Introduction to Statistical Data Science

Dr. Francois-Xavier Briol
Department of Statistical Science,
UCL

UNSUPERVISED LEARNING

Outline

- Unsupervised learning is often ill-defined, but it is basically the problem of inferring “interesting features” in the distribution of one variable or several variables.

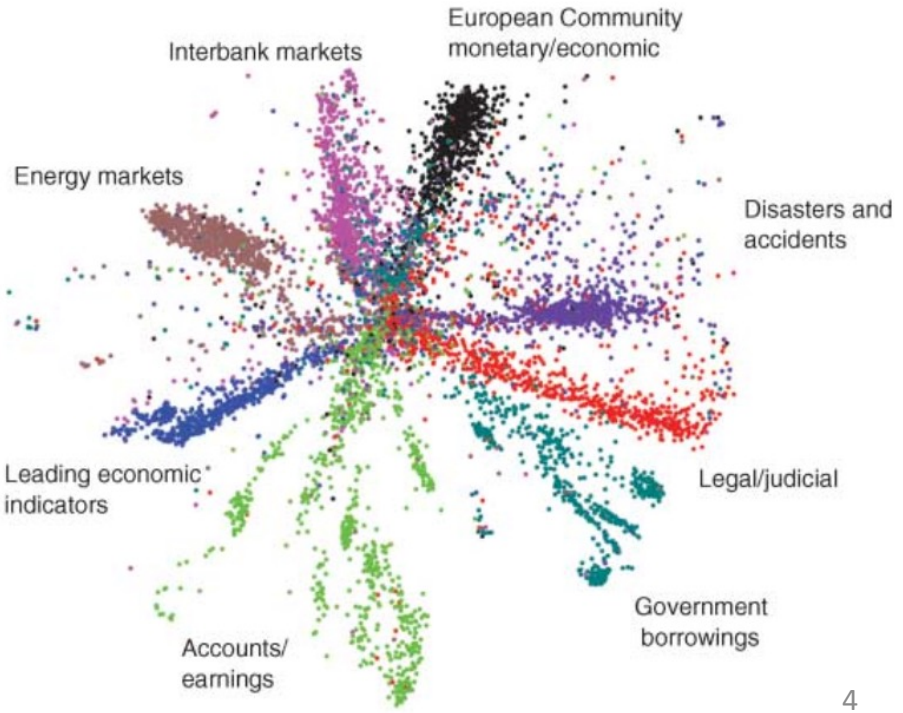
- There is no outcome variable to be predicted.

A Machine Learning Example

- Retrieval by content: find me stories similar to a given story
  - Story == news article == text document

Two-dimensional representation of a corpus of news articles, inferred from nothing but raw text. Labels shown were assigned by a human, but not used in the method!

"Similarity" is now given by Euclidean distance in this space.



Hinton and Salakhutdinov (2006). "Reducing the Dimensionality of Data with Neural Networks". *Science*.

More Standard Statistical Applications

- Is this data point “atypical”? What is typical and what is not? I need a pdf for that.
- Which dependencies exist in my system? Can I infer which proteins in a cell “talk to each other” directly?
- I have too many variables. Can I “preserve information” in my data using fewer variables, regardless of a possible prediction problem?
- I may only see part of my data: I’m missing some data that I postulate to exist but I cannot measure.

Outline

- Nonparametric Density estimation
- Dimensionality reduction
- Latent variable and mixture models, with applications to clustering

Density estimation
DENSITY ESTIMATION

Models and Likelihoods

- The problem of **density estimation** is the problem of estimating the probability distribution of a population.
  - Sometimes the term “density estimation” is used even if the data is discrete (by machine learning people, mostly)

- We saw likelihood functions, mainly in the context of regression and some simple models. Like this:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(x-\mu)^2}{2\sigma^2} \right\}$$

- We can use models to characterize observations as being “likely” or not.

Example: Outlier Detection

- Let's resurrect our old NHANES data!
  - An American "Health and Nutrition Examination Survey".
- Is an individual of height 1.95m "too unusual" for the population we are studying?
  - For instance, if we were looking at kids of a particular age, a height that is too unusual might indicate problems with the data.

Gaussian Density Estimation

- We fit parameters by maximum likelihood.

$$\hat{\mu} \approx 168, \quad \hat{\sigma} \approx 10.2$$

- Then we calculate tail area probabilities:

$$P(Y > 190; \mu = 168, \sigma = 10.2) \approx 0.02$$

- Is this an unlikely event? This is a problem-dependent conclusion (just like p-values). Regardless of it, you can (and must) explain how you got to that probabilistic statement.

Estimate using the Empirical Dist.

- Don't trust the Gaussian? What about using the empirical distribution?

$$P(Y > 190) \approx \frac{\#\text{people in sample with height greater than } 190}{n} \approx 0.0001$$

- The difference in estimate may or may not matter, depending on the application (see also, "value at risk").

- Avoiding the Gaussianity assumption is nice, but the empirical distribution itself has its shortcomings: it all boils down to the bias-variance trade-off.
  - Gaussian: (possibly) "high" bias, "low" variance
  - Empirical distribution: low bias, "high" variance

Nonparametric Density Estimation

- Estimating P(Y>190) using the empirical distribution in an example of a nonparametric estimate.
- Nonparametric refers to the fact that we do not use a parametric model; as a result, nonparametric methods tend to be much more widely applicable.
- **Warning:** this generality also means that they might not be as efficient when in cases where you actually know the parametric form of your density!
- We will now see two examples of nonparametric density estimation methods: histograms and kernel density estimates.

DENSITY ESTIMATION WITH HISTOGRAMS

Alternative: the Histogram Estimate

- Break the data in bins, calculate frequency of points falling in each bin. This will smooth the empirical distribution:
  - few bins = lot of smoothing (R demo)

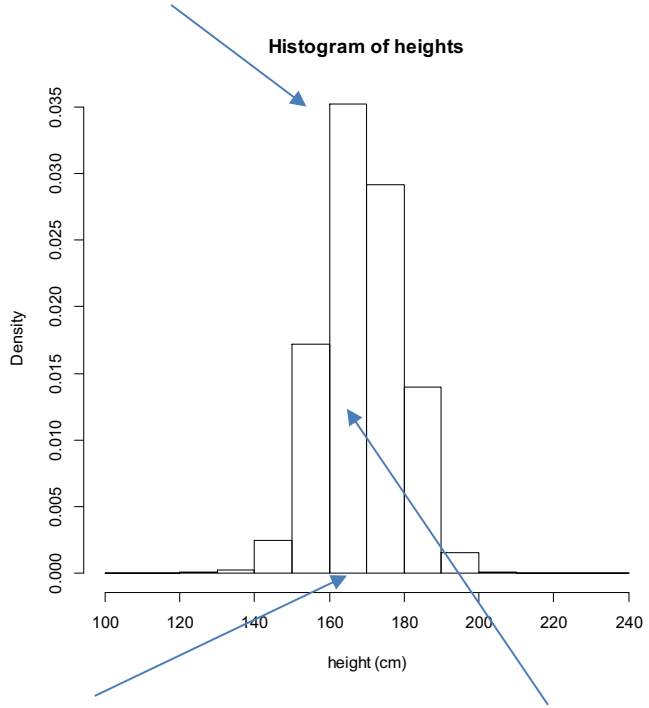
- There is no likelihood function: divide the range of your distribution into m bins of length h.
  - For simplicity, assume we know the maximum and minimum of our space (say 0 and 2.5 in the NHANES data, so $h = 2.5 / m$)
  - For each bin B_k, compute the proportion of points which fall in B_k.

Example: NHANES Height

$$\hat{p}_k = \frac{\#\text{points falling in bin } B_k}{n}$$

$$\hat{p}(x) = \begin{cases} \hat{p}_1/h & x \in B_1 \\ \hat{p}_2/h & x \in B_2 \\ \dots & \\ \hat{p}_m/h & x \in B_m \end{cases}$$

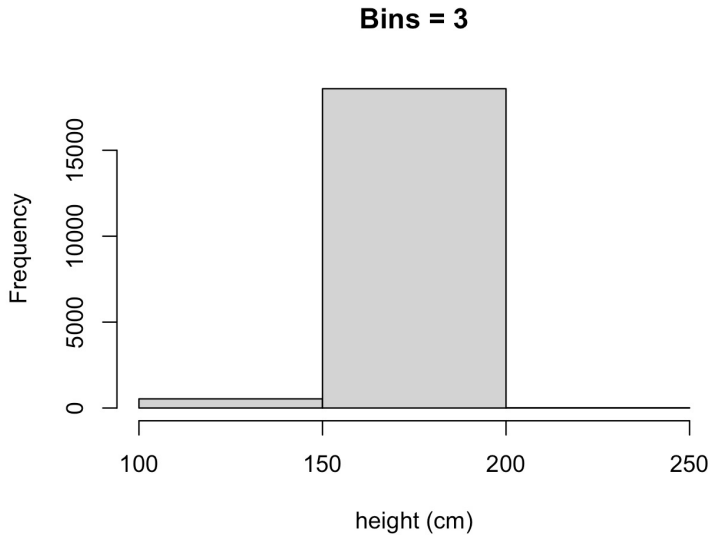
height is $\hat{p}_k/h$

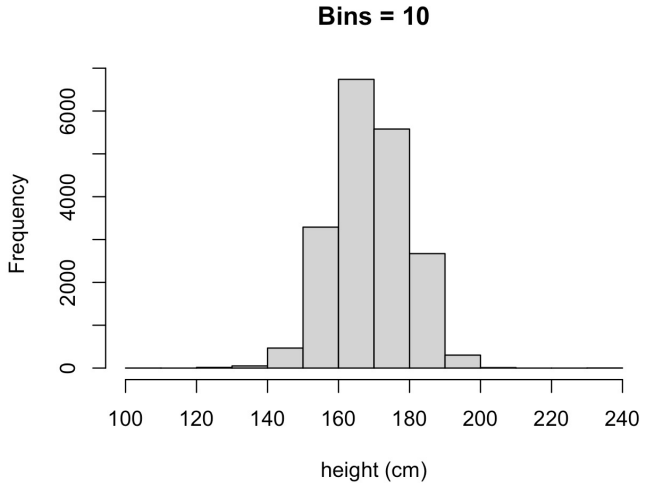


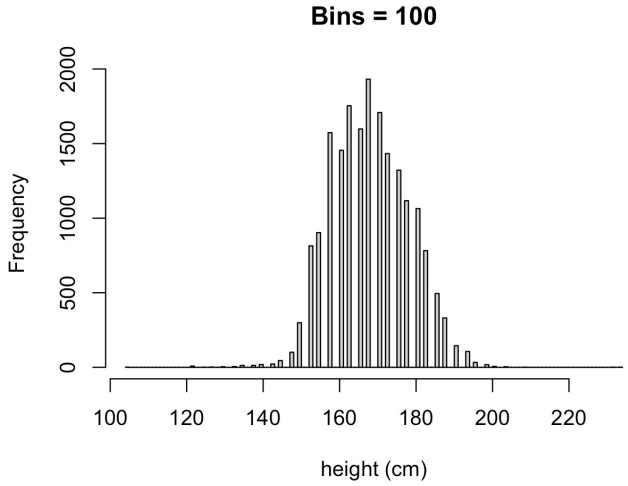
width is *h = 10*

area is $\hat{p}_k/h \times h = \hat{p}_k$

Number of Bins







Choosing *m*

- Cross-validation, duh!
- But what is the measure we are optimising? There is no outcome variable, and no likelihood.
- Option: mean-squared on the density.

$$
\begin{aligned}
L(m) &= \int (\hat{p}(x) - p(x))^2 dx \\
&= \int \hat{p}^2(x) dx - 2 \int \hat{p}(x) p(x) dx + \int p^2(x) dx \\
&\approx \int \hat{p}^2(x) dx - 2 \times \frac{1}{n} \sum_{i=1}^{n} \hat{p}_{(-i)}(x^{(i)}) + \text{constant}
\end{aligned}
$$

Does not depend on *m*

Easy to solve

This substitutes *p*(*x*).

