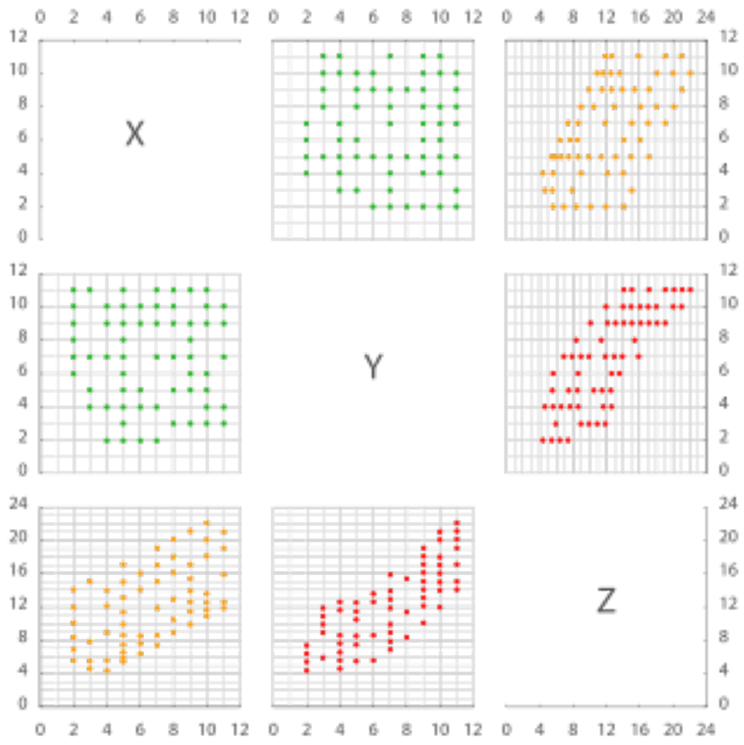
- No response variable y
- How do we visualise high-dimensional data?
- Can we uncover structure within the variables $X_1, X_2, \ldots, X_p$, i.e. subgroups?

- Unsupervised learning is useful as **exploratory data analysis**

Dimensionality reduction

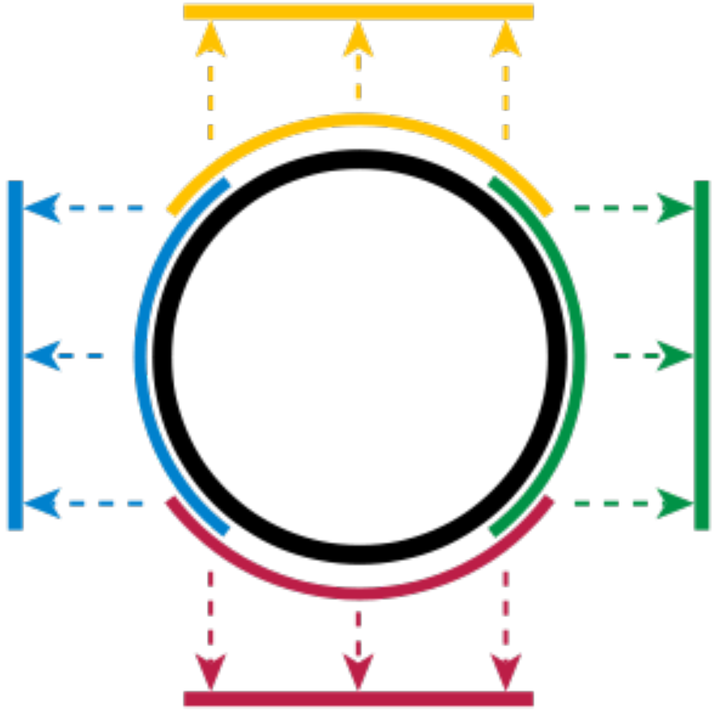
- Condense the data from p vectors down to 2 or 3 vectors
- This involves throwing away some information, but hopefully most of the information still remains in the lower dimensional space
- Does the data exist on a lower-dimensional manifold?

