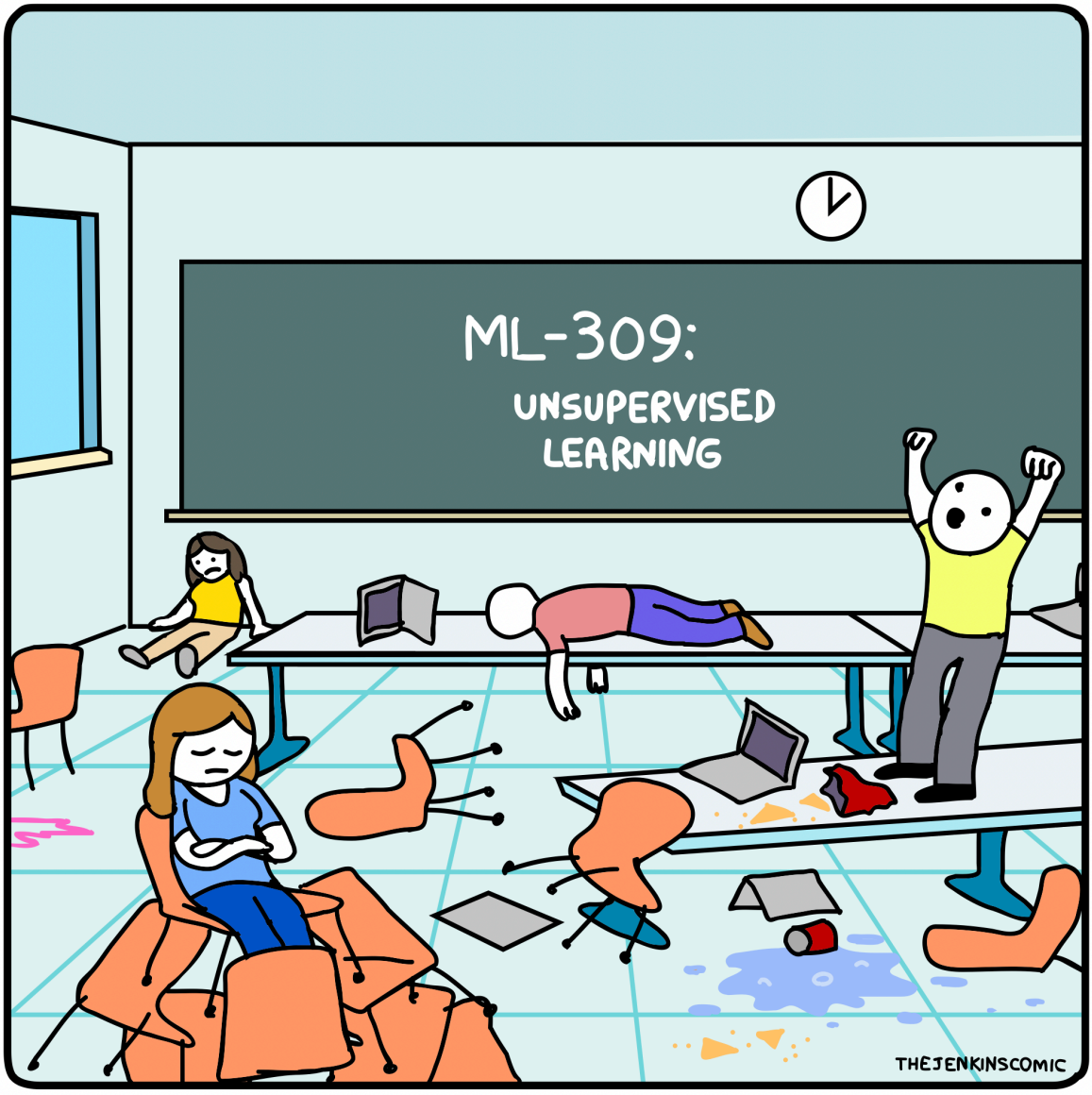
Chapter 10: Unsupervised Learning



Credit: https://thejenkinscomic.net/?id=366

This chapter will focus on methods intended for the setting in which we only have a set of features $X_1, \ldots, X_p$ measured on n observations.

We are not interested in prediction because we have no associated response Y.

Goal: discover interesting things about $X_1, \ldots, X_p$

- Is there an informative way to plot the data?
- Can we discover subgroups among the variables or the observations?