Histogram fit without point (*i*)

Confidence ~~Intervals~~ Bands

- We could look at each value *x* and build a confidence interval for *p*(*x*), but more generally we would like to bound the entire function at once, regardless of *x*.

- That is, find some *l*(*x*), *u*(*x*) such that

$$P(l(x) \leq p(x) \leq u(x) \text{ for all } x) \geq 1 - \alpha$$

for a given α, where the probability is over the datasets that can be used to build *l*(*x*) and *u*(*x*).

Confidence Bands

- For a fixed number *m* of bins, we can get a confidence band for a "histogramized" version of *p*(*x*), which we will call

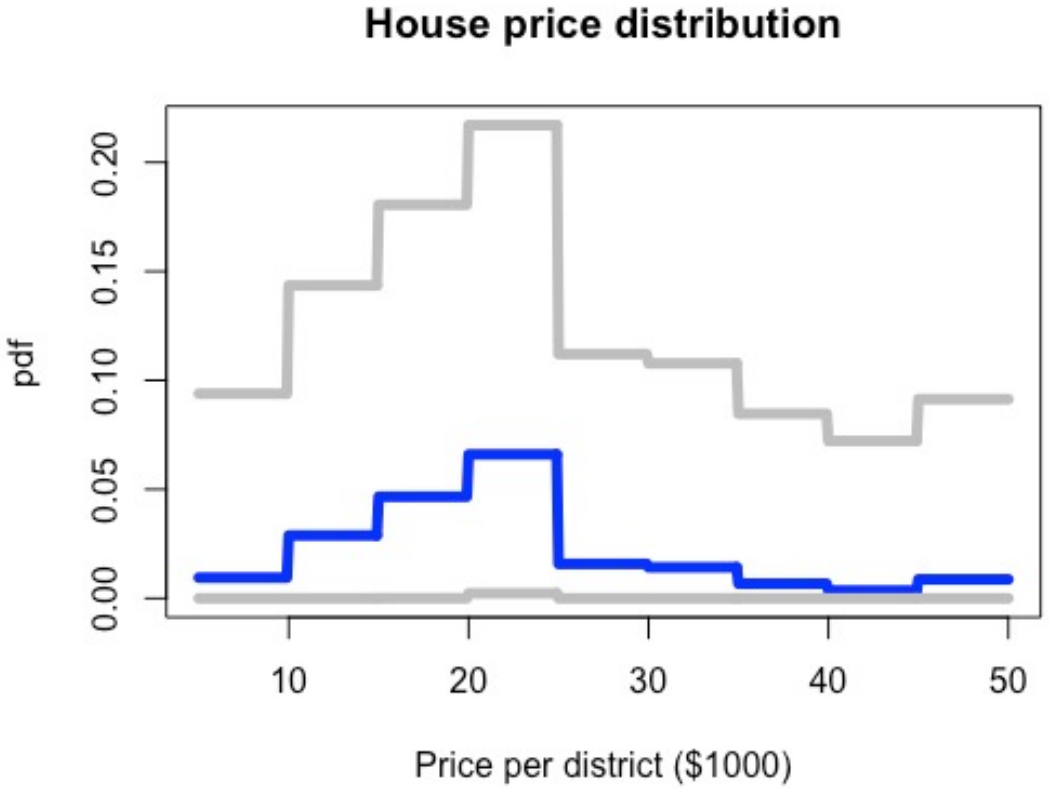
$$\tilde{p}(x) \equiv \frac{p_k}{h} \text{ for } x \in B_k, \text{ where } p_k \equiv \int_{B_k} p(x)dx$$

- Bizarre! What is this? I want my confidence band for *p*(*x*)!
  - There are limits of what we can do without a likelihood.
  - The (informal) idea is that as the data grows, cross-validation will pick higher and higher values for *m*, so in principle we can get arbitrarily close to *p(x)* if we feed in "enough" data.
  - The amount of uncertainty may be humbling (R demo).

(See Chapter 20 of Wasserman, if you want a formula and gory details.)

House Price Distribution

- Dataset of House prices in various districts Chicago.



KERNEL DENSITY ESTIMATION

Kernel Density Estimation

- Histograms are simple, but discontinuous. **Kernel density estimation** provides a smoother alternative.

- Function *K* is a kernel if:

$$\begin{aligned} K(x) &\geq 0 \\ \int K(x)dx &= 1 \\ \int xK(x)dx &= 0 \\ \int x^2K(x)dx &> 0 \end{aligned}$$

Basically, a density function of zero mean and positive variance.

Estimate

- Given a kernel, the estimator requires a choice of scaling parameter (called **bandwidth**):

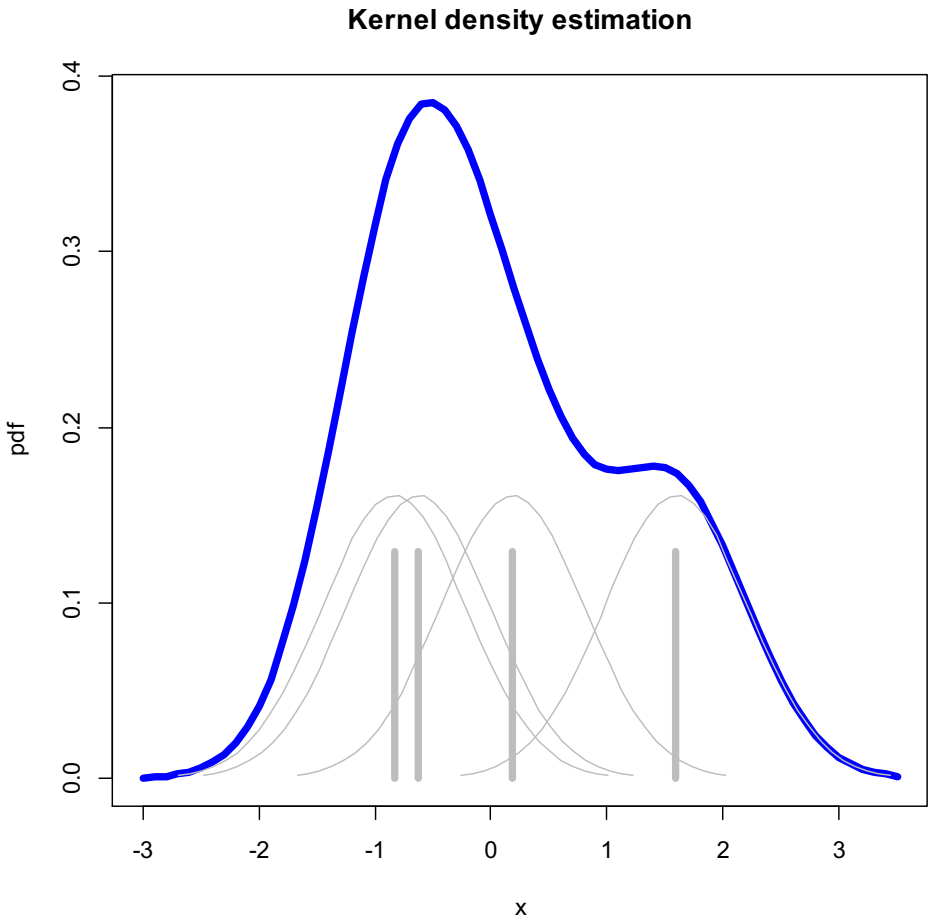
$$\hat{p}(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K\left(\frac{x - x^{(i)}}{h}\right).$$

- One possible choice of kernel is the standard Gaussian:

$$K(z) = \frac{1}{\sqrt{2\pi}}e^{-x^2/2}$$

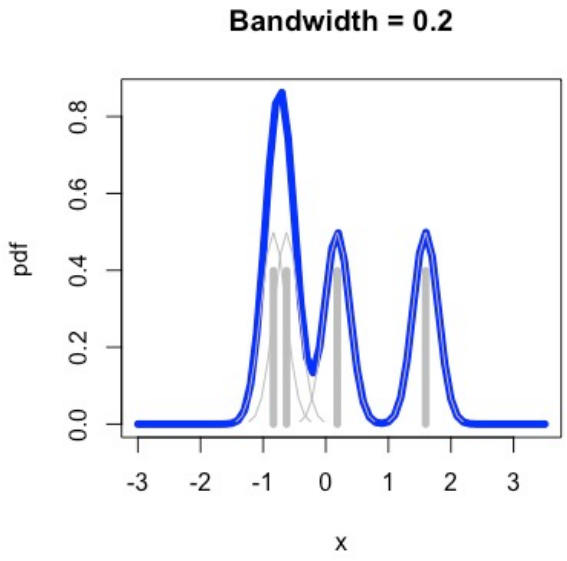
Example

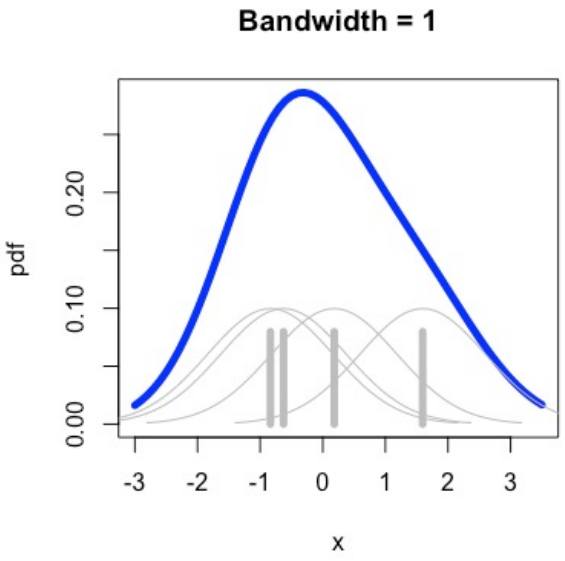
- Gaussian kernel, selection of bandwidth by cross-validation.