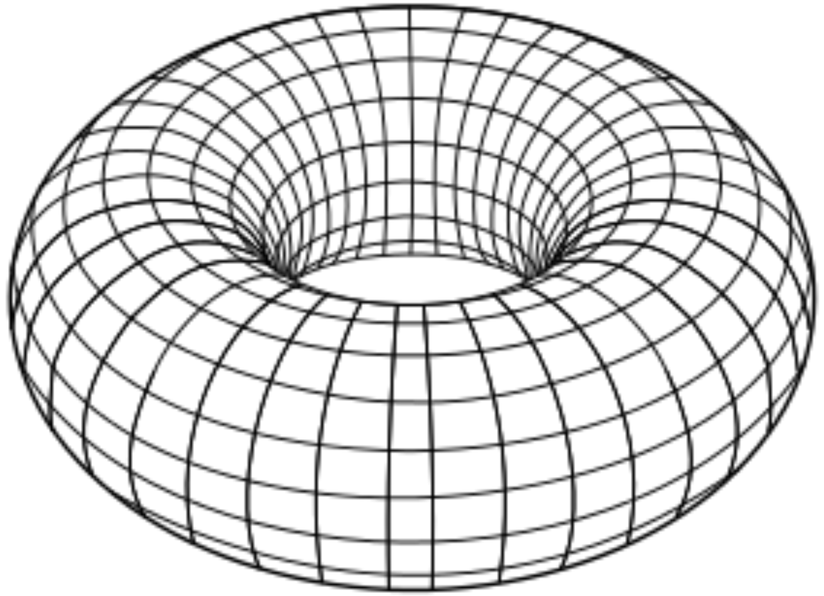




Manifolds

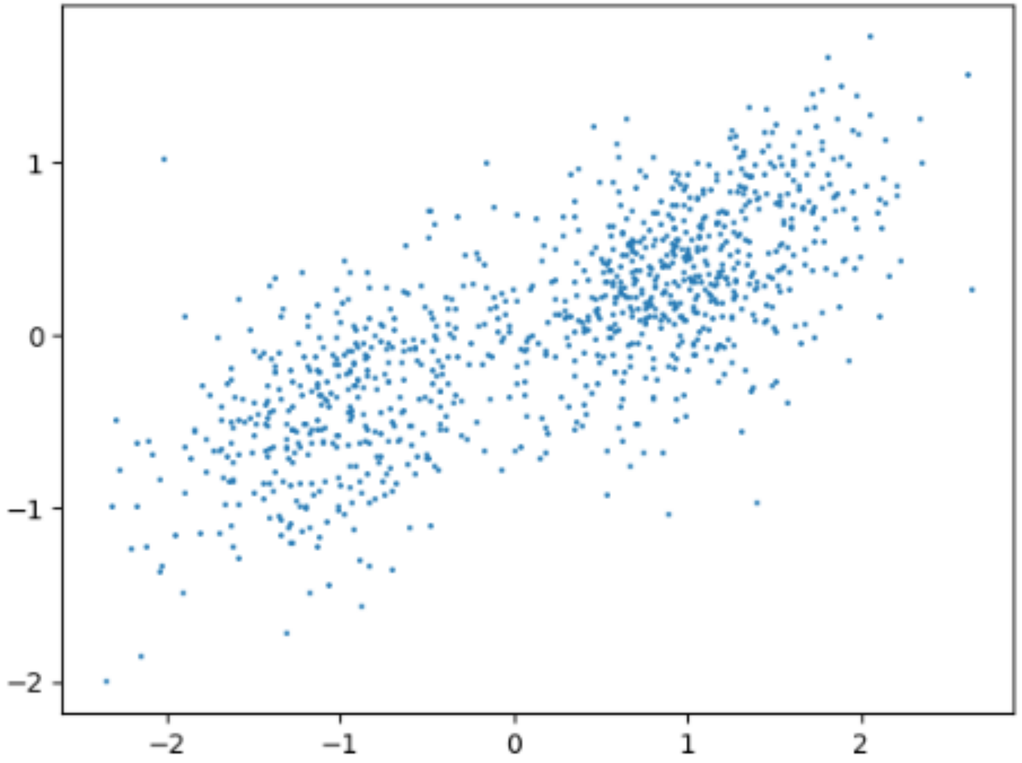
- Topological space that locally resembles Euclidean space at each
- **Manifold hypothesis**: many high-dimensional data sets in the real world lie on low-dimensional latent manifolds inside the high-dimensional space





Principal component analysis

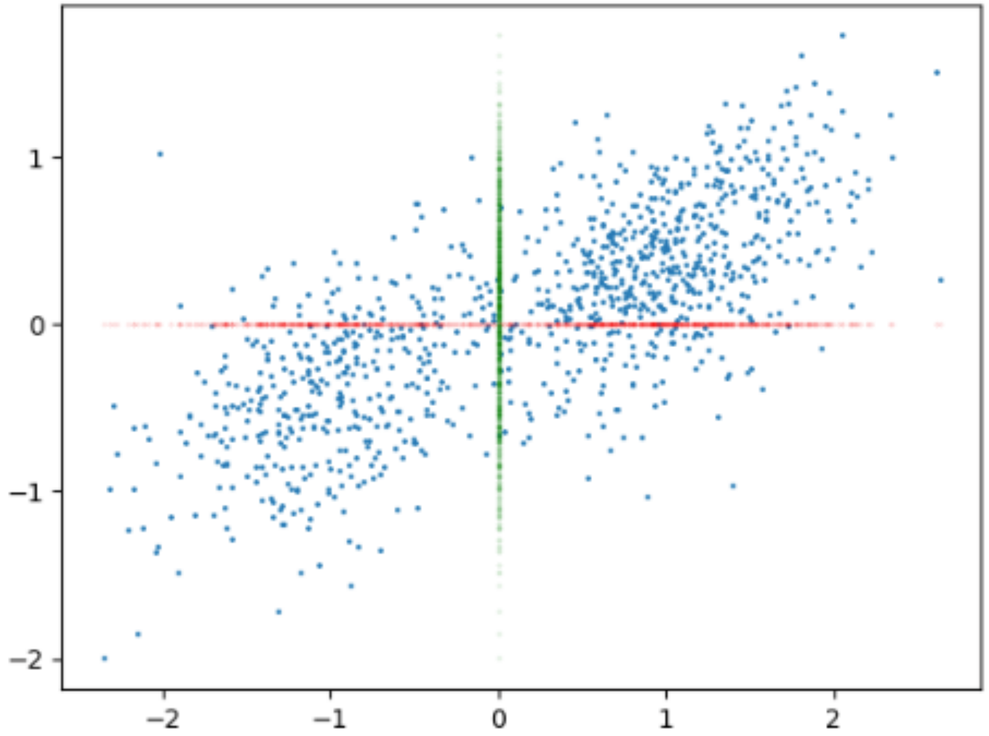
- Principal components are directions in the feature space along which the data are highly variable
- We can use principal components to summarise a set of highly-correlated variables (features) $X_1, X_2, \ldots, X_p$ using a smaller number of variables that explain most of the variability in the data



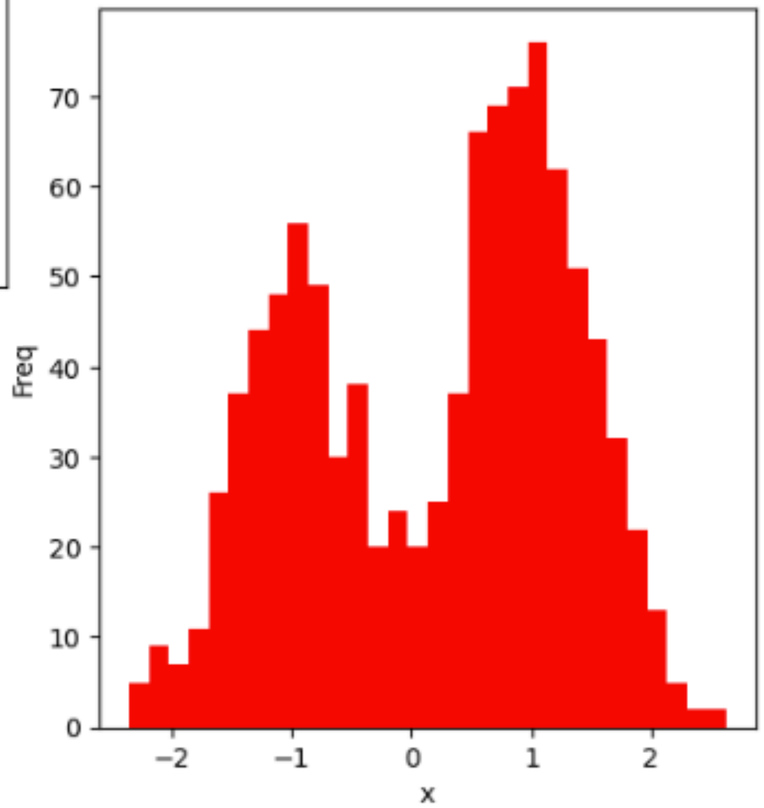
Principal component analysis

- Principal components are directions in the feature space along which the data are highly variable
- We can use principal components to summarise a set of highly-correlated variables (features) $X_1, X_2, \ldots, X_p$ using a smaller number of variables that explain most of the variability in the data
- How do we choose the new variables?

