

# Machine Learning and Empirical Economics

## *Economics:*

- Focus on one/few coefficients of interest (causal effect).
- Use one main specification (linear), show robustness to alternative specification and placebo tests.
- Model is evaluated on in-sample-properties (e.g.  $R^2$ ).

## *Machine learning:*

- Focus on prediction (and description).
- Use data-driven model selection to identify the most meaningful predictive variables.
- Model is evaluated out-of-sample (e.g. cross validation).

# Machine Learning in Economics

## *Prediction:*

- Measurement
  - Poverty from global images, crop yields, newspaper ideology, scraped price indexes in Argentina, job postings, ...
  - Prediction as input to policy

## *Causation:*

- Automated model-selection approach
  - Select control variables and function.
  - Identify group-fixed effects
- Heterogenous treatment effects

# The Supervised Learning Problem: Language

$Y$ , outcome

- dependent variable = *response, target*

$X$ ,  $p$  predictors

- independent variables = *features, inputs*

*Regression problem*,  $Y$  is continuous (e.g price).

*Classification problem*,  $Y$  takes values in a finite,

Unordered set (survived / died, digit 0-9, cancer class of tissue sample).

*Training data*  $(x_1, y_1), \dots, (x_N, y_N)$ . These are observations (*examples, instances*) of these measurements.

# The Supervised Learning Problem: Objective

On the basis of the training data we would like to:

- Accurately predict unseen test cases.
- Understand which inputs predict the outcome, and why.
- Assess the quality of our predictions and inferences.

# Unsupervised Learning

- No outcome variable, just a set of predictors (features) measured on a set of samples.
- Objective is fuzzier:
  - Find groups of observations that behave similarly, find features that behave similarly,
  - Reduce the dimensionality of the features in  $X$  while keeping most of the variation.
  - Difficult to know how well you are doing.
- Can be useful as a pre-processing step for supervised learning.

# Topics

## Unsupervised.

- Principal components
- Clustering

## Supervised

- Model Assessment and Selection
- Linear model selection and regularization
- Classification and text analysis
- Tree-based methods
- Deep Learning

# Readings

## *An Introduction to Statistical Learning with Applications in R*

- Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani
- Less technically advanced. Each chapter ends with an R lab, in which examples are developed.

## *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*

- More technical and covers a broader range of topics.

Both available free from authors' websites.

Online course:

<https://online.stanford.edu/courses/sohs-ystatslearning-statistical-learning>

# Unsupervised Learning

## Reading

*Introduction to Statistical Learning*: Chapter 12

*Elements of Statistical Learning*: Chapter 14



# Unsupervised Learning

## *Unsupervised vs Supervised Learning:*

- Most of this course focuses on *supervised learning* methods such as regression and classification.
- In that setting we observe both a set of features  $X_1, X_2, \dots, X_p$  for each object, as well as a response or outcome variable  $Y$ . The goal is then to predict  $Y$  using  $X_1, X_2, \dots, X_p$ .
- Here we instead focus on *unsupervised learning*, where we observe only the features  $X_1, X_2, \dots, X_p$ . We are not interested in prediction, because we do not have an associated response variable  $Y$ .

# The Goals of Unsupervised Learning

- The goal is to discover interesting things about the measurements: is there an informative way to visualize the data? Can we discover subgroups among the variables or among the observations?
- We discuss two methods:
  - *principal components analysis*, a tool used for data visualization or data pre-processing before supervised techniques are applied, and
  - *clustering*, a broad class of methods for discovering unknown subgroups in data.

# Principal Components Analysis

- PCA produces a low-dimensional representation of a dataset. It finds a sequence of linear combinations of the variables that have maximal variance, and are mutually uncorrelated.
- Produces derived variables for use in supervised learning  
Example: face recognition.
- Serves as a tool for data visualization  
Example: World Values Survey.

# Principal Components Analysis: details

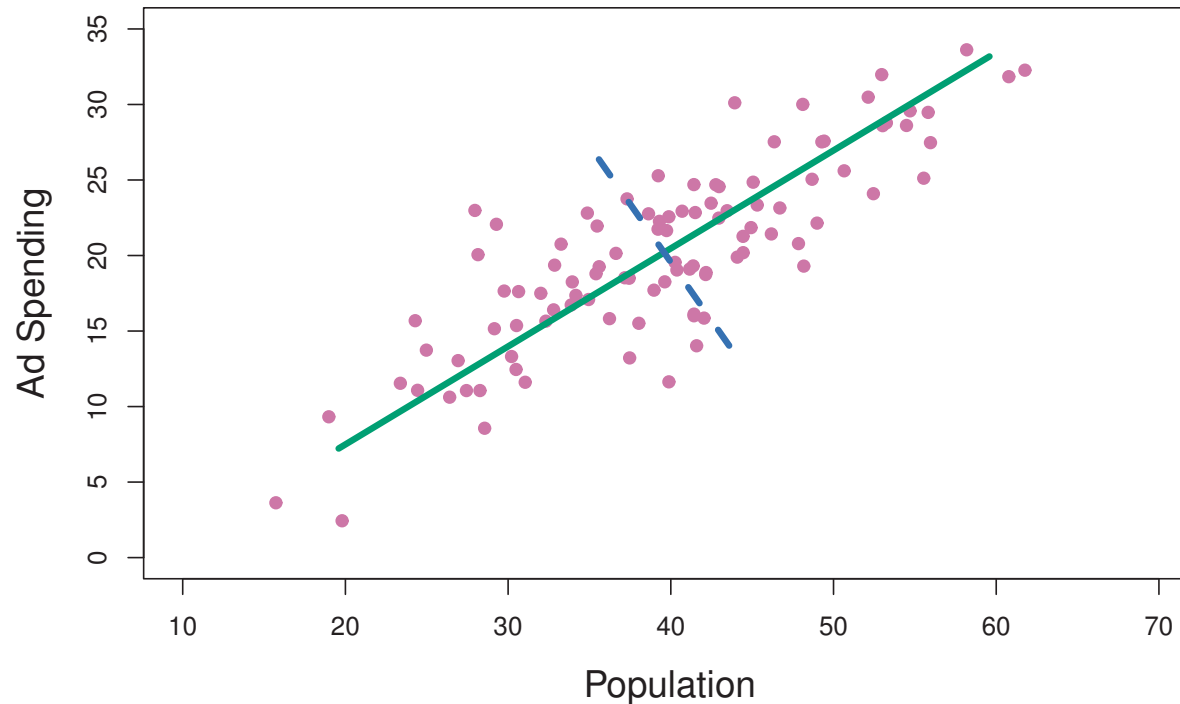
- The *first principal component* of a set of features  $X_1, X_2, \dots, X_p$  is the normalized linear combination of the features

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \dots + \phi_{p1}X_p$$

that has the largest variance. By *normalized*, we mean that  $\sum_{j=1}^p \phi_{j1}^2 = 1$ .

- We refer to the elements  $\phi_{11}, \dots, \phi_{p1}$  as the loadings of the first principal component; together, the loadings make up the principal component loading vector,  $\phi_1 = (\phi_{11} \ \phi_{21} \ \dots \ \phi_{p1})^T$ .
- We constrain the loadings so that their sum of squares is equal to one, since otherwise setting these elements to be arbitrarily large in absolute value could result in an arbitrarily large variance.

# PCA: example



The population size (**pop**) and ad spending (**ad**) for 100 different cities are shown as purple circles. The green solid line indicates the first principal component direction, and the blue dashed line indicates the second principal component direction.

# Computation of Principal Components

- Suppose we have a  $n \times p$  data set  $\mathbf{X}$ . Since we are only interested in variance, we assume that each of the variables in  $\mathbf{X}$  has been centered to have mean zero (that is, the column means of  $\mathbf{X}$  are zero).
- We then look for the linear combination of the sample feature values of the form

$$z_{i1} = \phi_{11}x_{i1} + \phi_{21}x_{i2} + \dots + \phi_{p1}x_{ip} \quad (1)$$

for  $i = 1, \dots, n$  that has largest sample variance, subject to the constraint that  $\sum_{j=1}^p \phi_{j1}^2 = 1$ .

- Since each of the  $x_{ij}$  has mean zero, then so does  $z_{i1}$  (for any values of  $\phi_{j1}$ ). Hence the sample variance of the  $z_{i1}$  can be written as  $\frac{1}{n} \sum_{i=1}^n z_{i1}^2$ .

## Computation: continued

- Plugging in (1) the first principal component loading vector solves the optimization problem

$$\underset{\phi_{11}, \dots, \phi_{p1}}{\text{maximize}} \frac{1}{n} \sum_{i=1}^n \left( \sum_{j=1}^p \phi_{j1} x_{ij} \right)^2 \quad \text{subject to} \quad \sum_{j=1}^p \phi_{j1}^2 = 1.$$

- This problem can be solved via a singular-value decomposition of the matrix  $\mathbf{X}$ , a standard technique in linear algebra.
- We refer to  $Z_1$  as the first principal component, with realized values  $z_{11}, \dots, z_{n1}$

# 1 Two interpretations of PCA: maximizing variance and minimizing residual.

Let  $X$  be a  $n \times p$  matrix containing  $n$  observations  $i$ , of  $p$  variables. Let  $x_i$  be a  $p \times 1$  vector with one such observation. These variables have been centered, so that

$$\frac{1}{n} \sum_{i=1}^n x_i = 0.$$

Hence,

$$\frac{1}{n} X'X = V$$

is the in-sample variance matrix.