1 The Challenge of Unsupervised Learning

Supervised learning is a well-understood area.

You now have a good grasp of supervised learning.

If you are asked to predict a binary response you have many ways to do it:

logistic regression, LDA, QDA, classification trees, RF, boosted trees, SVM

and a clear understanding of how to assess quality of your model:

cross-validation, validation on a test set

In contrast, unsupervised learning is often much more challenging.

more subjective, no simple goal for the analysis, e.g. prediction.

Unsupervised learning is often performed as part of an *exploratory data analysis*.

1st part of any analysis, before fitting models.

It can be hard to assess the results obtained from unsupervised learning methods.

No universally accepted mechanism for performing cross-validation or validation on a test set

Because there is no way for us to "check our work" with no response variable.

→ we don't know the true answer!

Techniques for unsupervised learning are of growing importance in a number of fields.

cancer research: assay gene expression levels in 100 patients and look for subgroups among cancer samples to better understand the disease.

online shopping: identify similar groups of shoppers and show preferential items that he/she might be particularly interested in.

My research: Entity resolution: many noisy databases without a unique identifying attribute → can we find matches or links?

2 Principal Components Analysis

We have already seen principal components as a method for dimension reduction.

When faced with a large set of correlated variables, we used principal components to summarise this set with a smaller number of representative variables that collectively explain most of the variability in the original data set.

PC direction = directions in feature space along which original data are highly variable

PCR = use principal components as predictors in a regression model instead of original predictors.

Principal Components Analysis (PCA) refers to the process by which principal components are computed and the subsequent use of these components to understand the data.

Unsupervised approach (involves only features $X_1, \ldots, X_p$, no response Y).

Apart from producing derived variables forr use in supervised learning, PCA also serves as a tool for data visualization.

Visualization of observations or of variables

2.1 What are Principal Components?

Suppose we wish to visualize n observations with measurements on a set of p features as part of an exploratory data analysis. $X_1, \ldots, X_p$

We could do this by examining 2D scatterplots of the data which contain n obs measured on 2 features.

$\Rightarrow \binom{p}{2} = \frac{p(p-1)}{2}$ scatterplots, e.g. w/ $p=10 \Rightarrow 45$ plots!

- Too many to look at.
- Likely no plot will be informative because they contain a small fraction of information present in the data.

For visualization in high dimension

Goal: We would like to find a low-dimensional representation of the data that captures as much of the information as possible.

Then plot the observations in low-dimensional space.

PCA provides us a tool to do just this.

finds low-dimensional representations of a data set that contain as much as possible of the variation (information).

Idea: Each of the n observations lives in p dimensional space, but not all of these dimensions are equally interesting.

PCA seeks a small number of dimensions that are as interesting as possible.

"interesting" = amount of information along each dimension.

Each dimension found by PCA is a linear combination of the p features.

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

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

normalized: $\sum_{j=1}^{p} \phi_{j1}^2 = 1$ (otherwise could result in arbitrarily large variance).

$\phi_{11}, \ldots, \phi_{p1}$ are called "loadings" of the first principal component

$\phi_1 = (\phi_{11}, \ldots, \phi_{p1})^T$ = "loading vector"

that has the largest variance.

Given a $n \times p$ data set $\boldsymbol{X}$, how do we compute the first principal component?

(1) Assume each variable has been centered (i.e. columns have mean zero) — only care about variances.

(2) look for linear combination of of the form

$$z_{i1} = \phi_{11} x_{i1} + \phi_{21} x_{i2} + \ldots + \phi_{p1} x_{ip}$$

w/ largest sample variance subject to $\sum_{j=1}^{p} \phi_{j1}^2 = 1$.

i.e. solve the following optimization problem: ← solved via eigen decomposition (outside scope of this course).

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

↑ can write this way because columns are centered.

$\Rightarrow \frac{1}{n} \sum_{i=1}^{n} x_{ij} = 0 \Rightarrow \frac{1}{n} \sum_{i=1}^{n} z_{i1} = 0.$

$\Rightarrow$ this is the sample variance of $z_{i1}, i = 1, \ldots, n$.

$z_{11}, \ldots, z_{n1}$ are called "scores" of the first principal component.

There is a nice geometric interpretation for the first principal component.

The loading vector ϕ_1 defines a direction in the feature space along which the data vary the most.