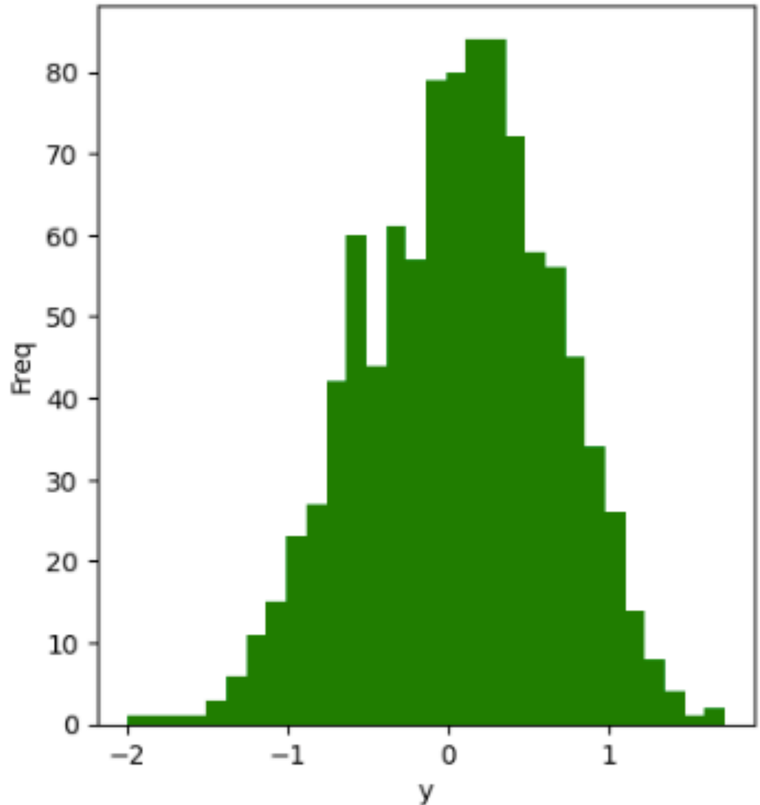
Principal component analysis



Variance = 1.230



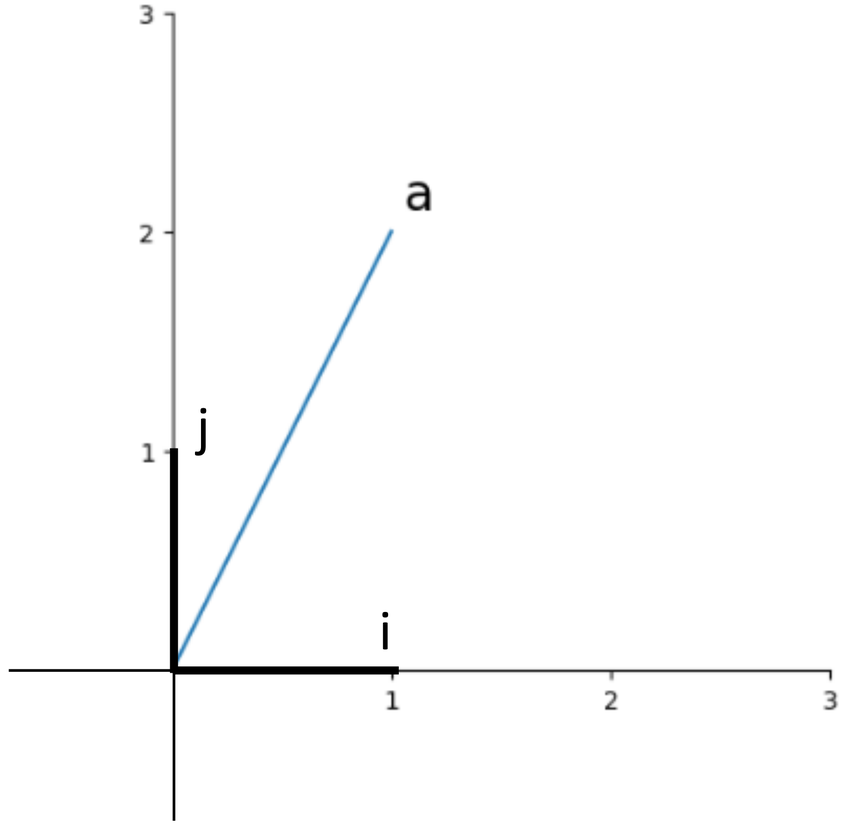
Variance = 0.353



Projection: Cartesian coordinates

- Coordinates are given with respect to basis vectors

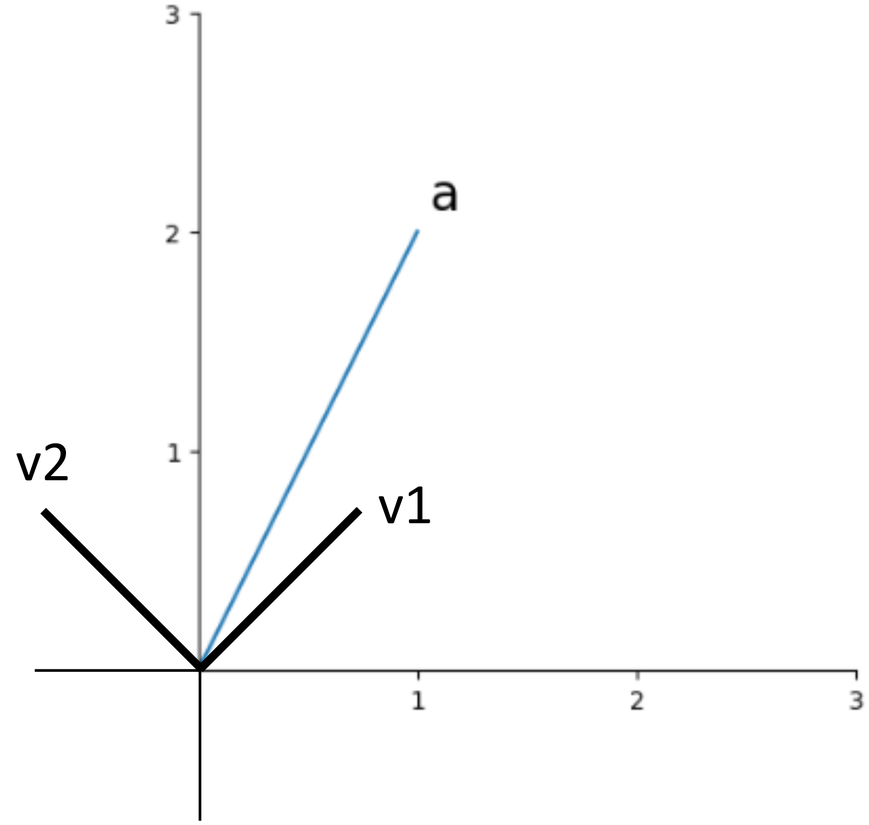
$$a = \begin{pmatrix} 1 \\ 2 \end{pmatrix} = 1i + 2j = 1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + 2 \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$



Projection: a different coordinate system

- Coordinates are given with respect to basis vectors

$$a = \lambda_1 v_1 + \lambda_2 v_2 = \frac{\lambda_1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \frac{\lambda_2}{\sqrt{2}} \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$



