STAT 621 Lecture Notes
Nonparametric Density Estimation

Objectives and Examples: In many situations, it is of interest to estimate the exact distribution that generated a random sample. Often we are willing to make some general assumptions about that distribution, and we use the data to estimate specific parameters. For example,

Sometimes, however, we want to make as few assumptions as possible about the form of the distribution. In this case, the objective is to estimate the entire function $f(x)$. Often the goal of a study is to determine what characteristics are possessed by a variable's distribution. Interesting questions might include:

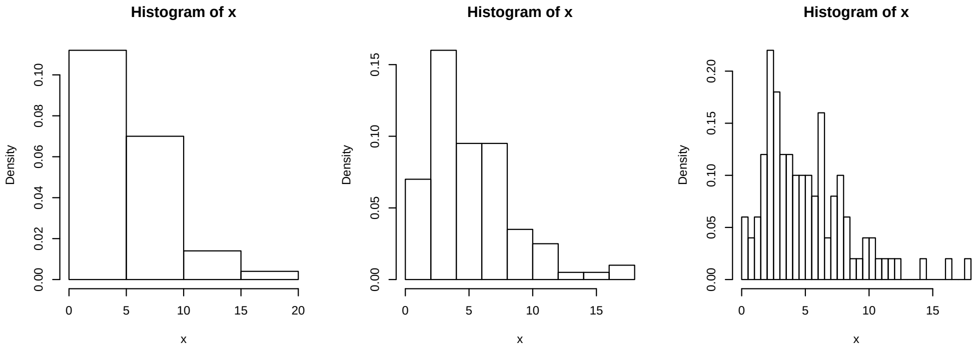
A Few Examples:

Histograms: Consider a random sample $X_1, \dots, X_n$. To construct a histogram, we group the data into M intervals, each having width h. Let B_j denote the jth interval and let m_j be the number of observations that fall in interval B_j. The height of the histogram at a point x is determined by the number of observations that fall in the same interval as x. We can write this mathematically as

$$\widehat{f}(x) = \frac{1}{nh} \sum_{j=1}^{M} m_j I(x \in B_j)$$

where $I(.)$ is the indicator function. Why the divisor nh?

Example: The following histograms are all plotted from the same set of data, 50 Chisquare(5) random variables. Bin width decreases from left to right.



Concerns:

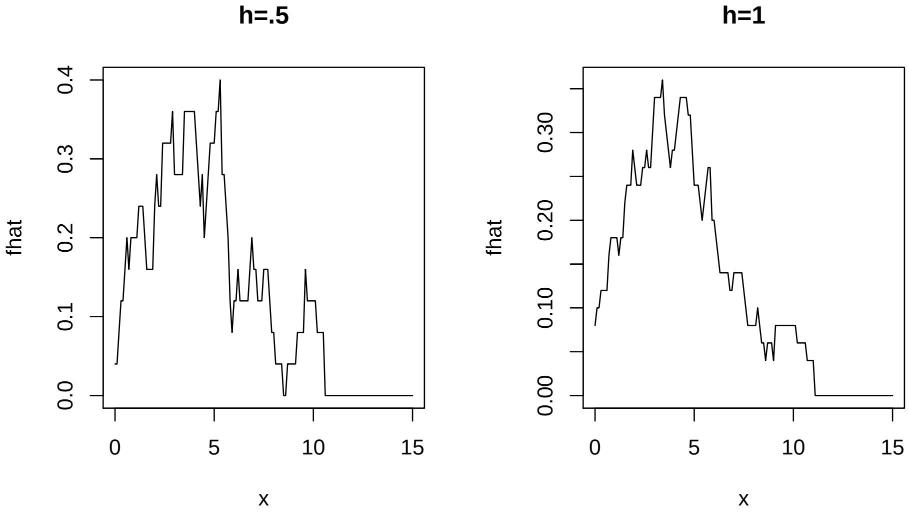
- Selecting bin width h
- Placement of bins is arbitrary – density at x may ignore nearby values.
- Estimate of density is zero in any interval that contains no observations
- It's not smooth!
- Others?