If we project n data points onto this direction we get the scores

$z_{11}, \ldots, z_{1n}$.

After the first principal component Z_1 of the features has been determined, we can find the second principal component, Z_2. The second principal component is the linear combination of $X_1, \ldots, X_p$ that has maximal variance out of all linear combinations that are uncorrelated with Z_1.

The second principal component scores are

$$z_{2i} = \phi_{12} x_{i1} + \ldots + \phi_{p2} x_{ip}$$

ϕ_2 = second principal component loading vector

Z_2 uncorrelated w/ Z_1

$\Leftrightarrow$

ϕ_2 orthogonal to ϕ_1

[If $p = 2$, in 2D space there is only one possibility for ϕ_2

But for $p > 2$ there are multiple orthogonal options.]

To find Z_2, solve similar optimization problem w/ additional constraint

$$\text{maximize}_{\phi_{21}, \ldots, \phi_{2p}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{2j} x_{ij} \right)^2 \right\}$$

subject to $\sum_{j=1}^{p} \phi_{2j}^2 = 1$ and $\sum_{j=1}^{p} \phi_{2j} \cdot \phi_{1j} = 0$ (ϕ_2 and ϕ_1 are orthogonal)

Once we have computed the principal components, we can plot them against each other to produce low-dimensional views of the data.

each of the 50 states, # arrests per 100,000 residents for 3 crimes.

```
str(USArrests)
```

```
## 'data.frame':    50 obs. of  4 variables:
##  $ Murder  : num  13.2 10 8.1 8.8 9 7.9 3.3 5.9 15.4 17.4 ...
##  $ Assault : int  236 263 294 190 276 204 110 238 335 211 ...
##  $ UrbanPop: int  58 48 80 50 91 78 77 72 80 60 ...
##  $ Rape    : num  21.2 44.5 31 19.5 40.6 38.7 11.1 15.8 31.9 25.8
...
```

% pop in state living in an urban area.

```
USArrests_pca <- USArrests |>
  prcomp(scale = TRUE, center = TRUE)

summary(USArrests_pca) # summary
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.5749 0.9949 0.59713 0.41645
## Proportion of Variance 0.6201 0.2474 0.08914 0.04336
## Cumulative Proportion  0.6201 0.8675 0.95664 1.00000
```

PVE

Cumulative PVE

First two PC explain 86.75% of variability in data
last two PC only explain ~13% => looking at first 2 pretty good summary.