Let  $\phi$  be a unit  $1 \times p$  vector (of pca "loadings").  $z_i = (x_i' \phi)$  is the length of the projection of  $x_i$  on  $\phi$ . This projection has mean zero since

$$\frac{1}{n} \sum_{i=1}^n (x_i' \phi) \phi = \left( \left( \frac{1}{n} \sum_{i=1}^n x_i \right)' \phi \right) \phi.$$

The squared residual distance from  $x_i$  to this projection is

$$\begin{aligned} \|x_i - (x_i' \phi) \phi\|^2 &= x_i' x_i - 2 (x_i' \phi)^2 + (x_i' \phi)^2 \phi' \phi \\ &= x_i' x_i - (x_i' \phi)^2 \end{aligned}$$

since  $\phi' \phi = 1$ . Summing squared residuals across observations,

$$\frac{1}{n} \sum_{i=1}^n \|x_i - (x_i' \phi) \phi\|^2 = \frac{1}{n} \left( \sum_{i=1}^n x_i' x_i - \sum_{i=1}^n (x_i' \phi)^2 \right).$$

The first part does not depend on  $\phi$ . To minimize the sum of squared residuals we should maximize

$$\frac{1}{n} \sum_{i=1}^n (x_i' \phi)^2 = \frac{1}{n} \sum_{i=1}^n z_i^2 = \text{Var}(z),$$

since  $z_i$  has mean zero. Hence, the linear projection that minimizes the squared residual is that with maximum variance.



## 2 Relationship to eigenvectors and eigenvalues.

$$\begin{aligned}\frac{1}{n} \sum_{i=1}^n (x_i' \phi)^2 &= \frac{1}{n} (X\phi)' (X\phi) \\ &= \phi' \left( \frac{1}{n} X' X \right) \phi \\ &= \phi' V \phi.\end{aligned}$$

We want to maximize  $\phi' V \phi$  subject to  $\phi' \phi = 1$ ,

$$\max L = \phi' V \phi - \lambda (\phi' \phi - 1).$$

$$\frac{\partial L}{\partial \lambda} = \phi' \phi - 1$$

$$\frac{\partial L}{\partial \phi} = 2V\phi - 2\lambda\phi$$

Setting the second derivative to zero yields

$$V\phi = \lambda\phi.$$

Hence,  $\phi$  is an eigenvector to  $V$  with eigenvalue  $\lambda$ . Since  $\phi' \phi = 1$

$$\lambda = \phi' V \phi,$$

the eigenvalue gives the variance of the projection of  $X$  onto the associated eigenvector (principal component)  $\phi$ . Since  $V$  is a symmetric matrix, the eigenvectors (principal components) will be orthogonal.

# Geometry of PCA

- The loading vector  $\phi_1$  with elements  $\phi_{11}, \phi_{21}, \dots, \phi_{p1}$  defines a direction in feature space along which the data vary the most.
- If we project the  $n$  data points  $x_1, \dots, x_n$  onto this direction, the projected values are the principal component scores  $z_{11}, \dots, z_{n1}$  themselves.

## Further principal components

- The second principal component is the linear combination of  $X_1, \dots, X_p$  that has maximal variance among all linear combinations that are *uncorrelated* with  $Z_1$ .
- The second principal component scores  $z_{12}, z_{22}, \dots, z_{n2}$  take the form

$$z_{i2} = \phi_{12}x_{i1} + \phi_{22}x_{i2} + \dots + \phi_{p2}x_{ip},$$

where  $\phi_2$  is the second principal component loading vector, with elements  $\phi_{12}, \phi_{22}, \dots, \phi_{p2}$ .

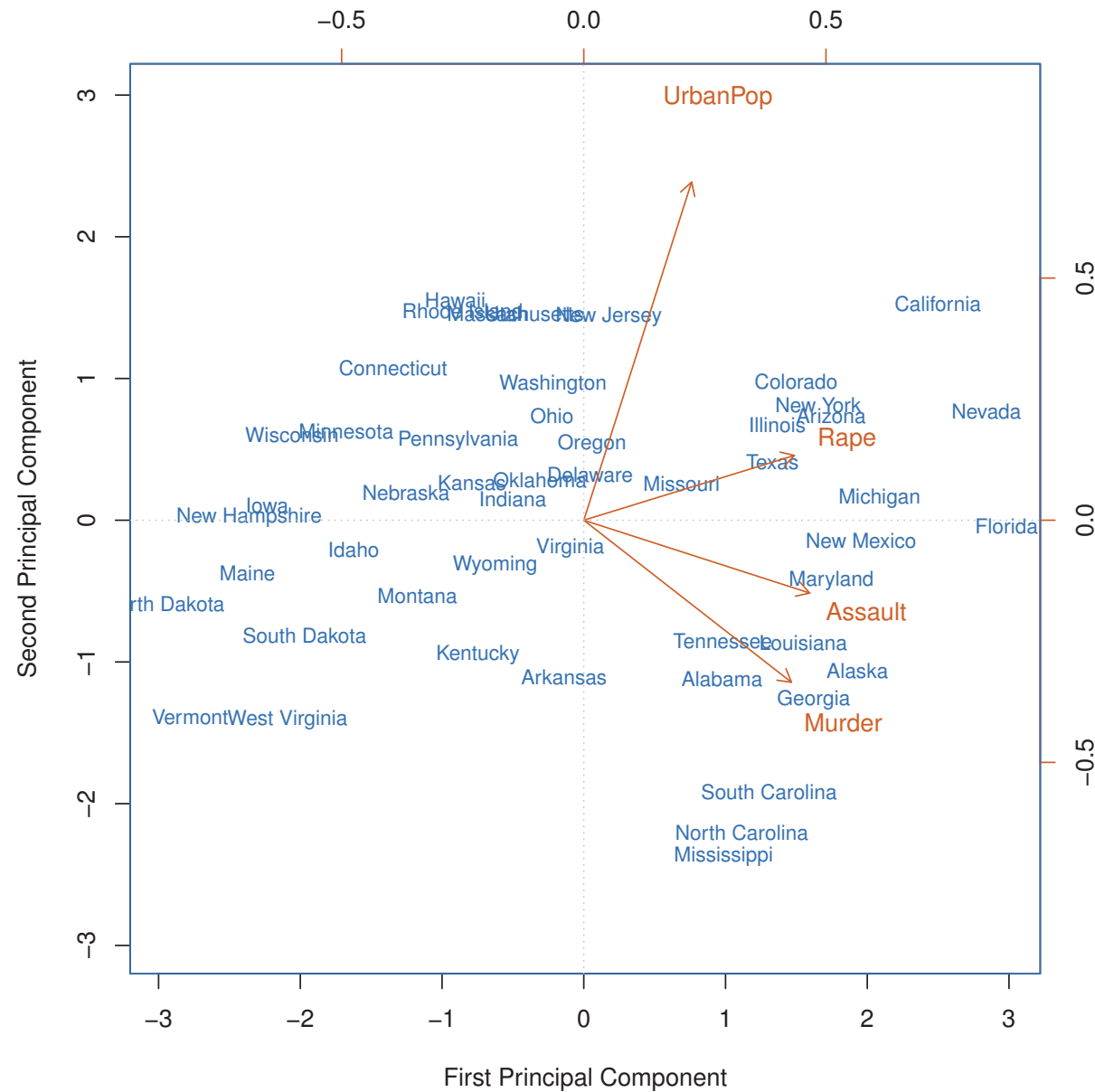
## Further principal components: continued

- It turns out that constraining  $Z_2$  to be uncorrelated with  $Z_1$  is equivalent to constraining the direction  $\phi_2$  to be orthogonal (perpendicular) to the direction  $\phi_1$ . And so on.
- The principal component directions  $\phi_1, \phi_2, \phi_3, \dots$  are the ordered sequence of right singular vectors of the matrix  $\mathbf{X}$ , and the variances of the components are  $\frac{1}{n}$  times the squares of the singular values. There are at most  $\min(n - 1, p)$  principal components.

# Illustration

- **USAarrests** data: For each of the fifty states in the United States, the data set contains the number of arrests per 100,000 residents for each of three crimes: **Assault**, **Murder**, and **Rape**. We also record **UrbanPop** (the percent of the population in each state living in urban areas).
- The principal component score vectors have length  $n = 50$ , and the principal component loading vectors have length  $p = 4$ .
- PCA was performed after standardizing each variable to have mean zero and standard deviation one.

# USAarrests data: PCA plot



## Figure details

The first two principal components for the USArrests data.