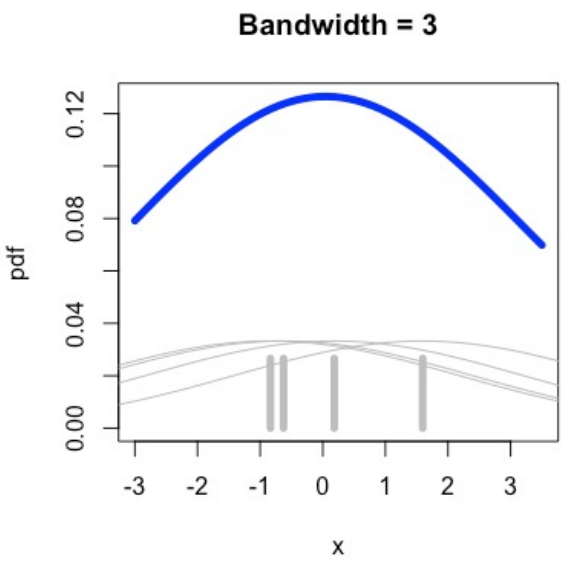


R demo: effect of bandwidth selection

Example


Bandwidth = 0.2
pdf
0.0
0.2
0.4
0.6
0.8
-3
-2
-1
0
1
2
3
x


Bandwidth = 1
pdf
0.00
0.10
0.20
-3
-2
-1
0
1
2
3
x


Bandwidth = 3
pdf
0.00
0.04
0.08
0.12
-3
-2
-1
0
1
2
3
x

Confidence Bands

- Same issue as histograms: confidence for a given bandwidth *h*, meaning confidence over a smoothed version of the truth.

$$\tilde{p}(x) \equiv \int \frac{1}{h} K\left(\frac{x-u}{h}\right) p(u) du$$

Actual density

- As in the histogram case, the idea is that cross-validation gives $h \to 0$ as $n \to \infty$.

(Again, for those interested, Chapter 20 of Wasserman gives details.)

Example

- Comparison of density estimate with histogram (left) and kernel density estimate (right).

MULTIVARIATE DENSITY ESTIMATION

Multivariate Density Estimation

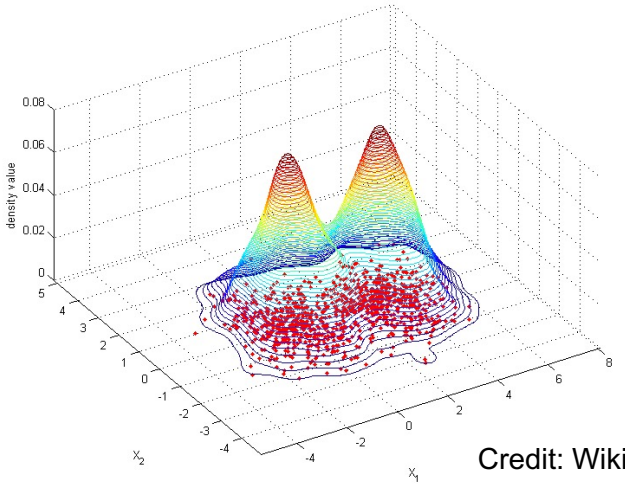
- The kernel idea again applies to *p*-dimensional vectors **x**:

$$\hat{p}(\mathbf{x}) \equiv \frac{1}{n} \sum_{i=1}^{n} K_h(\mathbf{x} - \mathbf{x}^{(i)})$$

$$K_h(\mathbf{x} - \mathbf{x}^{(i)}) = \frac{1}{nh_1h_2 \ldots h_p} \left\{ \prod_{j=1}^{p} K\left(\frac{x_j - x_j^{(i)}}{h_j} \right) \right\}$$

- Is there a catch?

Of course there is a catch.



Credit: Wikipedia

The Curse of Dimensionality

- On your right, the sample sizes required to ensure, as the dimension increases, a MSE less than 0.1 at **0**, when data comes from a multivariate Normal, but you do not assume it.

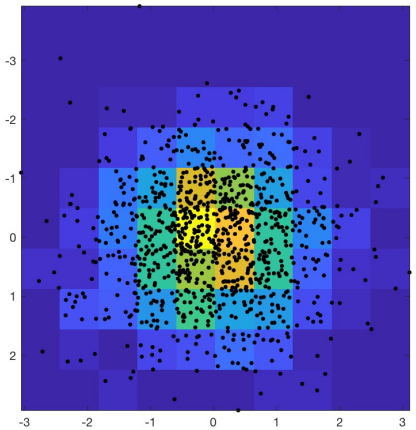
- For this problem, this is like saying that having 800,000+ observations in a ten-dimensional problem is as good as having 4 in one dimension!

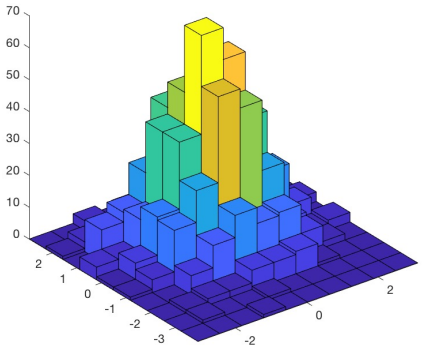
Dimension	Sample Size
1	4
2	19
3	67
4	223
5	768
6	2790
7	10,700
8	43,700
9	187,000
10	842,000

(Wasserman, Chapter 20)

Multivariate Histograms

- Similarly, we could consider multivariate histograms.
- This is equivalent to constructing one histogram per dimension. In that case, many of the intervals may end up having no data points!





Credit: MathWorks

Modelling Joint Distributions

- You will hardly be able to learn some big distribution $P(X_1, X_2, \ldots, X_p)$ with a purely nonparametric approach.
- A combination of domain knowledge and off-the-shelf ideas can go a long way.
- In what follows, we describe:
  - An example of a recipe for defining joint distributions.
  - The great classical example: the multivariate Gaussian in more detail.

Example from Exercise Sheet 1

- A blast from the past:

(h) Suppose that the number of distinct uranium deposits in a given area is a Poisson random variable with parameter $\mu = 10$. If, in a fixed period of time, each deposit is independently discovered with probability 1/50, find the probability that (i) exactly one, (ii) at least one and, (iii) at most one deposit is discovered during that time.

(Exercise sheet #1)

- We have *D* as the number of deposits and *Y* as the number of deposits discovered.

$$p(d, y) = p(d)p(y \mid d)$$

Maximum Likelihood

- If we have a dataset ($d^{(1)}$, $y^{(1)}$), … ($d^{(n)}$, $y^{(n)}$), how do we do maximum likelihood? Same old.

$$\log \prod_{i=1}^{n} p(d^{(i)}, y^{(i)}) \quad = \quad \log \prod_{i=1}^{n} p(d^{(i)}; \theta_1) p(y^{(i)} \mid d^{(i)}; \theta_2)$$

$$\sum_{i=1}^{n} \log p(d^{(i)}; \theta_1) + \sum_{i=1}^{n} \log p(y^{(i)} \mid d^{(i)}; \theta_2)$$

We can optimise these separately, using the standard stuff

In some models, this could be a GLM or a GAM, for instance

- This can be generalised to more variables.

“Canonical” Models

- Where could we find the “natural” (“canonical”) generalisation of models we have seen?
  - Multivariate Binomial?
  - Multivariate Poisson?
  - Multivariate Gaussian?...

- Surprisingly, few exist! No single agreeable way of defining a multivariate Poisson, for instance.

Multivariate Binomial

- Hinted at when we mentioned contingency tables.

- Say you have a dataset of three variables
  - X_1 = did customer buy milk? Yes/No (0/1)
  - X_2 = did customer buy bread? Yes/No (0/1)
  - X_3 = did customer buy coffee? Yes/No (0/1)

- Which parameters?
  - Literally, for each p(x_1, x_2, x_3), have a θ_{x1x2x3}.
  - Notice that

$$0 \leq \theta_{ijk} \leq 1, i \in \{0,1\}, j \in \{0,1\}, k \in \{0,1\} \qquad \sum_{i=0}^{1}\sum_{j=0}^{1}\sum_{k=0}^{1}\theta_{ijk} = 1$$

Multivariate Gaussian

- Our main focus in this chapter (hinted at before in Chapter 1 and Exercise Sheet #4, we never got in details).

- The pdf of a *p*-dimensional Gaussian consists of a *p x* 1 **mean vector** μ and a *p x p* **covariance matrix** $\sum$.

$$p(x_1, \ldots, x_p) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma}|^{1/2}} e^{-(\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)/2}$$

Example: Bivariate Gaussian with Zero Mean and Unit Variances

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Example: zero means

Variances

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix} = \begin{bmatrix} 1 & \sigma_{12} \\ \sigma_{21} & 1 \end{bmatrix}$$

Covariances: they are always symmetric.

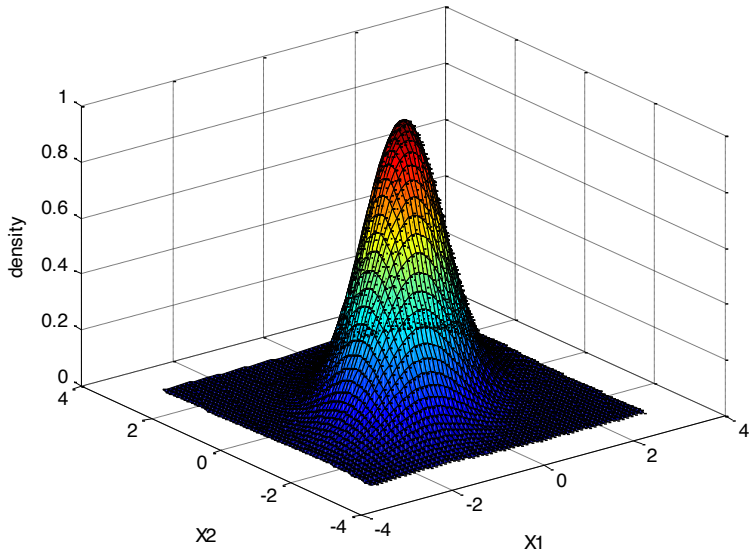
Explicit representation of symmetry.

