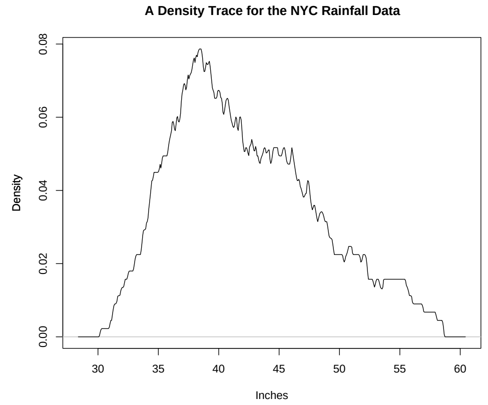
Statistics 120
Density Traces



Histograms

- Traditional histograms work with a fixed set of histogram cells.
- The height of each histogram bar provides a measure of the density of data values within the bar.
- The notion of data density is very useful and worth generalising.

The Quality of Histograms

- A moving-bar histogram provides information on $h(x)$ at all x values.
- A fixed bar histogram provides information on $h(x)$ only at its cell midpoints.
- Comparing both kinds of histograms shows just how much information is lost by a standard histogram.

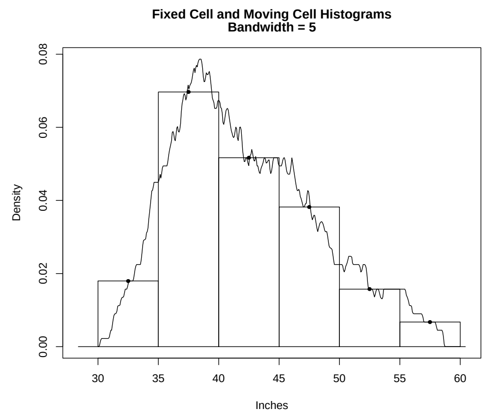
Histogram Density Estimates

- The height of bar in a relative frequency histogram provides a measure of the density of data points in the histogram cell that the bar is drawn over.
- If a cell centred at x has width w and contains k data points, the height of the bar is

$$h(x) = \frac{k}{n} \times \frac{1}{w}$$

which is directly proportional to the density of points in the interval.

$$\text{data density} = \frac{k}{w}$$



Terminology

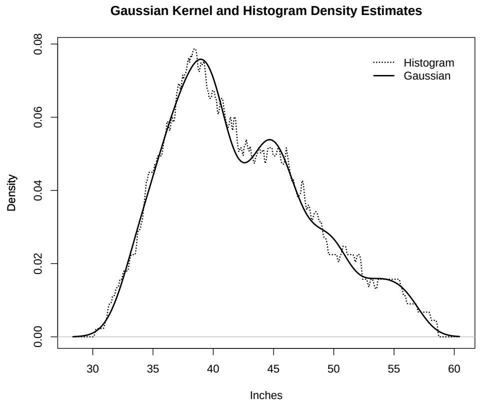
- The function $h(x)$ is called the *histogram estimate of data density*.
- The value of w is called the *bandwidth* of the estimate.
- $h(x)$ is defined for every x value.
- The area under $h(x)$ is 1.
- The graph of $h(x)$ plotted against x is called a *density trace*.

Lack of Smoothness

- Histogram density estimates have a very rough appearance.
- This is because points enter and leave the window (histogram cell) suddenly and this causes jumps in $h(x)$.
- When a point is within a distance $w/2$ of x, it contributes an amount $1/nw$ to the value of $h(x)$.
- When it is a greater distance away its contribution is 0.
- It is this sudden change in the contribution of points to $h(x)$ which makes histogram density traces so rough.

Smooth Density Estimates

- It is possible to make density traces smoother by changing the way points make a contribution to $h(x)$.
- Smooth density estimates work by making the contribution a point makes to $h(x)$ depend on its distance to x. A small distance means a large contribution and vice versa.

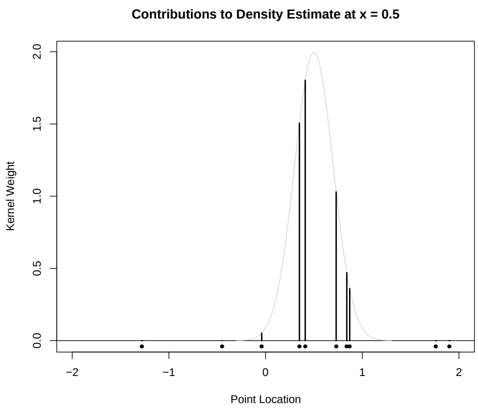


Smooth Density Estimates

- One way to achieve smoothness is to make the contribution of a value at y to $h(x)$ be $k(y-x)$, where $k(u)$ is a function which has a peak at $u = 0$ and falls away to zero as u increases in magnitude.
- The function $k(u)$ is called the kernel of the density estimate.
- The function $k(u)$ is usually taken to be symmetric about 0, positive, and to integrate to 1.
- The most common kernel function is the normal probability density function.

