

# Density and Distribution Estimation

Nathaniel E. Helwig

Assistant Professor of Psychology and Statistics  
University of Minnesota (Twin Cities)



Updated 04-Jan-2017

Copyright © 2017 by Nathaniel E. Helwig

# Outline of Notes

## 1) PDFs and CDFs

- Overview
- Estimation problem

## 2) Empirical CDFs

- Overview
- Examples

## 3) Histogram Estimates

- Overview
- Bins & breaks

## 4) Kernel Density Estimation

- KDE basics
- Bandwidth selection

# PDFs and CDFs

# Density Functions

Suppose we have some variable  $X \sim f(x)$  where  $f(x)$  is the **probability density function (pdf)** of  $X$ .

Note that we have two requirements on  $f(x)$ :

- $f(x) \geq 0$  for all  $x \in \mathcal{X}$ , where  $\mathcal{X}$  is the domain of  $X$
- $\int_{\mathcal{X}} f(x)dx = 1$

Example: normal distribution pdf has the form

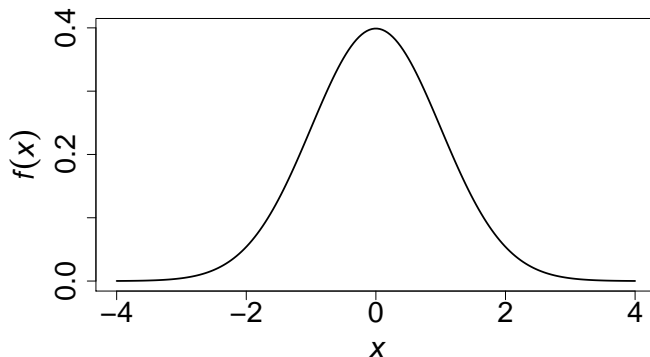
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

which is well-defined for all  $x, \mu \in \mathbb{R}$  and  $\sigma \in \mathbb{R}^+$ .

# Standard Normal Distribution

If  $X \sim N(0, 1)$ , then  $X$  follows a standard normal distribution:

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



# Probabilities and Distribution Functions (revisited)

Probabilities relate to the area under the pdf:

$$\begin{aligned}P(a \leq X \leq b) &= \int_a^b f(x)dx \\ &= F(b) - F(a)\end{aligned}\tag{2}$$

where

$$F(x) = \int_{-\infty}^x f(u)du\tag{3}$$

is the **cumulative distribution function (cdf)**.

Note:  $F(x) = P(X \leq x) \implies 0 \leq F(x) \leq 1$

# Problem of Interest

We want to estimate  $f(x)$  or  $F(x)$  from a sample of data  $\{x_i\}_{i=1}^n$ .

We will discuss three different approaches:

- Empirical cumulative distribution functions (ecdf)
- Histogram estimates
- Kernel density estimates



# Empirical Cumulative Distribution Function

# Empirical Cumulative Distribution Function

Suppose  $\mathbf{x} = (x_1, \dots, x_n)'$  with  $x_i \stackrel{\text{iid}}{\sim} F(x)$  for  $i \in \{1, \dots, n\}$ , and we want to estimate the cdf  $F$ .

The **empirical cumulative distribution function (ecdf)**  $\hat{F}_n$  is defined as

$$\hat{F}_n(x) = \hat{P}(X \leq x) = \frac{1}{n} \sum_{i=1}^n I_{\{x_i \leq x\}}$$

where

$$I_{\{x_i \leq x\}} = \begin{cases} 1 & \text{if } x_i \leq x \\ 0 & \text{otherwise} \end{cases}$$

denotes an indicator function.

# Some Properties of ECDFs

The ecdf assigns probability  $1/n$  to each value  $x_i$ , which implies that

$$\hat{P}_n(A) = \frac{1}{n} \sum_{i=1}^n I_{\{x_i \in A\}}$$

for any set  $A$  in the sample space of  $X$ .