Projection: dot product

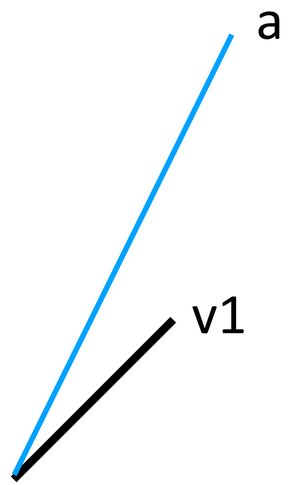
- If we project a onto v_1, what is the distance λ_1?
- Dot product:

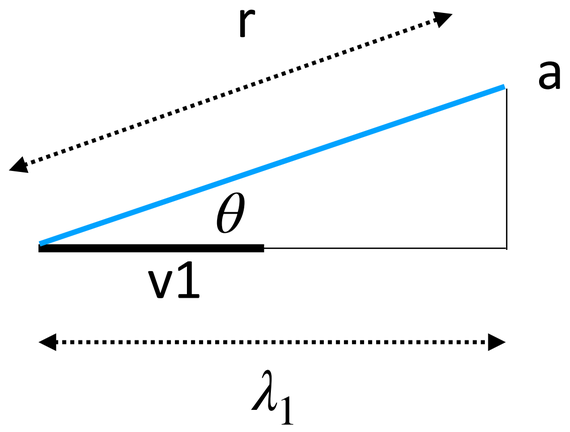
$$x = r\cos(\theta) = |a|\cos(\theta)$$

$$a \cdot v_1 = |a||v_1|\cos(\theta)$$

$$\lambda_1 = \frac{a \cdot v_1}{|v_1|}$$

- Choose v_1 with $|v_1| = 1$





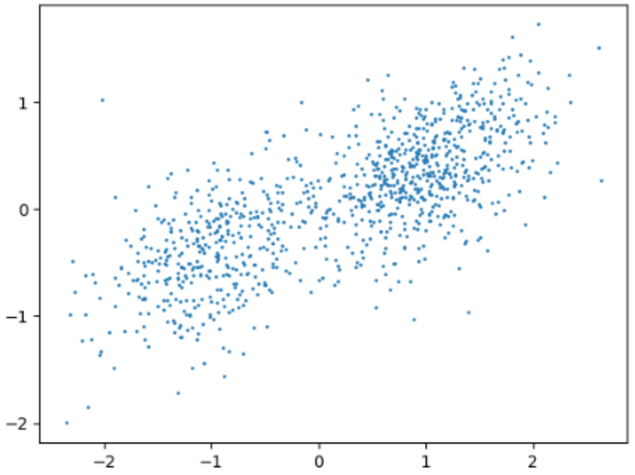
Projection

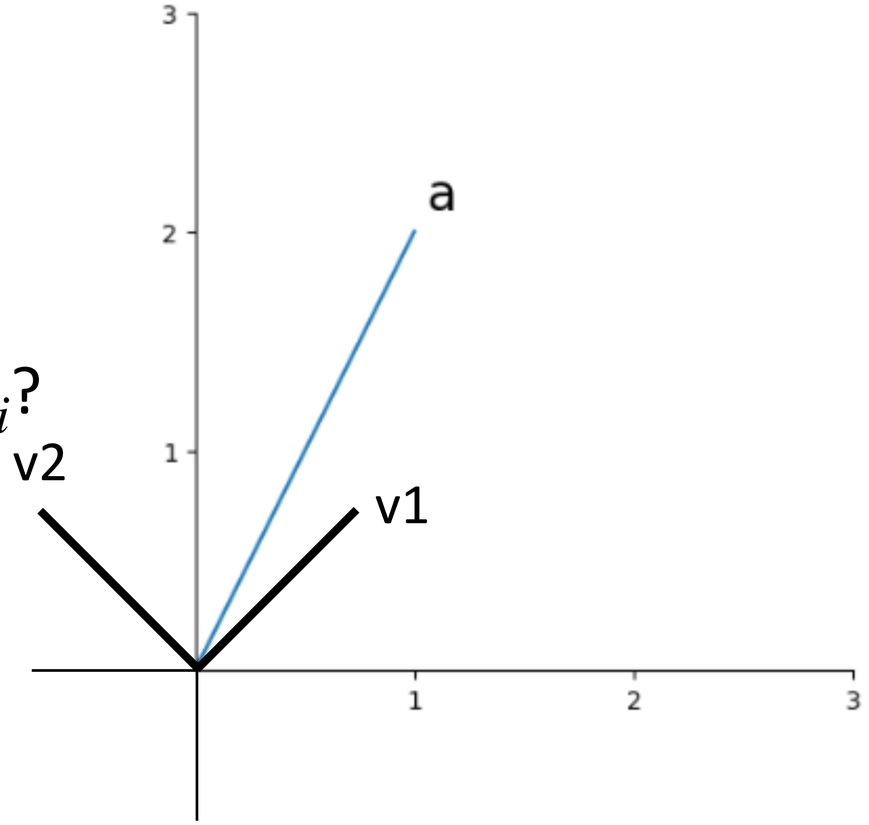
- Coordinates are given with respect to basis vectors

$$a = (a \cdot e_1)e_1 + (a \cdot e_2)e_2 \text{ (Cartesian)}$$

$$a = (a \cdot v_1)v_1 + (a \cdot v_2)v_2$$

- Can we choose v_1 so that $x_i \approx (x_i \cdot v_1)v_1$ for data x_i?





Maximal variance

- We want to choose a vector w so that it's as 'informative' as possible i.e. it maximises the variance of the projections of the data onto w.

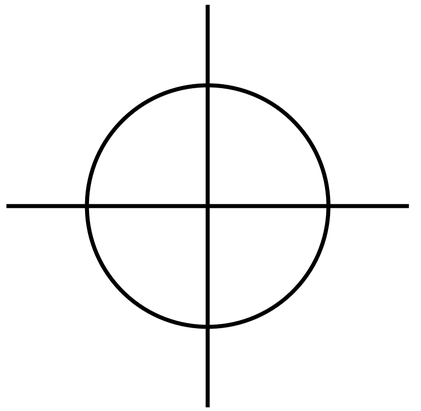
- Suppose $\alpha_i = w \cdot x_i = w^T x_i$ is the projection for observation x_i, where w has unit length. We want to maximise the variance of $\alpha = (\alpha_1, \ldots, \alpha_n)$.

$$\bar{\alpha} = \frac{1}{n}\sum_i \alpha_i = \frac{1}{n}\sum_i w^T x_i = w^T\left(\frac{1}{n}\sum_i x_i\right) = w^T\bar{x}$$

$$\text{var}(\alpha) = \frac{1}{n-1}\sum_i (\alpha_i - \bar{\alpha})^2 = \frac{1}{n-1}\sum_i (w^T x_i - w^T\bar{x})^2 = \frac{1}{n-1}\sum_i w^T(x_i - \bar{x})(x_i - \bar{x})^T w = w^T Q w$$

Maximal variance

- $\text{var}(\alpha) = w^T Q w$ is a **quadratic form**. This is maximised (for unit-length vector w) when $w = e_1$, which is the first **principal component**.



