# 450c -Principal Components Analysis

Apoorva Lal, Zuhad Hai

April 15, 2021

Stanford

Intuition

Mechanics and Implementation

# Questions?

# Intuition

# Scatterplot Matrices can get unwieldy


Source:Claus Wilke's data-viz course

# Big Picture: Unsupervised Learning

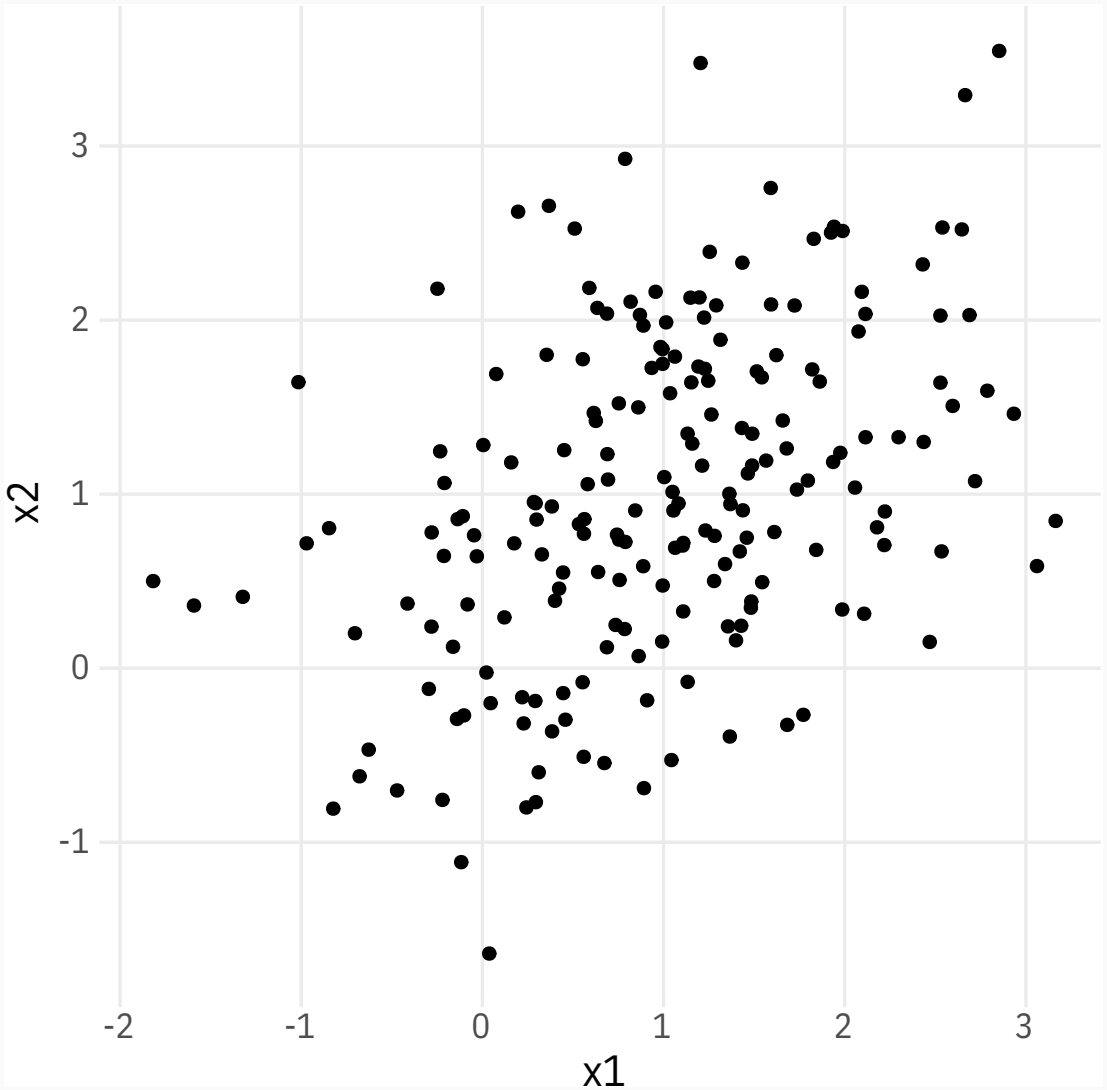
- Difference between supervised and unsupervised learning
- **supervised**: predict $y$ using $\mathbf{x}$
    - regression
    - random forests
    - LASSO
    - support vector machines
    - neural networks
- **unsupervised**: characterise $\mathbf{X}$
    - no response/label $y$; only have a big data matrix $\mathbf{X}$
        - categorise and cluster data (based on substantive knowledge)
        - principal components analysis
        - factor analysis - link
        - $k$-means clustering
        - scaling

# PCA: Intuition

- Introductory **unsupervised** technique
- Dimension reduction technique
- **Examples**
    - How can we order Democratic congressmen from most liberal to most conservative?
    - How can we rank vice-presidential candidates on different dimensions?
    - How can we classify speeches or votes?
- We don't have $y$s to construct a regression model.
- Need to infer *latent* structure