For any fixed value  $x$ , we have that

- $E[\hat{F}_n(x)] = F(x)$
- $V[\hat{F}_n(x)] = \frac{1}{n} F(x)[1 - F(x)]$

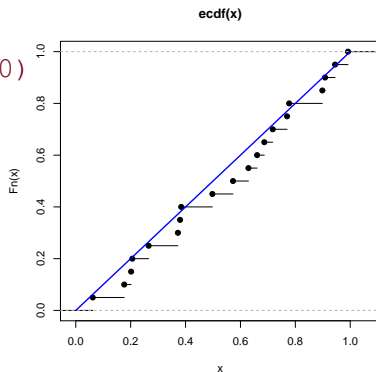
As  $n \rightarrow \infty$  we have that

$$\sup_{x \in \mathbb{R}} |\hat{F}_n(x) - F(x)| \xrightarrow{as} 0$$

which is the Glivenko-Cantelli theorem.

# ECDF: Example 1 (Uniform Distribution)

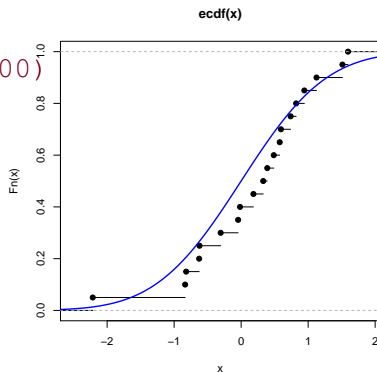
```
> set.seed(1)
> x = runif(20)
> xseq = seq(0,1,length=100)
> Fhat = ecdf(x)
> plot(Fhat)
> lines(xseq,punif(xseq),
+       col="blue",lwd=2)
```



Note that  $\hat{F}_n$  is a step-function estimate of  $F$  (with steps at  $x_i$  values).

# ECDF: Example 2 (Normal Distribution)

```
> set.seed(1)
> x = rnorm(20)
> xseq = seq(-3,3,length=100)
> Fhat = ecdf(x)
> plot(Fhat)
> lines(xseq,pnorm(xseq),
+       col="blue",lwd=2)
```



Note that  $\hat{F}_n$  is a step-function estimate of  $F$  (with steps at  $x_i$  values).

# ECDF: Example 3 (Bivariate Distribution)

Table 3.1 *An Introduction to the Bootstrap* (Efron & Tibshirani, 1993).

School	LSAT ( $y$ )	GPA ( $z$ )	School	LSAT ( $y$ )	GPA ( $z$ )
1	576	3.39	9	651	3.36
2	635	3.30	10	605	3.13
3	558	2.81	11	653	3.12
4	578	3.03	12	575	2.74
5	666	3.44	13	545	2.76
6	580	3.07	14	572	2.88
7	555	3.00	15	594	2.96
8	661	3.43			

Defining  $A = \{(y, z) : 0 < y < 600, 0 < z < 3.00\}$ , we have

$$\hat{P}_{15}(A) = (1/15) \sum_{i=1}^{15} I_{\{(y_i, z_i) \in A\}} = 5/15$$

# Histogram Estimates

# Histogram Definition

If  $f(x)$  is smooth, we have that

$$\begin{aligned}P(x - h/2 < X < x + h/2) &= F(x + h/2) - F(x - h/2) \\&= \int_{x-h/2}^{x+h/2} f(z)dz \approx hf(x)\end{aligned}$$

where  $h > 0$  is a small (positive) scalar called the **bin width**.

If  $F(x)$  were known, we could estimate  $f(x)$  using

$$\hat{f}(x) = \frac{F(x + h/2) - F(x - h/2)}{h}$$

but this isn't practical (b/c if we know  $F$  we don't need to estimate  $f$ ).