- Proof (Lagrangian method):

  Maximise $L(w, \gamma) = w^T Q w - \gamma(w^T w - 1)$

  $$\frac{\partial L}{\partial w} = 0 = 2Qw - 2\gamma w$$

  Optimal w satisfies $Qw = \gamma w$, i.e. w is an eigenvector with eigenvalue γ.

  Then $L(w, \gamma) = w^T(\gamma w) - \gamma(w^T w - 1) = \gamma$.

  This is maximised when $\gamma = \lambda_1$ (largest eigenvalue of Q).

Questions?

Format of the data

$$X = \begin{pmatrix} X_1 & X_2 & \cdots & X_p \end{pmatrix} = \begin{pmatrix} x_1^T \\ \vdots \\ x_n^T \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}$$

- PCA depends on scaling of each variable
- If we mean-centre each variable, it helps simplify the computation
- Standardise each variable (i.e. subtract mean, divide by variance) so that $\sum_{i=1}^{n} x_{ij} = 0$ and $\sum_{i=1}^{n} x_{ij}^2 = 1$ for all $j = 1, \ldots, p$.

Second principal component

- Greatest variance is from the first principal component (PC), second greatest variance from the second PC (orthogonal to the first) and so on

$$e_1 = \arg\max_{w^T w=1} \left(w^T X^T X w \right)$$

- Subtract the first PC from X and find the weight vector that maximises the variance from this new matrix $\hat{X}_1$:

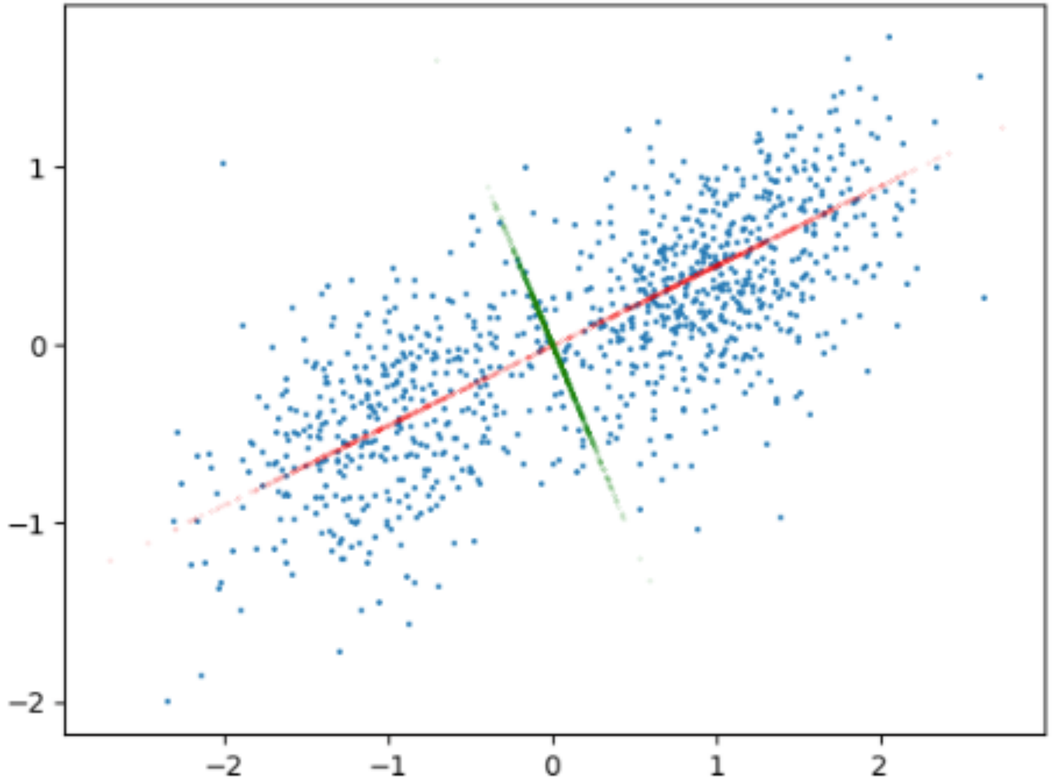
$$\hat{X}_1 = X - X e_1 e_1^T, \quad e_2 = \arg\max_{w^T w=1} \left(w^T \hat{X}_1^T \hat{X}_1 w \right), \quad \hat{X}_2 = X - X e_1 e_1^T - X e_2 e_2^T$$

- It turns out the principal components are the eigenvectors of $X^T X$

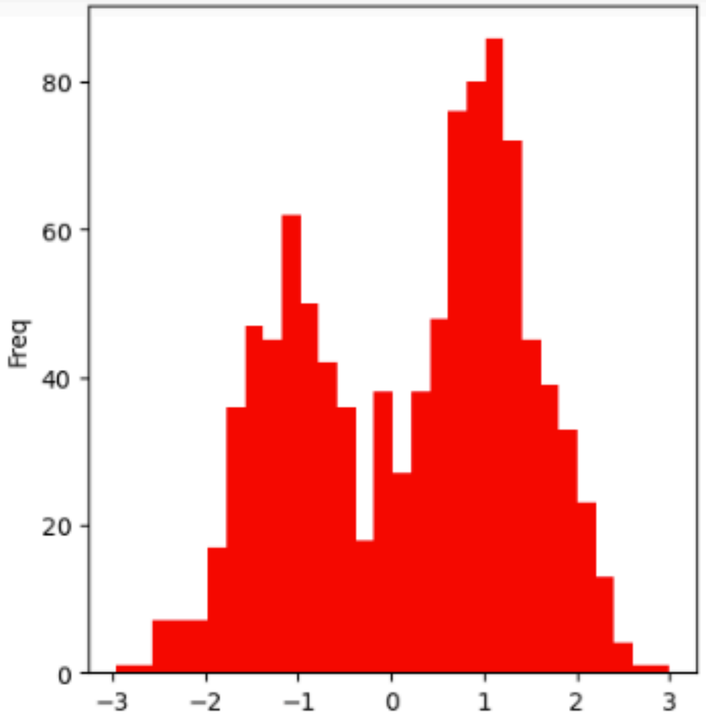
Empirical covariance and eigenvectors

- Q is the empirical covariance matrix of data X: $Q = X^T X/(n-1)$
- Real, symmetric matrix of size $p \times p$
- Therefore it has p orthonormal eigenvectors and eigenvalues, satisfying $Qe_i = \lambda_i e_i$.
- Orthonormal means orthogonal plus unit length, i.e. $e_i^T e_i = 1$, $e_i^T e_j = 0$ (if $i \neq j$)
- As Q is positive semi-definite, all eigenvalues are non-negative
- Order the eigenvalues by size $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p$
- Then the **principal components** are the eigenvectors $e_1, e_2, \ldots, e_p$