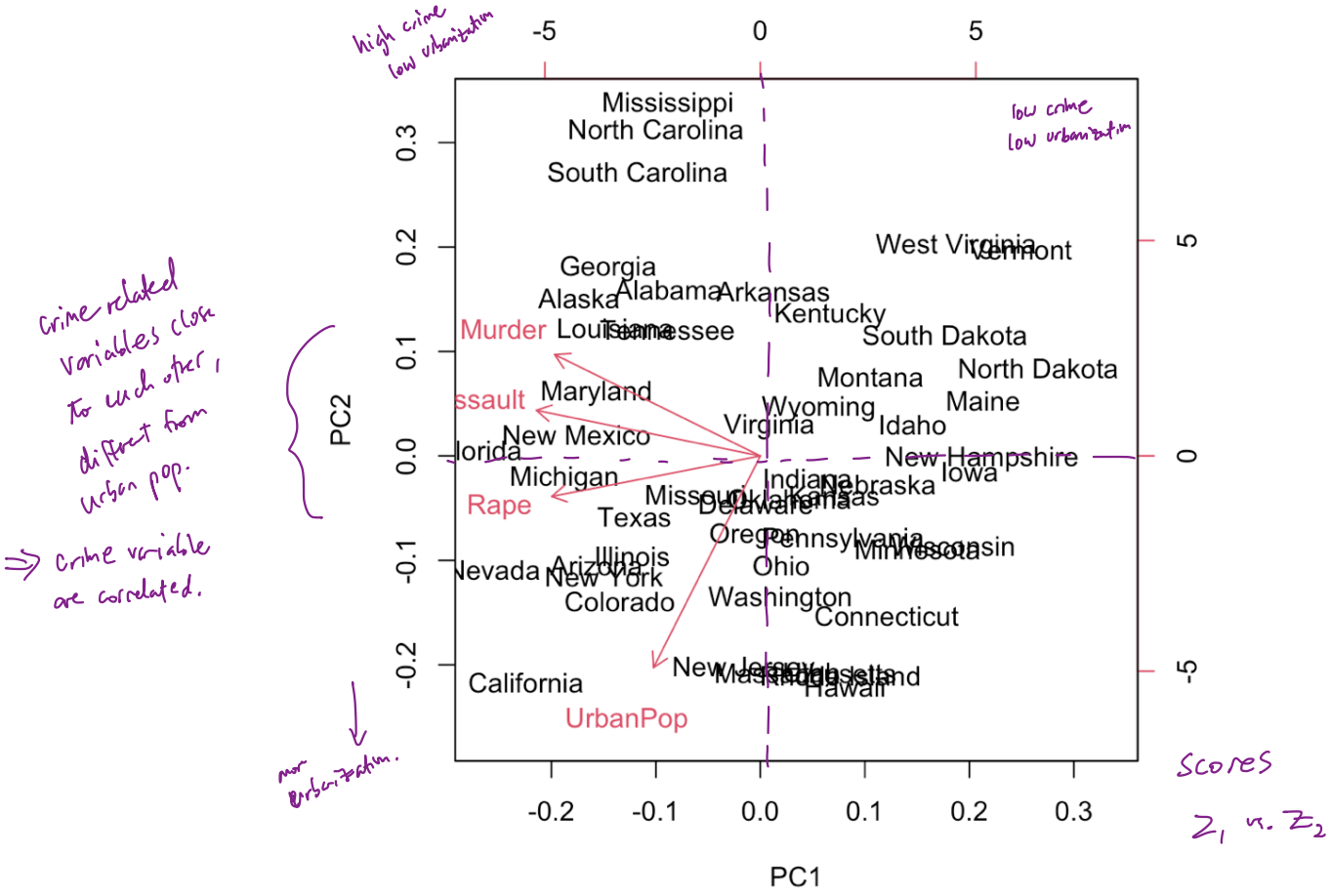
```
tidy(USArrests_pca, matrix = "loadings") |> # principal components
        loading matrix
  pivot_wider(names_from = PC, values_from = value)
```

ϕ_1 ϕ_2 ϕ_3 ϕ_4

```
## # A tibble: 4 × 5
##   column      `1`    `2`    `3`     `4`
##   <chr>     <dbl>  <dbl>  <dbl>   <dbl>
## 1 Murder   -0.536  0.418 -0.341  0.649
## 2 Assault  -0.583  0.188 -0.268 -0.743
## 3 UrbanPop -0.278 -0.873 -0.378  0.134
## 4 Rape     -0.543 -0.167  0.818  0.0890
```

```
## plot scores + directions
```

```
biplot(USArrests_pca)
```



Crime related variables close to each other, different from urban pop.

⇒ crime variable are correlated.

more urbanization.

scores z_1 vs. z_2

First loading places approx equal weight on 3 crime variables, less weight on UrbanPop.

⇒ this component ≈ measure of rate of serious crimes.

Second loading places most weight on Urban Pop ⇒ ≈ level of urbanization.

2.2 Scaling Variables

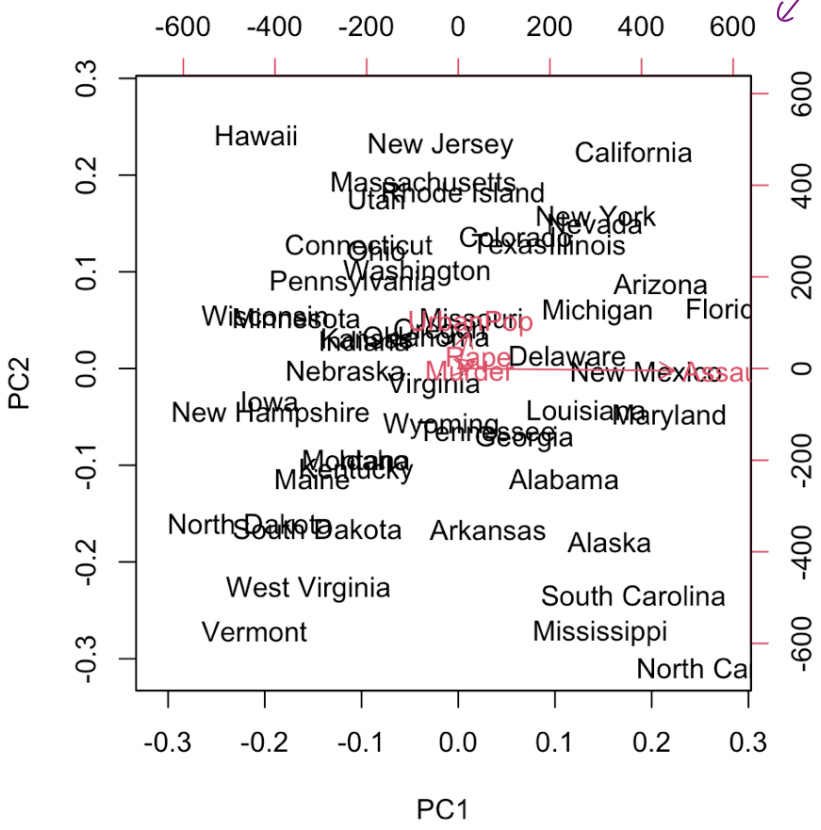
We've already talked about how when PCA is performed, the varriables should be centered to have mean zero.