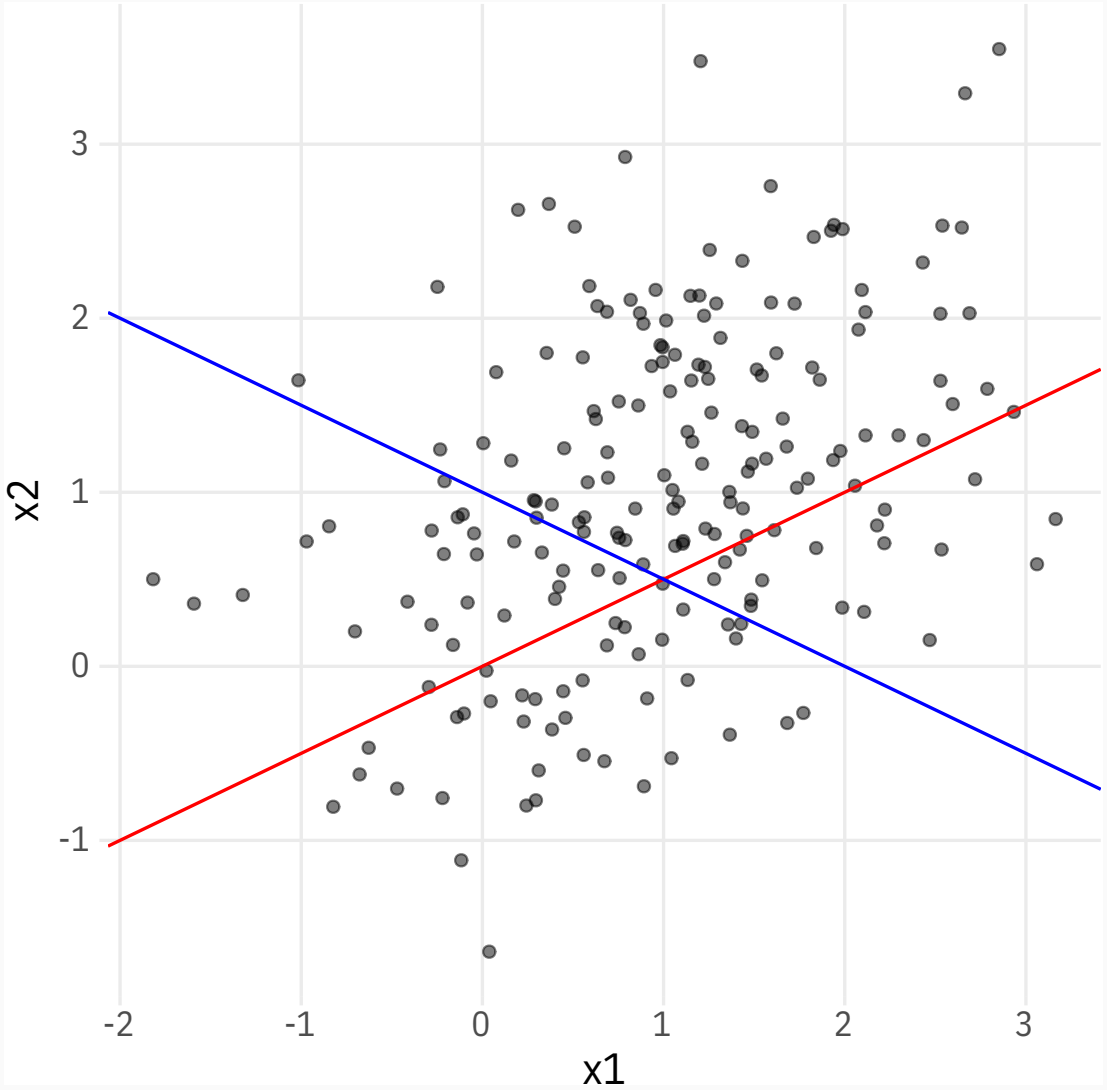
# PCA

- Suppose we have the following two-dimensional data, and want to reduce it to just one dimension:



# PCA (contd)

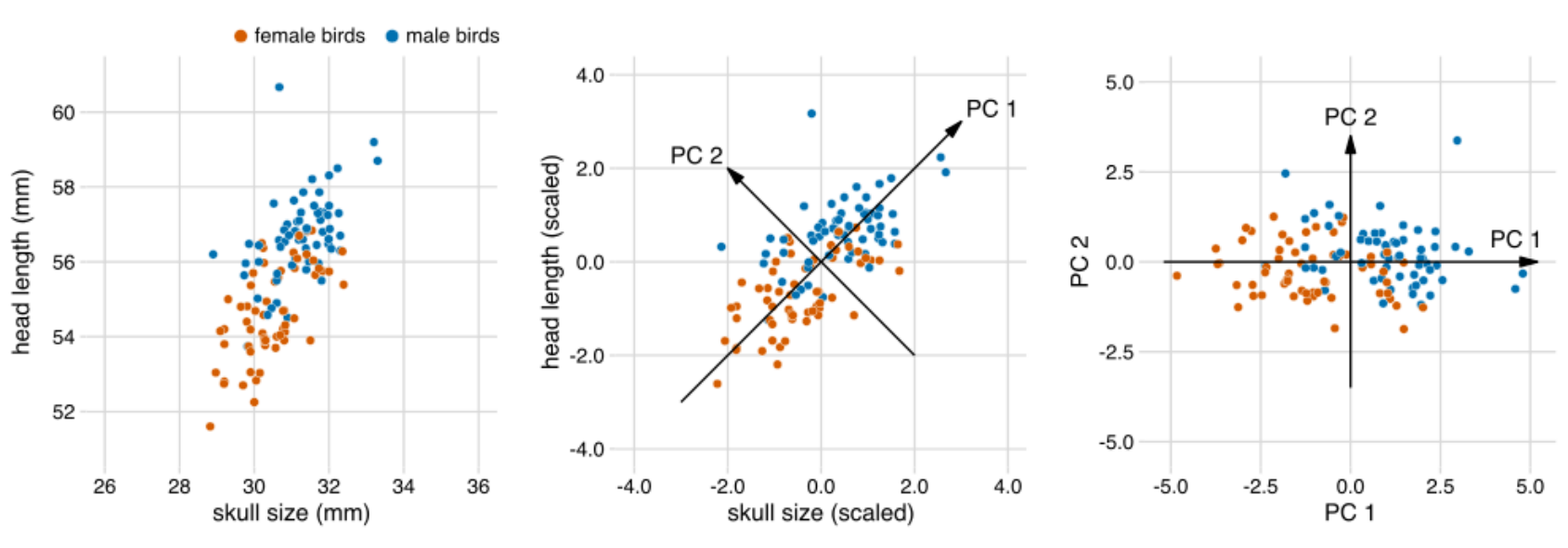
Which of these lines would be a better fit?



