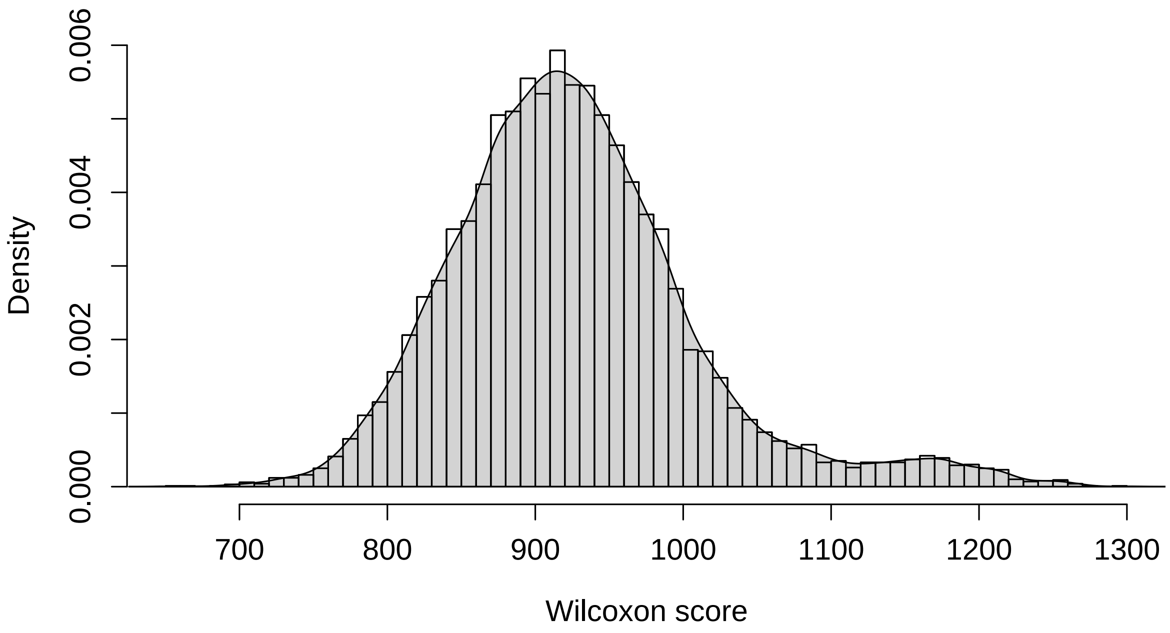
Introduction to Kernel Smoothing



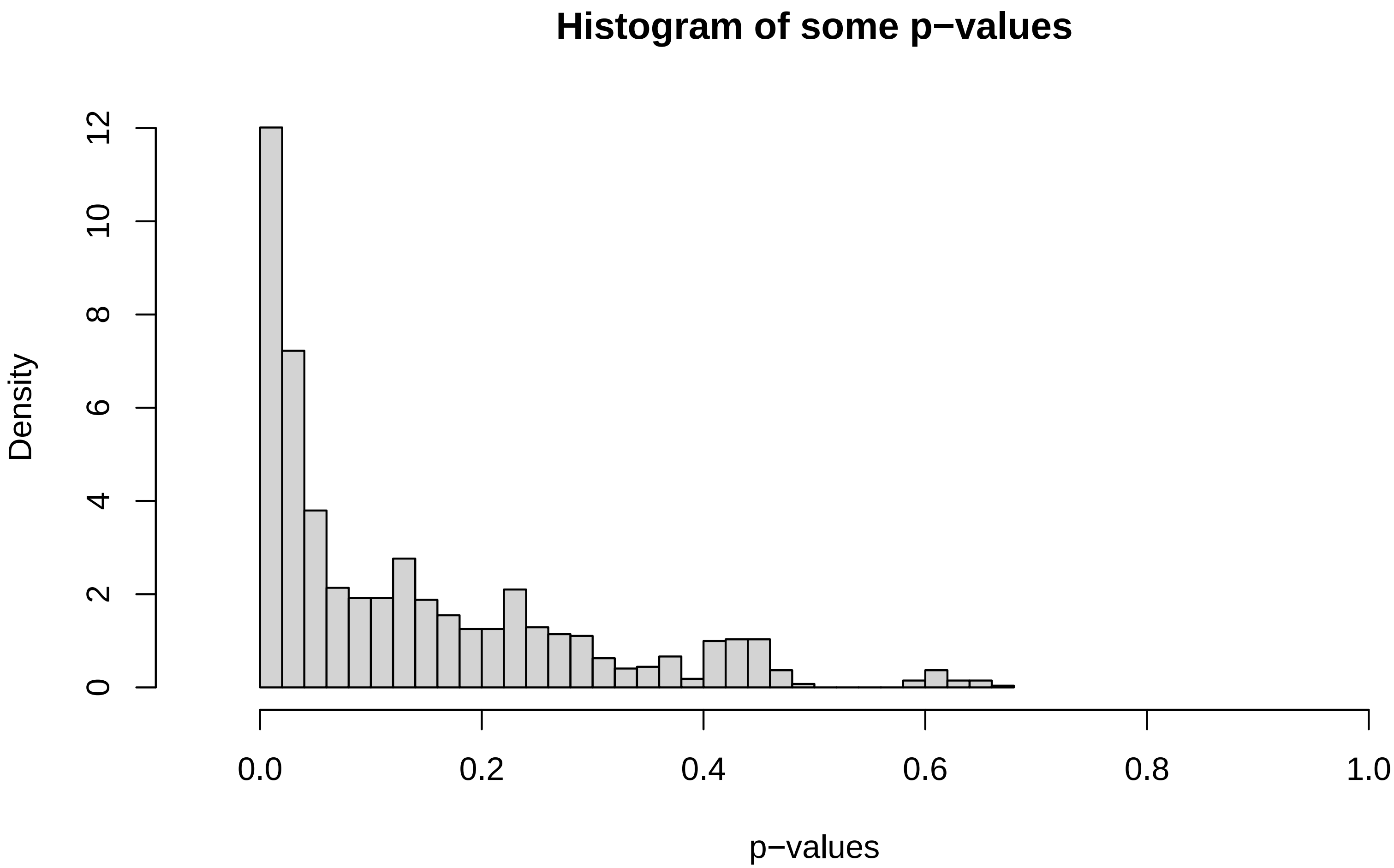
M. P. Wand & M. C. Jones

Kernel Smoothing

Monographs on Statistics and Applied Probability

Chapman & Hall, 1995.

Introduction



Introduction

- Estimation of functions such as regression functions or probability density functions.
- Kernel-based methods are most popular non-parametric estimators.
- Can uncover structural features in the data which a parametric approach might not reveal.

Univariate kernel density estimator

Given a random sample $X_1, \ldots, X_n$ with a continuous, univariate density f. The kernel density estimator is

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

with kernel K and bandwidth h. Under mild conditions (h must decrease with increasing n) the kernel estimate converges in probability to the true density.

