



THE UNIVERSITY OF
SYDNEY

Lecture 3: Density estimation

STAT5003

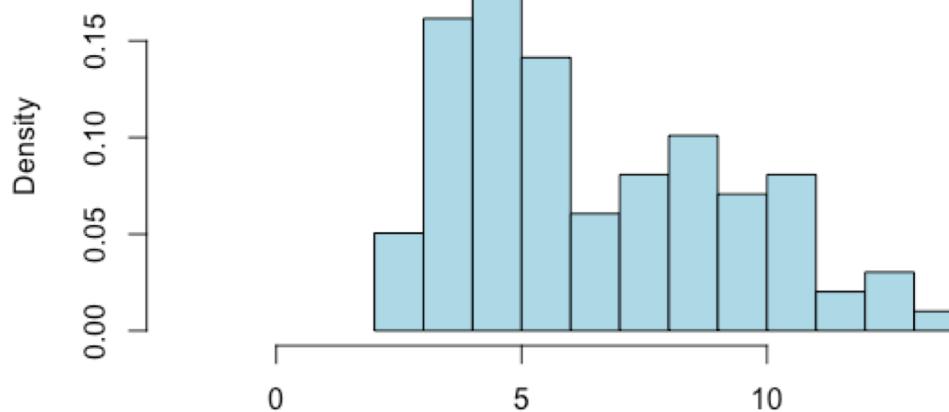
Pengyi Yang

Readings

- An Introduction to Statistical Learning with Applications in R
 - Chapter 7 : Moving Beyond Linearity. Sections 7.1 to 7.5

Density estimation

- In this section of the unit we will introduce density estimation and smoothing methods.



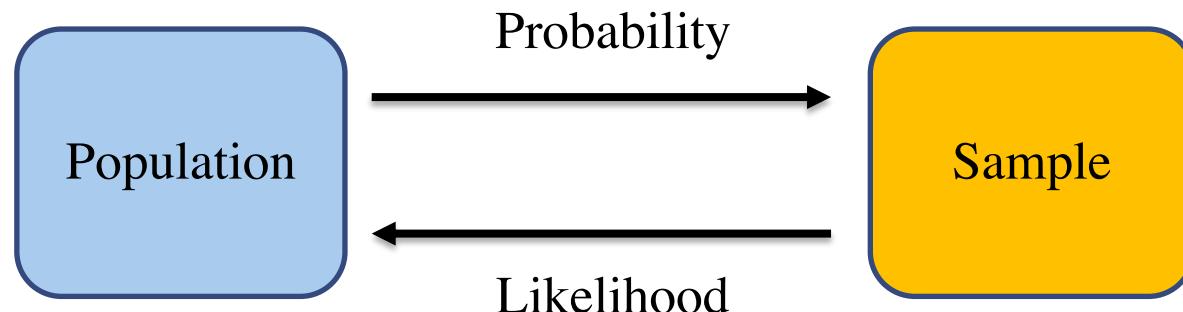
- There are two types of methods classified as *parametric* and *nonparametric*. Compared to *parametric* methods *nonparametric* methods lack of a formal statistical model.
- The methods are generally intended for *description* rather than formal *inference*.
- We learn methods to describe the probability distribution of univariate random variable and multivariate random variables.

Usage of estimating a density function

- In exploratory data analysis, an estimate of the density function can be used
 - to assess multimodality, skew, tail behaviour, etc.
 - in decision making, classification, and summarizing Bayesian posteriors
 - as a useful visualisation tool (a simple summary of a distribution)
- This week our concern is the estimation of a *density function* f using observations of random variables x_1, \dots, x_n sampled independently from f .

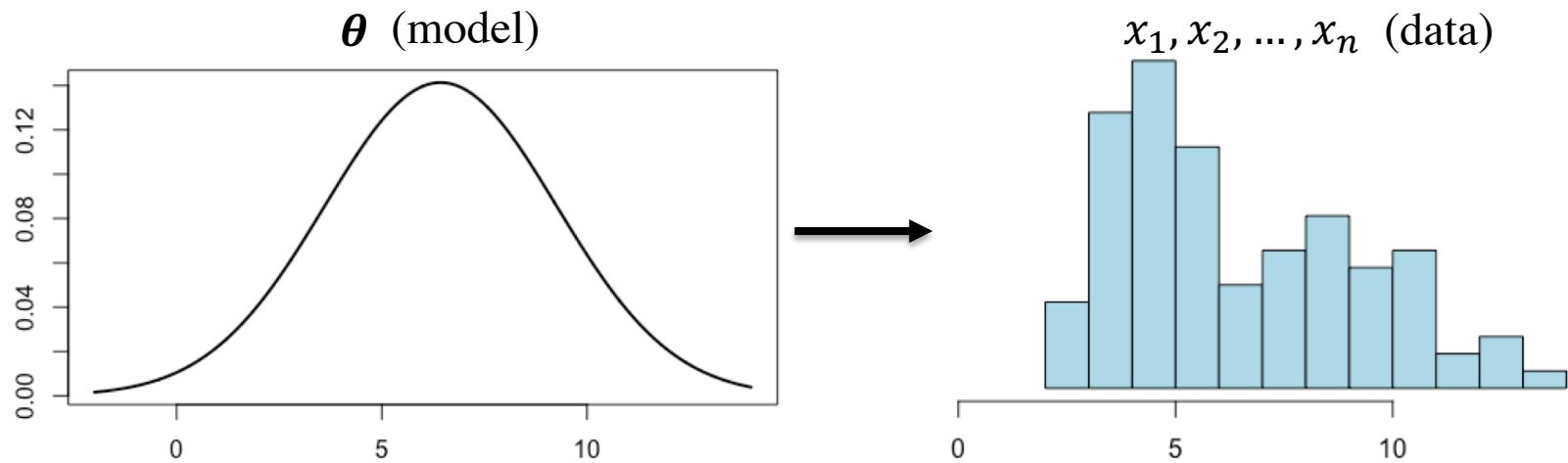
Parametric density estimation

- The *parametric* approach to a density estimation assumes a *parametric* model, $x_1, \dots, x_n \sim \text{i.i.d. } f_\theta$, (where θ is a low-dimensional parameter vector).
For example, assume $x_i \sim N(m, s)$
- We typically estimate parameters $\hat{\theta}$ using maximum likelihood.
density at x can be estimated as $f(x|\hat{\theta})$
- What is likelihood?