# PCA: Intuition (cont'd)

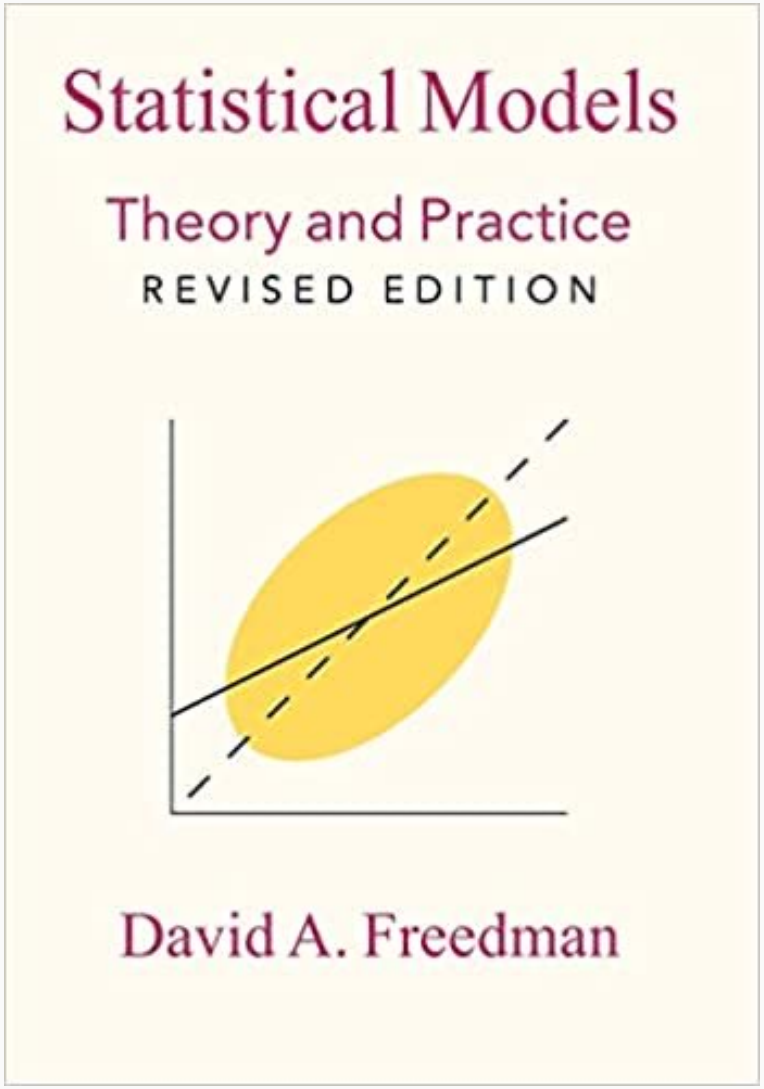
- The core idea behind PCA is to pick the vector through the dimensions along which most the variance in the data is represented
- This amounts to fitting a hyperplane that minimises distance to each point
- That way, we retain as much information as possible
- Conversely, we minimise the reconstruction error – because we maximise the amount of information that we retain.

# Basic Exercise Visualised




**Source: Fundamentals of Data Visualisation**

# Difference between OLS and PCA



# Mechanics and Implementation

# PCA: Mechanics

- Matrix $\mathbf{X}$ with dimensions $n \times p$.
- Objective is to reduce matrix to $K$ dimensions
- PCA dimensions denoted by $\mathbf{w}_k$
- Each data point reconstructed by observation-specific weight (*score*) $z_{ik}$ on dimensions $\mathbf{w}_k$.

$$\tilde{\mathbf{x}}_i = \sum_{k=1}^{K} z_{ik} \mathbf{w}_k$$

- **Objective Function**
  - Pick $\boldsymbol{\theta} := \mathbf{w}_k, z_{ik}$ as to minimise avg. reconstruction error:

$$\min_{\mathbf{w}, z_{ik}} \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{x}_i - \sum_{k=1}^{K} z_{ik} \mathbf{w}_k||^2$$