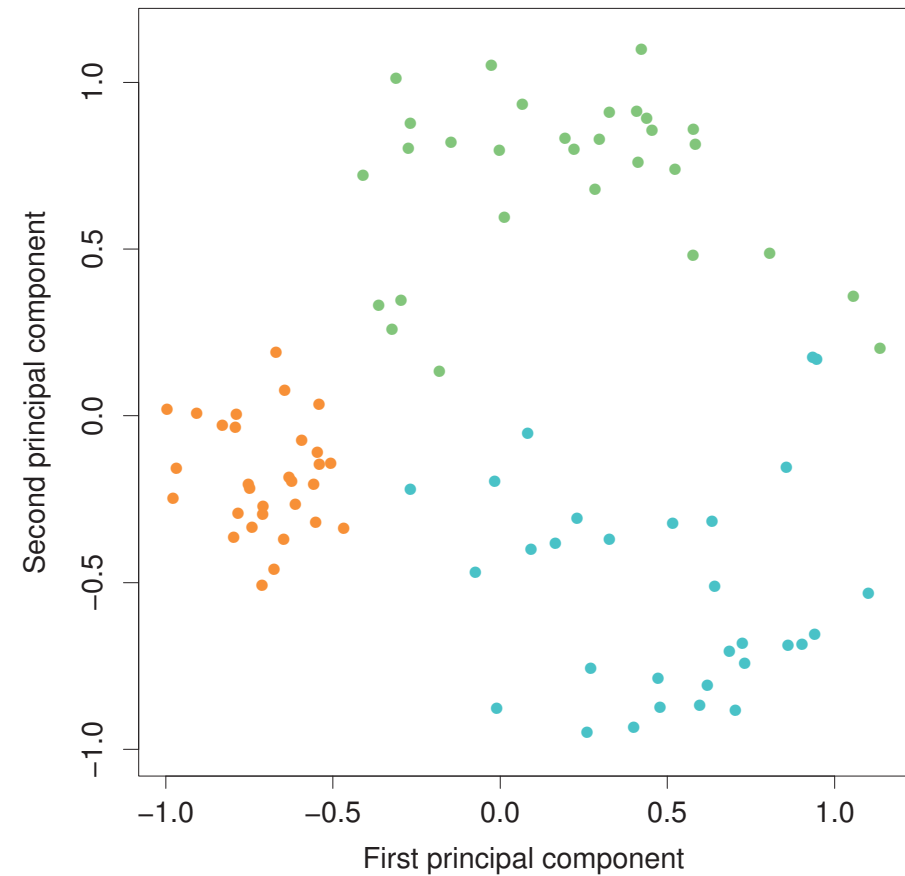
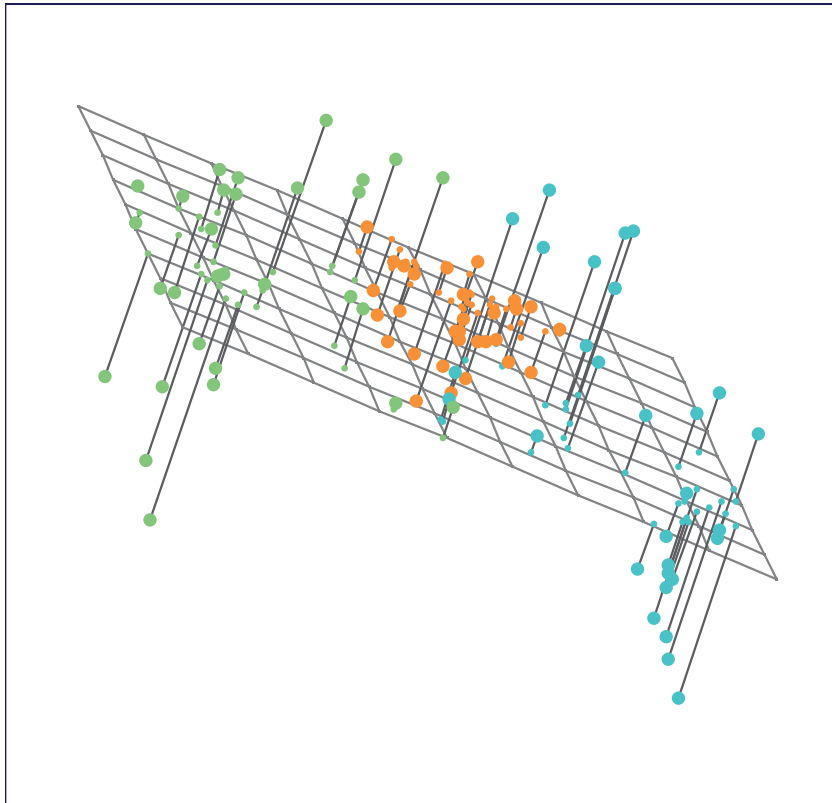
- The blue state names represent the scores for the first two principal components.
- The orange arrows indicate the first two principal component loading vectors (with axes on the top and right). For example, the loading for **Rape** on the first component is 0.54, and its loading on the second principal component 0.17 [the word **Rape** is centered at the point (0.54, 0.17)].
- This figure is known as a *biplot*, because it displays both the principal component scores and the principal component loadings.

## PCA loadings

	PC1	PC2
Murder	0.5358995	-0.4181809
Assault	0.5831836	-0.1879856
UrbanPop	0.2781909	0.8728062
Rape	0.5434321	0.1673186



# Another Interpretation of Principal Components

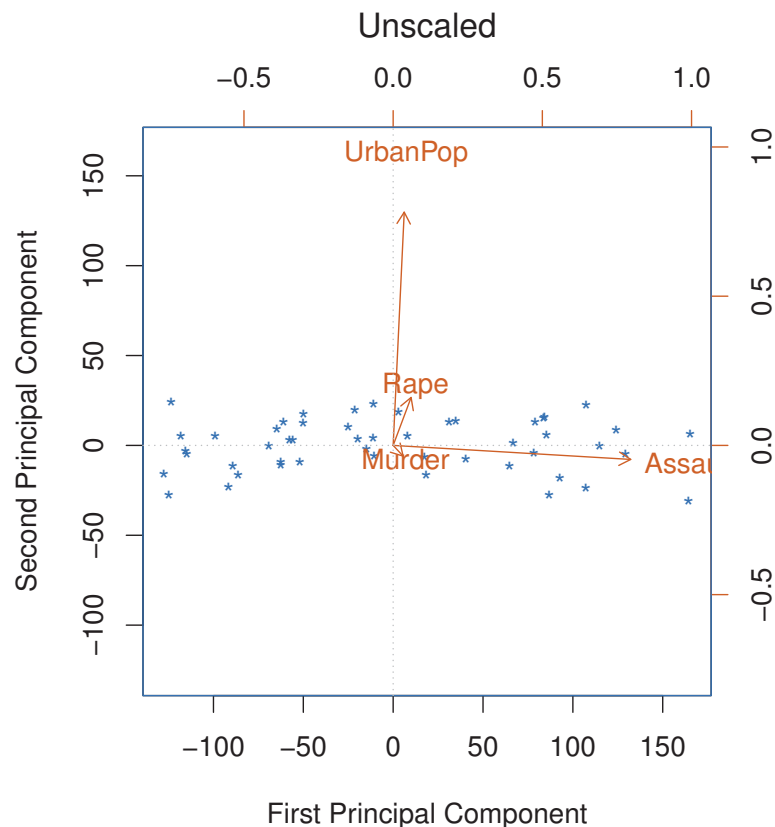
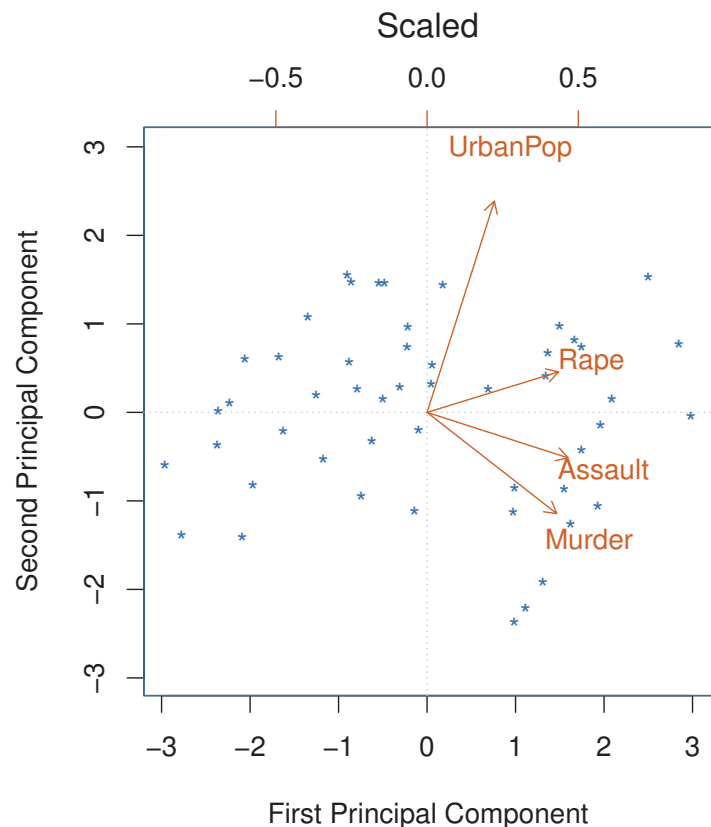


# PCA find the hyperplane closest to the observations

- The first principal component loading vector has a very special property: it defines the line in  $p$ -dimensional space that is *closest* to the  $n$  observations (using average squared Euclidean distance as a measure of closeness)
- The notion of principal components as the dimensions that are closest to the  $n$  observations extends beyond just the first principal component.
- For instance, the first two principal components of a data set span the plane that is closest to the  $n$  observations, in terms of average squared Euclidean distance.

# Scaling of the variables matters

- If the variables are in different units, scaling each to have standard deviation equal to one is recommended.
- If they are in the same units, you might or might not scale the variables.



# Proportion Variance Explained

- To understand the strength of each component, we are interested in knowing the proportion of variance explained (PVE) by each one.
- The *total variance* present in a data set (assuming that the variables have been centered to have mean zero) is defined as

$$\sum_{j=1}^p \text{Var}(X_j) = \sum_{j=1}^p \frac{1}{n} \sum_{i=1}^n x_{ij}^2,$$

and the variance explained by the  $m$ th principal component is

$$\text{Var}(Z_m) = \frac{1}{n} \sum_{i=1}^n z_{im}^2.$$

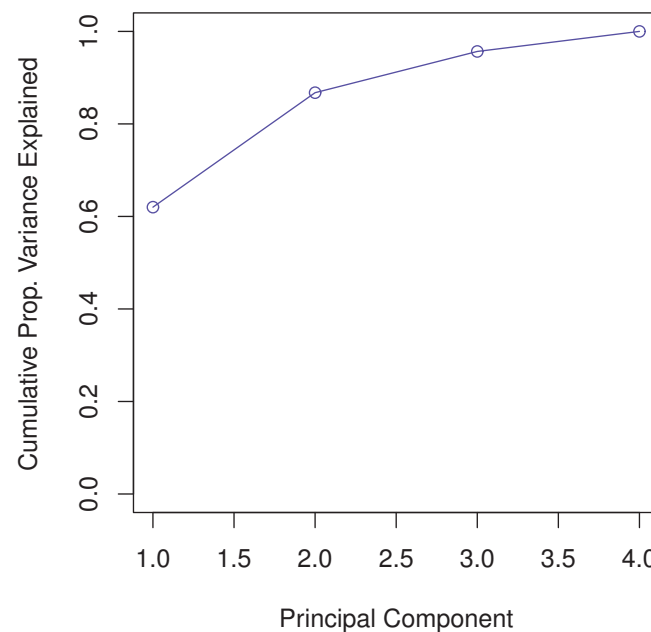
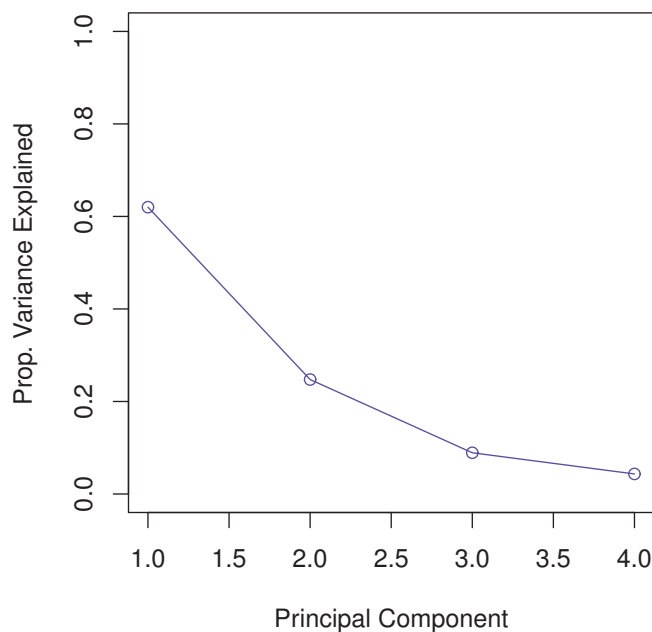
- It can be shown that  $\sum_{j=1}^p \text{Var}(X_j) = \sum_{m=1}^M \text{Var}(Z_m)$ , with  $M = \min(n - 1, p)$ .