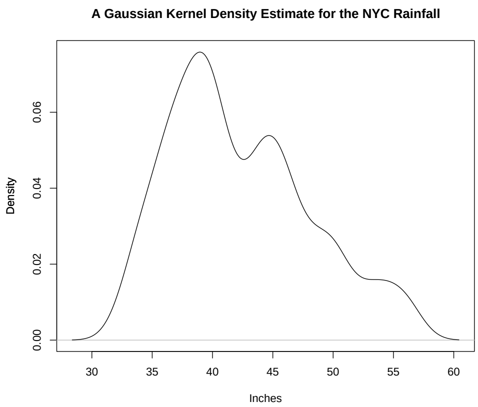
Bandwidth

- It is possible to vary the appearance of a histogram by varying its cell width.
- A similar effect is possible with kernel density estimates by varying how spread-out the kernel function is.
- The spread of a kernel is controlled by a scale parameter which is also called the bandwidth.



R Functions

- The R function `density` computes density estimates.
- A better option is to use the R "density" library (installed in the labs and available from the class web site).
- The library contains a function called `dtrace` which can be used to compute density traces.
- The estimates produced `dtrace` by can be plotted with the *plot* function.



R Examples

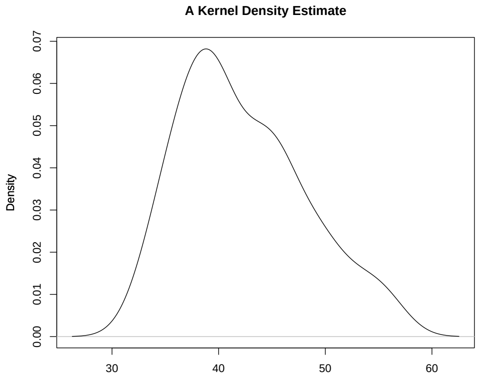
It is simple to construct density plots using R.

Long hand . . .

```
> d = dtrace(rain.nyc)
> plot(d, main = "A Kernel Density Estimate")
```

Or equivalently . . .

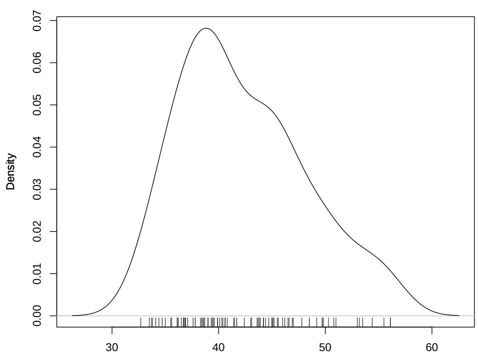
```
> plot(dtrace(rain.nyc))
> title(main = "A Kernel Density Estimate")
```



Showing the Data

The function *rug* can be used to draw vertical lines at the bottom of the plot at the locations of the data values (the result looks a little like the tassels on a Persian rug).

```
> plot(dtrace(rain.nyc))
> rug(x)
```



Control of Bandwidth

The default bandwidth chosen by R often produces quite good results, but sometimes it can be useful to try alternative values to see what the effect of more or less smoothing might be.

Example: Geyser Eruptions

The default bandwidth chosen by R often produces quite good results, but sometimes it can be useful to try alternative values to see what the effect of more or less smoothing might be.

We will look at bandwidth choice using some data on the length of eruptions of the “Old Faithful” geyser in Yellowstone National Park in the USA.

```
> data(faithful)
> attach(faithful)
```

Bandwidth for the Geyser Eruptions

We can leave R free to choose the bandwidth and determine the chosen bandwidth as follows:

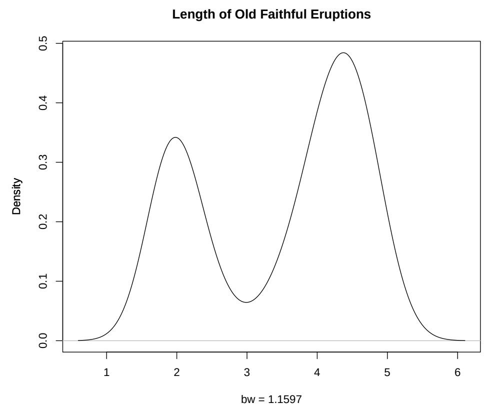
```
> d = dtrace(eruptions)
> d$bw
[1] 1.159702
```