# PCA: Mechanics (cont'd)

- Taking the FOC and simplifying yields the solution $\mathbf{w}_k^T \boldsymbol{\Sigma} \mathbf{w}_k$.
- where $\boldsymbol{\Sigma}$ is the empirical covariance matrix $\frac{1}{n}\mathbf{X}^\top \mathbf{X}$
- Minimising reconstruction error $\equiv$ Maximising variance of projected data [Slide 18]

$$\mathbf{w}_k^T \boldsymbol{\Sigma} \mathbf{w}_k = \lambda_k$$

- $\mathbf{w}_k^*$ is equal to the $k$ th eigenvector of $\Sigma$, and $z_{ik}^* = \mathbf{w}_k^T \mathbf{x}_i$.

# PCA: Mechanics (cont'd)

- Taking the FOC and simplifying yields the solution $\mathbf{w}_k^T \boldsymbol{\Sigma} \mathbf{w}_k$.
- where $\boldsymbol{\Sigma}$ is the empirical covariance matrix $\frac{1}{n}\mathbf{X}^\top \mathbf{X}$
- Minimising reconstruction error $\equiv$ Maximising variance of projected data [Slide 18]

$$\mathbf{w}_k^T \boldsymbol{\Sigma} \mathbf{w}_k = \lambda_k$$

- $\mathbf{w}_k^*$ is equal to the $k$ th eigenvector of $\Sigma$, and $z_{ik}^* = \mathbf{w}_k^T \mathbf{x}_i$.
- Remember that $\mathbf{A}\mathbf{w} = \lambda \mathbf{w}$.

# PCA: Mechanics (cont'd)

- Taking the FOC and simplifying yields the solution $\mathbf{w}_k^T \boldsymbol{\Sigma} \mathbf{w}_k$.
- where $\boldsymbol{\Sigma}$ is the empirical covariance matrix $\frac{1}{n}\mathbf{X}^\top \mathbf{X}$
- Minimising reconstruction error $\equiv$ Maximising variance of projected data [Slide 18]

$$\mathbf{w}_k^T \boldsymbol{\Sigma} \mathbf{w}_k = \lambda_k$$

- $\mathbf{w}_k^*$ is equal to the $k$ th eigenvector of $\Sigma$, and $z_{ik}^* = \mathbf{w}_k^T \mathbf{x}_i$.
- Remember that $\mathbf{A}\mathbf{w} = \lambda \mathbf{w}$.
    - The eigenvector $\mathbf{w}$ points out the vector in multidimensional space along which most of the variance-covariance matrix ( $\Sigma$ ) can be captured.

# PCA: Mechanics (cont'd)

- Taking the FOC and simplifying yields the solution $\mathbf{w}_k^T \boldsymbol{\Sigma} \mathbf{w}_k$.
- where $\boldsymbol{\Sigma}$ is the empirical covariance matrix $\frac{1}{n}\mathbf{X}^\top \mathbf{X}$
- Minimising reconstruction error $\equiv$ Maximising variance of projected data [Slide 18]

$$\mathbf{w}_k^T \boldsymbol{\Sigma} \mathbf{w}_k = \lambda_k$$

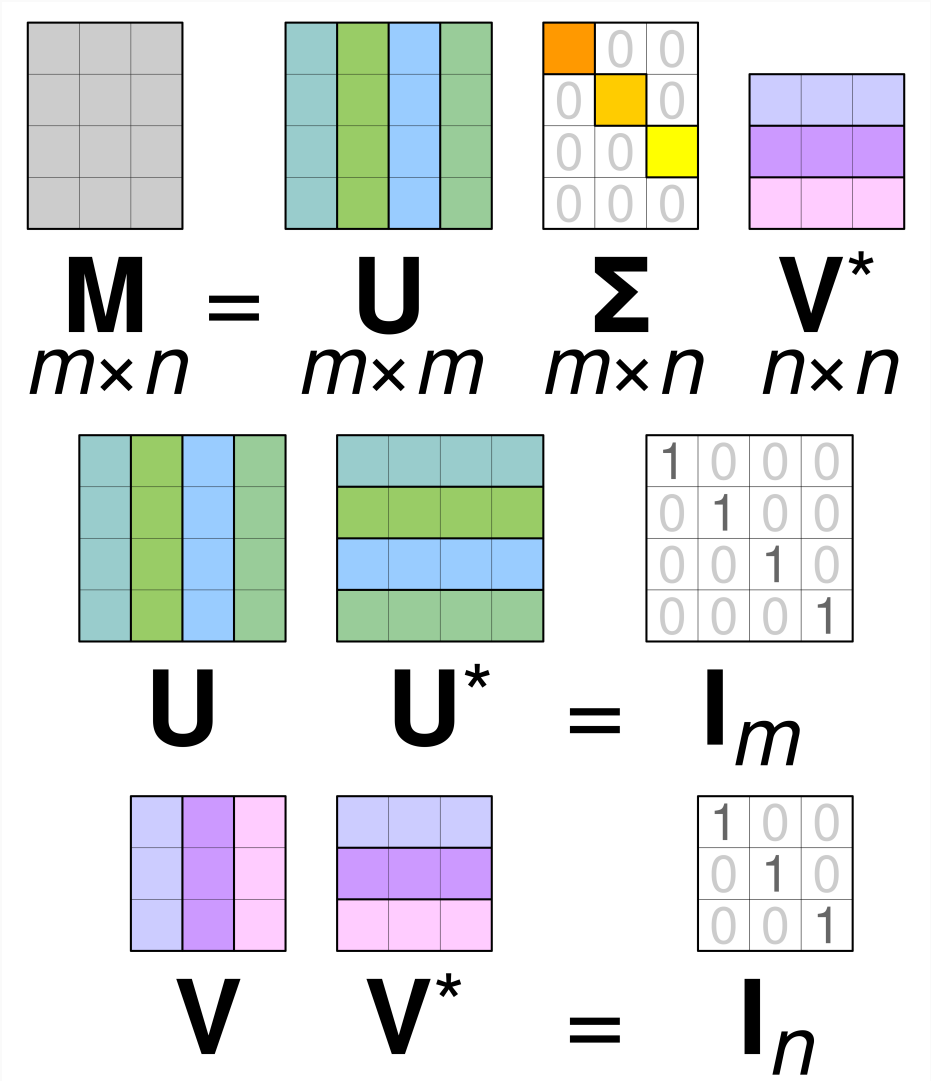
- $\mathbf{w}_k^*$ is equal to the $k$ th eigenvector of $\Sigma$, and $z_{ik}^* = \mathbf{w}_k^T \mathbf{x}_i$.
- Remember that $\mathbf{A}\mathbf{w} = \lambda \mathbf{w}$.
    - The eigenvector $\mathbf{w}$ points out the vector in multidimensional space along which most of the variance-covariance matrix ( $\Sigma$ ) can be captured.
    - we're rotating the coordinate system as to remove the correlation between the covariates.