## Proportion Variance Explained: continued

- Therefore, the PVE of the  $m$ th principal component is given by the positive quantity between 0 and 1

$$\frac{\sum_{i=1}^n z_{im}^2}{\sum_{j=1}^p \sum_{i=1}^n x_{ij}^2}.$$

- The PVEs sum to one. We sometimes display the cumulative PVEs.



# How many principal components should we use?

If we use principal components as a summary of our data, how many components are sufficient?

- No simple answer to this question, as cross-validation is not available for this purpose.
  - *Why not?*
  - When could we use cross-validation to select the number of components?
- the “scree plot” on the previous slide can be used as a guide: we look for an “elbow”.

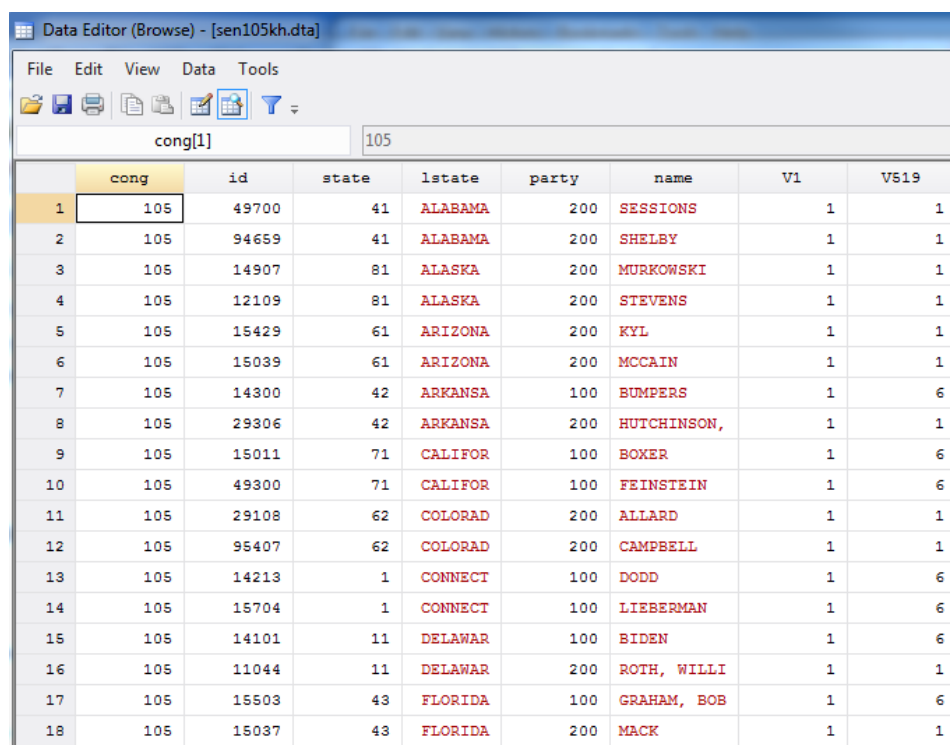
# 1 PCA examples with many variables

1. Latent semantic indexing: PCA on word frequencies.

- All pages that talk about e.g. farming do not contain the words "farming". Construct a matrix of term frequencies x pages. Do PCA on the word frequency vectors. Index pages that are most similar in their PCA projections to pages containing the word "farming".

2. Voting in the US Senate.

- 105th Senate: 612 roll call votes by each of 100 Senators.
- How can we characterize the voting of a Senator?
  - PCA: what linear combination of votes has the largest variance / best predict the actual voting of each Senator?



Data Editor (Browse) - [sen105kh.dta]								
File Edit View Data Tools								
congr[1] 105								
	congr	id	state	lstate	party	name	V1	V519
1	105	49700	41	ALABAMA	200	SESSIONS	1	1
2	105	94659	41	ALABAMA	200	SHELBY	1	1
3	105	14907	81	ALASKA	200	MURKOWSKI	1	1
4	105	12109	81	ALASKA	200	STEVENS	1	1
5	105	15429	61	ARIZONA	200	KYL	1	1
6	105	15039	61	ARIZONA	200	MCCAIN	1	1
7	105	14300	42	ARKANSAS	100	BUMPERS	1	6
8	105	29306	42	ARKANSAS	200	HUTCHINSON,	1	1
9	105	15011	71	CALIFORNIA	100	BOXER	1	6
10	105	49300	71	CALIFORNIA	100	FEINSTEIN	1	6
11	105	29108	62	COLORADO	200	ALLARD	1	1
12	105	95407	62	COLORADO	200	CAMPBELL	1	1
13	105	14213	1	CONNECTICUT	100	DODD	1	6
14	105	15704	1	CONNECTICUT	100	LIEBERMAN	1	6
15	105	14101	11	DELAWARE	100	BIDEN	1	6
16	105	11044	11	DELAWARE	200	ROTH, WILLI	1	1
17	105	15503	43	FLORIDA	100	GRAHAM, BOB	1	6
18	105	15037	43	FLORIDA	200	MACK	1	1

Principal components/correlation	Number of obs	=	100
	Number of comp.	=	99
	Trace	=	604
Rotation: (unrotated = principal)	Rho	=	1.0000

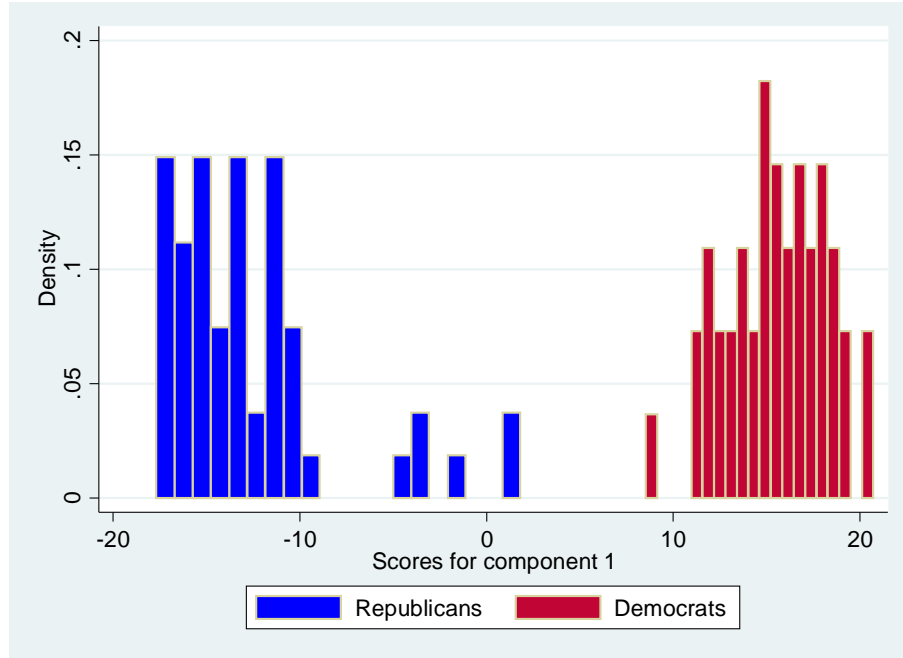
Component	Eigenvalue	Difference	Proportion	Cumulative
Comp1	214.87	190.72	0.3557	0.3557
Comp2	24.1495	9.19101	0.0400	0.3957
Comp3	14.9585	1.16161	0.0248	0.4205
Comp4	13.7969	1.92706	0.0228	0.4433
Comp5	11.8698	.524444	0.0197	0.4630

- What is the first dimension, responsible for 36% of the variance voting?