Parametric approach – maximum likelihood

$f(x_1, x_2, \dots, x_n | \boldsymbol{\theta})$ Probability of observing x_1, x_2, \dots, x_n given parameter(s) $\boldsymbol{\theta}$



$$f(x_1, x_2, \dots, x_n | \boldsymbol{\theta}) = f(x_1 | \boldsymbol{\theta}) \cdot f(x_2 | \boldsymbol{\theta}) \cdot \dots \cdot f(x_n | \boldsymbol{\theta}) = \prod_{i=1}^n f(x_i | \boldsymbol{\theta})$$

Likelihood function is defined as $L(\boldsymbol{\theta} | x_i) = \prod_{i=1}^n f(x_i | \boldsymbol{\theta})$

Because maximise log-likelihood is often easier so we commonly maximise the following:

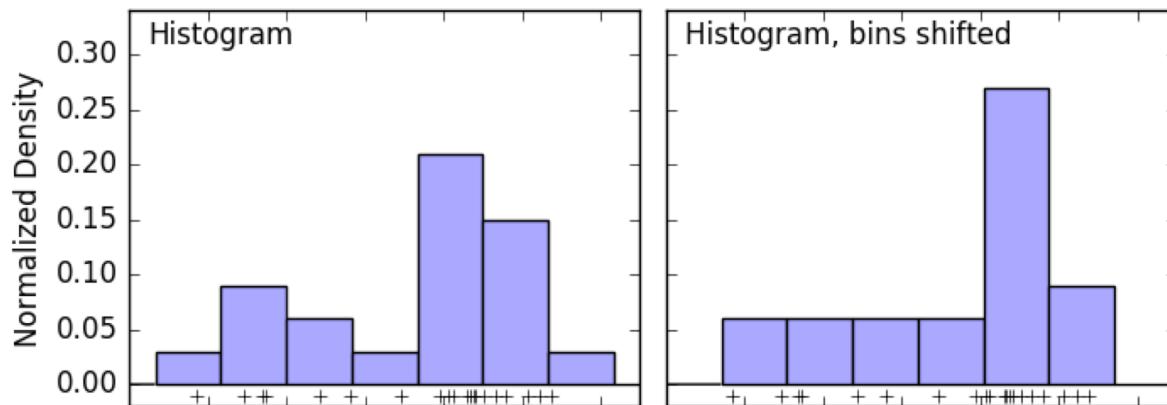
$$L(\boldsymbol{\theta} | x_i) = \prod_{i=1}^n f(x_i | \boldsymbol{\theta}) \rightarrow \ln L(\boldsymbol{\theta} | x_i) = \sum_{i=1}^n \ln f(x_i | \boldsymbol{\theta})$$

Non-parametric density estimation

- *Danger with parametric approach:*
When the assumed model f_θ is incorrect this approach can lead to serious inferential errors.
- *Nonparametric* approaches to density estimation
 - ✓ assume very little about the form of f .
 - ✓ use *local information* to estimate f at a point x .
- *Histograms are*
 - ✓ one type of nonparametric density estimators
 - ✓ *piecewise* constant density estimators
 - ✓ produced automatically by most software packages

Histograms

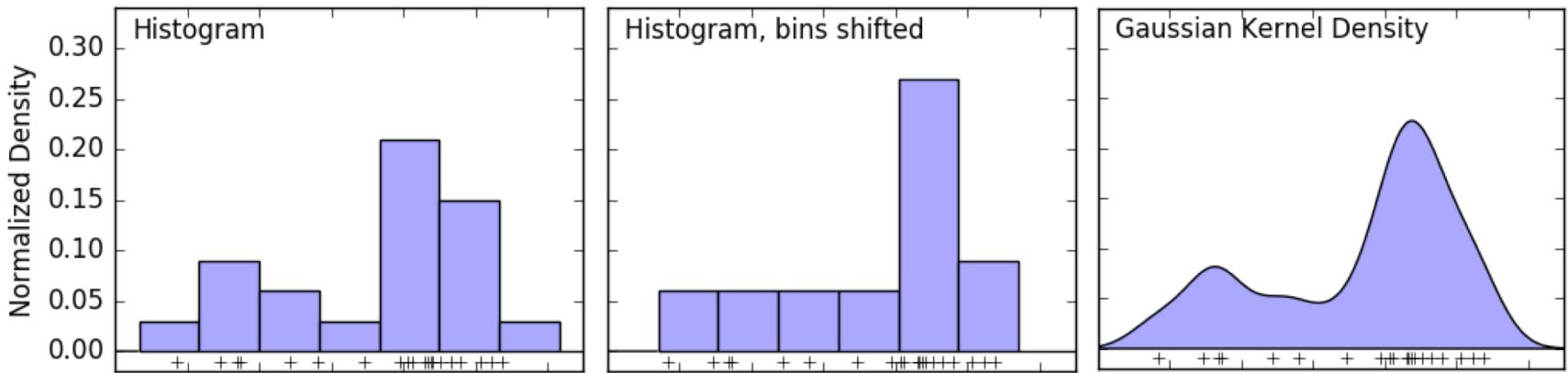
- A histogram can be used as a simple visualization of data;
- In a histogram bins are first defined and then the number of data points within each bin is used to determine the height of the bar plot.
- Two histograms of the same data are shown below (how would you interpret the data according to each of these two histograms?):



- The number of bins are slightly different (7 vs 6)
- Different interpretations of the data.