# Singular Value Decomposition (SVD)

- Every matrix $\mathbf{X}$ can be written as

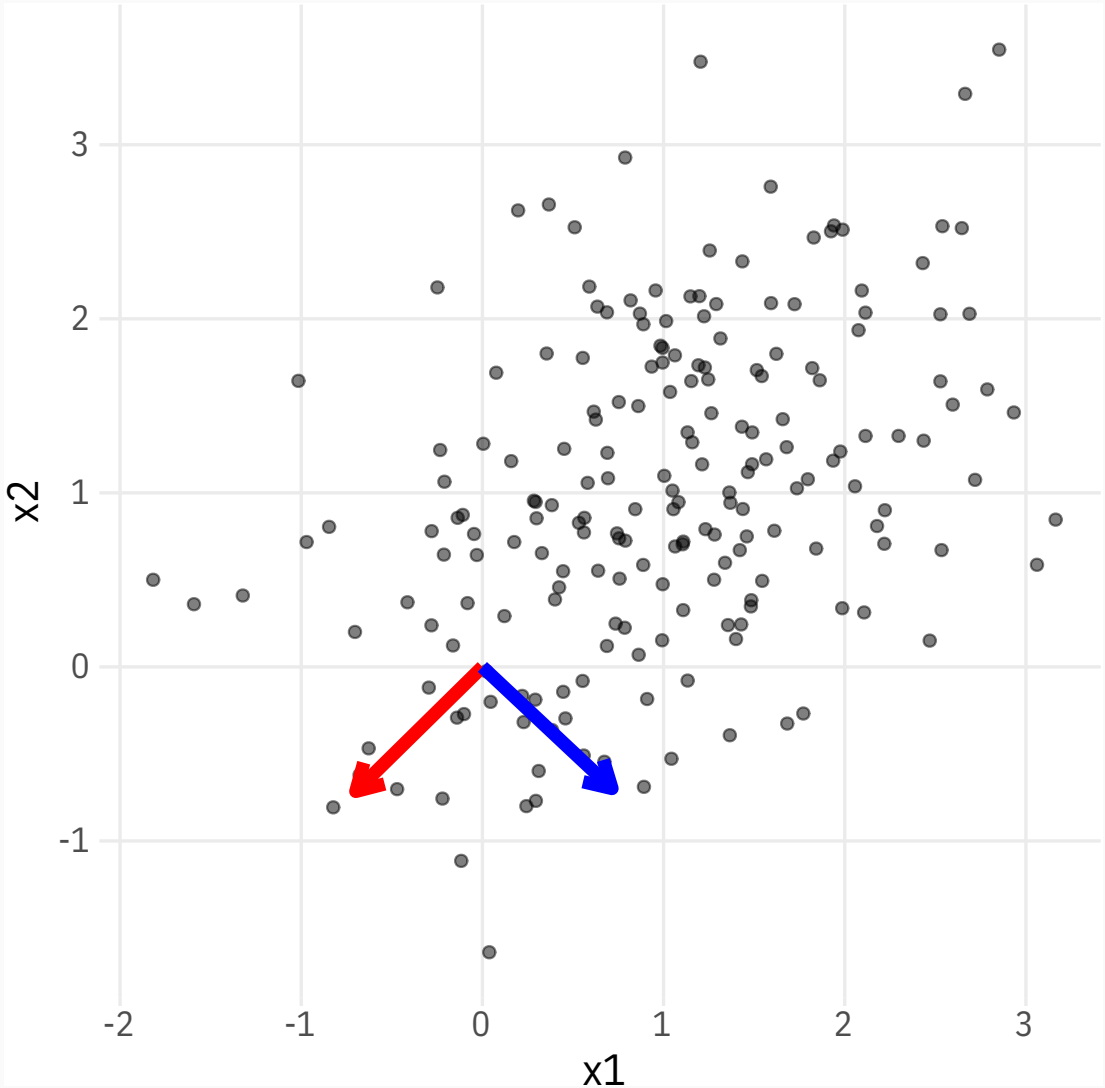
$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$$