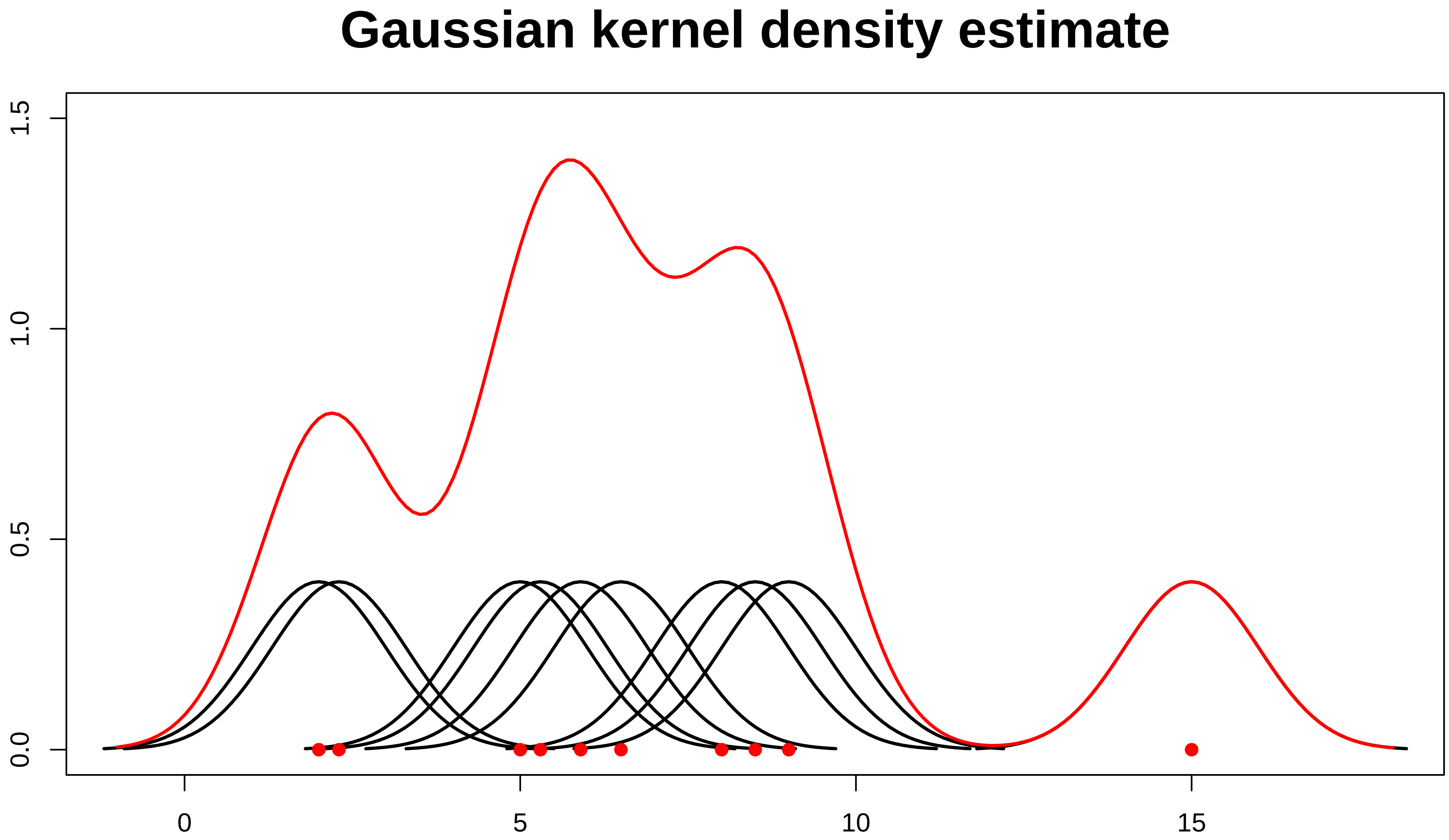
The kernel K

– Can be a proper pdf. Usually chosen to be unimodal and symmetric about zero.

$\Rightarrow$ Center of kernel is placed right over each data point.

$\Rightarrow$ Influence of each data point is spread about its neighborhood.

$\Rightarrow$ Contribution from each point is summed to overall estimate.