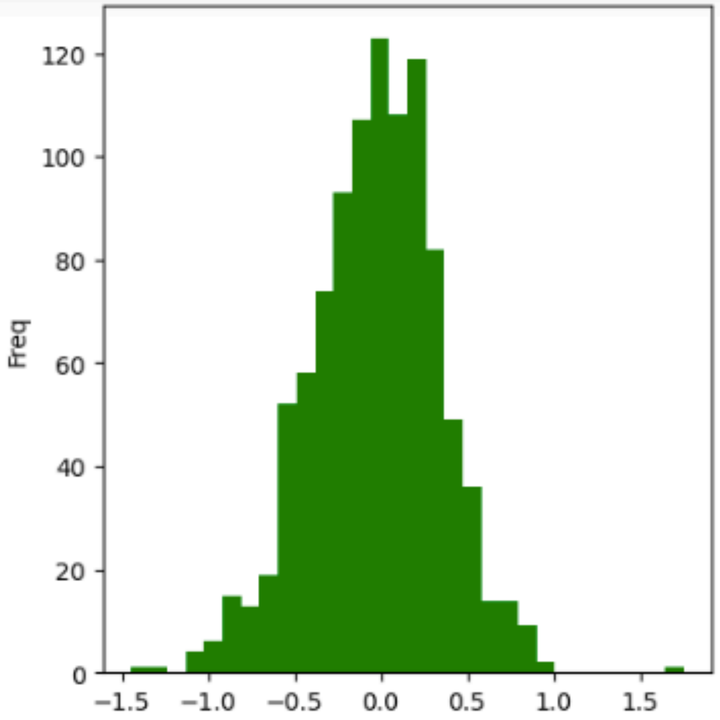
Principal component analysis



Variance = 1.449



Variance = 0.134



Projection: dot product

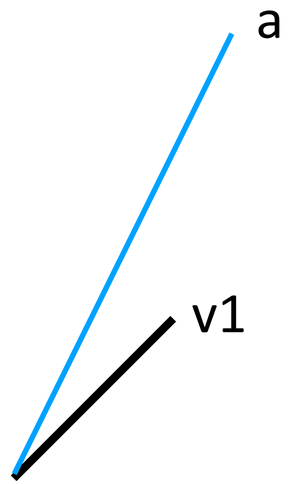
- If we project a onto v_1, what is the distance λ_1?
- Dot product:

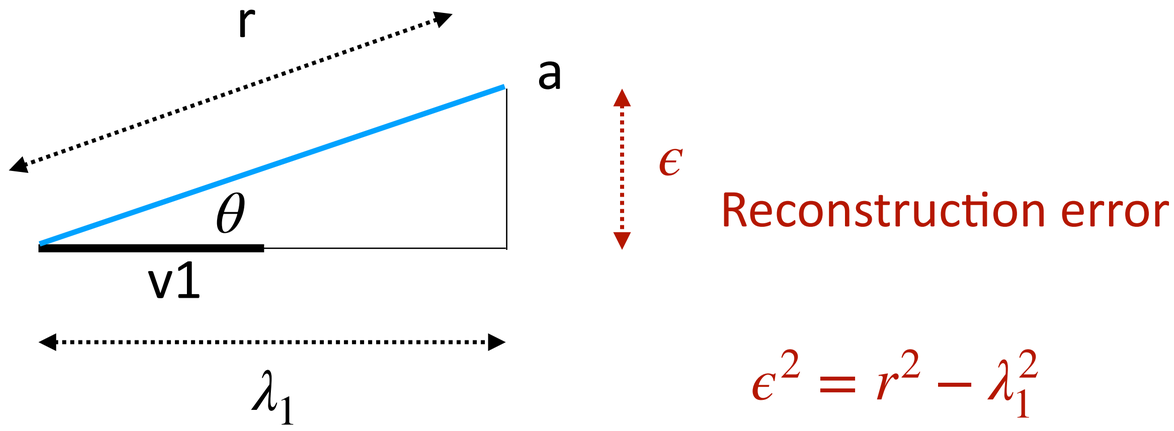
$$x = r\cos(\theta) = |a|\cos(\theta)$$

$$a \cdot v_1 = |a||v_1|\cos(\theta)$$

$$\lambda_1 = \frac{a \cdot v_1}{|v_1|}$$

- Choose v_1 with $|v_1| = 1$





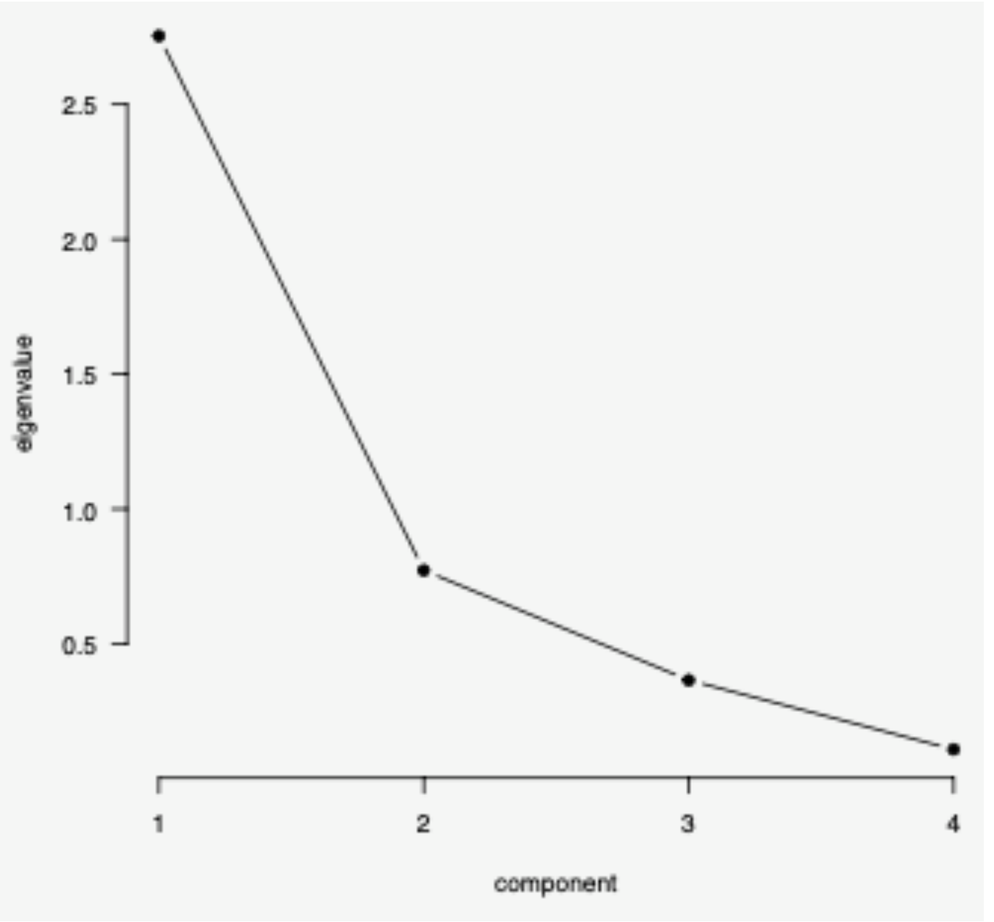
Reconstruction error

$$\epsilon^2 = r^2 - \lambda_1^2$$

Maximising variance is the same as minimising the reconstruction error

Scree plot

- Plot the eigenvalues λ_i against i
- The total variance V of the data X is the sum of eigenvalues of the covariance matrix
- The i^{th} eigenvalue accounts for λ_i / V of the variance.
- The first $m < p$ principal components account for $\sum_{i=1}^{m} (\lambda_i / V)$ of the total variance
- Choose m to capture enough of the variance