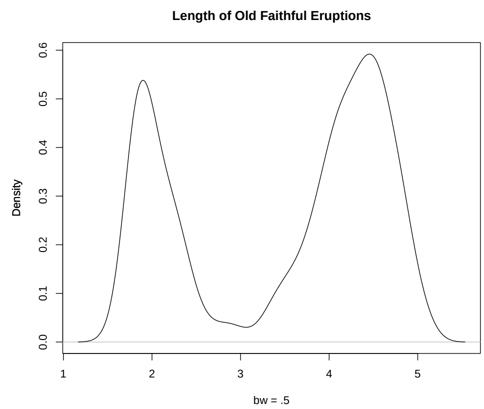
Plots for this bandwidth can be produced as follows.

```
> plot(d, xlab = paste("bw =", d$bw))
```

We can also produce plots for other bandwidths. E.g.

```
> plot(dtrace(eruptions, bw = 0.5))
> title(xlab = "bw = .5")
```





Comparing Distributions

- Density traces provide a good way of comparing the distribution of two batches of values.
- All that is necessary is to superimpose the two (or more) density traces on the same graph.
- This example is about comparing the levels of ozone from two areas in metropolitan New York (Yonkers and Stamford).
- Ozone is a pollutant which is formed when sunlight shines on to car exhaust emissions. It is implicated in respiratory and cardiac health problems (particularly asthma).

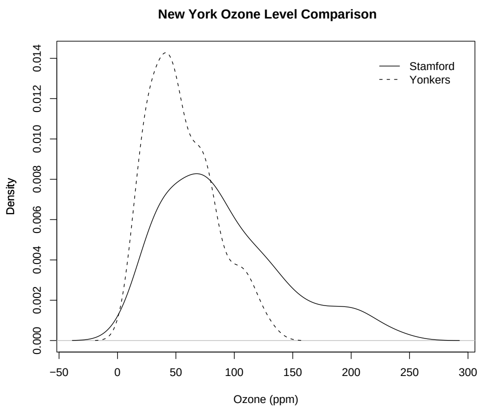
Graphical Comparison Using Density Traces

Read in and clean the data. The `na.omit` statements omt any missing values.

```
> ozone = read.table("ozone.dat", header = TRUE)
> stamford = na.omit(ozone$stamford)
> yonkers = na.omit(ozone$yonkers)
```

Compute the density estimates for the Stamford and Yonkers values. We will need to compute the ranges for the plot.

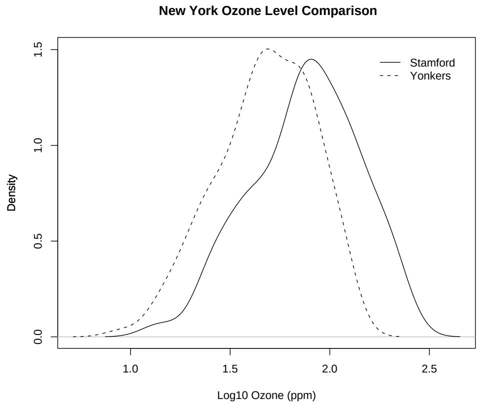
```
> d = dtrace(list(Stamford = stamford,
                  Yonkers = yonkers))
> plot(d, lty = c("solid", "dashed"),
       main = "New York Ozone",
       xlab = "Ozone (ppm)")
```



Data Transformation

- The previous plot indicates that the ozone concentrations in Stamford are a multiple of those in Yonkers (about 1.25 times).
- We can check this by transforming to a logarithmic scale – a multiplicative effect will be transformed to a shift.
- We can do this as follows:

```
> d = dtrace(list(Stamford = log10(stamford),
                  Yonkers = log10(yonkers)))
> plot(d, lty = c("solid", "dashed"),
       main = "New York Ozone",
       xlab = "Log10 Ozone (ppm)")
```



Relative Ozone Patterns

The graphs show that the distributions of ozone levels are related by

$$\log_{10}\text{Stamford} = \log_{10}\text{Yonkers} + 0.25.$$

In raw terms this means

$$\text{Stamford} = 1.78 \times \text{Yonkers}.$$

In in other words, ozone levels in Stamford are close to double those of Yonkers.