Gaussian kernel density estimate
0.0
0.5
1.0
1.5
0
5
10
15

The bandwidth h

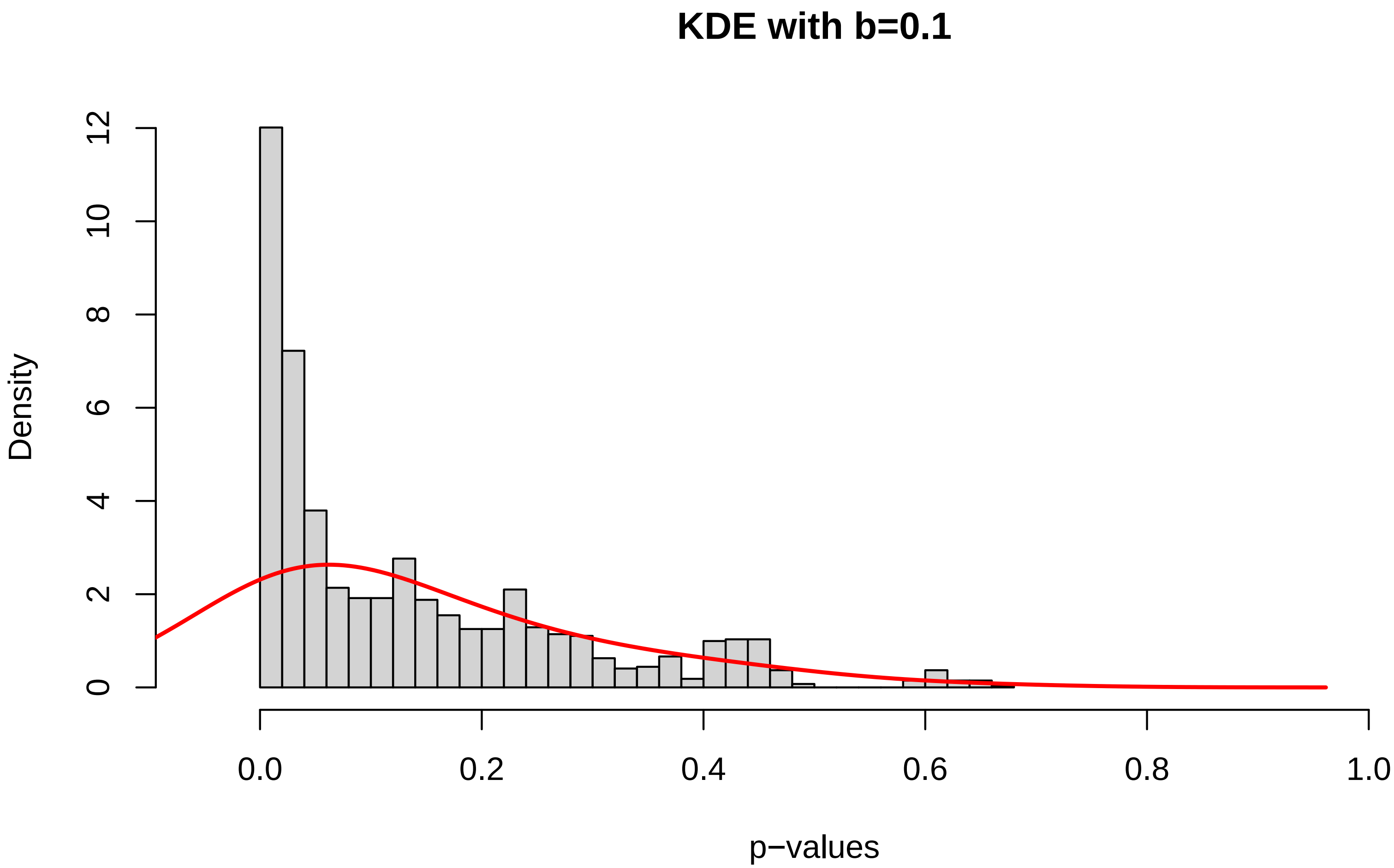
– Scaling factor.

– Controls how wide the probability mass is spread around a point.