Demonstrate histogram and parametric
method

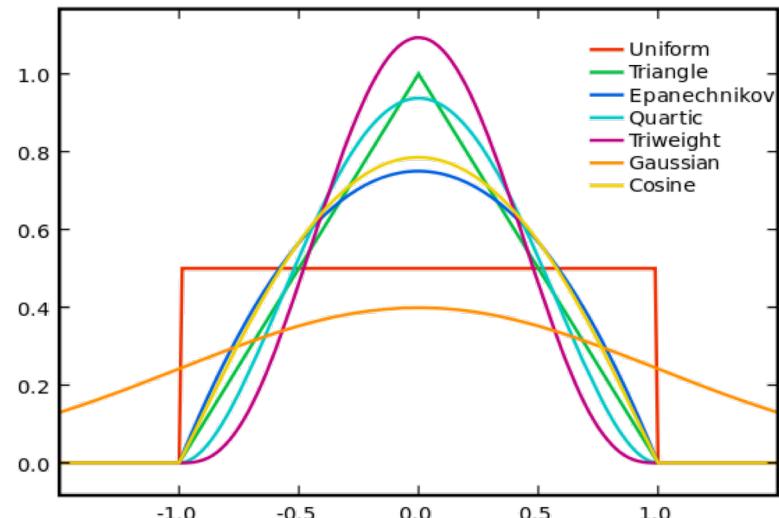
Refining histogram



- Right plot uses a smoother kernel *Gaussian kernel density estimate*
- each point contributes to a Gaussian curve to the total
 - a smooth density estimate derived from the data,
 - a powerful non-parametric model of the distribution of points.

Kernel

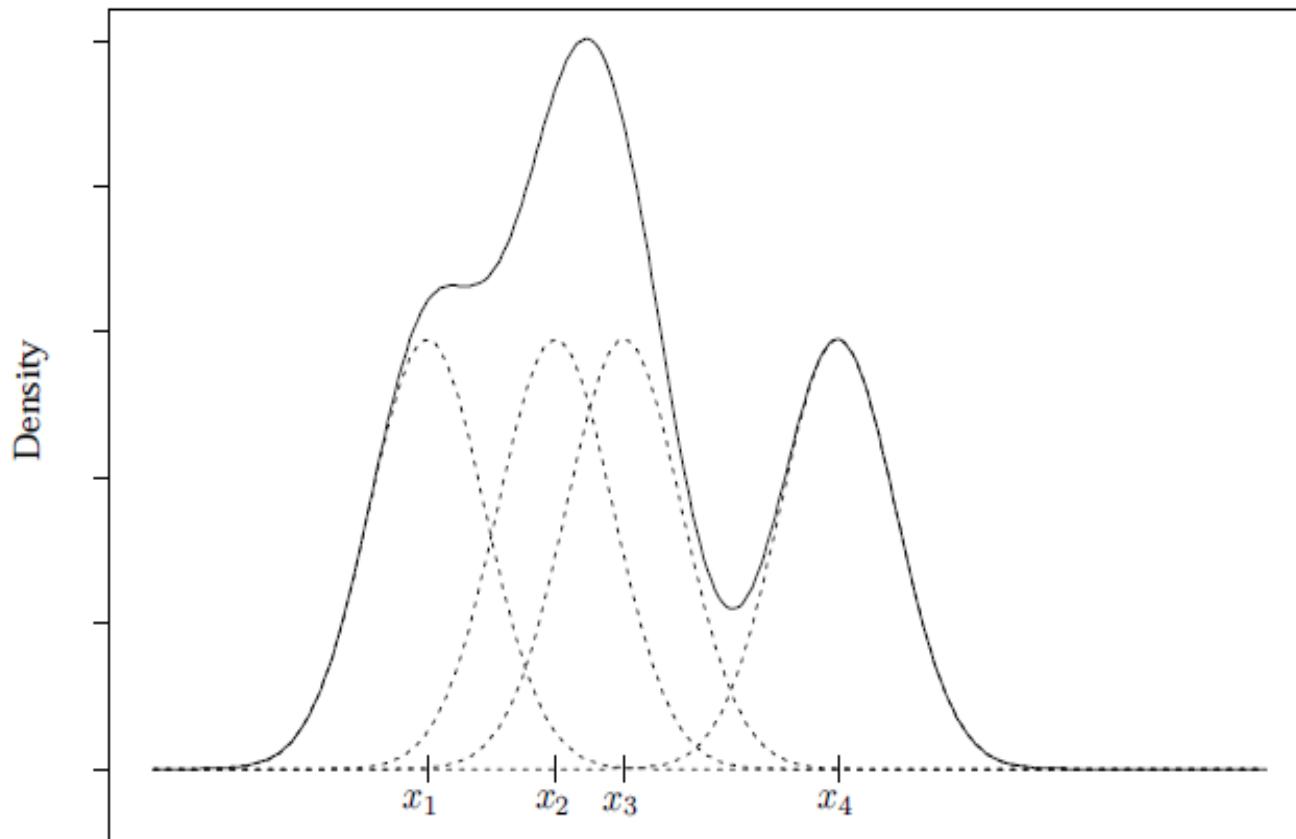
- A kernel is a special type of probability density function (PDF) which is *symmetric*.
- A kernel is a function and has the following properties
 - ✓ non-negative
 - ✓ real-valued
 - ✓ symmetric
 - ✓ its definite integral over its support set must equal to 1
- Some common PDFs are kernels:
e.g. Uniform(-1,1) and standard normal distributions.