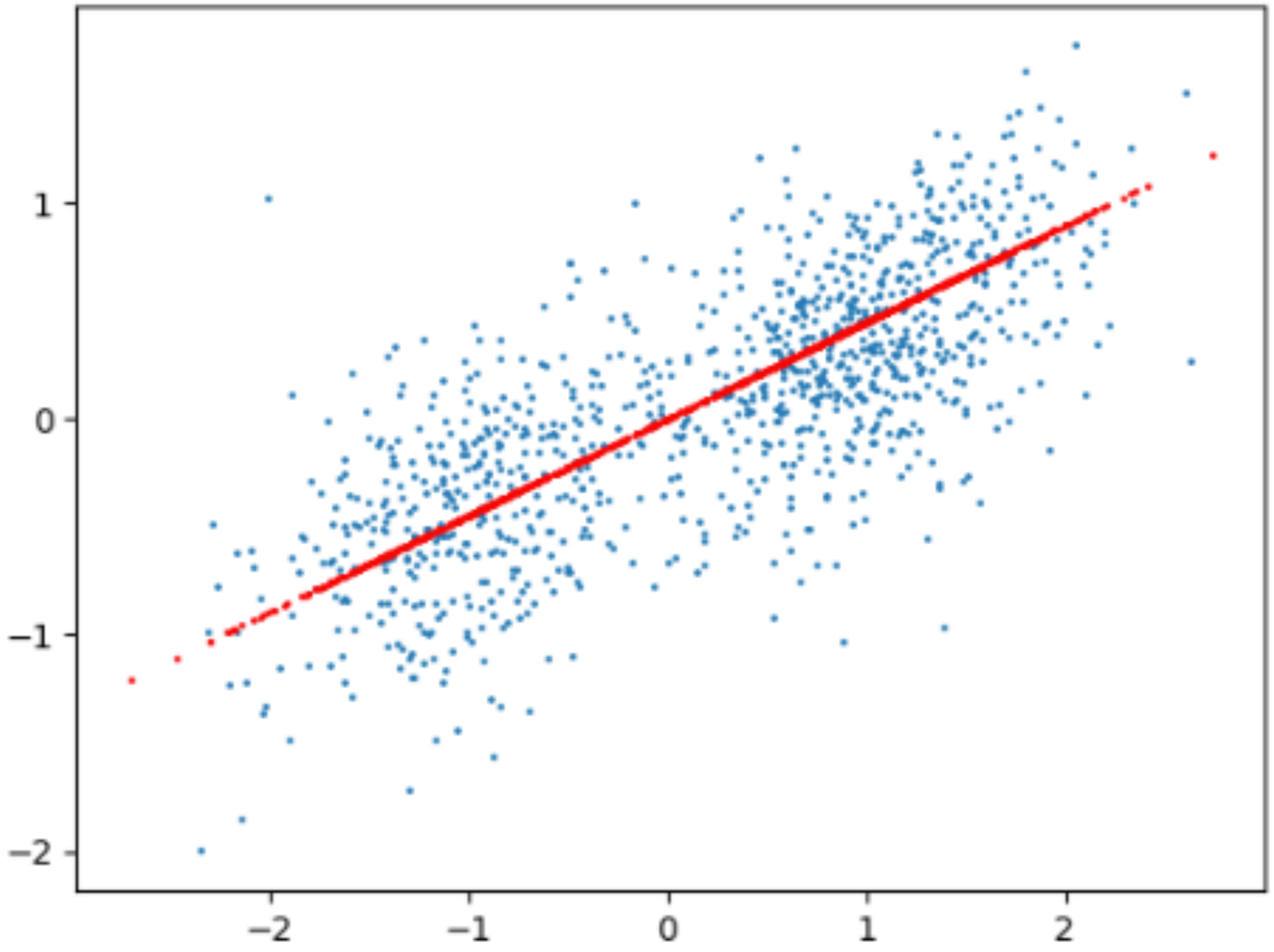
PCA for data compression

- We could think of PCA as projecting the data onto a lower-dimensional subspace (e.g. 2d to 1d)
- If we want to compress the data X, we could calculate the principal components and keep only $m < p$ of them. The representation of X in this space is $Z = XW_m$, where $W_m = (e_1 \quad e_2 \ldots \quad e_m)$.

  $$x_i^T = (x_{i1}, \ldots, x_{ip}), \quad z_i^T = (z_{i1}, \ldots, z_{im}) = \left(\sum_{j=1}^{n} x_{ij} W_{j1}, \ldots, \sum_{j=1}^{n} x_{ij} W_{jm} \right)$$

- The PCA reconstruction is $Y = XW_m W_m^T$. We have discarded information, so this is a lossy transformation! $y_i^T = (y_{i1}, \ldots, y_{ip}) = \left(\sum_{j=1}^{m} z_{ij} W_{1j}, \ldots, \sum_{j=1}^{m} z_{ij} W_{nj} \right)$

PCA for data compression



PCA and information

- Dimensionality reduction usually results in a loss of information
- Using PCA for dimensionality reduction tends to minimise information loss, under certain conditions (i.e. Gaussian noise)
- E.g. if $X = S + N$ (S is the information signal and N is Gaussian noise), then PCA minimises an upper bound on the **information loss**, defined as $I(X, S) - I(Z, S)$, where $Z = XW_m$

Singular value decomposition

- What happens if the number of variables p is very large? (assume X mean-centred)
- The empirical covariance matrix is $p \times p$, which is expensive to calculate

- A non-negative number s is a singular value of X if and only if there are unit-length vectors u (length n) and v (length p) such that $Xv = su$ and $X^T u = sv$.
- $X = U\Lambda V^T$, where U is a $(n \times n)$ orthogonal matrix with $UU^T = I$, V is a $(p \times p)$ orthogonal matrix with $VV^T = I$, and Λ is a $(n \times p)$ matrix with singular values s_i in decreasing order on the diagonal.

Singular value decomposition

- The empirical covariance matrix is
  $Q = X^T X/(n-1) = VSU^T US^T V^T/(n-1) = VSS^T V^T/(n-1)$
- This means the columns of V are the eigenvectors and the eigenvalues are $\lambda_i = s_i^2/(n-1)$.
- Truncated SVD calculates $X \approx U_m \Lambda_m V_m^T$, where columns of U_m and rows of V_m and diagonal terms of Λ_m correspond to the largest m singular values.
- The SVD matrices V and S are easier to compute than Q, when p is large.