Slight Improvement: Local Estimator. Here the estimated density at a point x is computed by counting the number of observations that fall within a certain distance h of x.

$$\widehat{f}(x) = \frac{1}{2nh} \sum_{i=1}^{n} I\left(|x - X_i| < h\right)$$

Example: The local histogram estimator for the Chisquare(5) distribution. Still not very smooth.

```
n=50
X=rchisq(n,5)
x=seq(0,15,.1)
h=1
fhat=rep(NA,length(x))
for (j in 1:length(x)) fhat[j]=sum(abs(x[j]-X)<h)/(n*h)
plot(x,fhat,type=’l’)
title("h=1")
```



Kernel Density Estimators

A kernel density estimator takes the local approach a step further. The indicator function $I(.)$ is replaced with a smooth function,

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

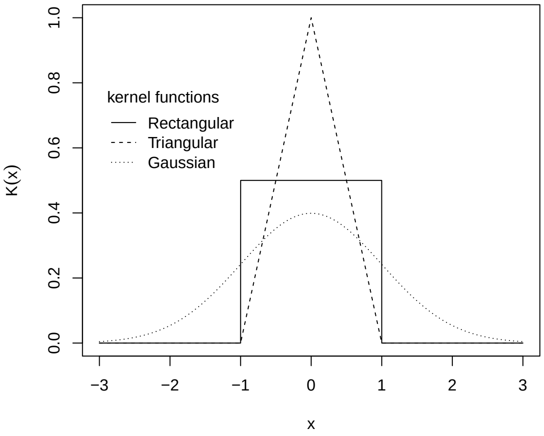
The kernel function $K(.)$ is usually selected to be a probability density function with the following properties:

- Strictly positive
- Symmetric
- Twice-differentiable

These properties ensure that $\widehat{f}(x)$ will also be a pdf.

Here are some examples of kernel functions. Code from *Handbook of Statistical Analysis Using R* on the R-project website.

```
rec <- function(x) (abs(x) < 1) * 0.5
tri <- function(x) (abs(x) < 1) * (1 - abs(x))
gauss <- function(x) 1/sqrt(2*pi) * exp(-(x^2)/2)
x <- seq(from = -3, to = 3, by = 0.001)
plot(x, rec(x), type = "l", ylim = c(0,1), lty = 1,  ylab = expression(K(x)))
lines(x, tri(x), lty = 2)
lines(x, gauss(x), lty = 3)
legend(-3, 0.8, legend = c("Rectangular", "Triangular", "Gaussian"), lty = 1:3,
  + title = "kernel functions",    bty = "n")
```



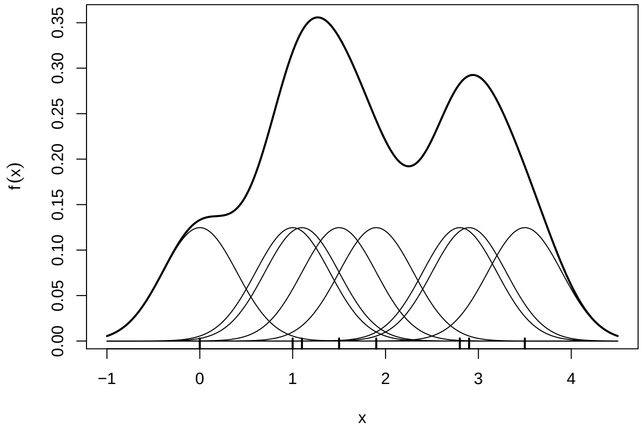
A Closer Look. Let's consider the estimator when the kernel function is the standard normal pdf, also called the *Gaussian Kernel.* A single term is

$$\frac{1}{h}\phi\left(\frac{x - X_i}{h}\right).$$

What is this function? What happens as we sum it over all the $X's$?