Kernel density estimation

- Kernel density estimation is a non-parametric approach for estimating the probability density function (*pdf*) of a continuous random variable.
- It is non-parametric because it does not assume any underlying distribution for the *variable* (a.k.a. X does not need to be assumed to follow any specific distribution).
- Essentially, at every data point, a kernel function is created with the point at its centre.
- → the kernel is symmetric around the point.
- The *pdf* is estimated by adding all of these kernel functions and dividing by the number of data to ensure that it satisfies
 - ✓ every possible value of the *pdf* is non-negative.
 - ✓ the definite integral of the *pdf* over its support set equals 1.

Normal kernel density estimate



Normal kernel density estimate (solid) and kernel contributions (dotted) for the sample x_1, \dots, x_4 . The kernel density estimate at any x is the sum of the kernel contributions centered at each x_i .

Kernel density estimator

- The simple density estimator that weights all points within h of x equally.

$$\hat{f}(x) = \frac{1}{2hn} \sum_{i=1}^n 1_{\{|x-X_i| < h\}},$$

- A univariate *kernel density estimator* allows a more flexible weighting scheme, fitting

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

where

K is a *kernel function* and

h is a fixed number, called the *bandwidth*, *window width*, or *smoothing parameter*.

Constructing a kernel density estimator

1. Choose a kernel; the common ones are normal (Gaussian), uniform (rectangular) and triangular (*kernel functions are positive everywhere and symmetric at zero*).
2. At each point, X_i , build the scaled kernel function

where

$$\frac{1}{h} K \left[\frac{(x-X_i)}{h} \right]$$

$K[]$ is the chosen kernel function,
the parameter h is the *bandwidth*, window width or *smoothing parameter*.

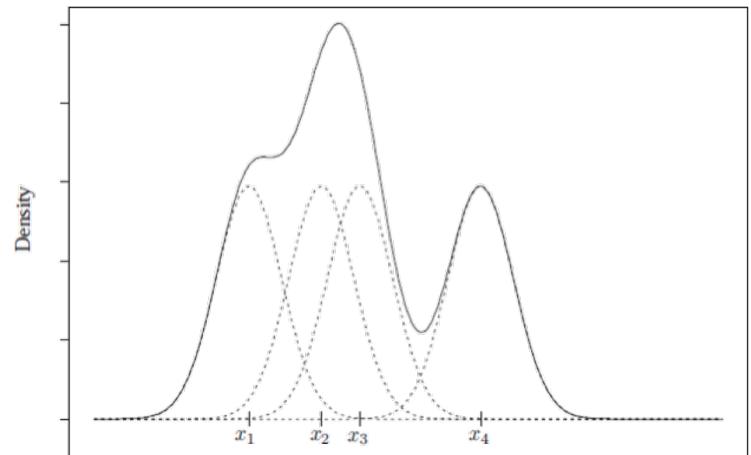
3. Add the individual scaled kernel functions and divide by n ;

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^n K \left[\frac{(x-X_i)}{h} \right]$$

- ✓ this places a probability of $1/n$ to each x_i
- ✓ it also ensures that the kernel density estimate integrates to 1

Constructing kernel density estimate

- Figure on the right illustrates how a kernel density estimate is constructed from a sample of four univariate observations, x_1, \dots, x_4 .



- Centred at each observed data point is a scaled kernel: in this case, *a normal density function divided by 4*. (These contributions are shown with the dotted lines).
- Summing the contributions yields the estimate \hat{f} (solid line).

A sum of “bumps”

- **Intuitively, a kernel density estimate is a sum of “bumps”.**
- A “bump” is assigned to every point, and the size of the “bump” represents the probability assigned at the *neighbourhood of values* around that point;
- thus, if the data set contains
 - 2 data points at $x = 1.5$
 - 1 data point at $x = 0.5$
- then the “bump” at $x=1.5$ is twice as big as the “bump” at $x=0.5$.
- Each “bump” is centred at the point, and spreads out symmetrically to cover the neighbouring values around the point.

Choice of kernels

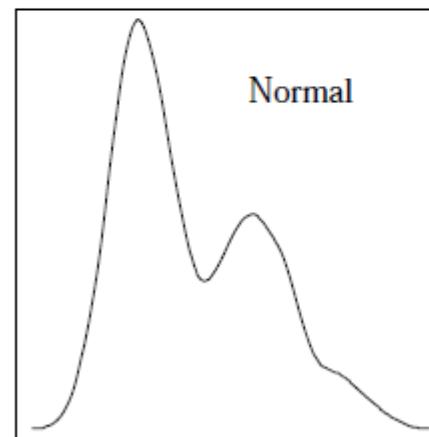
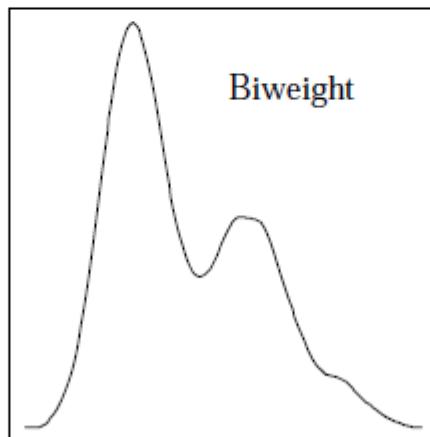
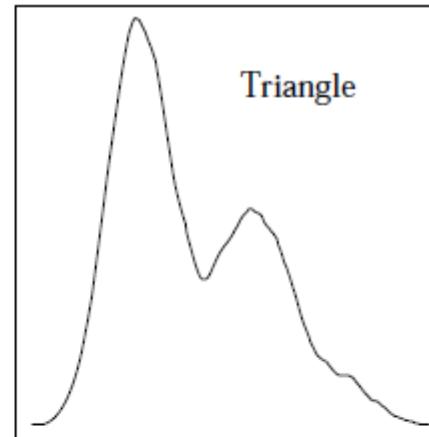
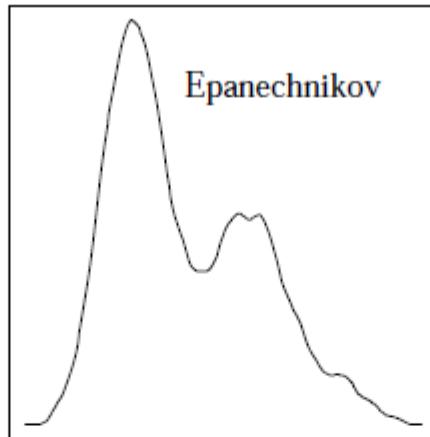
- Kernel density estimation requires specification of two components: the kernel and the bandwidth.
- The shape of the kernel has much less influence on the results than does the bandwidth.
- Some choices for kernel functions are,