Example: unit variances

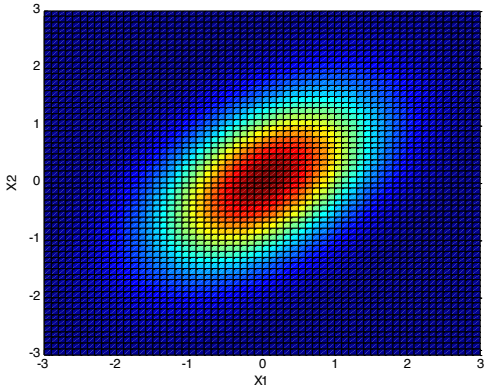
- Notice that $\sum$ is a symmetric matrix:
  - $\sigma_{jk} = \sigma_{kj}$ for any two variables X_j and X_k.

Examples

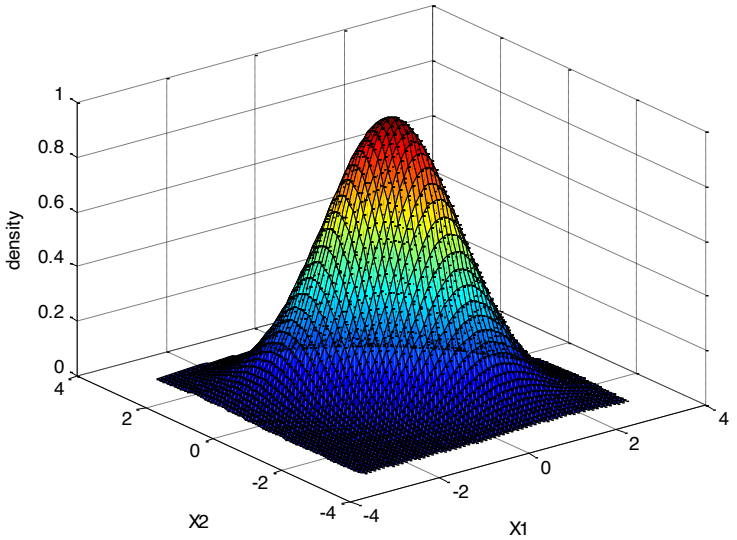
Zero mean, unit variance, $\sigma_{12} = 0.5$

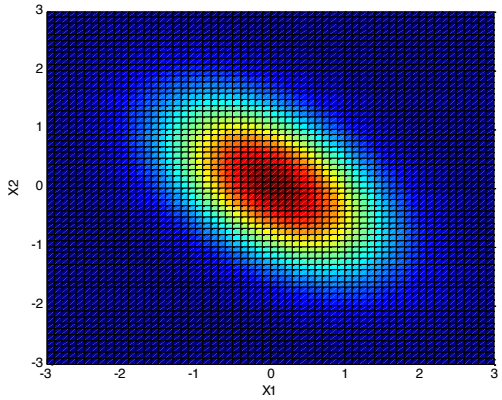


Contour plot



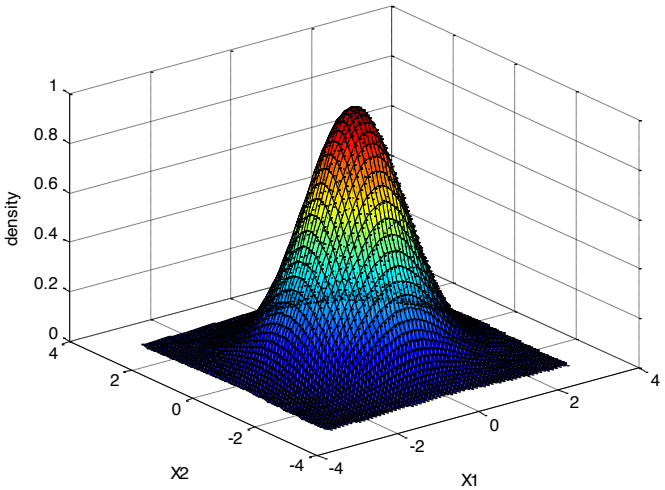
Zero mean, unit variance, $\sigma_{12} = -0.5$

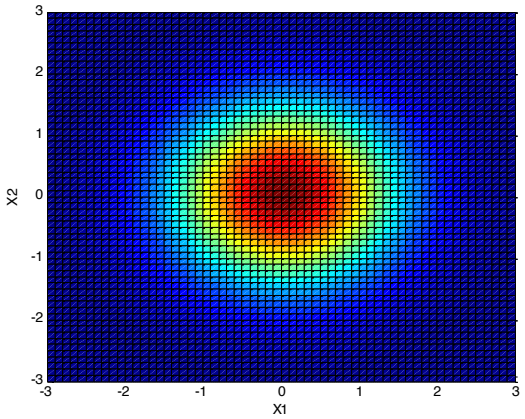




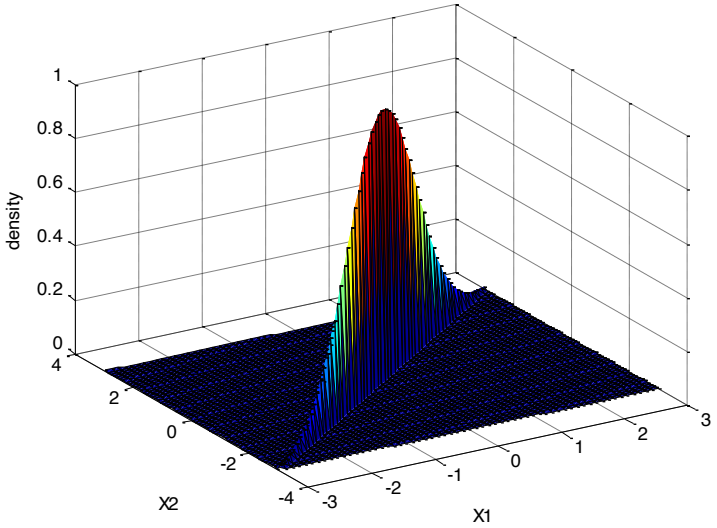
Examples

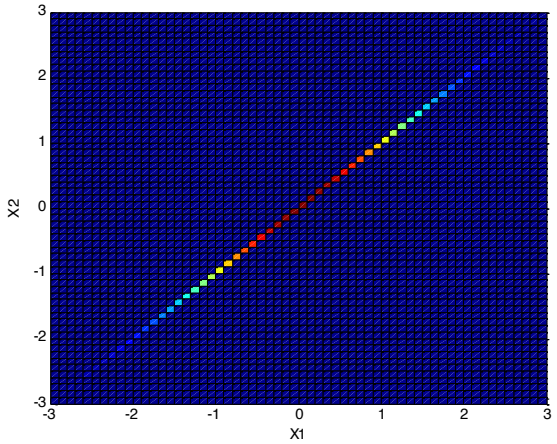
Zero mean, unit variance, $\sigma_{12} = 0$





Zero mean, unit variance, $\sigma_{12} \cong 1$





Interpretation and Estimation

- Just like variance, the name "covariance" applies both to the following particular summary of any distribution and parameters of a multivariate Gaussian:

$$\sigma_{ij} \equiv E[(X_i - \mu_i)(X_j - \mu_j)]$$

- Maximum likelihood for a multivariate Gaussian gives

$$\hat{\mu}_j = \frac{1}{n} \sum_{i=1}^{n} x_j^{(i)}$$

$$\hat{\sigma}_{jk} = \frac{1}{n} \sum_{i=1}^{n} (x_j^{(i)} - \hat{\mu}_j)(x_k^{(i)} - \hat{\mu}_k)$$

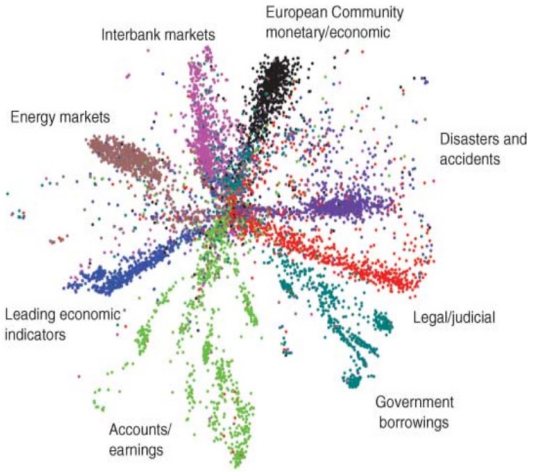
Summary

- Density estimation is a useful tool to answer questions about our data.
- Nonparametric approaches are very useful as they do not require us to fix a restrictive parametric model
- …but they will struggle in high-dimensions.
- In those cases, it might be better off to revert back to parametric models.

DIMENSIONALITY REDUCTION WITH PCA

Dimensionality Reduction

- "Represent the information" in your data with a smaller number of variables.
  - It is a type of data compression.



- Needed:
  - a way of quantifying the information preserved
  - a rule to establish a trade-off of compression vs information loss
  - a family of representations. For instance: compress *p* variables into 1 by a linear transformation.

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

Principal Component Analysis (PCA)

- One of the earliest methods for dimensionality reduction based on linear transformations.
- For all that follows, let's assume our given variables have zero (empirical) mean.
- Idea: maximise sample variance of the transformation, subject to it being "normalized".

$$\max_{\phi_{11},\ldots,\phi_{p1}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1} x_j^{(i)} \right)^2 \right\} \text{ subject to } \sum_{j=1}^{p} \phi_{j1}^2 = 1$$

Why?

Outcome

- So if we have a *n* x *p* dataset, we end up with a *n x* 1 dataset instead.

$$\mathbf{X} = \begin{bmatrix} X_1^{(1)} & X_2^{(1)} & \dots & X_p^{(1)} \\ X_1^{(2)} & X_2^{(2)} & \dots & X_p^{(2)} \\ \dots & \dots & \dots & \dots \\ X_1^{(n)} & X_2^{(n)} & \dots & X_p^{(n)} \end{bmatrix} \xrightarrow{\text{PCA, single variable}} \mathbf{Z} = \begin{bmatrix} Z_1^{(1)} \\ Z_1^{(2)} \\ \dots \\ Z_1^{(n)} \end{bmatrix}$$

- Values $Z_1^{(i)}$ are also called the **scores**, or **projections**, of the data.