# Histogram Definition (continued)

If  $F(x)$  is unknown we could estimate  $f(x)$  using

$$\hat{f}_n(x) = \frac{\hat{F}_n(x + h/2) - \hat{F}_n(x - h/2)}{h} = \frac{\sum_{i=1}^n I_{\{x_i \in (x-h/2, x+h/2]\}}}{nh}$$

which uses previous formula with the ECDF in place of the CDF.

More generally, we could estimate  $f(x)$  using

$$\hat{f}_n(x) = \frac{\sum_{i=1}^n I_{\{x_i \in I_j\}}}{nh} = \frac{n_j}{nh}$$

for all  $x \in I_j = (c_j - h/2, c_j + h/2]$  where  $\{c_j\}_{j=1}^m$  are chosen constants.

# Histogram Bins/Breaks

Different choices for  $m$  and  $h$  will produce different estimates of  $f(x)$ .

Freedman and Diaconis (1981) method:

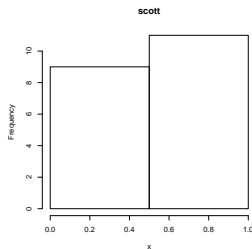
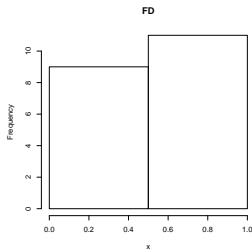
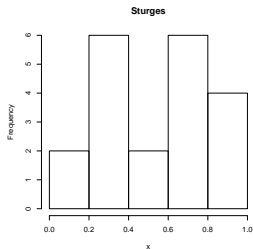
- Set  $h = 2(\text{IQR})n^{-1/3}$  where IQR = interquartile range
- Then divide range of data by  $h$  to determine  $m$

Sturges (1929) method (default in R's `hist` function):

- Set  $m = \lceil \log_2(n) + 1 \rceil$  where  $\lceil x \rceil$  denotes ceiling function
- May oversmooth for non-normal data (i.e., use too few bins)

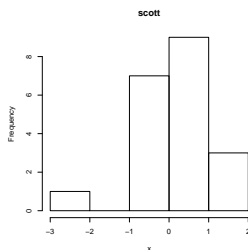
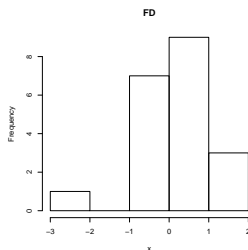
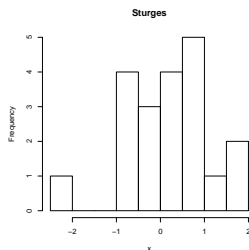
# Histogram: Example 1

```
> par(mfrow=c(1,3))  
> set.seed(1)  
> x = runif(20)  
> hist(x,main="Sturges")  
> hist(x,breaks="FD",main="FD")  
> hist(x,breaks="scott",main="scott")
```



# Histogram: Example 2

```
> par(mfrow=c(1,3))  
> set.seed(1)  
> x = rnorm(20)  
> hist(x,main="Sturges")  
> hist(x,breaks="FD",main="FD")  
> hist(x,breaks="scott",main="scott")
```



# Kernel Density Estimation

# Kernel Function: Definition

A **kernel function**  $K$  is a function such that...

- $K(x) \geq 0$  for all  $-\infty < x < \infty$
- $K(-x) = K(x)$
- $\int_{-\infty}^{\infty} K(x)dx = 1$

In other words,  $K$  is a non-negative function that is symmetric around 0 and integrates to 1.