Kernel	Equation
Uniform	$k_0(u) = \frac{1}{2}1(u \leq 1)$
Epanechnikov	$k_1(u) = \frac{3}{4}(1 - u^2)1(u \leq 1)$
Biweight	$k_2(u) = \frac{15}{16}(1 - u^2)^21(u \leq 1)$
Triweight	$k_3(u) = \frac{35}{32}(1 - u^2)^31(u \leq 1)$
Gaussian	$k_\phi(u) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right)$

- See also:

[https://en.wikipedia.org/wiki/Kernel_\(statistics\)#Kernel_functions_in_common_use](https://en.wikipedia.org/wiki/Kernel_(statistics)#Kernel_functions_in_common_use)

Different kernel estimates for “bimodal.dat”



Bimodal density, cont.

- Previous figure shows kernel density estimates for the data with equally weighted mixture of $N(4, 12)$ and $N(9, 22)$ densities.
 - All the bandwidths were set at 0.69 for the canonical kernels of each shape.
- The results for all the kernels are qualitatively the same.
- Even quite different kernels can be scaled to produce such similar results that the choice of kernel is unimportant.

Kernel density estimation in R

- The `density()` function in R computes the values of the kernel density estimate.
- Applying the `plot()` function to an object created by `density()` will plot the estimate.
- Applying the `summary()` function to the object will reveal useful statistics about the estimate.

Demonstrate kernel density estimation

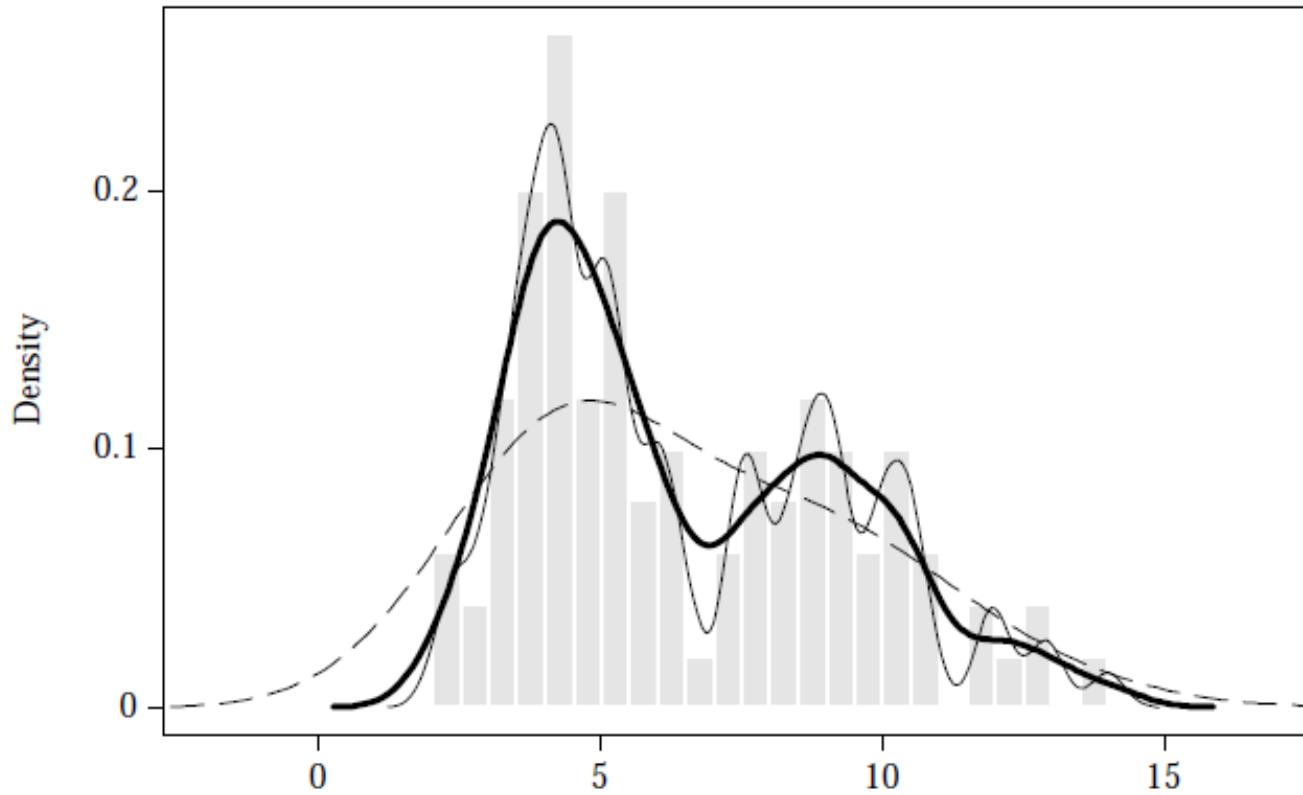
Value chosen for the bandwidth

- The density estimator

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^n K\left[\frac{(x-X_i)}{h}\right]$$

- is a *fixed-bandwidth kernel density estimator* since h is constant.
- The value chosen for the bandwidth exerts a strong influence on the estimator \hat{f} .
- If **h is too small**, the density estimator will tend to assign probability density too locally near observed data
→ a wiggly estimated density function with many false modes.
- If **h is too large**, the density estimator will spread probability density contributions too diffusely
→ smooths away important features of f .