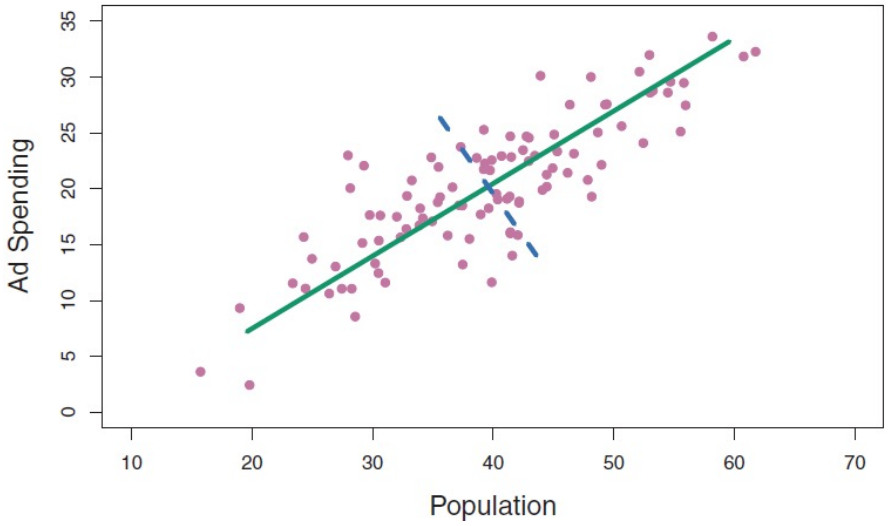
- **By construction**, this rule gives the single linear transformation of your data with maximum variance.

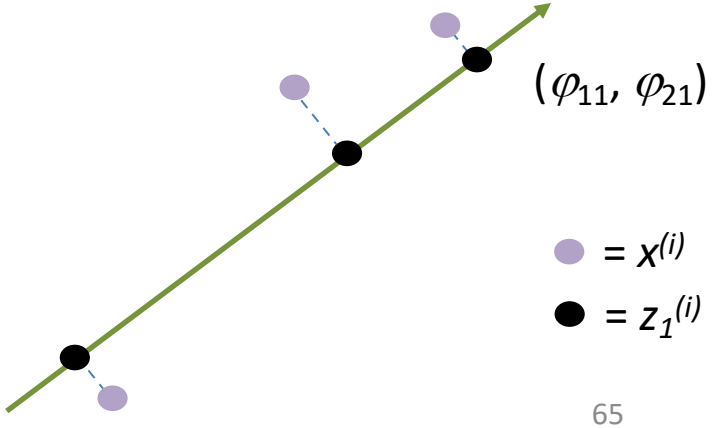
Interpretation

- Consider $p = 2$ with the *Advertising* data of ISLR. Let's reduce (*Population*, *Ad Spending*) to a single variable.

- Essentially, we find a direction (vector φ_{11}, φ_{21}) so that **the projection of the data into it will be of maximum variance**.

(ISLR, Chapters 6 and 10)





Adding More Components

- It will follow the same structure

$$Z_2 = \phi_{12}X_1 + \phi_{22}X_2 + \cdots + \phi_{p2}X_p$$

but we will need it to be "as different as possible" from Z_1.

  - PCA uses correlation as a measure of "difference"

$$\max_{\phi_{12},\ldots,\phi_{p2}} \left\{ \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{j2}x_j^{(i)}\right)^2 \right\} \text{ subject to } \sum_{j=1}^{p}\phi_{j2}^2 = 1 \text{ and } \sum_{j=1}^{p}\phi_{j1}\phi_{j2} = 0$$

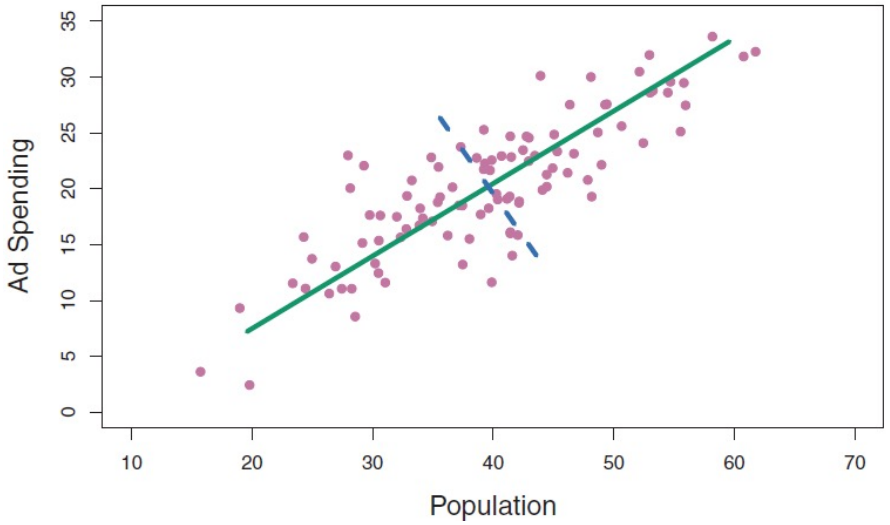
Vectors are perpendicular, this is enough for uncorrelated $z_1^{(i)}$ and $z_2^{(i)}$.

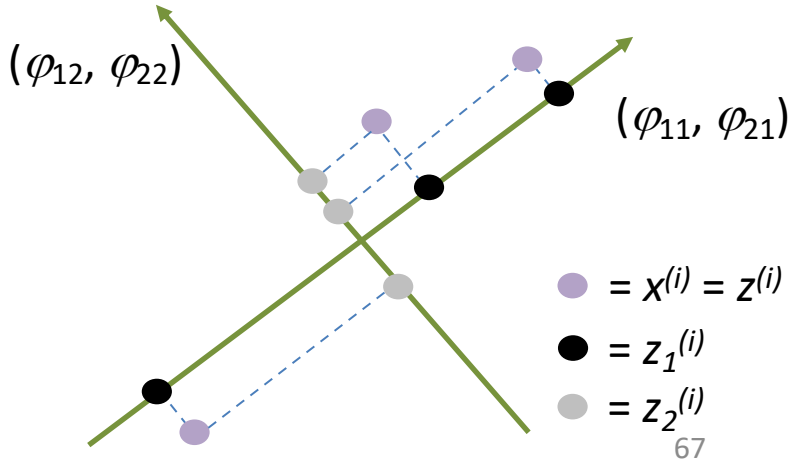
Interpretation

- This adds more dimensions to projection vector $\mathbf{z}^{(i)}$. With two dimensions, the new dataset is

$$\mathbf{Z} = \begin{bmatrix} Z_1^{(1)} & Z_2^{(1)} \\ Z_1^{(2)} & Z_2^{(2)} \\ \cdots & \cdots \\ Z_1^{(n)} & Z_2^{(n)} \end{bmatrix}$$

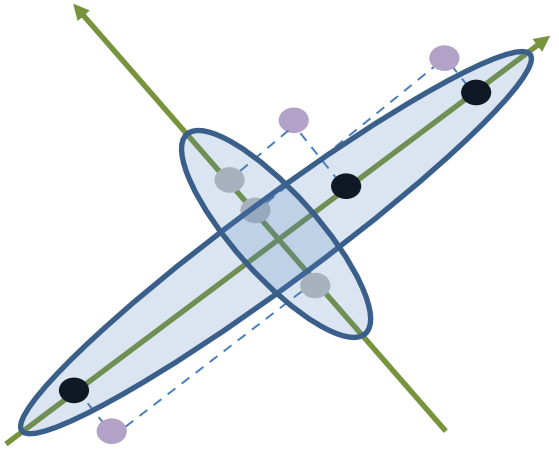
- If the original data was two-dimensional, then **Z** = **X**!





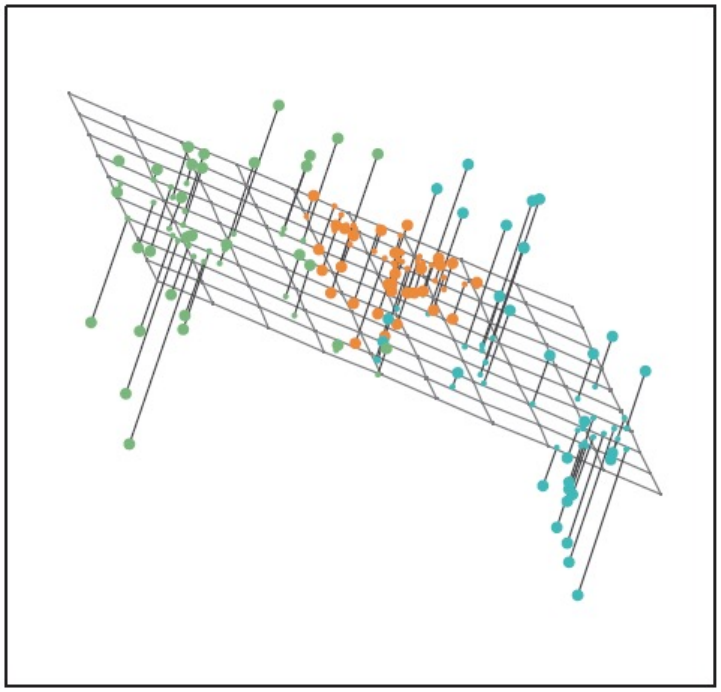
Interpretation

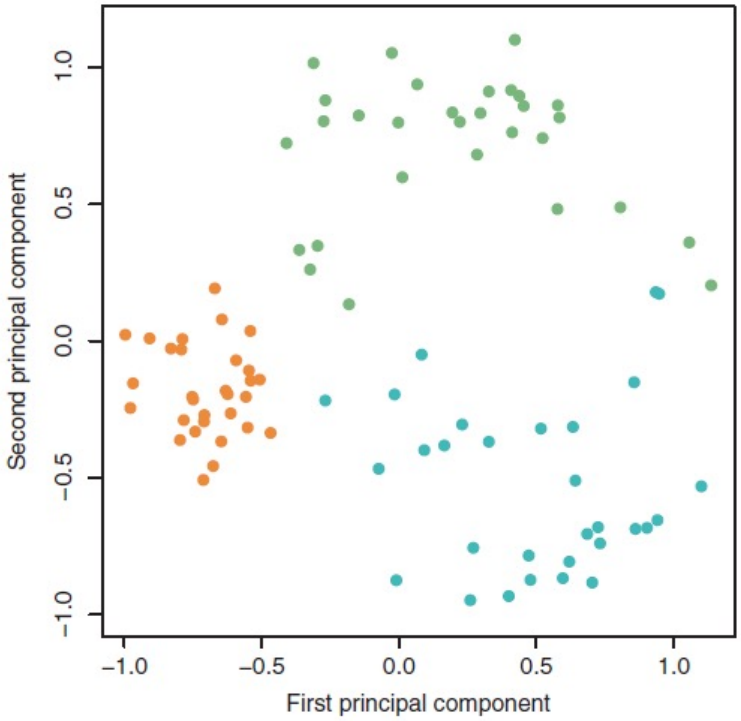
- Notice that the empirical variance around the points projected into the first principal component is greater than the one in the second.
- As it should be, by construction.
- With p variables, Z_3, Z_4, …, Z_p are defined in an analogous way: make each Z_m uncorrelated with all those preceding it.