  where $\mathbf{U} \in \mathbb{R}^{n \times n}$, $\mathbf{\Sigma} \in \mathbb{R}^{n \times p}$, $\mathbf{V} \in \mathbb{R}^{p \times p}$.
- $\mathbf{U}^\top\mathbf{U} = \mathbf{I}_n$ are left singular vectors
- $\mathbf{V}^\top\mathbf{V} = \mathbf{I}_p$ are right singular vectors
- $\mathbf{\Sigma} = \text{diag}(\sigma_1, \ldots, \sigma_{\min\{n,p\}})$, where $\sigma_i > \sigma_{i+1}$
- Note: computers often store only $\min\{n, p\}$ dimensions
- *Principal components of* $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}$




- case for SVD

# PCA: Viz with simulated data



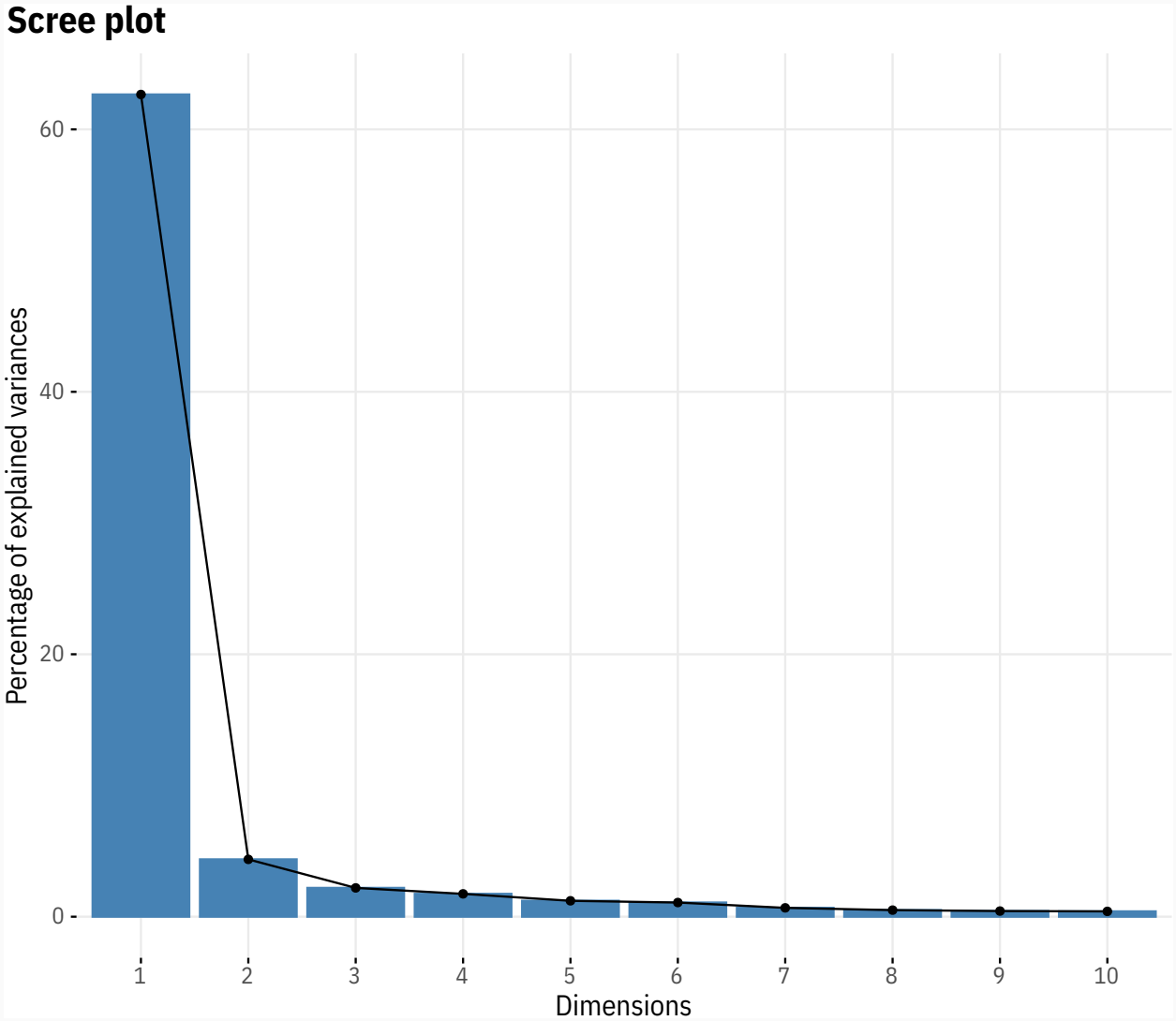
# PCA: Implementation on Voting data

```
load("house_113.RData")
votes <- rc$votes[-1, ] %>% as_tibble() # removes Obama
votes <- votes %>% mutate_all(~ case_when(. %in% 1:3 ~ 1, TRUE ~ 0))
house <- rc$legis.data[-1,] %>% as_tibble()
pca <- prcomp(votes)
pca %>% glimpse
```

```
## List of 5
##  $ sdev    : num [1:444] 12.52 3.31 2.34 2.09 1.74 ...
##  $ rotation: num [1:1202, 1:444] -0.0367 -0.0374 -0.0379 0.0371 -0.038 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:1202] "Vote 1" "Vote 2" "Vote 3" "Vote 4" ...
##   .. ..$ : chr [1:444] "PC1" "PC2" "PC3" "PC4" ...
##  $ center  : Named num [1:1202] 0.495 0.505 0.511 0.437 0.514 ...
##   ..- attr(*, "names")= chr [1:1202] "Vote 1" "Vote 2" "Vote 3" "Vote 4" ...
##  $ scale   : logi FALSE
##  $ x       : num [1:444, 1:444] -0.984 -3.705 -11.935 -12.019 -10.986 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:444] "PC1" "PC2" "PC3" "PC4" ...
##  - attr(*, "class")= chr "prcomp"
```