Also, the results depend on whether variables have been scaled (to have the same sd).

This is in contrast to other methods we've seen before.

e.g. linear regression when we multiply a variable by c, the corresponding coefficient is changed by a factor of $1/c$.

same data as before, didn't scale.



Variables are measured in different units:
crime: #/100,000 ppl
Urban pop: percentage.

Undesirable for PCA to depend on something as arbitrary as scale $\Rightarrow$ scale each variable to have same sd.

UNLESS: all variables are measured on the same units $\Rightarrow$ might not want to scale them.

2.3 Uniqueness

Each principal component loading vector is unique, up to a sign flip.

⇒ software should result in same pr. comp. loading vectors, but sign might flip.

Flipping signs has no effect since direction doesn't change.

Similarly, the score vectors are unique up to a sign flip.

$$Var(Z) = Var(-Z)$$

2.4 Proportion of Variance Explained

We have seen using the `USArrests` data that e can summarize 50 observations in 4 dimensions using just the first two principal component score vectors and the first two principal component vectors.

Question:

How much information in a given data set is lost by projecting the observations onto 1st 2 Prin comp vectors?

More generally, we are interested in knowing the *proportion of vriance explained (PVE)* by each principal component.

Total variance in data set: $\sum_{j=1}^{p} Var(X_j) = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2$

Variance explained by m^{th} prin. comp: $\frac{1}{n} \sum_{i=1}^{n} z_{im}^2 = \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{jm} x_{ij} \right)^2$

PVE by m^{th} prin comp: $\dfrac{\sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{jm} x_{ij} \right)^2}{\sum_{j=1}^{p} \sum_{i=1}^{n} x_{ij}^2}$ (positive quantity)

Cumulative PVE for 1st M components: sum PVE first M.

2.5 How Many Principal Components to Use

In general, a $n \times p$ matrix $\boldsymbol{X}$ has $\min(n-1, p)$ distinct principal components.

We are probably not interested in all of them.

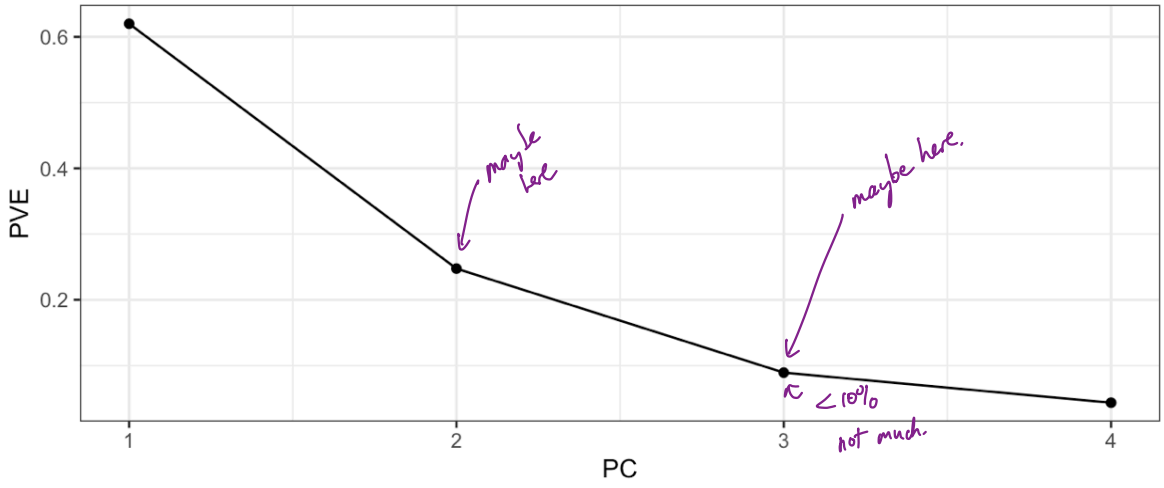
Rather, we would like to just use the first few principal components in order to visualize or interpret the data.