Projection into 2 Components

- Synthetic data (ISLR, Chapter 10)





Example: USA Arrests Data (ISLR)

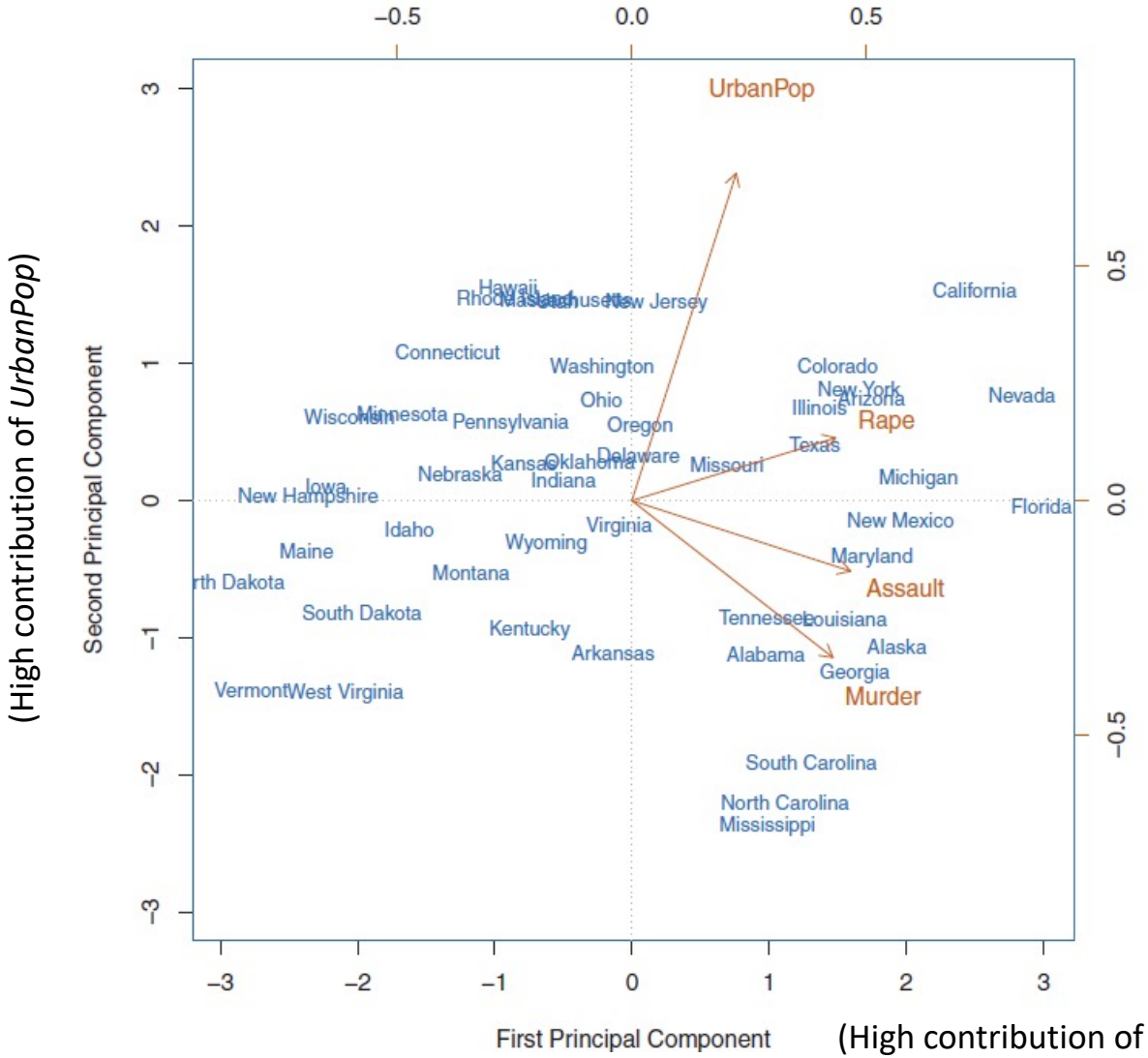
- For each of the $n = 50$ states in the US, this includes number of arrests per 100,000 residents for crimes of *Assault*, *Murder* or *Rape*.

- Also, percentage of population (*UrbanPop*) living in urban areas, per state.

- Goal: visualize how states differ.
  - scale the data to zero empirical mean/one standard deviation, do PCA with two components, get $n \times 2$ reduced data matrix **Z**, plot it.

- For each variable j, also plot the vector $(\varphi_{j1}, \varphi_{j2})$.

Result

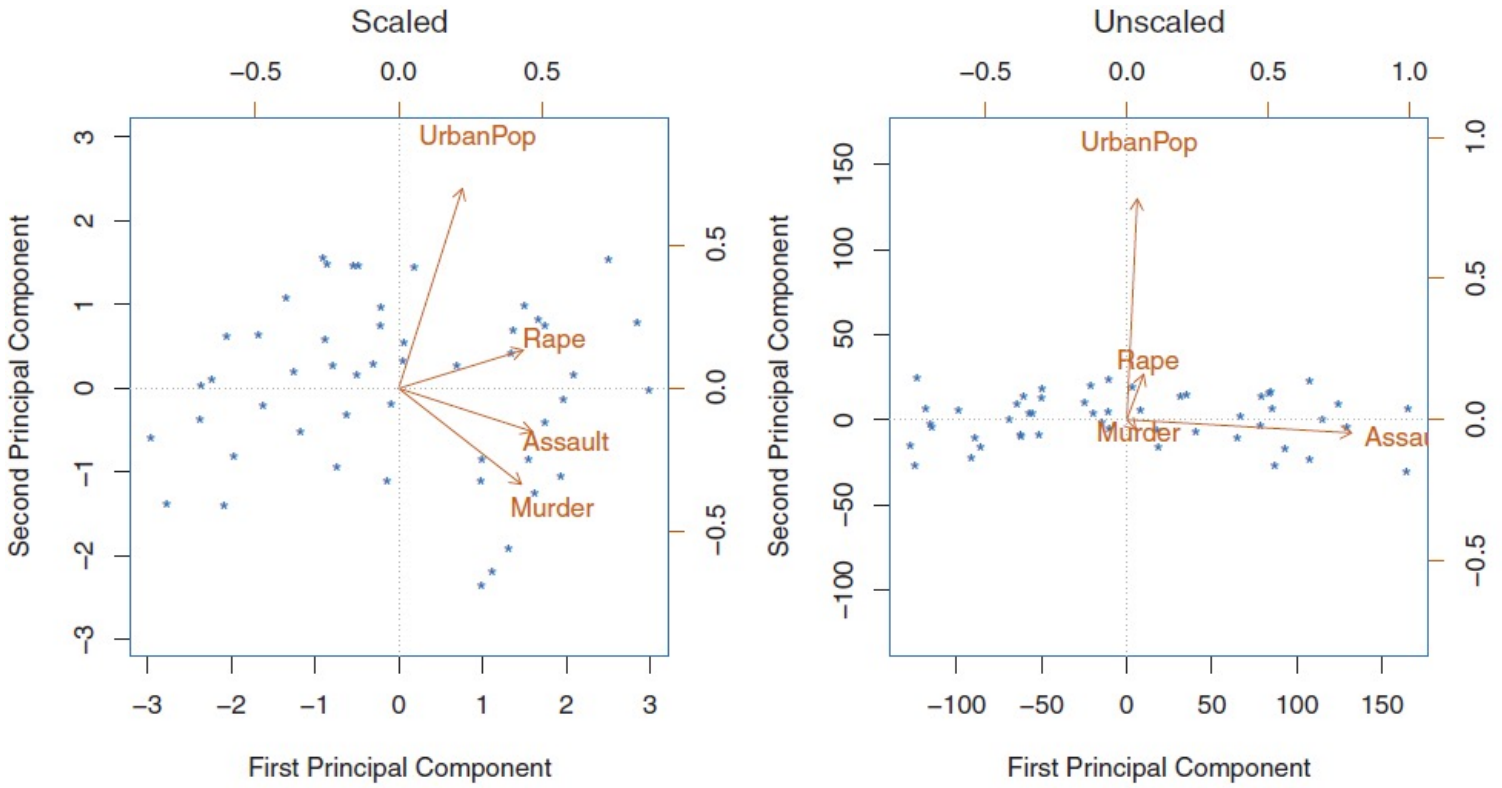


(High contribution of *UrbanPop*)

(High contribution of combination of crimes)

Other Practical Issues

- **PCA results depend on how you scale your data.** In hindsight this is obvious, since the maximisation of variance means we will put more weight in variables of higher variance.



Other Practical Issues

- We may want to choose the number of components for the goal of using **Z** as input to some other statistical analysis.
  - For instance, regression for prediction purposes.

- How to select it?
  - Cross-validation is an option, but even then we might want to reduce the possibilities to a more manageable set.