# Visualising Model Fit

```
fviz_eig(pca, ggtheme = lal_plot_theme())
```



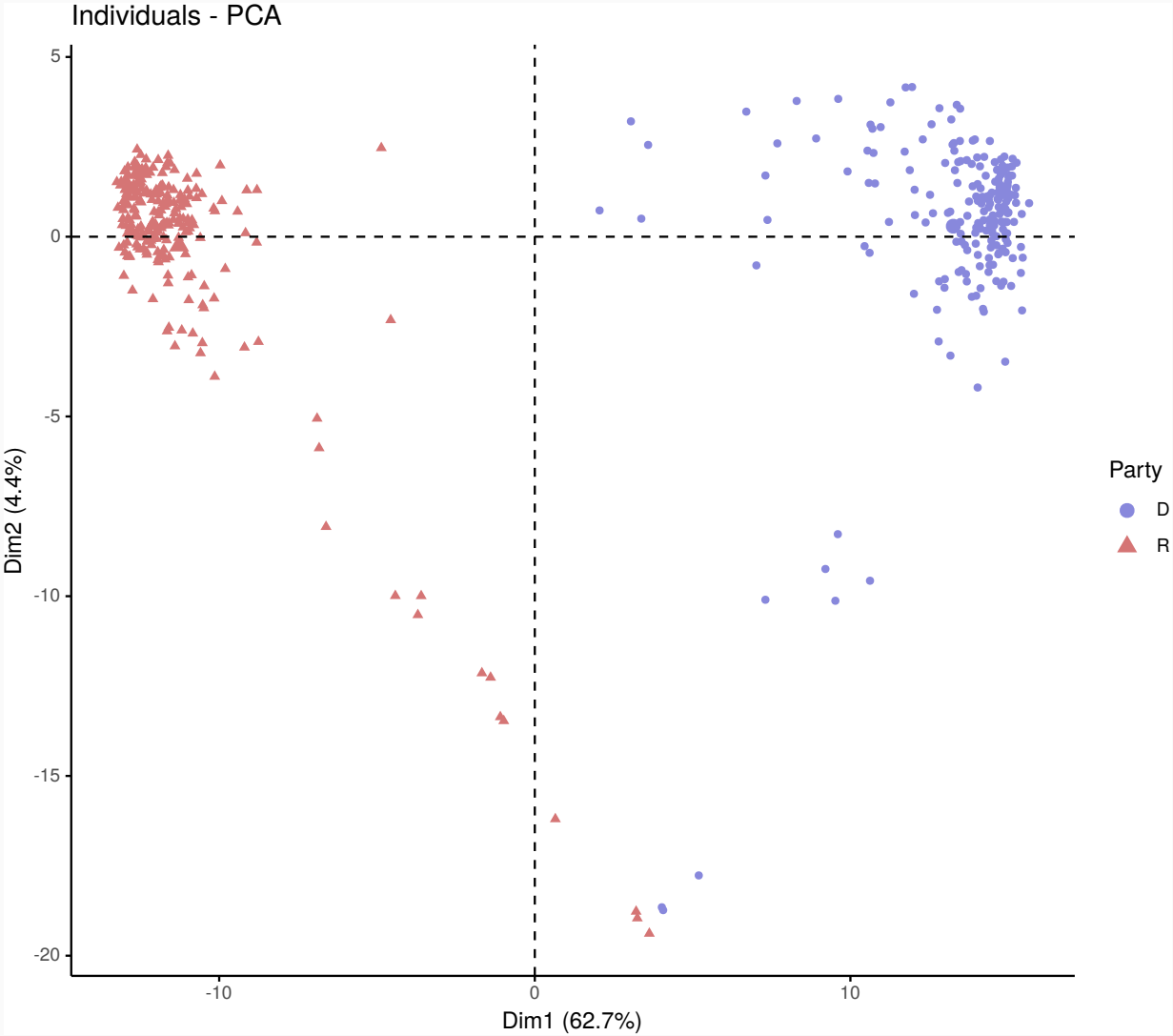
# Generating PCs for units

```
house <- house %>% mutate(PC1 = pca$x[,1])
house %>% arrange(PC1) %>% head(n = 5)
```

| state | icpsrState | cd | icpsrLegis | party | partyCode | PC1 |
|---|---|---|---|---|---|---|
| OH | 24 | 1 | 29550 | R | 200 | -13.24 |
| TX | 49 | 19 | 20353 | R | 200 | -13.21 |
| NC | 47 | 13 | 21349 | R | 200 | -13.17 |
| OH | 24 | 5 | 20755 | R | 200 | -13.13 |
| MO | 34 | 7 | 21150 | R | 200 | -13.10 |