# Kernel Function: Examples

A simple example is the uniform (or box) kernel:

$$K(x) = \begin{cases} 1 & \text{if } -1/2 \leq x < 1/2 \\ 0 & \text{otherwise} \end{cases}$$

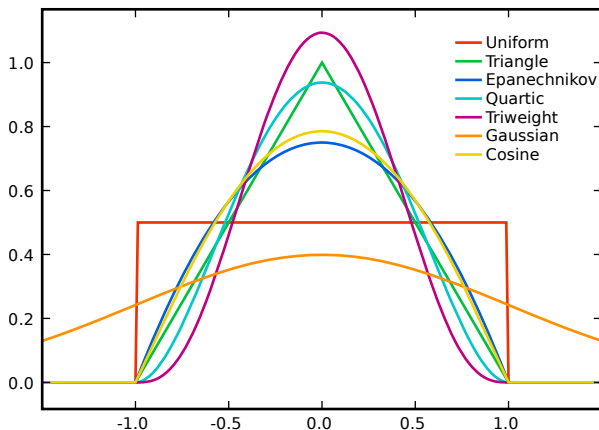
Another popular kernel function is the Normal kernel (pdf) with  $\mu = 0$  and  $\sigma$  fixed at some constant:

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

We could also use a triangular kernel function:

$$K(x) = 1 - |x|$$

# Kernel Function: Visualization



From <http://upload.wikimedia.org/wikipedia/commons/4/47/Kernels.svg>



# Scaled and Centered Kernel Functions

If  $K$  is a kernel function, then the scaled version of  $K$

$$K_h(x) = \frac{1}{h} K\left(\frac{x}{h}\right)$$

is also a kernel function, where  $h > 0$  is some positive scalar.

We can center a scaled kernel function at any data point  $x_i$ , such as

$$K_h^{(x_i)}(x) = \frac{1}{h} K\left(\frac{x - x_i}{h}\right)$$

to create a kernel function that is symmetric around  $x_i$ .

# Kernel Density Estimate: Definition

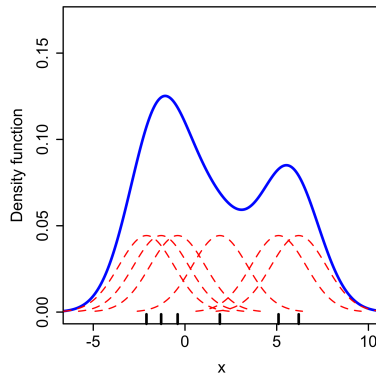
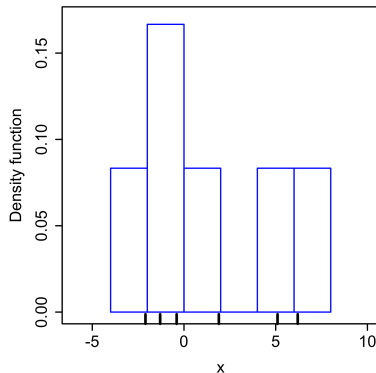
Given a random sample  $x_i \stackrel{\text{iid}}{\sim} f(x)$ , the **kernel density estimate** of  $f$  is

$$\begin{aligned}\hat{f}(x) &= \frac{1}{n} \sum_{i=1}^n K_h^{(x_i)}(x) \\ &= \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right)\end{aligned}$$

where  $h$  is now referred to as the **bandwidth** (instead of bin width).

Using the uniform (box) kernel, the KDE reduces to histogram estimate using ECDF in place of CDF.

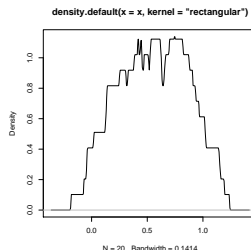
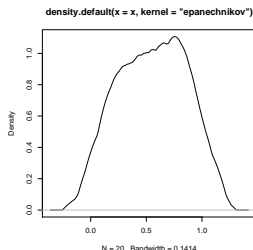
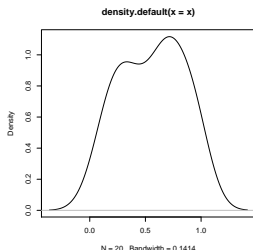
# Kernel Density Estimate: Visualization