- In the four votes with the highest loading, all Democrats vote on one side and all Republicans on the other.
- Predict the projection (score) of each Senator's voting vector on the first principal component

$$z_i = \sum_{v=1}^{612} \phi_v x_{vi} = \sum_{v=1}^{612} \text{loading}_v * \text{senator\_vote}_{vi}$$



- Possible interpretation: The first principal component corresponds to left-right ideology (Democrat-Republican).
  - The score is the Senators ideology score  $z_i$  (based on voting) and the
  - The vote loading,  $\phi_v$ , is the ideological loading of vote  $v$ .
- PCA computes the linear projections that minimizes

$$\sum_{i=1}^{100} \sum_{v=1}^{612} (\hat{v}_{vi} - \text{senator\_vote}_{vi})^2.$$

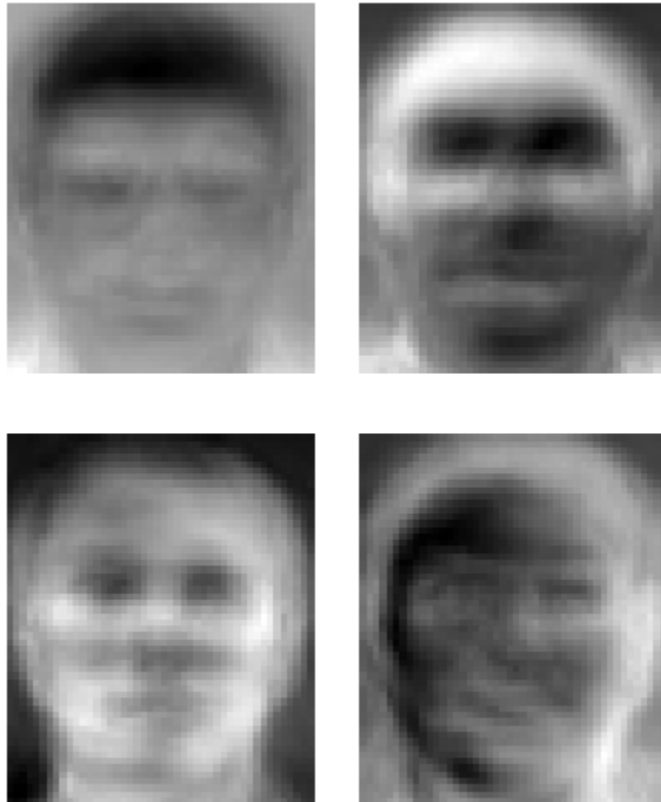
where the Senator's projected vote on  $\hat{v}_{vi} = z_i \phi_v$ .

# Even more dimensions: Face recognition

- Large set of digitized images of human faces is taken under the same lighting conditions.
- The images are normalized to line up the eyes and mouths.
- The eigenvectors of the covariance matrix of the statistical distribution of face image vectors are then extracted.
- These eigenvectors are called eigenfaces.

# Eigenfaces

- The principal eigenface looks like a bland androgynous average human face



# PCA 3: Face recognition



`coefficients =`

```
{-6.85693, 23.7498, -11.4515, -3.43352, 5.24749, -7.1615,  
8.09015, -9.7205, -0.660834, -2.4148, -10.3942, 3.33424,  
2.94988, -2.75981, 3.02687, -2.4499, -2.09885, -5.98832,  
-4.22564, -0.65014, 2.20144, -5.43782, -9.61821, -3.25227,  
7.49413, -0.145002, 7.61483, -0.696994, -3.7731, 3.23569,  
-1.78853, 0.0400116, -3.86804, -2.02456, 2.20949, -1.86902,  
1.23445, 0.140996, 0.698304, -0.420466, 2.30691, 3.70434,  
1.02417, 0.382809, 0.413049, -0.994902, 0.754145, 0.363418,  
-0.383865, 1.46379, 1.96381, -2.90388, -2.33381, -0.438939,  
-0.30523, -0.105925, 0.665962, -0.729409, -1.28977, 0.150497,  
0.645343, 0.30724, -1.04942, 1.0462, -0.60808, 0.333288,  
1.09659, -1.38876, 0.33875, 0.278604, 1.0632, -0.0446148,  
0.24526, -0.283482, -0.236843, 0.312122};
```

# Clustering

- *Clustering* refers to a very broad set of techniques for finding *subgroups*, or *clusters*, in a data set.
- We seek a partition of the data into distinct groups so that the observations within each group are quite similar to each other,
- To make this concrete, we must define what it means for two or more observations to be *similar* or *different*.
- Indeed, this is often a domain-specific consideration that must be made based on knowledge of the data being studied.

# PCA vs Clustering

- PCA looks for a low-dimensional representation of the observations that explains a good fraction of the variance.
- Clustering looks for homogeneous subgroups among the observations.

# Clustering for Market Segmentation

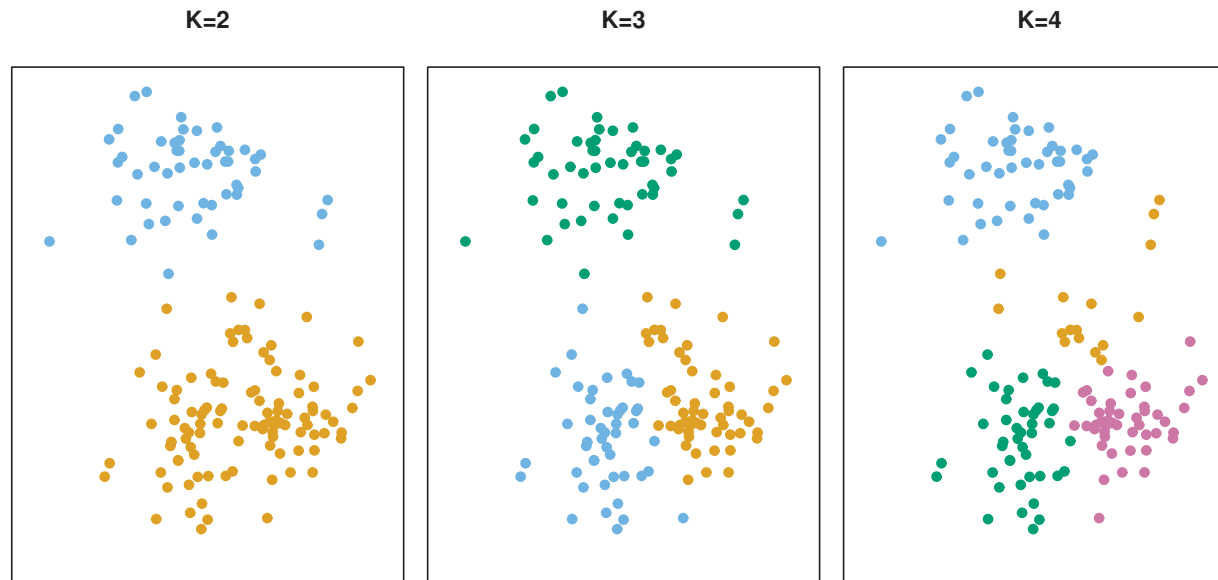
- Suppose we have access to a large number of measurements (e.g. median household income, occupation, distance from nearest urban area, and so forth) for a large number of people.
- Our goal is to perform *market segmentation* by identifying subgroups of people who might be more receptive to a particular form of advertising, or more likely to purchase a particular product.
- The task of performing market segmentation amounts to clustering the people in the data set.

## Two clustering methods

- In *K-means clustering*, we seek to partition the observations into a pre-specified number of clusters.
- In *hierarchical clustering*, we do not know in advance how many clusters we want; in fact, we end up with a tree-like visual representation of the observations, called a *dendrogram*, that allows us to view at once the clusterings obtained for each possible number of clusters, from 1 to  $n$ .



# $K$ -means clustering



A simulated data set with 150 observations in 2-dimensional space. Panels show the results of applying  $K$ -means clustering with different values of  $K$ , the number of clusters. The color of each observation indicates the cluster to which it was assigned using the  $K$ -means clustering algorithm. Note that there is no ordering of the clusters, so the cluster coloring is arbitrary. These cluster labels were not used in clustering; instead, they are the outputs of the clustering procedure.

## Details of $K$ -means clustering

Let  $C_1, \dots, C_K$  denote sets containing the indices of the observations in each cluster. These sets satisfy two properties:

1.  $C_1 \cup C_2 \cup \dots \cup C_K = \{1, \dots, n\}$ . In other words, each observation belongs to at least one of the  $K$  clusters.
2.  $C_k \cap C_{k'} = \emptyset$  for all  $k \neq k'$ . In other words, the clusters are non-overlapping: no observation belongs to more than one cluster.

For instance, if the  $i$ th observation is in the  $k$ th cluster, then  $i \in C_k$ .

## Details of $K$ -means clustering: continued

- The idea behind  $K$ -means clustering is that a *good* clustering is one for which the *within-cluster variation* is as small as possible.
- The within-cluster variation for cluster  $C_k$  is a measure  $WCV(C_k)$  of the amount by which the observations within a cluster differ from each other.
- Hence we want to solve the problem

$$\underset{C_1, \dots, C_K}{\text{minimize}} \left\{ \sum_{k=1}^K WCV(C_k) \right\}. \quad (2)$$

- In words, this formula says that we want to partition the observations into  $K$  clusters such that the total within-cluster variation, summed over all  $K$  clusters, is as small as possible.

## How to define within-cluster variation?

- Typically we use Euclidean distance

$$\text{WCV}(C_k) = \frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2, \quad (3)$$

where  $|C_k|$  denotes the number of observations in the  $k$ th cluster.

- Combining (2) and (3) gives the optimization problem that defines  $K$ -means clustering,

$$\underset{C_1, \dots, C_K}{\text{minimize}} \left\{ \sum_{k=1}^K \frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2 \right\}. \quad (4)$$

# $K$ -Means Clustering Algorithm

1. Randomly assign a number, from 1 to  $K$ , to each of the observations. These serve as initial cluster assignments for the observations.
2. Iterate until the cluster assignments stop changing:
  - 2.1 For each of the  $K$  clusters, compute the cluster *centroid*.  
The  $k$ th cluster centroid is the vector of the  $p$  feature means for the observations in the  $k$ th cluster.
  - 2.2 Assign each observation to the cluster whose centroid is closest (where *closest* is defined using Euclidean distance).