We want smallest # possible to get a good understanding of our data.

How many?

No one simple answer.

We typically decide on the number of principal components required by examining a *scree plot*.



looking for an "elbow" where plot drops sharply.

This is ad hoc, but the question of how many is "enough" is not well defined.

depends on problem, the data, and your goals.

unsupervised EDA

Usually plot first two PCs and look for "interesting" patterns. If there are none, probably won't be interesting in later components.
If first two are interesting, keep looking!

For supervised PCR

There is a good way to choose # of components: CV.

2.6 Other Uses for Principal Components

We've seen previously that we can perform regression using the principal component score vectors as features for dimension reduction.

Many statistical techniques can be easily adapted to use the $n \times M$ matrix whose columns are the first $M << p$ principal components. Instead of the full $n \times p$ dataset X

e.g. other types of regression, classification, clustering (next)

This can lead to *less noisy* results.

Since usually the signal is concentrated in first few PCs.

3 Clustering

Clustering refers to a broad set of techniques for finding *subgroups* in a data set.

We seek to portion observations into distinct groups so that

- observations within a group are similar
- observations in different groups are dissimilar

need to define
Depends on the domain!

For instance, suppose we have a set of n observations, each with p features. The n observations could correspond to tissue samples for patients with breast cancer and the p features could correspond to measurements, collected for each tissue sample.

- clinical measurements, e.g. tumor grade or stage.
- gene expression measurements.

diverse in character

We may have reason to believe there is heterogeneity among the n observations.

e.g. different unknown subtype of cancer.

This is *unsupervised* because

We are trying to discover structure (distinct clusters)

This is different from a supervised problem which has a goal of prediction.

Both clustering and PCA seek to simplify the data via a small number of summaries.

- PCA – find a low dimensional representation of observations that explain a good fraction of variance.
- Clustering – find homogenous subgroups among observations.

Since clustering is popular in many fields, there are many ways to cluster.

We focus on 2 best known clustering approaches

- K-means clustering

We seek to partition the observations into a pre-specified # of clusters.

- Hierarchical clustering

We do not know in advance how many clusters we want.

Obtain clusterings for $1, \ldots, n$ clusters and view these in a dendrogram.

In general, we can cluster (1) observations on the basis of features or we can cluster (2) features on the basis of observations.

(1) identify subgroups among observations

(2) discover subgroups among features.

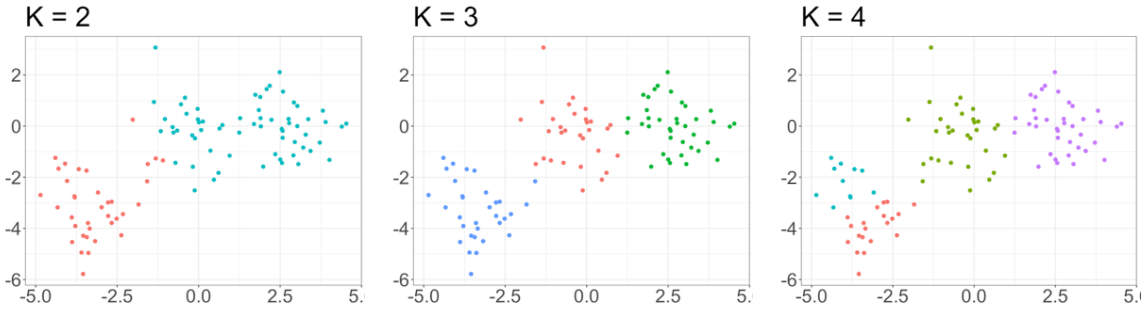
We will focus on (1), but to perform (2) just transpose data matrix X.