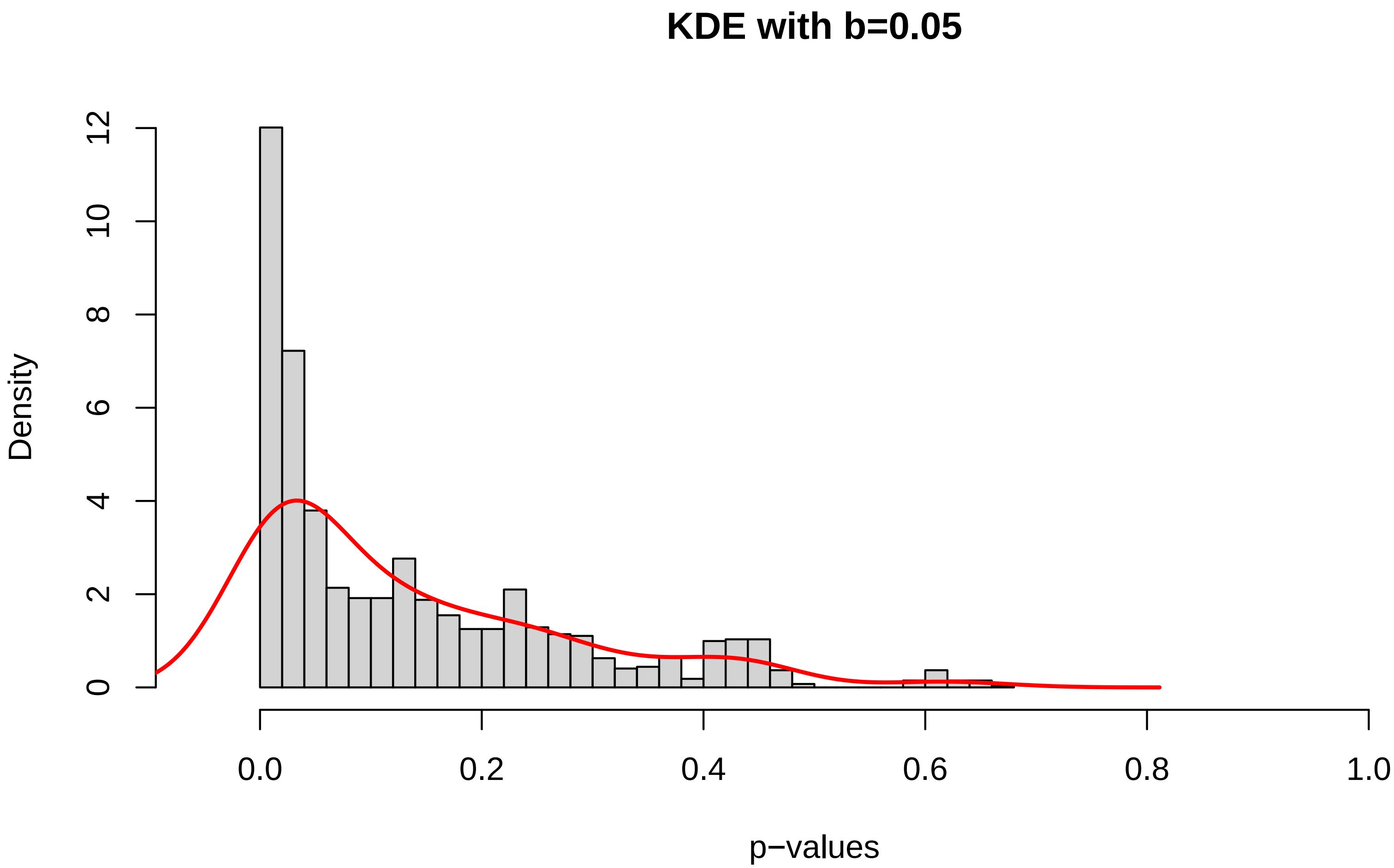
– Controls the smoothness or roughness of a density estimate.

$\Rightarrow$ Bandwidth selection bears danger of under- or oversmoothing.