From [http://en.wikipedia.org/wiki/Kernel\\_density\\_estimation](http://en.wikipedia.org/wiki/Kernel_density_estimation)

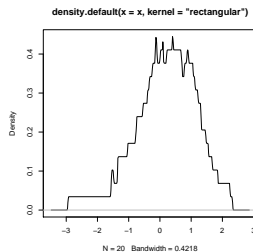
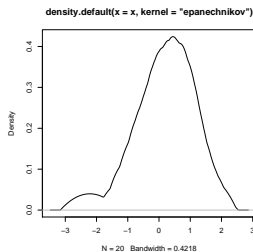
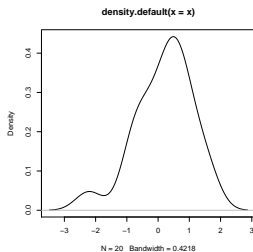
# Kernel Density Estimate: Example 1

```
> set.seed(1)
> x = runif(20)
> kde = density(x)
> plot(kde)
> kde = density(x, kernel="epanechnikov")
> plot(kde)
> kde = density(x, kernel="rectangular")
> plot(kde)
```



# Kernel Density Estimate: Example 2

```
> set.seed(1)
> x = rnorm(20)
> kde = density(x)
> plot(kde)
> kde = density(x, kernel="epanechnikov")
> plot(kde)
> kde = density(x, kernel="rectangular")
> plot(kde)
```

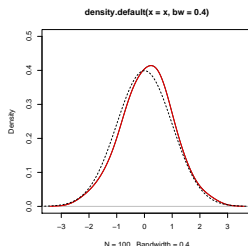


# KDE with Gaussian Kernel

```
kdenorm <- function(x,bw,q=NULL) {  
  if(is.null(q)) {  
    q = seq(min(x)-3*bw, max(x)+3*bw, length.out=512)  
  }  
  nx = length(x)  
  nq = length(q)  
  xmat = matrix(q,nq,nx) - matrix(x,nq,nx,byrow=TRUE)  
  denall = dnorm(xmat/bw) / bw  
  denhat = apply(denall,1,mean)  
  list(x=q, y=denhat, bw=bw)  
}
```

# Kernel Density Estimate: Example 3

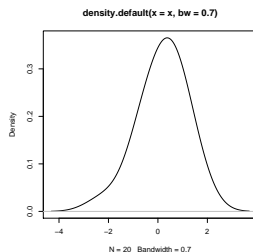
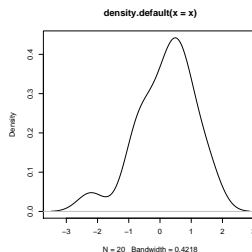
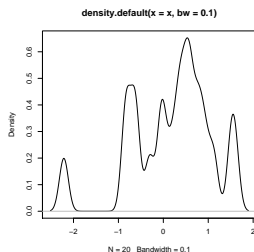
```
> dev.new(width=6,height=6,noRStudioGD=TRUE)
> set.seed(1)
> x = rnorm(100)
> plot(density(x,bw=0.4),ylim=c(0,0.5))
> kde = kdenorm(x,bw=0.4)
> lines(kde,col="red")
> lines(seq(-4,4,l=500),dnorm(seq(-4,4,l=500)),lty=2)
```



# The Bandwidth Problem

Kernel density estimate  $\hat{f}(x)$  requires us to select the bandwidth  $h$ .

Different values of  $h$  can produce vastly different estimates  $\hat{f}(x)$ .





# Mean Integrated Squared Error

The Mean Integrated Squared Error (MISE) between a function  $f$  and its estimate  $\hat{f}_h$  is

$$MISE(f, \hat{f}_h) = E \left\{ \int (f - \hat{f}_h)^2 \right\}$$

For a kernel function  $K$ , the asymptotic MISE is

$$\frac{\int K^2}{nh} + \frac{\sigma_K^4 h^4 \int (f'')^2}{4}$$

where  $\sigma_K^2 = \int x^2 K(x) dx$  is the kernel variance.