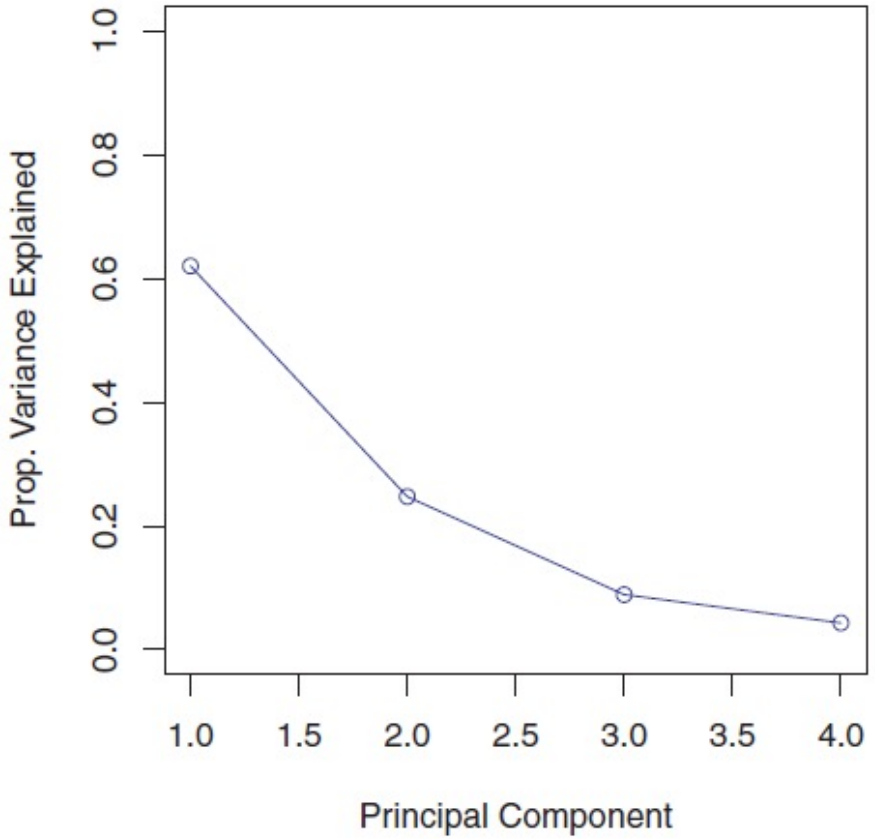
Other Practical Issues

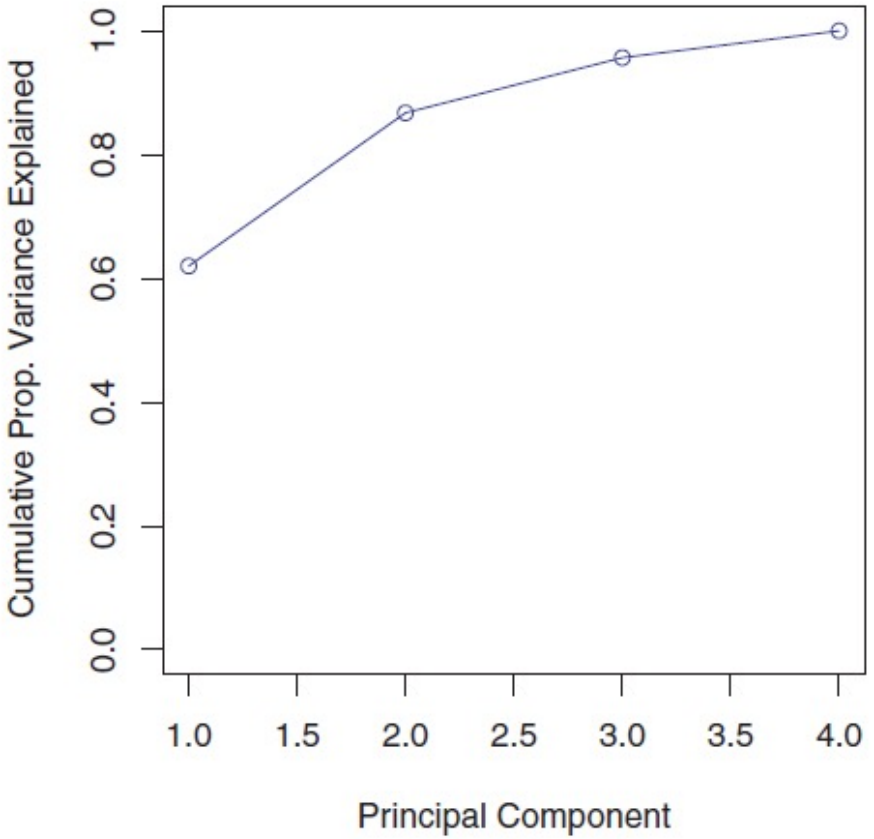
- One heuristic is the proportion of variance explained (*PVE*) by *m* components, contrasted to total variance (*TV*). For zero-mean data,

$$TV \equiv \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} (x_j^{(i)})^2 \qquad PVE(m) \equiv \frac{\frac{1}{n}\sum_{i=1}^{n} (z_m^{(i)})^2}{TV}$$

- We can plot *PVE(m)* against *m*, visualize where it is not worthwhile to add more components.

Example: USA Arrests Data





- The plot on the left is sometimes called a **scree plot**.

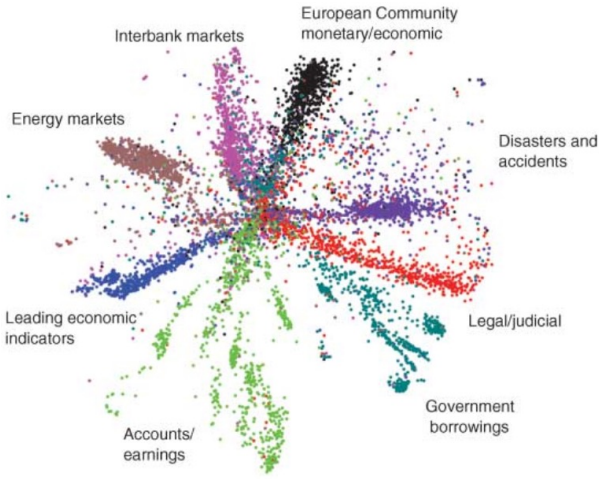
Other Practical Issues

- PCA is particularly useful for multivariate Gaussian data. Sometimes unclear how good it is otherwise.
  - Alternatives: make data "Gaussian-like" by marginally transforming variables; use non-linear PCA (a story for another day), neural networks etc.
- It is possible to directly pipeline PCA with regression to optimise parameters in a different way.
  - See Principal Components Regression, Section 6.3 of ISLR.
  - In the non-linear case, this is basically what a multilayer perceptron is.
- Just like any statistical method, it is possible to have confidence intervals on PCA coefficients. They are less obvious to derive. Johnson and Wichern (see reading list) discuss this with more detail if you are interested.

CLUSTERING WITH LATENT VARIABLE MODELS

Clustering

- Our final unsupervised learning task is clustering.
- More generally: find "natural" groups of data points.
  - Example: market segmentation, socio-economic stratification, etc.
- Two approaches: latent variable models & K-means.



Latent Variable Models

- Before discussing clustering, we do a small detour to discuss latent variable models…
- We already discussed some concepts when discussing GLMs (e.g. the propensities in ordered logit models).
- A latent variable model is a model where some variables are not in the data.
- Clustering does not need require the use of models, but latent variable models can motivate clustering.
  - Interpreting the outcome, is interpreting latent variable assignments.

Mixture of Gaussians

- Let *X* be a scalar discrete variable taking values in 1, 2, …, *K* for a given *K*. Parameters:

$$\theta_k \equiv P(X = k)$$

- Let our observations (Y_1, …, Y_p) be given independently by

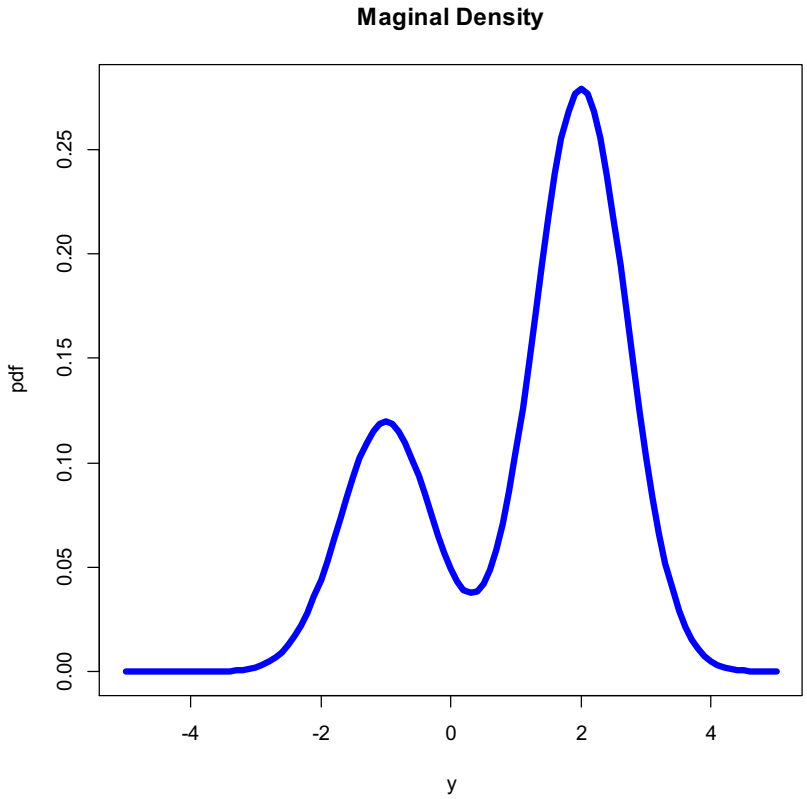
$$Y_j \mid X = k \sim N(\mu_k, \sigma^2), j = 1, 2, \ldots, p$$

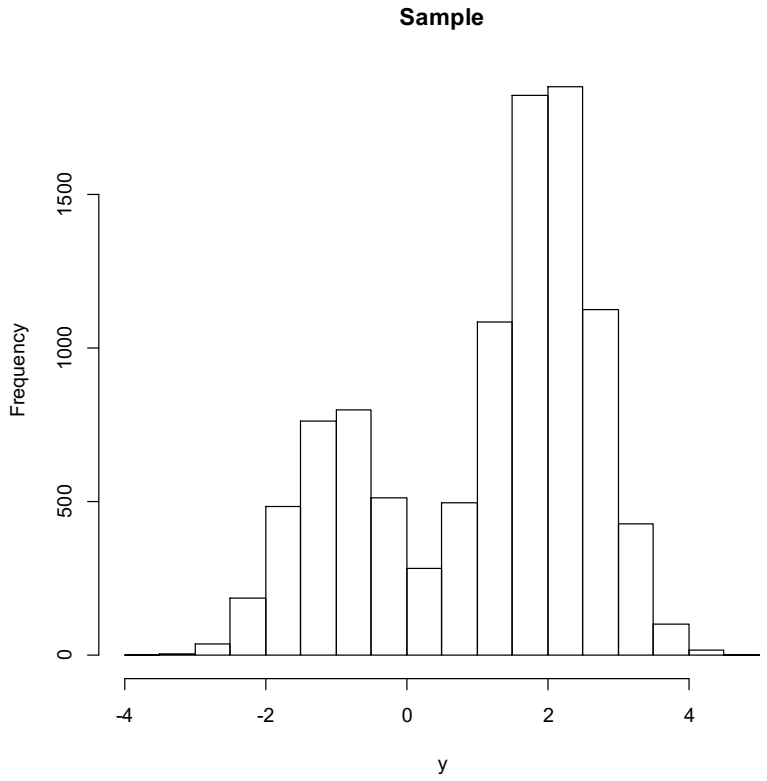
- This is a type of **Gaussian mixture model**. We could also generalize this to account for different variance for each cluster.

- The latent variable is the X variable indicating assignment to one of the Gaussians.

Example

- $p = 1, K = 2.$





Fitting

- The marginal likelihood is given by remembering that

$$p(\mathbf{y}) = \sum_{k=1}^{K} p(\mathbf{y} \mid X = k) P(X = k)$$

$$\prod_{i=1}^{n} p(\mathbf{y}^{(i)}) = \prod_{i=1}^{n} \left(\sum_{k=1}^{K} \theta_k \left\{ \prod_{j=1}^{p} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y_j^{(i)} - \mu_k)^2} \right\} \right)$$

$p(X^{(i)} = k)$ $\qquad$ $p(\mathbf{y}^{(i)} \mid X^{(i)} = k)$

- We can then maximise this marginal likelihood to get estimates of $\{\mu_k\}, \sigma^2, \{\theta_k\}$. In general this can be challenging: the common approach is to use and EM algorithm.