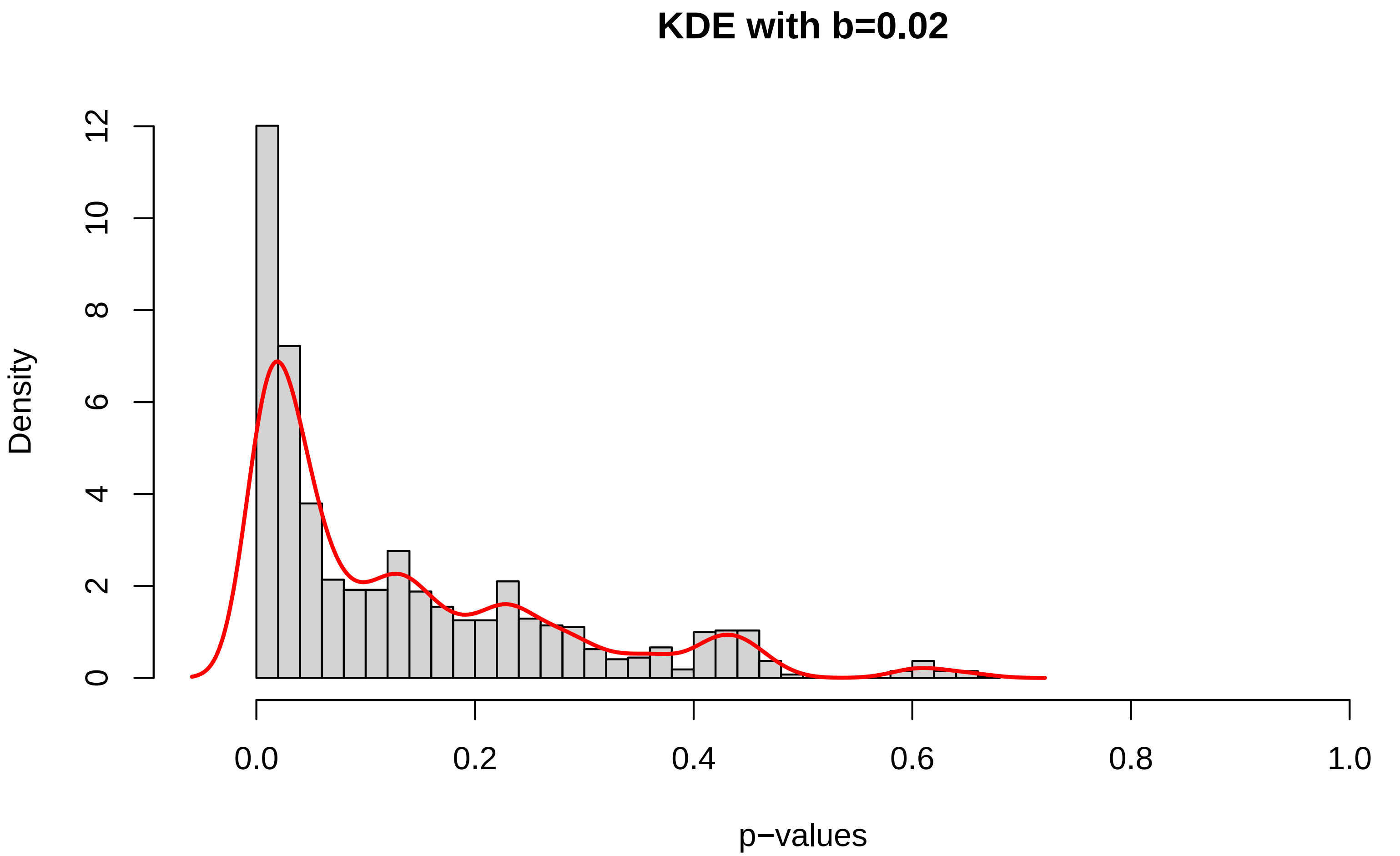
From over- to undersmoothing



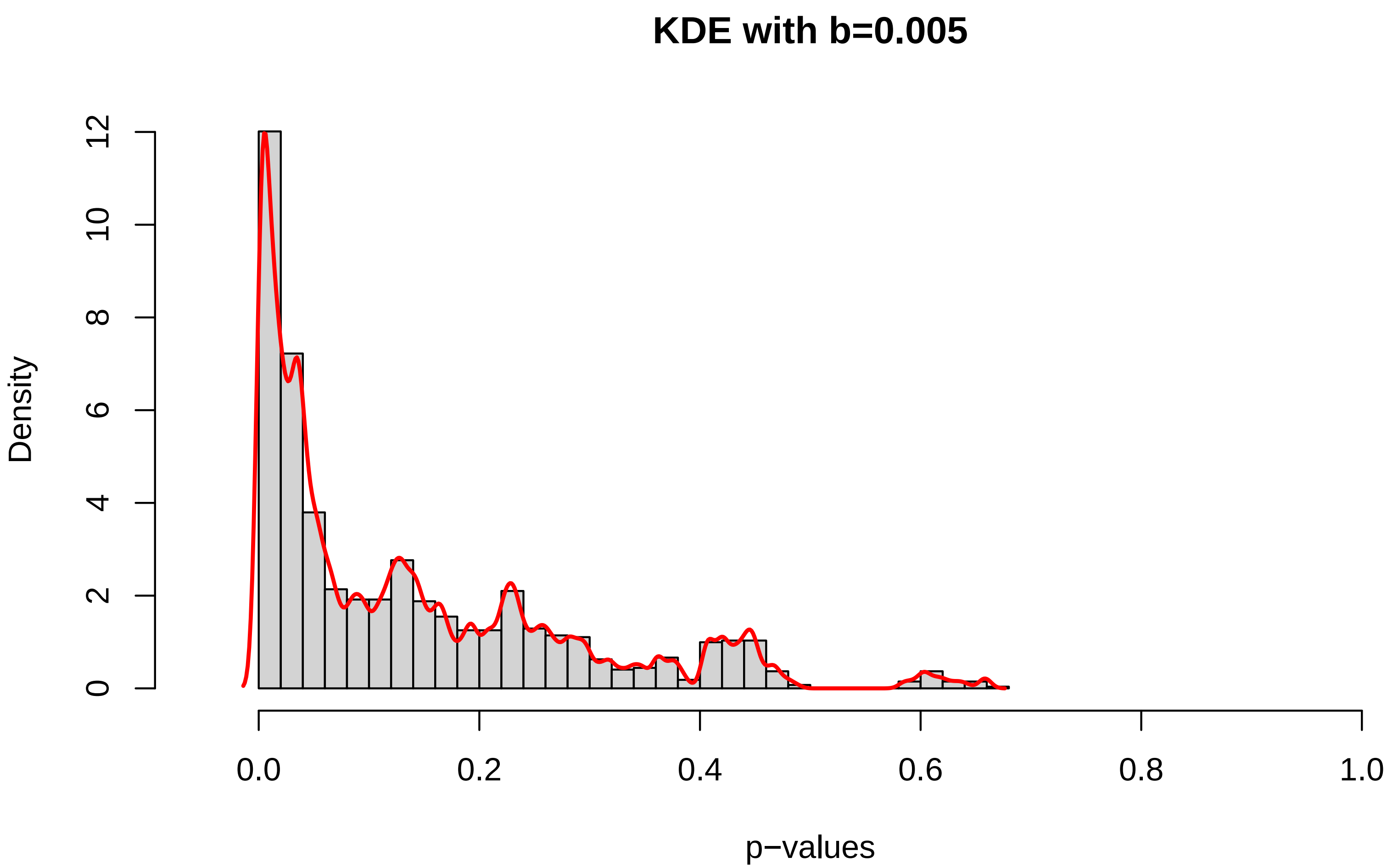
From over- to undersmoothing



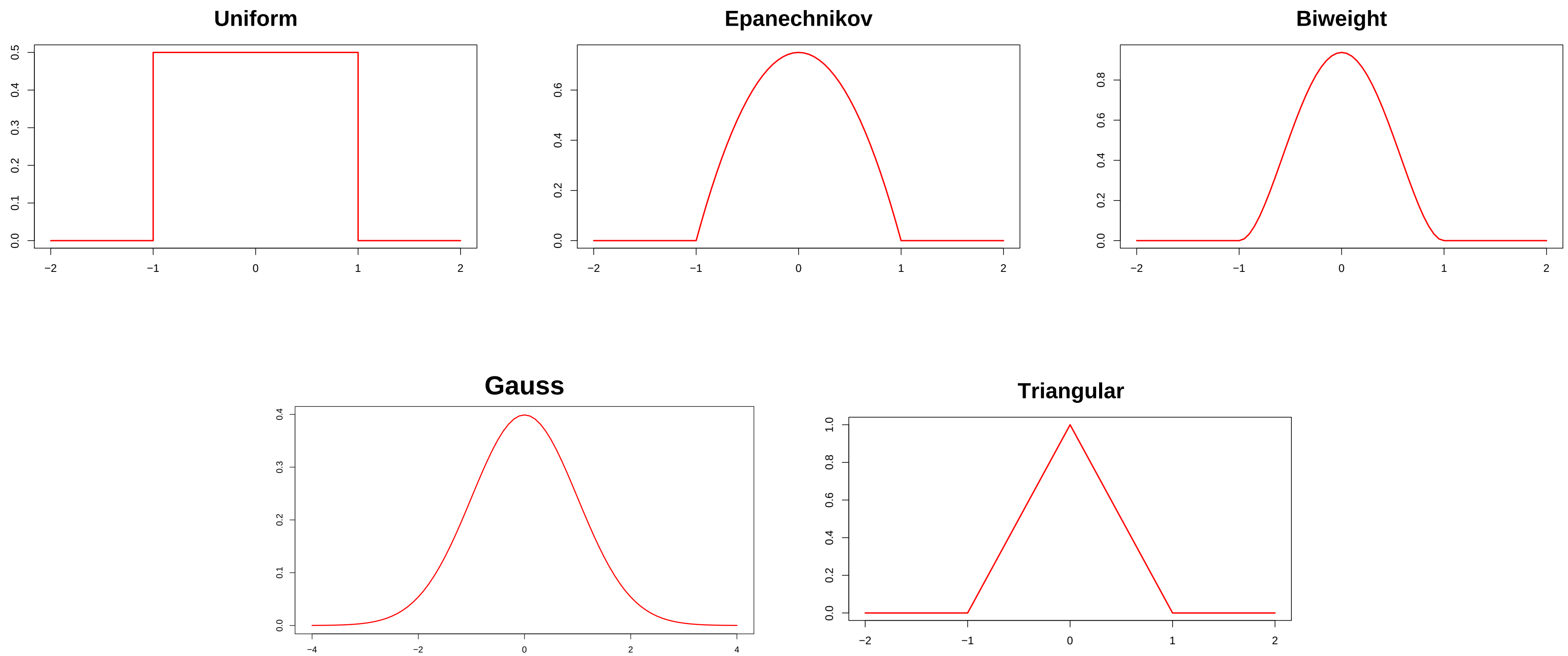
From over- to undersmoothing



From over- to undersmoothing



Some kernels



Some kernels

$$K(x,p) = \frac{(1-x^2)^p}{2^{2p+1}B(p+1,p+1)} 1_{\{|x|<1\}}$$

with $B(a,b) = \Gamma(a)\Gamma(b)/\Gamma(a+b)$.

- $p = 0$: Uniform kernel.
- $p = 1$: Epanechnikov kernel.
- $p = 2$: Biweight kernel.

Kernel efficiency

– Perfomance of kernel is measured by MISE (mean integrated squared error) or AMISE (asymptotic MISE).

– Epanechnikov kernel minimizes AMISE and is therefore optimal.

– Kernel efficiency is measured in comparison to Epanechnikov kernel.

Kernel	Efficiency
Epanechnikov	1.000
Biweight	0.994
Triangular	0.986
Normal	0.951
Uniform	0.930

$\Rightarrow$ Choice of kernel is not as important as choice of bandwidth.