Value of bandwidth affects smoothness



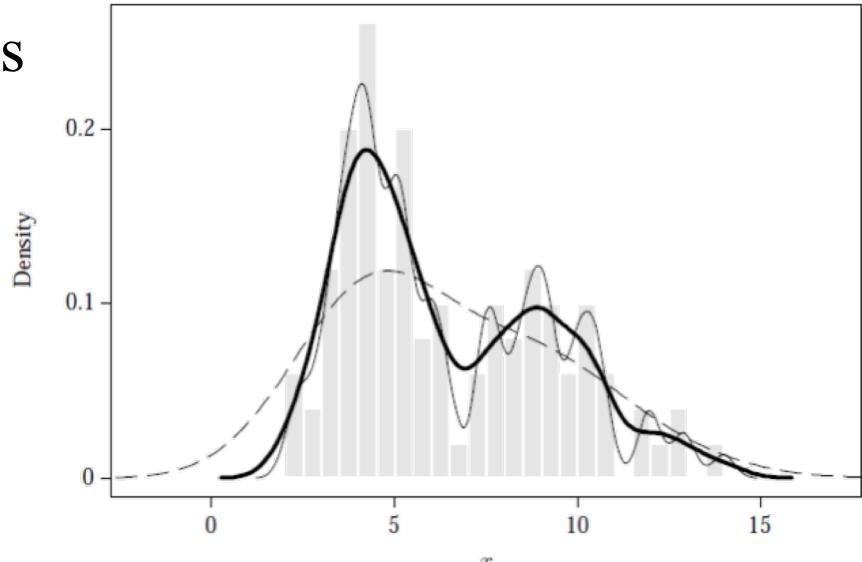
- In above, the bandwidths are 0.3, 0.625, and 1.875
- Demonstrate the above effect in R.

Bimodal density in the above example

- The effect of bandwidth is shown in previous slide.
- This histogram shows a sample of 100 points from an equally weighted mixture of $N(4, 12)$ and $N(9, 22)$ densities.
- Three density estimates that use a standard normal kernel are superimposed, with different bandwidths, h :
 - $h = 1.875$ (*dashed*) is clearly *too large* → oversmooth density estimate that fails to reveal the bimodality of f .
 - $h = 0.3$ (*solid*) is too small a bandwidth → undersmoothing, the density estimate is too wiggly with many false modes.
 - $h = 0.625$ (*heavy*) is adequate → represents main features of f while suppressing most effects of sampling variability.

Choice of bandwidth

- The bandwidth parameter controls the smoothness of the density estimate.
- → the bandwidth determines the trade-off between the *bias* and *variance of estimator* \hat{f} .
- ✓ A **small bandwidth** produces a density estimator with wiggles indicative of *high variability* caused by under smoothing.
- ✓ A **large bandwidth** causes important features of f to be smoothed away, thereby causing *bias*.
- (the trade-off between the *bias* and the *variance of an estimator* is important in nearly all kinds of model selection, including density estimation, smoothing, regression, and classification)



Integrated squared error (ISE)

- To evaluate \hat{f} as an estimator of f over the entire range of support, one could use the *integrated squared error* (ISE):

$$\text{ISE}(h) = \int_{-\infty}^{\infty} [\hat{f}(x) - f(x)]^2 dx$$

the ISE(h) gives the square of the distances from $\hat{f}(x)$ to $f(x)$ for the *observed data*.

- To discuss the generic properties of an estimator, without reference to a particular observed sample, it is more sensible to average ISE(h) over all samples that might be observed.
- This gives the *mean integrated squared error* (MISE):

$$\text{MISE}(h) = E\{\text{ISE}(h)\}$$

See <http://demonstrations.wolfram.com/VarianceBiasTradeoff/>

Cross-validation

- Cross-validation, is a model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set.
- It is mainly used when one wants to estimate how accurately a *predictive model* will perform in practice.
- In a prediction problem, a model is usually given
 - ✓ a dataset of *known data* on which training is run (*training dataset*),
 - ✓ and a dataset of *unknown data* (or *first seen* data) against which the model is tested (*testing dataset*).