# Mean Integrated Squared Error (continued)

The Mean Integrated Squared Error (MISE) can be written as

$$\begin{aligned} MISE(f, \hat{f}_h) &= E \left\{ \int (f - \hat{f}_h)^2 \right\} \\ &= E \int_{-\infty}^{\infty} f(x)^2 dx - 2E \int_{-\infty}^{\infty} f(x) \hat{f}_h(x) dx + E \int_{-\infty}^{\infty} \hat{f}_h(x)^2 dx \end{aligned}$$

To minimize the MISE, we want to minimize

$$E \int_{-\infty}^{\infty} \hat{f}_h(x)^2 dx - 2E \int_{-\infty}^{\infty} f(x) \hat{f}_h(x) dx$$

with respect to  $\hat{f}_h$  (really it is with respect to the bandwidth  $h$ ).

# Fixed Bandwidth Methods

Goal: find some optimal  $h$  to use in the KDE  $\hat{f}$ .

A typical choice of bandwidth is:  $h = cn^{-1/5} \min(\hat{\sigma}, (IQR)1.34)$

- Set  $c = 0.90$  to use `bw="nrd0"` in R's density function (default)
- Set  $c = 1.06$  to use `bw="nrd"` in R's density function
- Assumes  $f$  is normal, but can provide reasonable bandwidths for non-normal data

Could also use cross-validation where we estimate  $f$  holding out  $x_i$

- Can do unbiased MISE minimization (R's `bw="ucv"`)
- Or can do biased MISE minimization (R's `bw="bcv"`)
- Need to consider bias versus variance trade-off

# MISE and Cross-Validation

An unbiased estimate of the first term is given by

$$\int_{-\infty}^{\infty} \hat{f}_h(x)^2 dx$$

which is evaluated using numerical integration techniques.

It was shown by Rudemo (1982) and Bowman (1984) that an unbiased estimate of the second term is

$$-\frac{2}{n} \sum_{i=1}^n \hat{f}_h^{(i)}(x_i)$$

where  $\hat{f}_h^{(i)}(x) = \frac{1}{(n-1)h} \sum_{j \neq i} K\left(\frac{x-x_j}{h}\right)$  is leave-one-out estimate of  $\hat{f}_h$ .

# MISE and Cross-Validation (in practice)

Note that we can write

$$\begin{aligned}\hat{f}_h^{(i)}(x_i) &= \frac{1}{(n-1)h} \sum_{j \neq i} K\left(\frac{x_i - x_j}{h}\right) \\ &= \frac{n}{n-1} \left[ \hat{f}_h(x_i) - \frac{K(0)}{nh} \right]\end{aligned}$$

which implies that our unbiased CV problem is

$$\min_h \int_{-\infty}^{\infty} \hat{f}_h(x)^2 dx - \frac{2}{n-1} \sum_{i=1}^n \left[ \hat{f}_h(x_i) - \frac{K(0)}{nh} \right]$$

# MISE and Cross-Validation (R code)

Note: this code is for demonstration only!

```
intfun <- function(ix,x,bw) kdenorm(x,bw,ix)$y^2
kdecv <- function(bw,x){
  lo = min(x)-3*bw
  up = max(x)+3*bw
  ival = integrate(intfun,x=x,bw=bw,lower=lo,upper=up)$value
  nx = length(x)
  ival - (2/(nx-1))*sum( kdenorm(x,bw,x)$y - dnorm(0)/(nx*bw) )
}
```

Could use `optimize` function to find minimum:

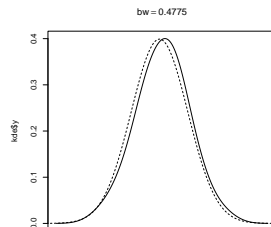
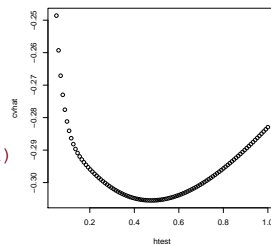
```
> set.seed(1)
> x = rnorm(100)
> optimize(kdecv,interval=c(0.05,1),x=x)$minimum
[1] 0.4756956
```

# Cross-Validation Example

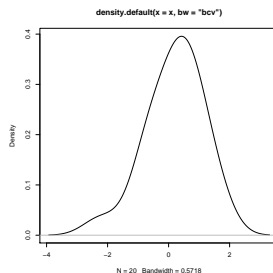
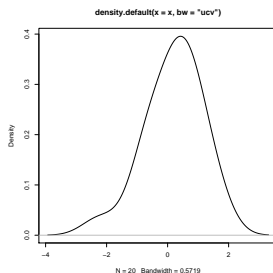
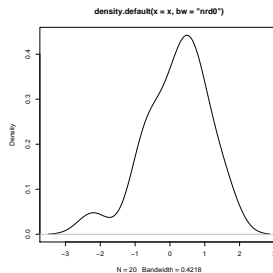
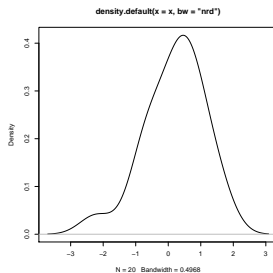
```

> dev.new(width=6,height=12,noRStudioGD=TRUE)
> par(mfrow=c(2,1))
> set.seed(1)
> xseq=seq(-4,4,length.out=500)
> x=rnorm(100)
> cvhat=rep(0,101)
> htest=seq(0.05,1,length.out=101)
> for(j in 1:101)cvhat[j]=kdecv(htest[j],x)
> plot(htest,cvhat)
> bwwhat=htest[which.min(cvhat)]
> kde=kdenorm(x,bw=bwwhat)
> plot(kde,main=bquote(bw==.(kde$bw)),type="l")
> lines(xseq,dnorm(xseq),lty=2)

```



# Fixed Bandwidths Examples





# Variable Bandwidth Methods

Previous rules use a fixed (constant) bandwidth  $h$  for all points  $x$ .

- Only reasonable if we have relatively uniform spread of  $x_i$  points

Having lots of data around  $x_i$  should lead to better estimate of  $f(x_i)$ .

- Need to use a smaller bandwidth for dense placements of data

Having little data around  $x_i$  should lead to worse estimate of  $f(x_i)$ .

- Need to use a larger bandwidth for sparse placements of data

# Variable Bandwidth KDE Form

Given a random sample  $x_i \stackrel{\text{iid}}{\sim} f(x)$ , the variable bandwidth KDE of  $f$  is

$$\begin{aligned}\hat{f}(x) &= \frac{1}{n} \sum_{i=1}^n K_{h_i}^{(x_i)}(x) \\ &= \frac{1}{n} \sum_{i=1}^n \frac{1}{h_i} K\left(\frac{x - x_i}{h_i}\right)\end{aligned}$$

where  $h_i$  is the *variable bandwidth*.

Following Silverman (1986), we typically assume that  $h_i = \lambda_i h$  where

- $h$  is fixed bandwidth from a pilot estimate  $\hat{f}_p$
- $\lambda_i = \left[ \hat{f}_p(x_i) / \left( \prod_{j=1}^n \hat{f}_p(x_j) \right)^{1/n} \right]^{-\alpha}$  where  $\alpha$  is sensitivity parameter

# Variable Bandwidth KDE in R

Can use `akj` function in `quantreg` package.

```
> library(quantreg)
> set.seed(1)
> x=rnorm(20)
> xs=sort(x)
> xseq=seq(-5,5,length=100)
> kde=akj(xs,xseq)
> plot(xseq,kde$dens)
```