3.1 K-Means Clustering

Simple and elegant approach to parition a data set into K distinct, non-overlapping clusters.

We must first specify how many clusters K, then K-means assigns each observation to one of the clusters.

e.g. n = 100 observation clustering into k clusters using p = 2 features.



The K-means clustering procedure results from a simple and intuitive mathematical problem. Let $C_1, \ldots, C_K$ denote sets containing the indices of observations in each cluster. These satisfy two properties:

e.g. if observation i is in cluster k, $i \in C_k$

1. $C_1 \cup C_2 \cup \cdots \cup C_K = \{1, \ldots, n\}$.
   each observation belongs to one of the K clusters.

2. $C_k \cap C_{k'} = \emptyset \quad \forall \; k \neq k'$
   the clusters are non overlapping.

Idea: good clustering is one for which the within-cluster variation is small as possible.

The *within-cluster variation* for cluster C_k is a measure of the amount by which the observations within a cluster differ from each other.

Call this $W(C_k)$

Then we want to solve the problem:

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} W(C_k) \right\}$$

← want to partition observations into K clusters s.t. total within-cluster variation is minimized.

To solve this, we need to define within-cluster variation.

Many ways we could do it

Most common way: squared euclidean distance

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

↑ # obs in cluster C_k.

This results in the following optimization problem that defines K-means clustering:

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

← objective.

This is very difficult to solve exactly: $\approx K^n$ ways to partition n obs into K clusters.

A very simple algorithm has been shown to find a local optimum — "pretty good solution" — to this problem:

1. randomly assign a number from 1 to K to each observation
   these are initial cluster assignments for observations.
2. Iterate until cluster assignments stop changing:
   (a) For each of the K clusters, compute the cluster centroid
   ← vector of p feature means for observations in each cluster.
   (b) assign each observation to closest centroid cluster.
   ← euclidean distance.

Algorithm is guaranteed to decrease value of objective at each step.

When cluster assignment stops changing, guaranteed to have reached a local minimum
↳ not global! ⇒ clustering depends on initialization (step 1.).

⇒ run algorithm multiple times from different initializations and choose clustering w/ smallest objective function.

Problem: we must choose 'K'! (e.g. "Dunn index").

3.2 Hierarchical Clustering

One potential disadvantage of K-means clustering is that it requires us to specify the number of clusters K. *Hierarchical clustering* is an alternative that does not require we commit to a particular K.

ahead of time.

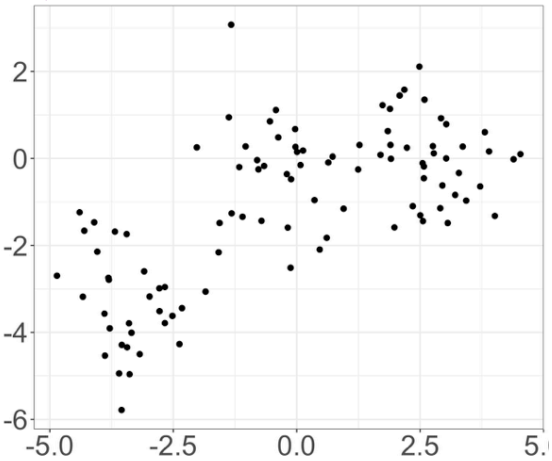
hierarchical clustering results in a tree-based representation of our observations.

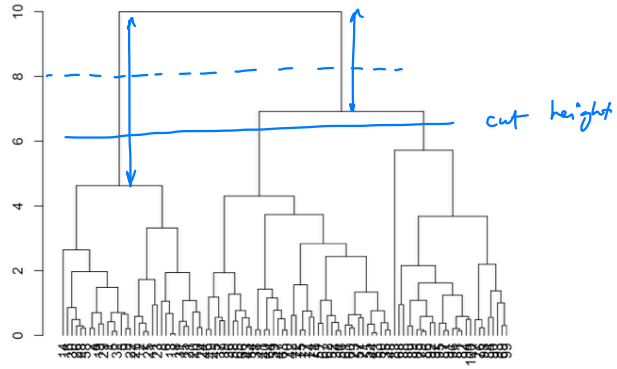
We will discuss *bottom-up* or "*agglomerative*" clustering. clusters getting bigger.

start w/ each observation is its own cluster and merge clusters/observations until all observations are in a single clustering (n clusters → 1 cluster).