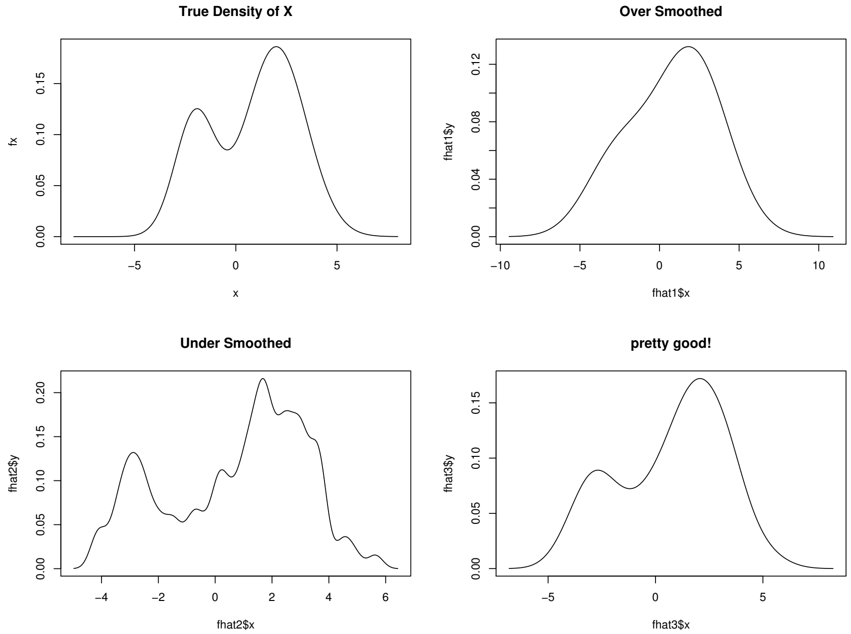
Here's a picture. Again code from *Handbook of Statistical Analysis Using R.*

```
x <- c(0, 1, 1.1, 1.5, 1.9, 2.8, 2.9, 3.5)
n <- length(x)
xgrid <- seq(from = min(x) - 1, to = max(x) + 1, by = 0.01)
h <- 0.4
bumps <- sapply(x, function(a) gauss((xgrid - a)/h)/(n *h))
plot(xgrid, rowSums(bumps), ylab = expression(hat(f)(x)), type = "l", xlab = "x", lwd = 2)
rug(x, lwd = 2)
out <- apply(bumps, 2, function(b) lines(xgrid, b))
```



The Bandwidth

As with a histogram, h has a large influence on the estimated density function. It is often referred to as the smoothing parameter. A few examples:



True Density of X

Over Smoothed

Under Smoothed

pretty good!

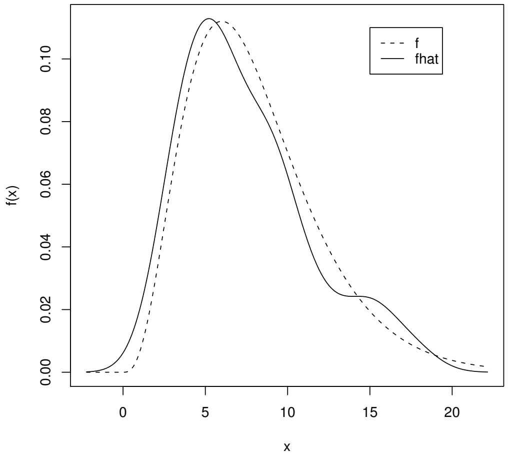
Evaluating the Estimator and Selecting a Bandwidth

Recall the general definition of the Mean Squared Error of an estimator $\widehat{\theta}$.

For our estimator $\widehat{f}(x)$, at a given point x, we could compute the Mean Squared Error,

However we'd rather have a global measure of the performance for all x's. This is given by the Mean Integrated Squared Error, Consider this example. Use the picture to describe the MISE.

```
 x=rchisq(50,8)
 fhat=density(x)
 names(fhat)
[1] "x"      "y"      "bw"     "n"  "call"  "data.name"   "has.na"
 f=dchisq(fhat$x,8)
 plot(fhat$x,fhat$y,type="l",ylim=c(0,max(f,fhat$y)),xlab="x",ylab="f(x)")
 lines(fhat$x,dchisq(fhat$x,8),lty=2)
 legend(15,.11,c("f","fhat"),lty=c(2,1))
```



One can show that the MISE of $\widehat{f}(x)$ can be written as

$$\text{MISE} = \frac{1}{nh}\int K^2(x)dx + \frac{h^4}{4}\mu_2\int\{f^{”}(x)\}^2dx$$

where $\mu_2 = \int x^2K(x)dx$. Consider this expression. We would like for MISE $\to 0$ as $n \to \infty$. How can we control MISE? Note the variance-bias trade-off.