# Properties of the Algorithm

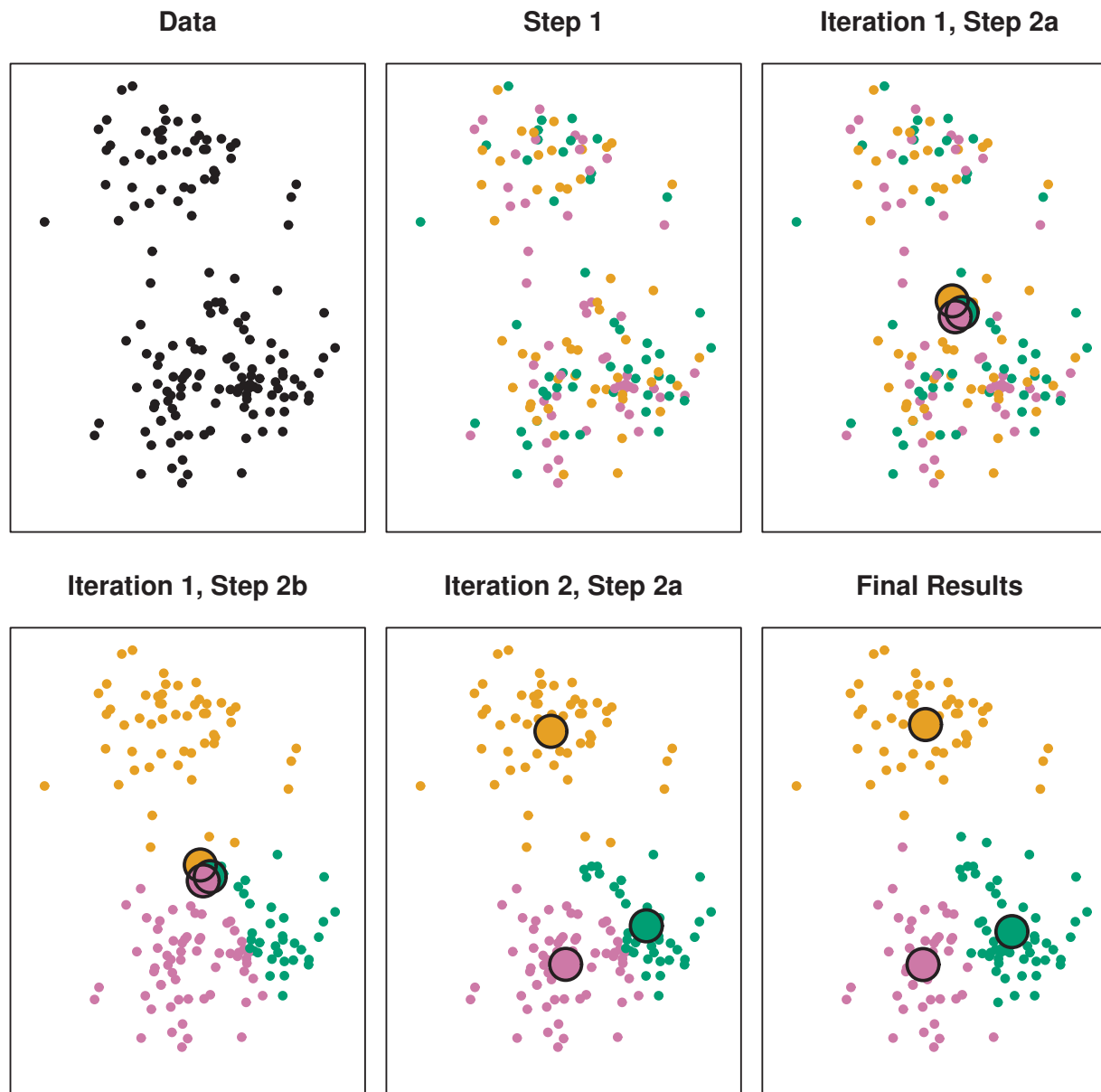
- This algorithm is guaranteed to decrease the value of the objective (4) at each step. *Why?* Note that

$$\frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2 = 2 \sum_{i \in C_k} \sum_{j=1}^p (x_{ij} - \bar{x}_{kj})^2,$$

where  $\bar{x}_{kj} = \frac{1}{|C_k|} \sum_{i \in C_k} x_{ij}$  is the mean for feature  $j$  in cluster  $C_k$ .

- however it is not guaranteed to give the global minimum.  
*Why not?*

# Example



## Details of Previous Figure

The progress of the K-means algorithm with  $K=3$ .

- *Top left:* The observations are shown.
- *Top center:* In Step 1 of the algorithm, each observation is randomly assigned to a cluster.
- *Top right:* In Step 2(a), the cluster centroids are computed. These are shown as large colored disks. Initially the centroids are almost completely overlapping because the initial cluster assignments were chosen at random.
- *Bottom left:* In Step 2(b), each observation is assigned to the nearest centroid.
- *Bottom center:* Step 2(a) is once again performed, leading to new cluster centroids.
- *Bottom right:* The results obtained after 10 iterations.



# Example: different starting values



## Details of Previous Figure

$K$ -means clustering performed six times on the data from previous figure with  $K = 3$ , each time with a different random assignment of the observations in Step 1 of the  $K$ -means algorithm.

Above each plot is the value of the objective (4).

Three different local optima were obtained, one of which resulted in a smaller value of the objective and provides better separation between the clusters.

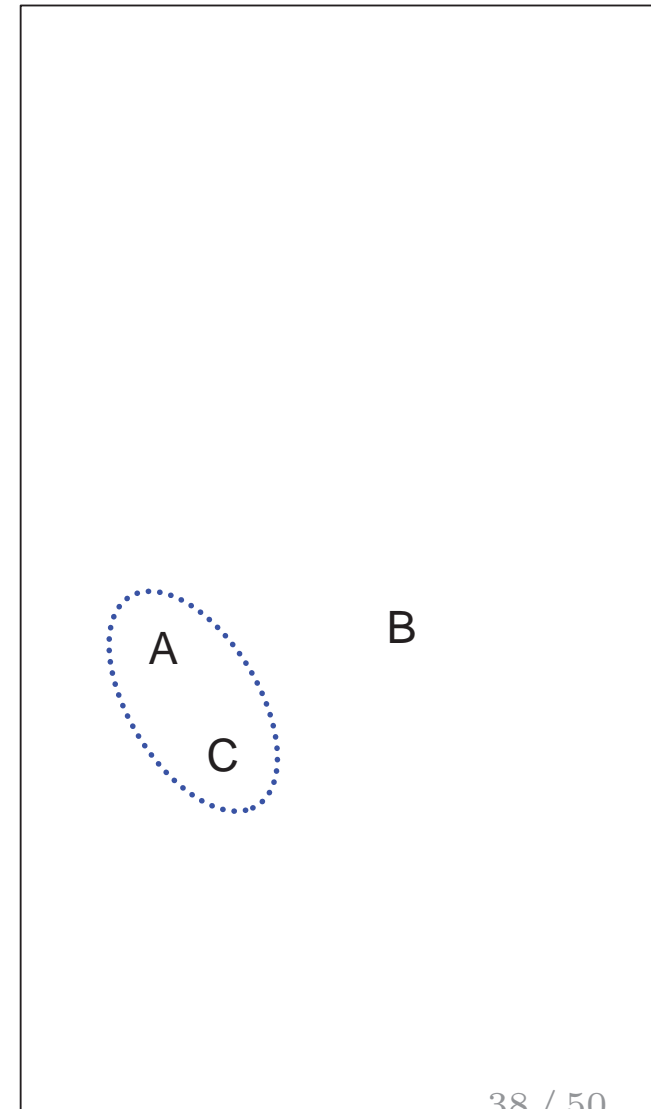
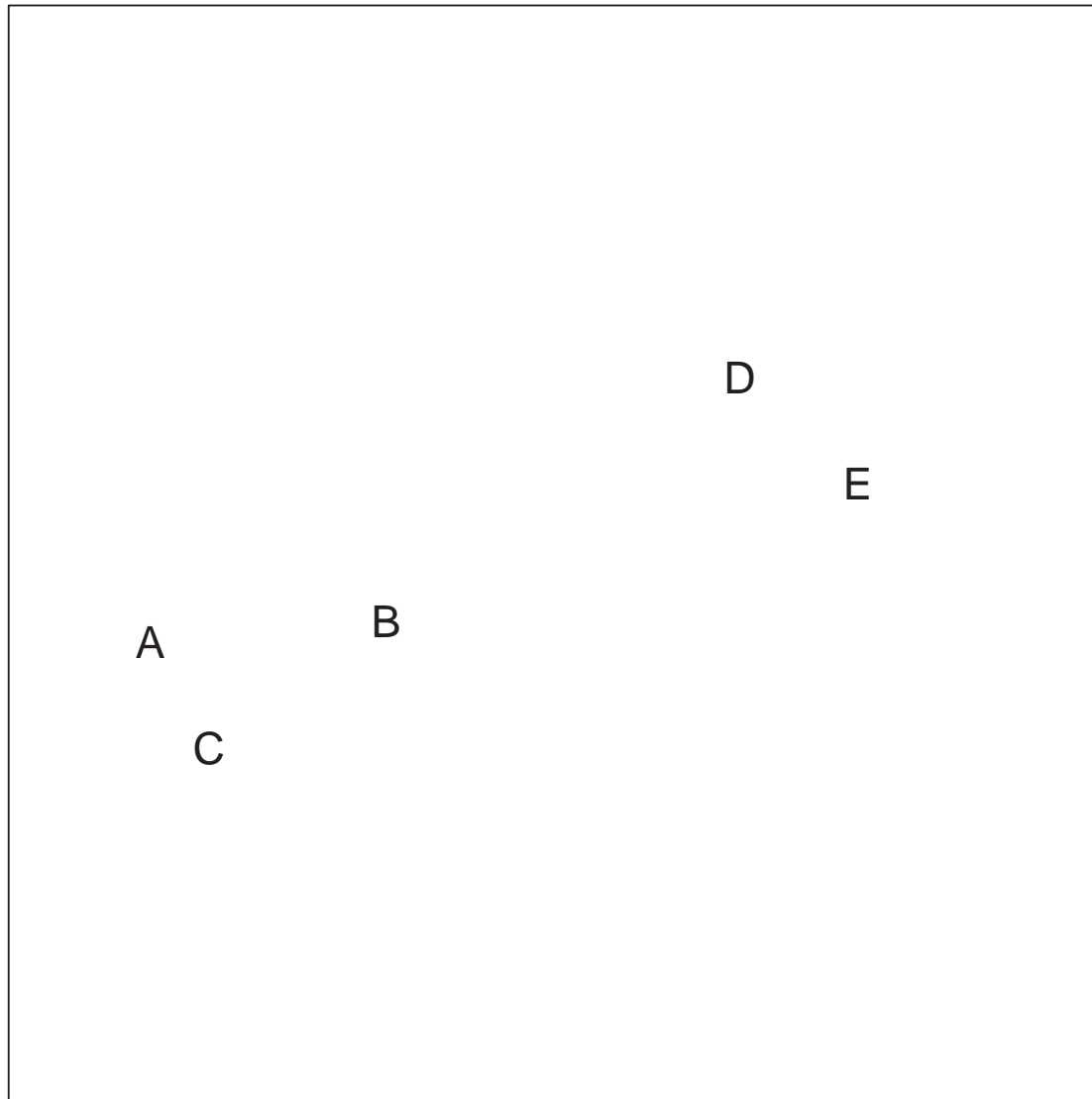
Those labeled in red all achieved the same best solution, with an objective value of 235.8

# Hierarchical Clustering

- $K$ -means clustering requires us to pre-specify the number of clusters  $K$ . This can be a disadvantage (later we discuss strategies for choosing  $K$ )
- *Hierarchical clustering* is an alternative approach which does not require that we commit to a particular choice of  $K$ .
- In this section, we describe *bottom-up* or *agglomerative* clustering. This is the most common type of hierarchical clustering, and refers to the fact that a dendrogram is built starting from the leaves and combining clusters up to the trunk.