Power iteration and QR algorithm

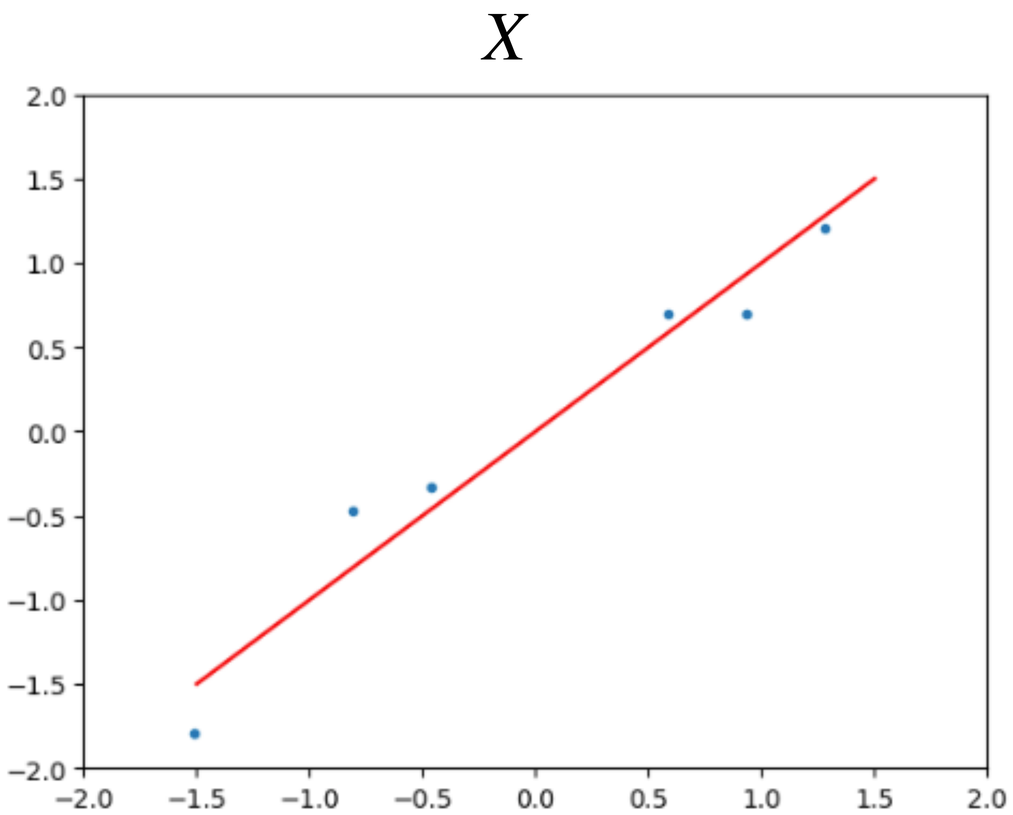
- Power iteration is an eigenvalue algorithm to estimate the largest eigenvalue λ_1 of A, i.e. $Aw_1 = \lambda_1 w_1$
- Start with a vector b_0 and update by the recurrence relation $b_{k+1} = \frac{Ab_k}{\|Ab_k\|}$.

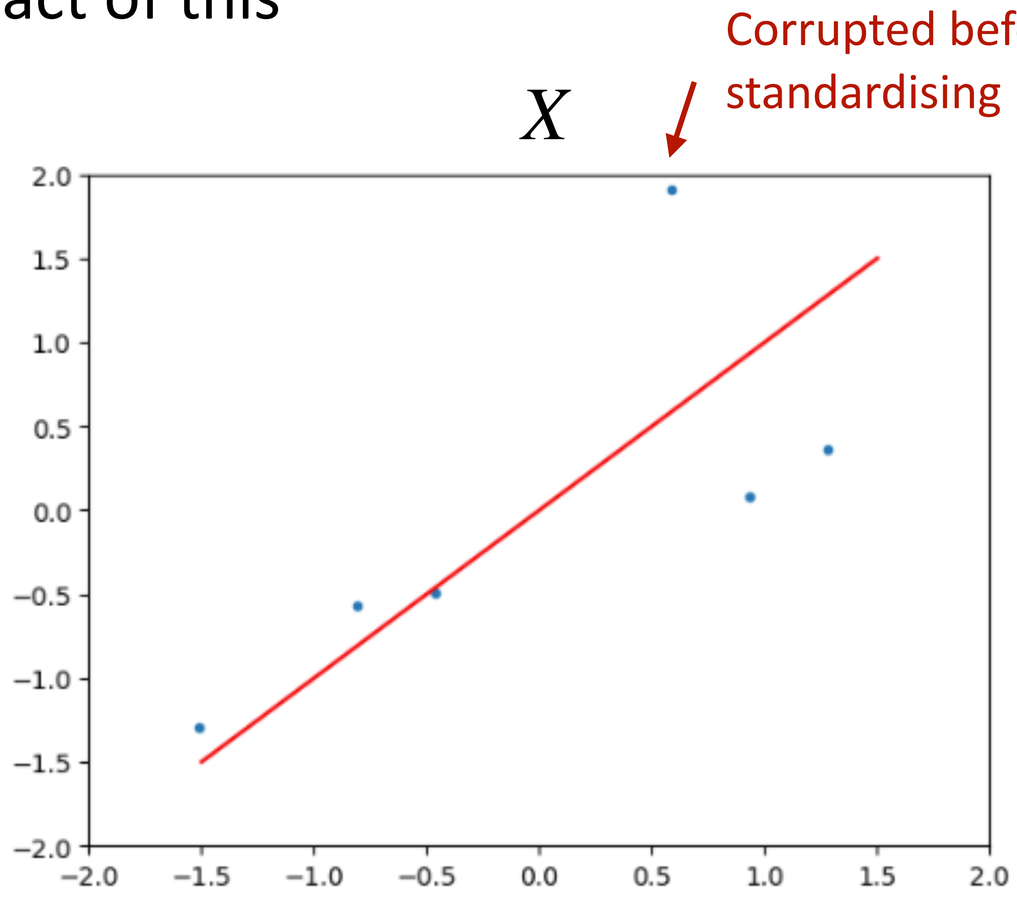
  Then b_k converges to λ_1 as $k \to \infty$ (assuming $b_0 \cdot w_1 \neq 0$)

- More sophisticated version: use the QR-decomposition to form a sequence of matrices $A_0, A_1, \ldots, A_k, \ldots$ (with $A = A_0$). This converges to a triangular matrix that has the eigenvalues of A on the diagonal.

Outliers

- PCA can be sensitive to outliers
- Robust PCA are methods to reduce the impact of this





Questions?

- Feel free to email me at te269@cam.ac.uk

Next time

- Non-linear dimensionality reduction
- Multidimensional scaling
- t-SNE
- UMAP
- Laplacian eigenmaps