Modified KDEs

– Local KDE: Bandwidth depends on x.

– Variable KDE: Smooth out the influence of points in sparse regions.

– Transformation KDE: If f is difficult to estimate (highly skewed, high kurtosis), transform data to gain a pdf that is easier to estimate.

Bandwidth selection

– Simple versus high-tech selection rules.

– Objective function: MISE/AMISE.

– R-function `density` offers several selection rules.

`bw.nrd0, bw.nrd`

– Normal scale rule.

– Assumes f to be normal and calculates the AMISE-optimal bandwidth in this setting.

– First guess but oversmoothes if f is multimodal or otherwise not normal.

`bw.ucv`

- Unbiased (or least squares) cross-validation.
- Estimates part of MISE by leave-one-out KDE and minimizes this estimator with respect to h.
- Problems: Several local minima, high variability.

`bw.bcv`

- Biased cross-validation.
- Estimation is based on optimization of AMISE instead of MISE (as `bw.ucv` does).
- Lower variance but reasonable bias.

`bw.SJ(method=c("ste", "dpi"))`

- The AMISE optimization involves the estimation of density functionals like integrated squared density derivatives.

- `dpi`: Direct plug-in rule. Estimates the needed functionals by KDE. Problem: Choice of pilot bandwidth.

- `ste`: Solve-the-equation rule. The pilot bandwidth depends on h.

Comparison of bandwidth selectors

- Simulation results depend on selected true densities.
- Selectors with pilot bandwidths perform quite well but rely on asymptotics $\Rightarrow$ less accurate for densities with "sharp features" (e.g. multiple modes).
- UCV has high variance but does not depend on asymptotics.
- BCV performs bad in several simulations.
- Authors' recommendation: DPI or STE better than UCV or BCV.

KDE with Epanechnikov kernel and DPI rule