# Hierarchical Clustering: the idea

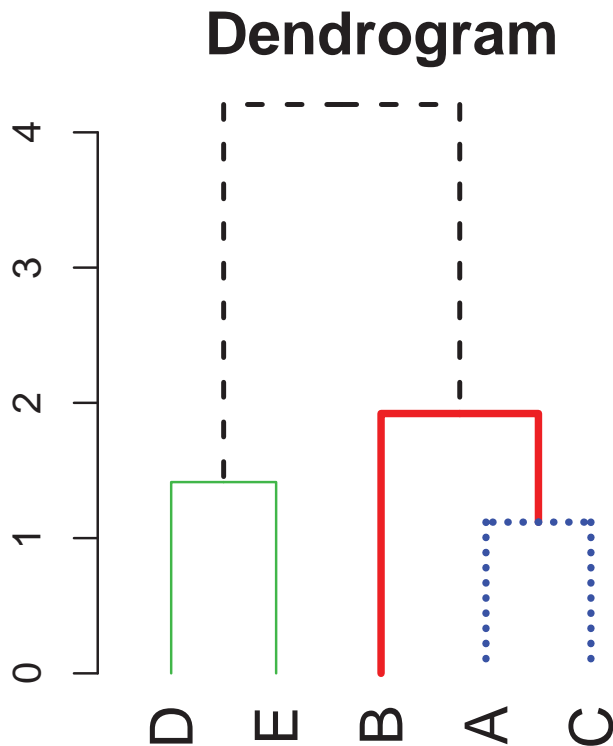
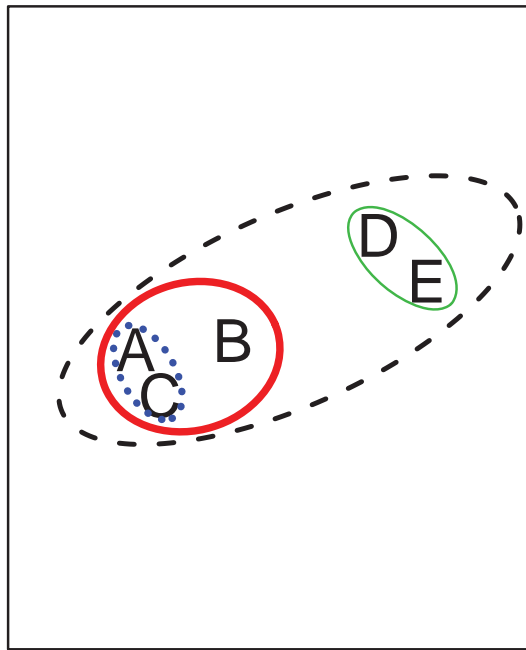
Builds a hierarchy in a “bottom-up” fashion...



# Hierarchical Clustering Algorithm

The approach in words:

- Start with each point in its own cluster.
- Identify the **closest** two clusters and merge them.
- Repeat.
- Ends when all points are in a single cluster.



### K-means clustering: specification design for causal estimates

- "Grouped Patterns of Heterogeneity in Panel Data", Bonhomme and Manresa (EMA 2015)
- Many studies in economics include group-fixed effects: at what groups should we place these?
- Select group structure to minimize squared residual with respect to all possible groupings of the cross-sectional units.

$$\left(\hat{\theta}, \hat{\alpha}, \hat{\gamma}\right) = \arg \min \sum_{i=1}^N \sum_{t=1}^T \left(y_{it} - x'_{it}\theta - \alpha_{g_{it}}\right)^2,$$

where  $\gamma$  is a partition of individuals into groups.

- Algorithm 1: For given group-fixed effects  $\alpha$  and coefficients  $\theta$ , assign each individual to the group with lowest squared residual

$$\hat{g}_i(\theta, \alpha) = \arg \min_{g \in \{1, 2, \dots, G\}} \sum_{t=1}^T \left(y_{it} - x'_{it}\theta - \alpha_{gt}\right)^2.$$

$$\left(\hat{\theta}, \hat{\alpha}\right) = \arg \min \sum_{i=1}^N \sum_{t=1}^T \left(y_{it} - x'_{it}\theta - \alpha_{\hat{g}_i(\theta, \alpha)t}\right)^2.$$

Iterate until convergence.

- Solution depends on starting values. Draw starting values at random and select lowest squared residual grouping.
- Number of groups selected by information criteria.

- Asymptotic properties. Suppose true data generating process is

$$y_{it} = x'_{it}\theta^0 + \alpha_{g^0_{it}} + \nu_{it},$$

where  $g^0_i$  denotes true group membership.

- Consistency: as  $N$  (cross-sectional units) and  $T$  (time periods) both go to infinity,

$$\hat{\theta} \xrightarrow{p} \theta^0$$

$$\frac{1}{NT} \sum_{i=1}^N \sum_{t=1}^T \left( \hat{\alpha}_{\hat{g}_{it}} - \alpha_{g^0_{it}} \right)^2 \xrightarrow{p} 0$$

under some regularity conditions.  $N \rightarrow \infty$  needed for  $\alpha$  estimates,  $T \rightarrow \infty$  needed for  $g$  estimates.

- Distribution: as  $N$ ,  $T$  and  $N/T^v \rightarrow \infty$  for some  $v > 0$  then group membership estimation does not affect inference. OLS s.e.'s are correct conditional on final group membership.  
(The estimated group membership indicators are uniformly consistent for the population ones under these conditions.)

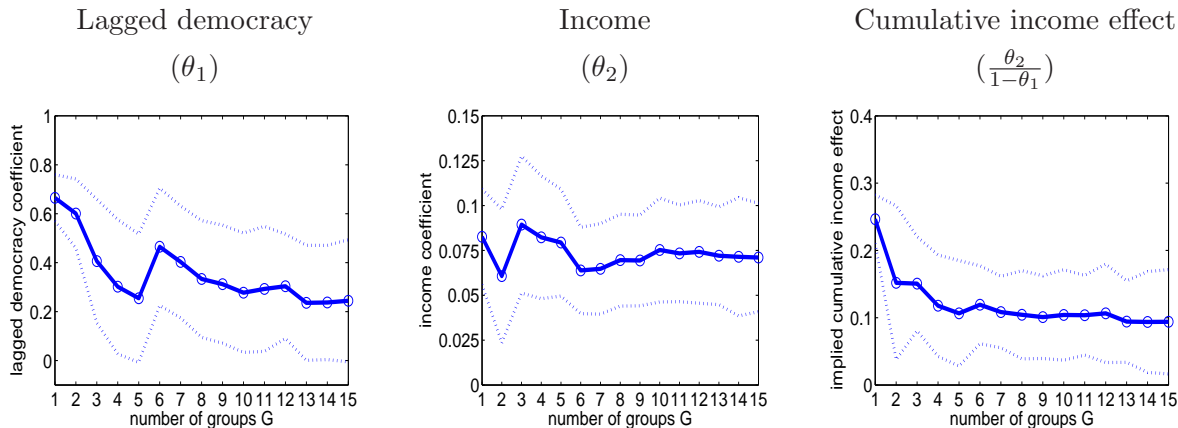
- Example: "Income and Democracy", Acemoglu et al. (AER, 2008)

$$democracy_{it} = \theta_1 democracy_{it-1} + \theta_2 \log GDP_{it-1} + \alpha_{g_{it}} + \nu_{it}.$$

- Finite sample properties based on simulated data.
  - Small probabilities of group misclassification (less than 10% when  $G = 3$  and  $G = 5$ ), and moderate biases on parameters.

Show Figure 1 and Figure 2.