Theoretically we can select h so that MISE $\rightarrow 0$ as $n \rightarrow \infty$. Specifically,

$$h_{opt} = g(f^{"}(x))n^{-1/5}.$$

where g is a known function. Discuss: Is this practical? What does this expression tell you about necessary sample sizes for these kind of estimators?

Some Options:

(1) Normal Reference Bandwidth: Use a normal density as a replacement in $g\{f^{"}(x)\}$. In that case, we find

$$h_{opt} = 1.06\widehat{\sigma}n^{-1/5}$$

This may not be appropriate if the true density is far from normal.

(2) Cross Validation: Choose h to minimize the expression

$$M(h) = \int \widehat{f}^2(x)dx - 2n^{-1}\sum \widehat{f}_{(-i)}(X_i).$$

(3) Bootstrap bandwidth estimator.

Example: The R data set `faithful` contains the waiting time (min) between eruptions and the duration (min) of eruptions for the Old Faithful geyser in Yellowstone National Park.

```
head(faithful)
  eruptions waiting
1     3.600      79
2     1.800      54
3     3.333      74
4     2.283      62
5     4.533      85
6     2.883      55
...
Etc.
```

First let's estimate the density using the normal reference bandwith. We'll try a few different kernel functoins. The R function `density` computes the KDE. The option `bw="nrd"` requests the normal reference bandwidth. Use the `kernel=""` option to choose a kernel. Choices are `"gaussian"`, `"epanechnikov"`, `"rectangular"`, `"triangular"`, `"biweight"`,`"cosine"`, `"optcosine"`.

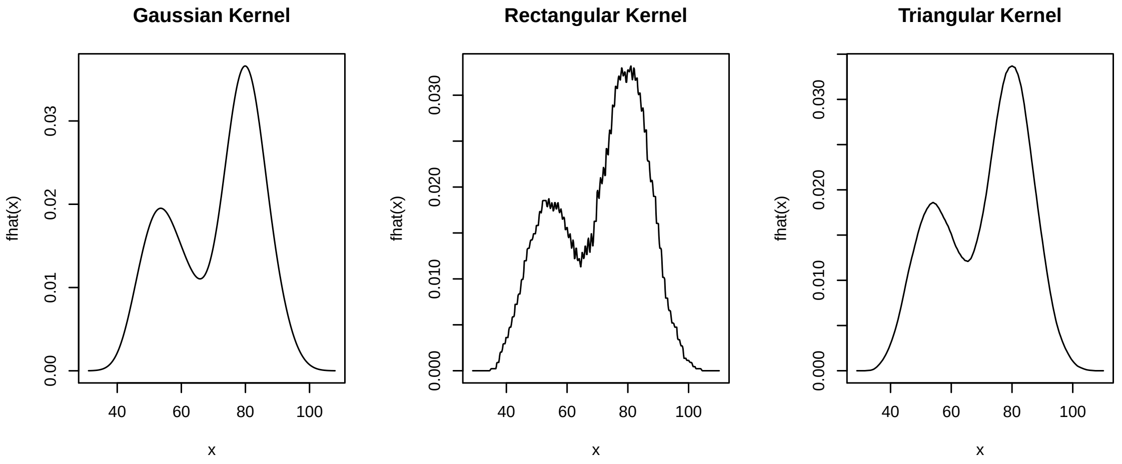
```
par(mfrow=c(1,3))
fhat1.gauss=density(faithful$waiting, kernel="gaussian", bw="nrd")
names(fhat1.gauss)

[1] "x"         "y"         "bw"        "n"         "call"      "data.name" "has.na"

plot(fhat1.gauss$x,fhat1.gauss$y,type=’l’,xlab=’x’,ylab=’fhat(x)’)
title("Gaussian Kernel")

fhat1.rect=density(faithful$waiting, kernel="rectangular", bw="nrd")
plot(fhat1.rect$x,fhat1.rect$y,type=’l’,xlab=’x’,ylab=’fhat(x)’)
title("Rectangular Kernel")

fhat1.tri=density(faithful$waiting, kernel="triangular", bw="nrd")
plot(fhat1.tri$x,fhat1.tri$y,type=’l’,xlab=’x’,ylab=’fhat(x)’)
title("Triangular Kernel")
```