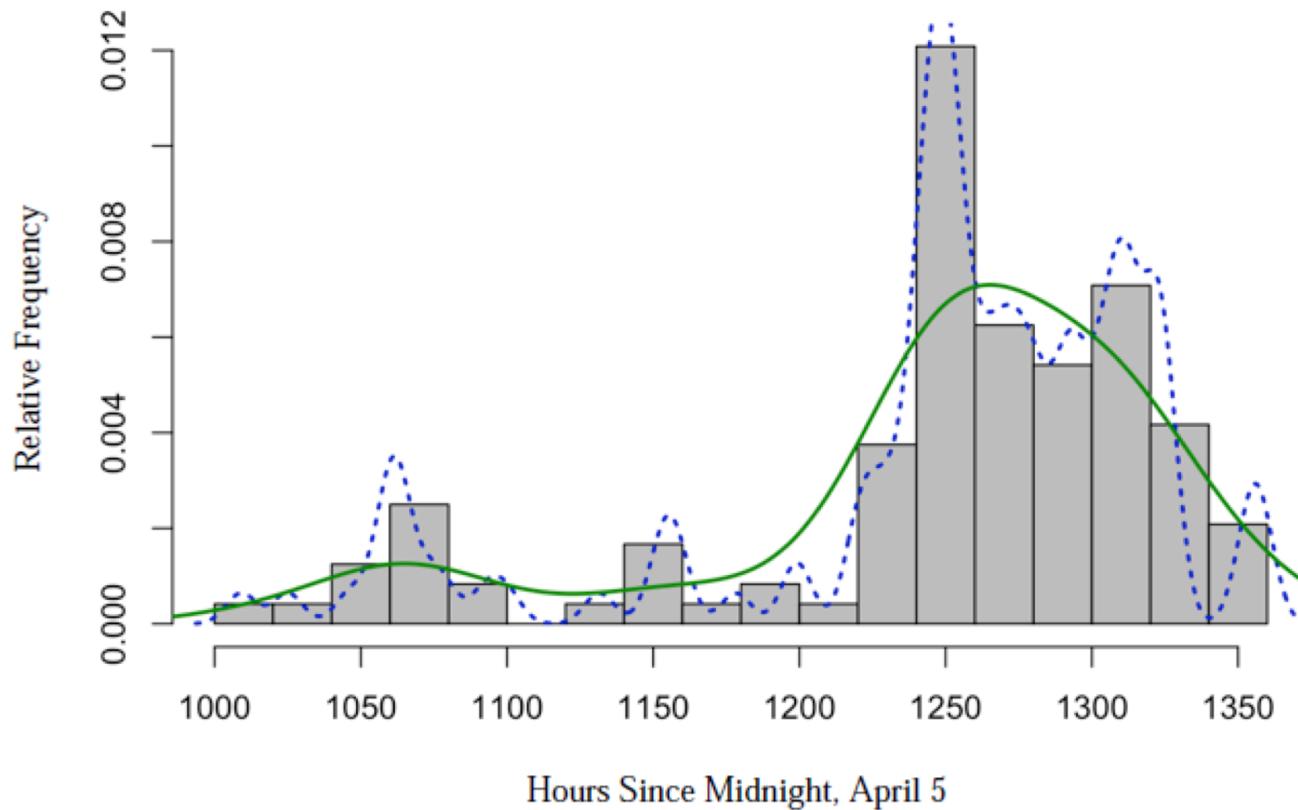
Goal of cross-validation

- The goal of cross validation is to define a dataset to “test” the model in the training phase (i.e., the *validation dataset*), in order to
 - limit problems like overfitting,
 - give an insight on how the model will generalise to an independent dataset (i.e., an unknown dataset, for instance from a real problem), etc.
- One round of cross-validation involves
 - ✓ partitioning a sample of data into complementary subsets,
 - ✓ performing the analysis on one subset (called the *training set*), and
 - ✓ validating the analysis on the other subset (called the *validation set* or *testing set*).

Why use cross-validation

- Many bandwidth selection strategies begin by relating h to some measure of the quality of \hat{f} as an estimator of f .
- The quality is quantified by some $Q(h)$, whose estimate, $\widehat{Q}(h)$, is optimized to find h .
- If $\widehat{Q}(h)$ evaluates the quality of \hat{f} based on how well it fits the observed data → observed data are being used twice:
 - ✓ once to calculate \hat{f} from the data and a
 - ✓ second time to evaluate the quality of \hat{f} as an estimator of f .
- Double use of the data provides an overoptimistic view of the quality of the estimator leading to overfitting.
- → Cross-validation provides a remedy to this problem.

Demonstration of CV for bandwidth selection



Using cross-validation criteria

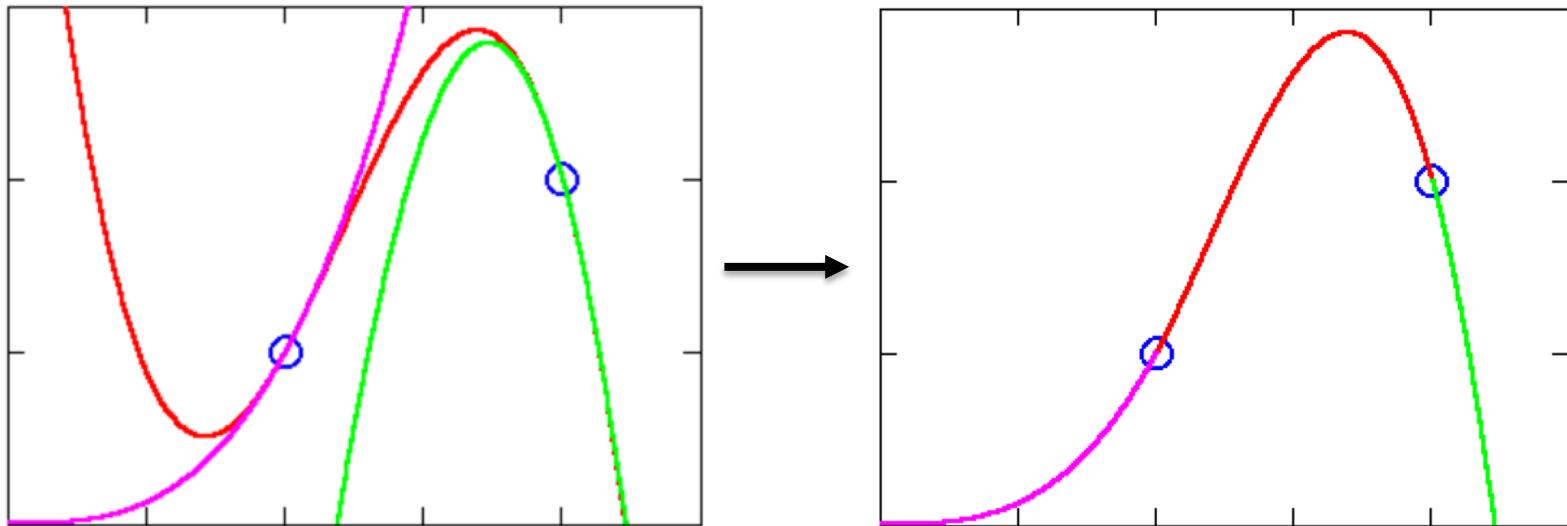
- Previous figure shows the results of kernel density estimates for these data using the *normal kernel* (*see previous slide*).
 1. The result from using a $h = 5.18$ (dotted curve)
→ bad, the bandwidth is clearly too small.
 2. Minimizing **Biased CV(h)** with respect to h yields $h = 26.43$ (solid line)
→ the better option in this case, emphasizes only the most prominent features of the data distribution.
- Biased CV [BCV(h)]: a type of CV that optimises asymptotic MISE (AMISE) instead of MISE. Lower variance but reasonable bias. (See additional reference)
- Perhaps a bandwidth around 26 would be preferable.