# Latent Dimensions

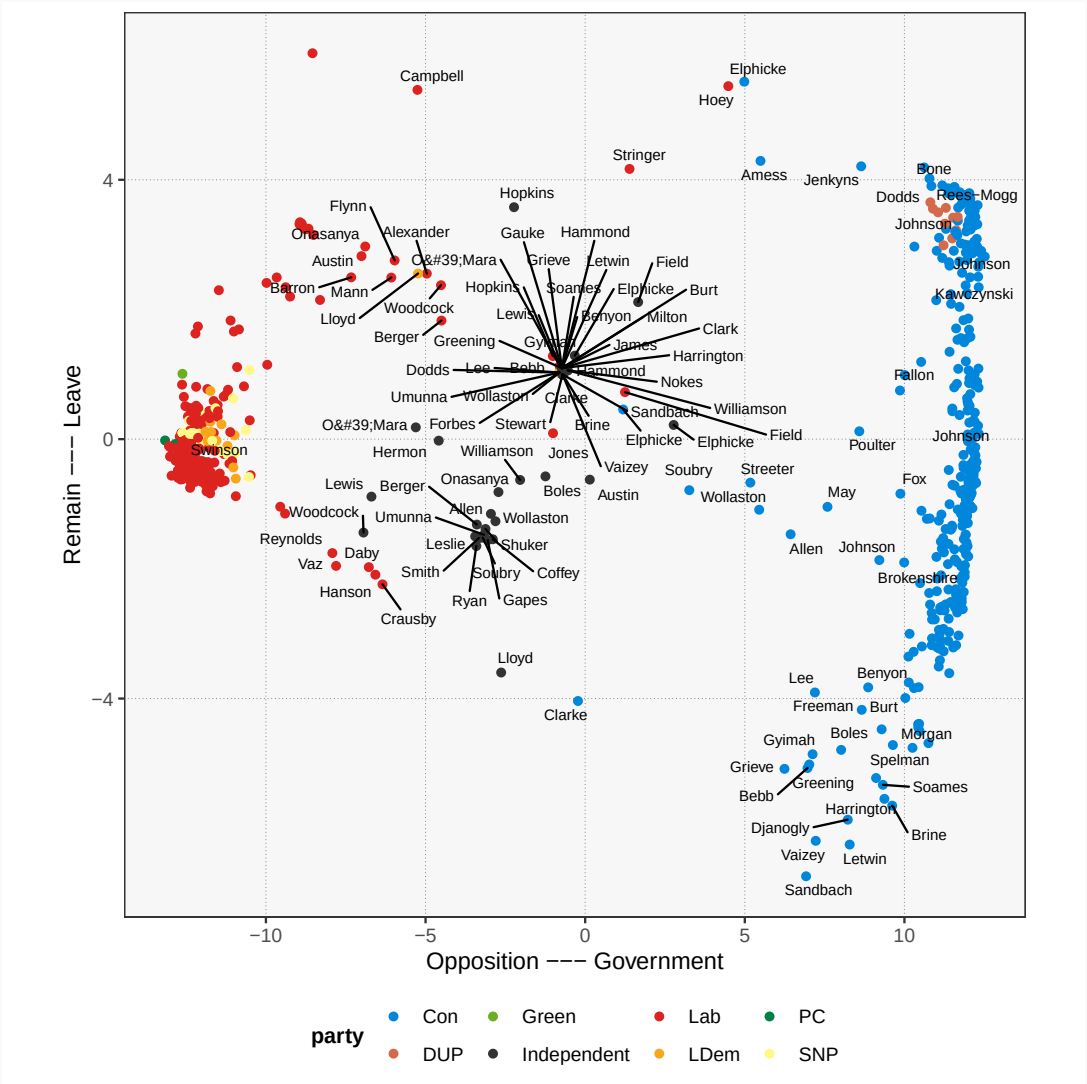
f



# Example 2: Analysing Brexit

- Roll-call example: We know that legislators in parliamentary systems predominantly vote along party lines
- But 2017-2019 UK Parliament was unusual: many, many rebellions with respect to Brexit
- Have a $n \times p$ votes matrix with $n$ MPs and $p$ divisions.
- Code an `Aye` vote as `1`, a `No` vote as `-1`, and an abstention as `0`.
- Use PCA to reduce dimensionality

# PCA: Implementation (cont'd)



# Sparse Matrices

- For many text/ml applications, data is very large and very sparse
    - Netflix problem
- Use irlba in such settings

# Summary

- PCA is a **dimension reduction** technique
- Convenient, but often not ideal:
    - Interpretation of principal components? Typically ad-hoc
    - Information loss
    - No easy way for categorical classification
- Next couple of weeks: more classification methods