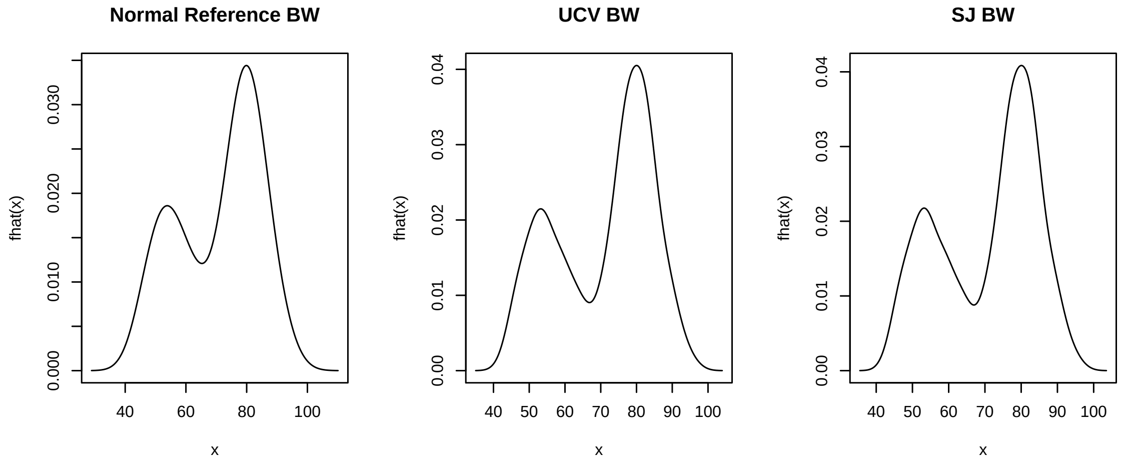
Now let’s try some different bandwidth estimators. We’ll use the Gaussian kernel. Recall our estimate `fhat1.gauss` used this kernel with the normal reference bandwidth. Let’s also try cross validation. The option `bw="bw.ucv"` implements (unbiased) cross validation. Another option is `bw="bw.SJ"` implements the methods of Sheather and Jones (1991) to select the bandwidth using pilot estimation of derivatives.

```
# First estimate each function
fhat1.gauss=density(faithful$waiting, kernel="gaussian", bw="nrd")
fhat2.gauss=density(faithful$waiting, kernel="gaussian", bw="ucv")
fhat3.gauss=density(faithful$waiting, kernel="gaussian", bw="SJ")

#  Look at bandwidth estimates
data.frame(fhat1.gauss$bw, fhat2.gauss$bw, fhat3.gauss$bw)

  fhat1.gauss.bw fhat2.gauss.bw fhat3.gauss.bw
       4.696458       2.658185       2.504371

# plot the estimates
par(mfrow=c(1,3))
plot(fhat1.gauss$x,fhat1.gauss$y,type=’l’,xlab=’x’,ylab=’fhat(x)’)
title("Normal Reference BW")
plot(fhat2.gauss$x,fhat2.gauss$y,type=’l’,xlab=’x’,ylab=’fhat(x)’)
title("UCV BW")
plot(fhat3.gauss$x,fhat3.gauss$y,type=’l’,xlab=’x’,ylab=’fhat(x)’)
title("SJ BW")
```


Normal Reference BW
fhat(x)
x
UCV BW
fhat(x)
x
SJ BW
fhat(x)
x