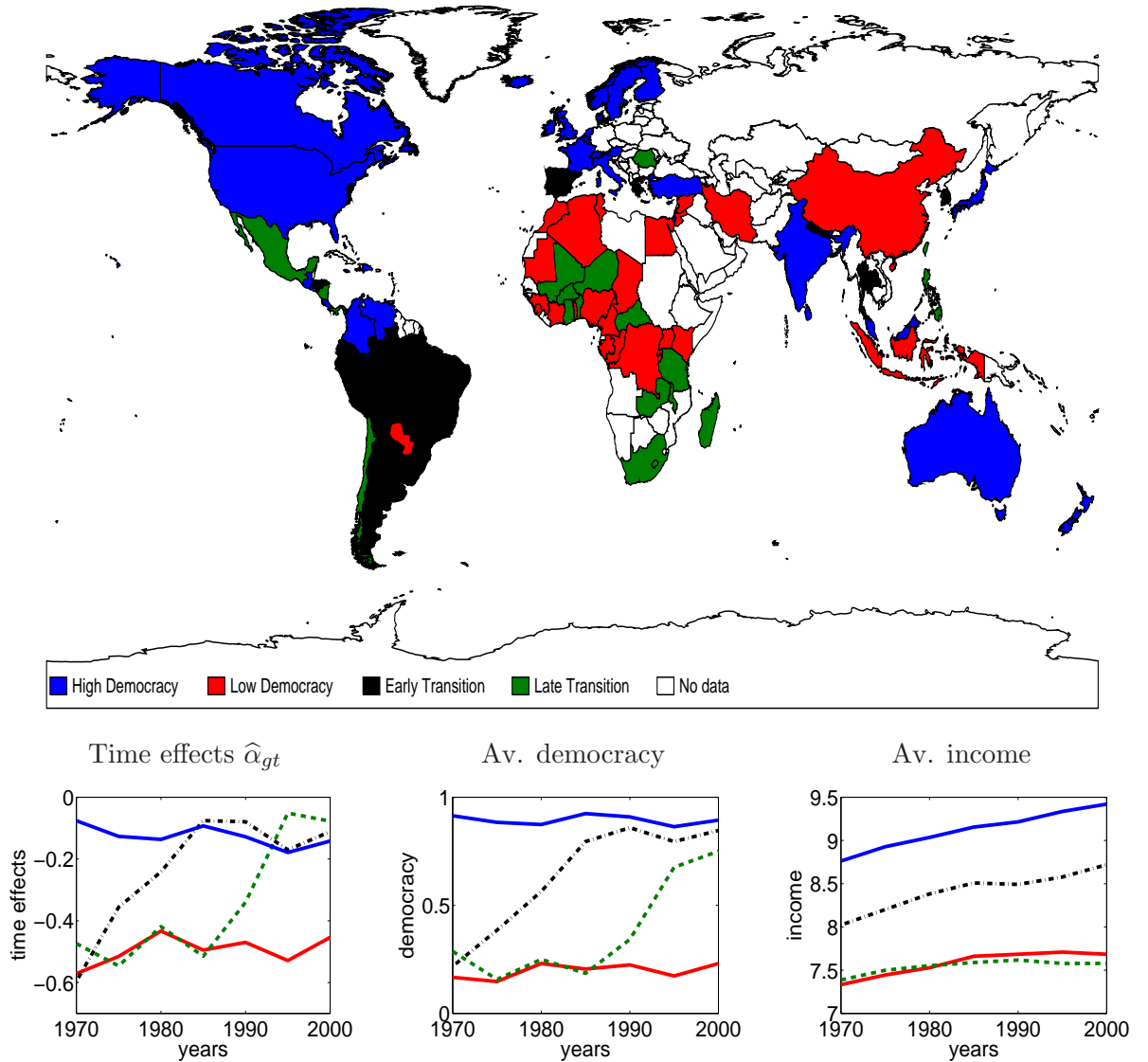
Figure 1: Coefficients of income and lagged democracy



*Note: Balanced panel from Acemoglu et al. (2008). The x-axis shows the number of groups  $G$  used in estimation, the y-axis reports parameter values. 95%-confidence intervals clustered at the country level are shown in dashed lines. Confidence intervals are based on bootstrapped standard errors (100 replications). Details on the computation are provided in the supplementary appendix.*



Figure 2: Patterns of heterogeneity,  $G = 4$



*Note:* See the notes to Figure 1. On the bottom panel, the left graph shows the group-specific time effects  $\hat{\alpha}_{gt}$ . The other two graphs show the group-specific averages of democracy and lagged log-GDP per capita, respectively. Calendar years (1970-2000) are shown on the x-axis. Light solid lines correspond to Group 1 (“high-democracy”), dark solid lines to Group 2 (“low-democracy”), light dashed lines to Group 3 (“early transition”), and dark dashed lines to Group 4 (“late transition”). The top panel shows group membership. The list of countries by group is given in the supplementary appendix.

# Conclusions

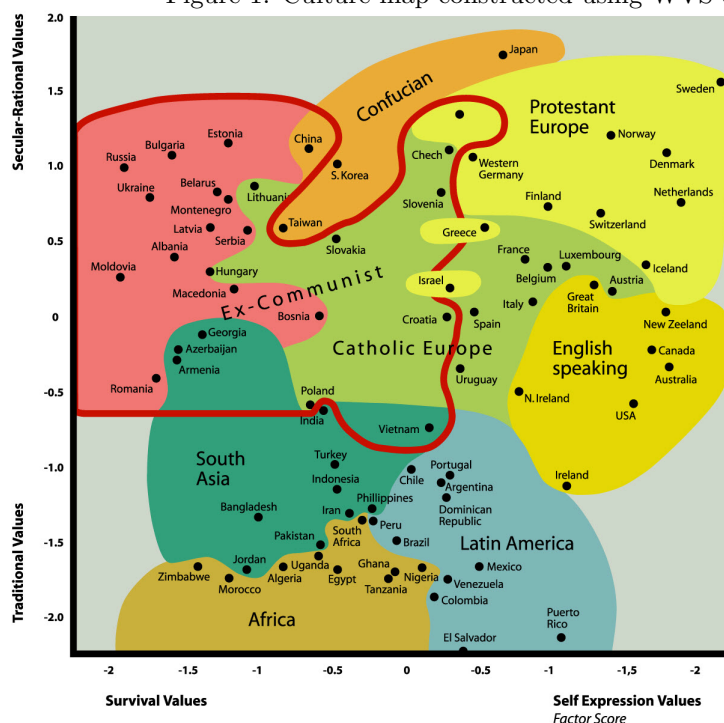
- *Unsupervised learning* is important for understanding the variation and grouping structure of a set of unlabeled data, and can be a useful pre-processor for supervised learning
- It is intrinsically more difficult than *supervised learning* because there is no gold standard (like an outcome variable) and no single objective (like test set accuracy).
- It is an active field of research, with many recently developed tools such as *self-organizing maps*, *independent components analysis* and *spectral clustering*.  
See *The Elements of Statistical Learning*, chapter 14.

# The World Value Survey and Unsupervised Learning

In this problems set, we use data from the World Value Survey (WVS). The WVS is a global research project that has tracked values and beliefs across the world since 1981. Unsupervised learning methods applied to the WVS underlie many well-known findings about variations in values and beliefs across countries, for example the famous culture maps like the one in Figure 1.

For this exercise, we will use the WVS to explore unsupervised learning methods such as cluster analysis and principal component analysis.

Figure 1: Culture map constructed using WVS-data



## 1 Download data and documentation

The dataset and the codebook from the World Value Survey can be downloaded from <http://www.worldvaluessurvey.org/WVSDocumentationWV6.jsp>. Download both. The dataset can be downloaded in many formats. I downloaded it in R, but if you prefer, you can also do the analysis in Stata or Python.

To download the data you will need to submit your name and institution. Write that it will be used for instruction, and as project description write "Problem set for unsupervised learning in Economics" or similarly.

The dataset will be downloaded as an .Rdata-file. If you double-click on it an R-session will open with the dataset loaded. Type `ls()` and hit enter, and you will see that a data frame called "WV6\_Data.R" is active in the session.

The data contains 90350 observations on individuals from 60 countries.

## 2 Creating the data set

Use the codebook to identify the country variable, and a large number (>50) of variables that measures values in a one-dimensional ordered way. An example of a one-dimensional value variable is variable "V8" which asks people to rate the importance of work from very important to not at all important. Marital status and number of children are not value variables. The choices about aims of a country is not a one-dimensional ordered variable.

Take the country average of each of your chosen variables, and collapse on country levels. Make sure that you recode missing values and exclude them (check codebook) before taking the country averages. Also make sure there are no missing variables in the collapsed country values (all questions are not included for all countries). Normalize the values (but try the analysis without normalizing the values as well. Are there any problems? Why?).

## 3 Analysis

Use your cleaned dataset to try out unsupervised learning. Use cluster analysis to try out which questions naturally belong together and which countries natural belong together. Interpret the results with words.

Use principal component analysis to find how answers to different questions covaries. Interpret the principal components. What countries score high on the different principal components? What variables do the principal components load strongly on? Find a way to use principal component analysis to do a cultural plot like the one in Figure 1.

Interpret the cultural map briefly. For example, does your analysis confirm the importance of self-expression vs survival values, and secular-rational vs traditional? Given the patterns you see in the data, would you choose the interpret anything differently? Do you find alternative patterns in the data that are possible to capture in words?

## 4 Submission

For the submission, you should send a code which fully replicates your results provided that one can load the WVS dataset. There should be a plot of the culture map, and some illustrative visualization of how different questions and country cluster, as well as of a culture map. There should also be a text that interpret the clusters and principal components in words.

## 5 Advice for packages and functions

For data manipulation, I warmly recommend the dplyr-package which has made it much simpler to manipulate data sets in R. Replacing all negative value with missing values and taking the country averages excluding the missing values takes three lines of code with dplyr. See <https://www.youtube.com/watch?v=jWjqLW-u3hc> for a tutorial or try learning it by doing.

The package "countrycode" is useful for converting the ISO3-numeric code to country names and 3-letter ISO codes.

The functions kmeans and prcomp can be used for clustering and principal components analysis. Read up the documentation on the functions and understand how the returned objects relate to the theory you have learnt in class. See <http://www-bcf.usc.edu/~gareth/ISL/Chapter%2010%20Labs.txt> for coding examples.

Plotting can be done beautifully with ggplot2 but you have to incur a fixed cost to learn it. This is advisable if you plan to keep working with R. The package lattice is somewhat easier, but you can also solve the exercise using the standard R plotting tools.

The commands for cluster analysis and principal component analysis in stata are kmeans and pca. Remember to set seed before executing the kmeans command so that the results are replicable.

Useful Stata commands to see factor loadings after running the pca command:

```
matrix define A=e(L)
svmat2 A, rnames(V)
gen aA1=abs(A1)
gsort -aA1
```

To transpose a data set in Stata from (row=country, column=question) to (row=question, column=country), use

```
reshape long varlist, i(country) j(question)
reshape wide varlist, i(question) j(country)
```