Clustering

- One interpretation of clustering (there are others) is: can we recover which mixture component generated each data point?

- More precisely, we would:
  - First fit our latent variable model to our dataset. i.e. we are fitting the parameters $\{\mu_k\}, \sigma^2, \{\theta_k\}$
  - Assign each data point a label depending on the probability that it belongs to cluster k:

$$P(X^{(i)} = k|y^{(i)}) = \frac{P(X^{(i)} = k)p(y^{(i)}|X^{(i)} = k)}{\sum_{j=1}^{K} P(X^{(i)} = j)p(y^{(i)}|X^{(i)} = j)}$$

CLUSTERING WITH K-MEANS

K-Means

- An algorithm originally motivated by grouping points by Euclidean distance. Find a partition (C_1, …, C_K) of {1, …, n} to solve

$$\min_{C_1,\ldots,C_K} \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j=1}^{p} (y_j^{(i)} - \bar{y}_{jk})^2$$

Average of y_j among points in C_k

- Note that we are computing the l2 norm between data points and the center of each cluster.

Solving K-Means

- Solving this exactly is pretty hard!
  - Exhaustive search: K possibilities for each of the n points, resulting on n^K possibilities
    - greater than number of particles in the observable universe for most problems.

- Practical solution:
  1. Allocate points to clusters randomly
  2. Find averages for each cluster
  3. Optimise cluster assignments for given averages
  4. Repeat 2-4 to convergence
  5. Repeat 1-4 a few times with different starting points, get best solution.

The Algorithm

Algorithm 10.1 *K-Means Clustering*

1. Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations.

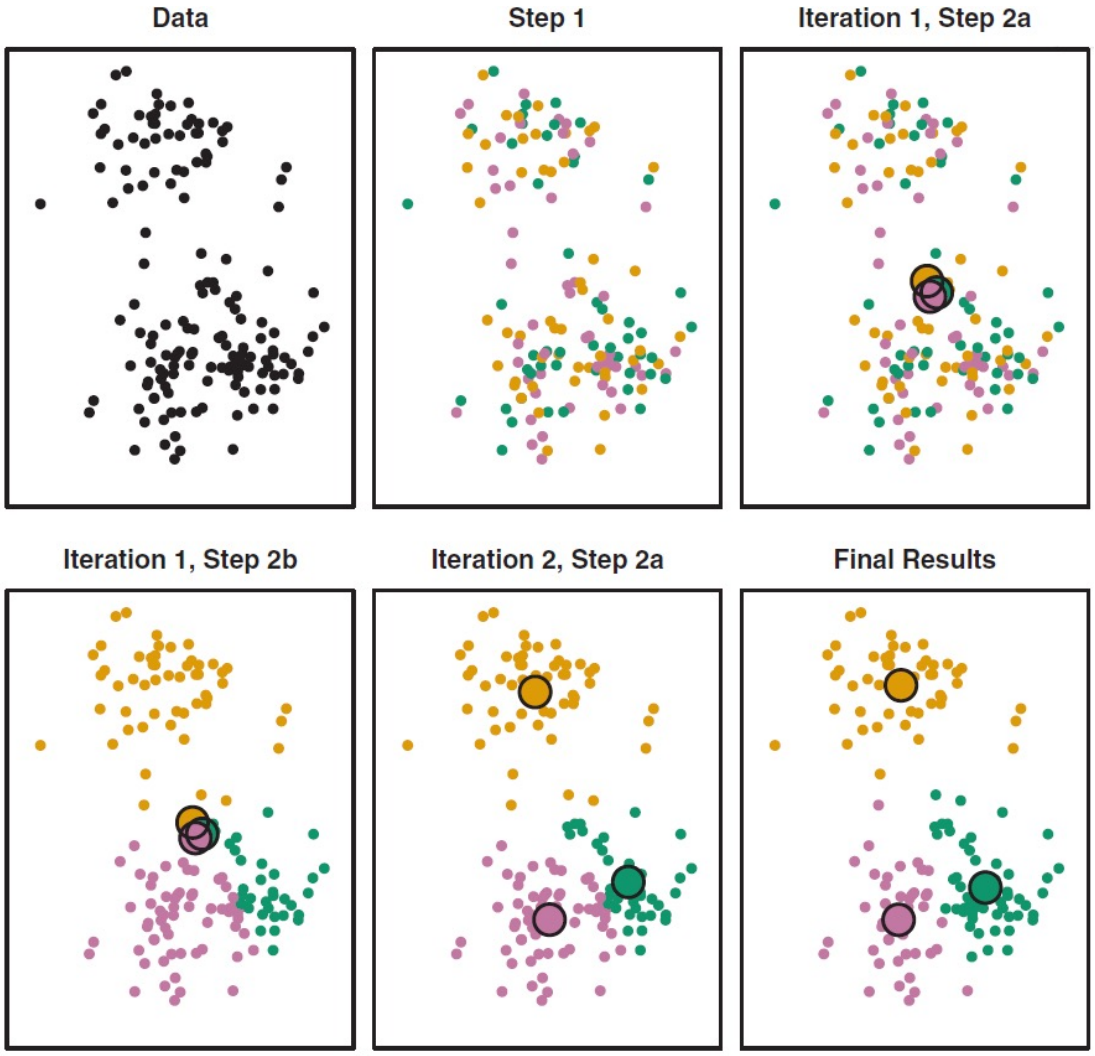
2. Iterate until the cluster assignments stop changing:

   (a) For each of the K clusters, compute the cluster *centroid*. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.

   (b) Assign each observation to the cluster whose centroid is closest (where *closest* is defined using Euclidean distance).

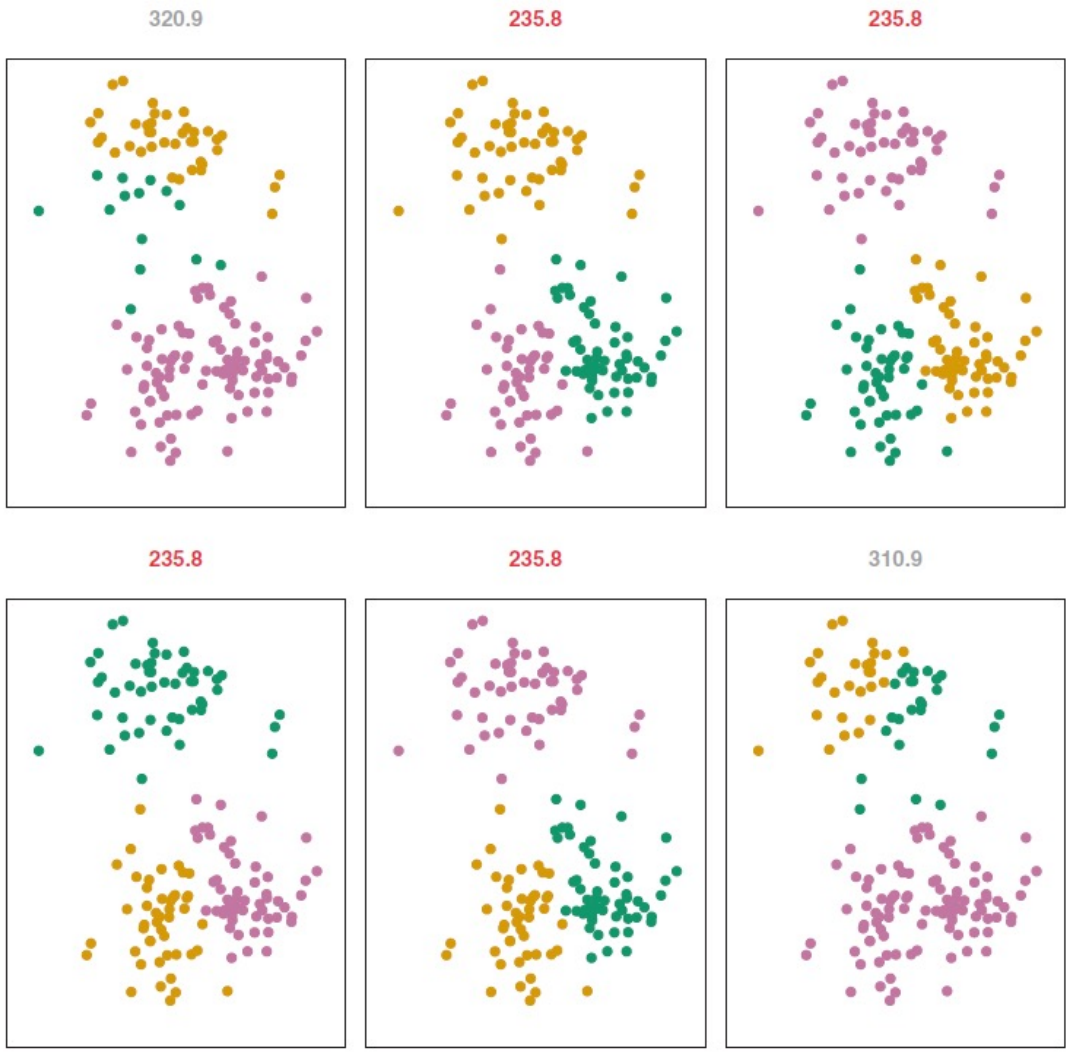
(ISLR, Chapter 10)

Illustration, $K = 3$



(ISLR, Chapter 10)

Practical Issues: Random Restarts



Practical Issues: Choice of *K*

- Many ways. Here are a few:
  - In principle, with an explicit likelihood, penalties like AIC/BIC could be applied.
    - Not necessarily a good thing. Do we believe in the mixture of Gaussians model?
  - "Compression" point of view:
    - Clustering is just extreme dimensionality reduction: one scalar per data point, the cluster assignment! Choose the largest *K* we feel happy with.
    - "Acceptable reconstruction error": to avoid wasting resources with large *K*, allow for smaller *K* if the average/maximum distance of the points to the cluster mean is below some domain-specific error.

Practical Issues: Validation

- Like much of unsupervised learning, this is not easy to validate.

- All sorts of sensitivity analyses:
  - How does clustering change given a subset of the data? (a type of bootstrapping)
  - How does it change given a subset of the variables? How much variance do the unused variables have within each cluster?

- I'm afraid there is no easy, domain-independent, answer here.

Take-Home Messages

- From a statistical perspective, we can see unsupervised learning as estimating joint distributions.
  - Sometimes merely as a tool to facilitate visualization or supervised learning.
- However, this is not the whole story, as we would like to characterize “features” of such distributions.
- Outliers, independencies and latent variables (including cluster assignments and PCA projections) are one way of describing what these features are, but they may come with strong assumptions too.

END OF THE BEGINNING