3.2.1 Dendrograms

some simulated data as before n=100, p=2



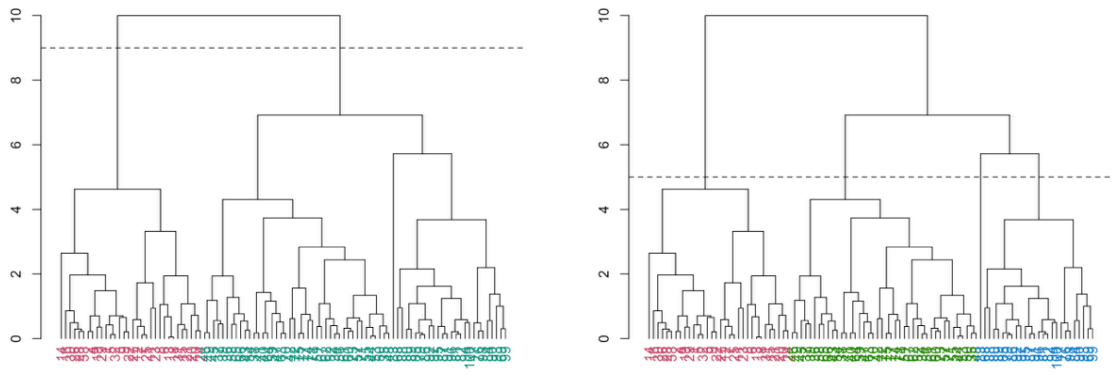


Each *leaf* of the dendrogram represents one of the 100 simulated data points.

As we move up the tree, leaves begin to fuse into branches, which correspond to observations that are similar to each other.

For any two observations, we can look for the point in the tree where branches containing those two observations are first fused.

How do we get clusters from the dendrogram?

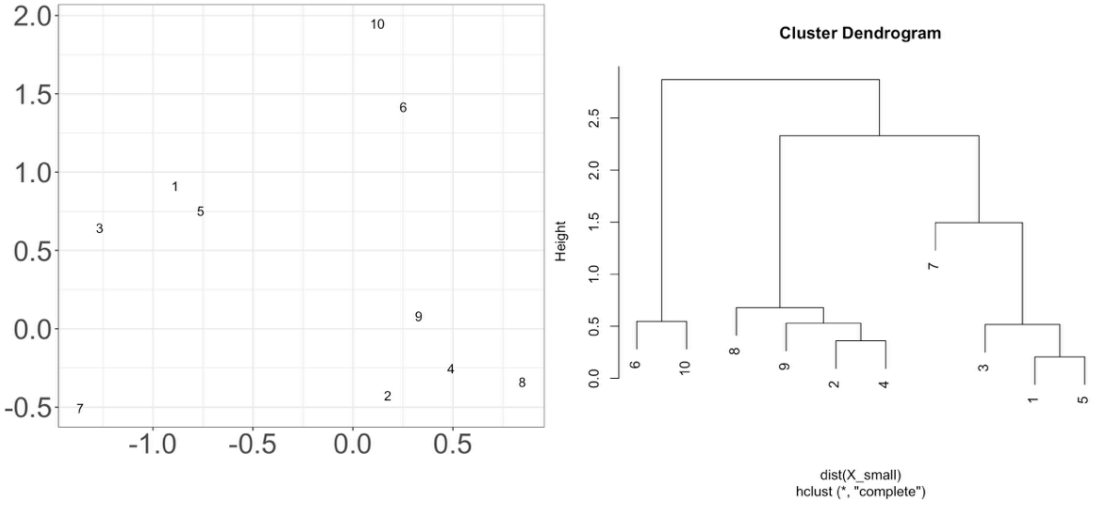


The term *hierarchical* refers to the fact that clusters obtained by cutting the dendrogram at a given height are necessarily nested within the clusters obtained by cutting the dendrogram at a greater height.

3.2.2 Algorithm

First, we need to define some sort of *dissimilarity* metric between pairs of observations.

Then the algorithm proceeds iteratively.



More formally,

One issue has not yet been addressed.

How do we determine the dissimilarity between two clusters if one or both of them contains multiple observations?

1.

2.

3.

4.

3.2.3 Choice of Dissimilarity Metric

3.3 Practical Considerations in Clustering

In order to perform clustering, some decisions should be made.

-
-
-

Each of these decisions can have a strong impact on the results obtained. What to do?