Spline methods

Cubic spline

- A *cubic spline* is a piecewise cubic function that is everywhere twice continuously differentiable but whose third derivative may be discontinuous at a finite number of pre-specified *knots*.
- A *cubic spline* is a function created from cubic polynomials on each *between-knot* interval by pasting them together twice continuously differentiable at the knots.
- <https://www.youtube.com/watch?v=f4iNbNRKZKU>
<http://wmueller.com/precalculus/families/splines.html>

Visually, cubic spline “concatenate” cubic polynomial lines

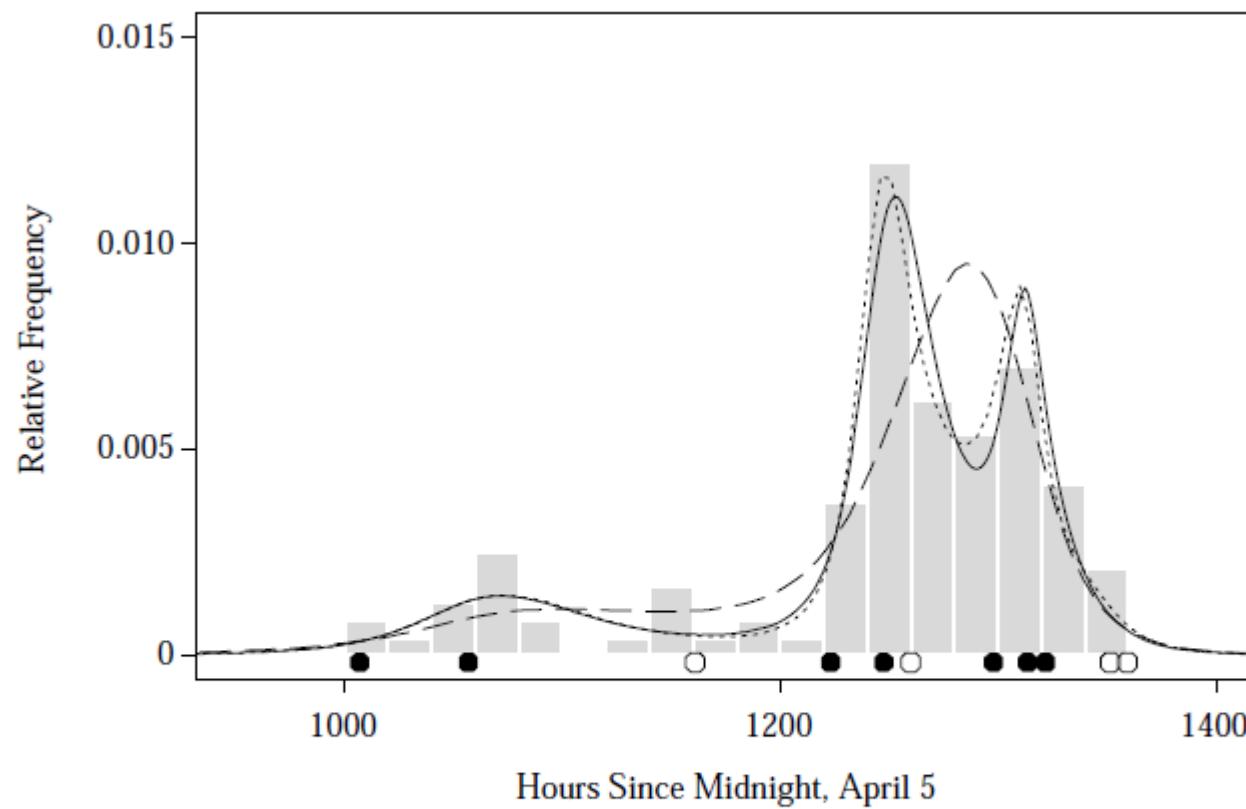


Cubic spline for density estimation

- Cubic spline can be used for estimating an unknown density function f based on sample data.
- One approach is to use maximum likelihood estimation to estimate $\log(f)$ by a cubic spline that have a finite number of pre-specified **knots**.
 - The knots are placed at selected order statistics of the sample data.
 - The number of knots can be determined either by a simple rule or by minimizing BIC (**Bayes information criterion**).
- The method works well both in obtaining smooth estimates and in picking up small details.

Logspline density estimate

Kooperberg and Stone's logspline density estimation approach estimates the log of f by a cubic spline



Whale migration example, continued

- Estimation of local modes can sometimes be a problem if there are too few knots or if they are poorly placed.
- The other lines in previous figure show the logspline density estimates with two other choices for the knots.
- The very poor estimate (dashed line) was obtained using 6 knots.
- The other estimate (dotted line) was obtained using all 11 knots shown in the figure with either hollow or solid dots.

Useful references and videos

- C. Kooperberg and C. J. Stone. Logspline density estimation for censored data. *Journal of Computational and Graphical Statistics*, 1:301–328, 1992
- https://cran.r-project.org/web/packages/HSAUR/vignettes/Ch_density_estimation.pdf



- <https://www.youtube.com/watch?v=QSNN0no4dSI> (first 53 mins)
- https://www.youtube.com/watch?v=WlK8PdHL_qc
- <https://www.youtube.com/watch?v=f4iNbNRKZKU>