Examining and Comparing Distributions

Example: Yearly Precipitation in New York City

The following table shows the number of inches of (melted) precipitation, yearly, in New York City, (1869-1957).

43.6	37.8	49.2	40.3	45.5	44.2	38.6	40.6	38.7	46.0
37.1	34.7	35.0	43.0	34.4	49.7	33.5	38.3	41.7	51.0
54.4	43.7	37.6	34.1	46.6	39.3	33.7	40.1	42.4	46.2
36.8	39.4	47.0	50.3	55.5	39.5	35.5	39.4	43.8	39.4
39.9	32.7	46.5	44.2	56.1	38.5	43.1	36.7	39.6	36.9
50.8	53.2	37.8	44.7	40.6	41.7	41.4	47.8	56.1	45.6
40.4	39.0	36.1	43.9	53.5	49.8	33.8	49.8	53.0	48.5
38.6	45.1	39.0	48.5	36.7	45.0	45.0	38.4	40.8	46.9
36.2	36.9	44.4	41.5	45.2	35.6	39.9	36.2	36.5	

The annual rainfall in Auckland is 47.17 inches, so this is quite comparable.

Plots for a Collection of Numbers

- Often we have no idea what features a set of numbers may exhibit.
- Because of this it is useful to begin examining the values with general purpose tools.
- In this lecture we'll examine a class of tools which give information about the distribution of a set of values.

Stem-and-Leaf Plots

```
> stem(rain.nyc, scale = .5)

  The decimal point is 1 digit(s) to the right of the |

  3 | 344444
  3 | 55666677777778888889999999999
  4 | 00000001111222233444444
  4 | 55555666677778999
  5 | 0000113344
  5 | 666
```

The argument `scale=.5` is use above above to compress the scale of the plot. Values of `scale` greater than 1 can be used to stretch the scale.

(It only makes sense to use values of `scale` which are 1, 2 or 5 times a power of 10.

Stem-and-Leaf Plots

- Stem and leaf plots are very “busy” plots, but they show a number of data features.
  - The location of the bulk of the data values.
  - Whether there are outliers present.
  - The presence of clusters in the data.
  - Skewness of the distribution of the data .
- It is possible to retain many of these good features in a less “busy” kind of plot.

Histograms

- Histograms provide a way of viewing the general distribution of a set of values.
- A histogram is constructed as follows:
  - The range of the data is partitioned into a number of non-overlapping “cells”.
  - The number of data values falling into each cell is counted.
  - The observations falling into a cell are represented as a “bar” drawn over the cell.

Types of Histogram

Frequency Histograms

The height of the bars in the histogram gives the number of observations which fall in the cell.

Relative Frequency Histograms

The area of the bars gives the proportion of observations which fall in the cell.

Warning: *Drawing frequency histograms when the cells have different widths misrepresents the data*.

Histograms in R

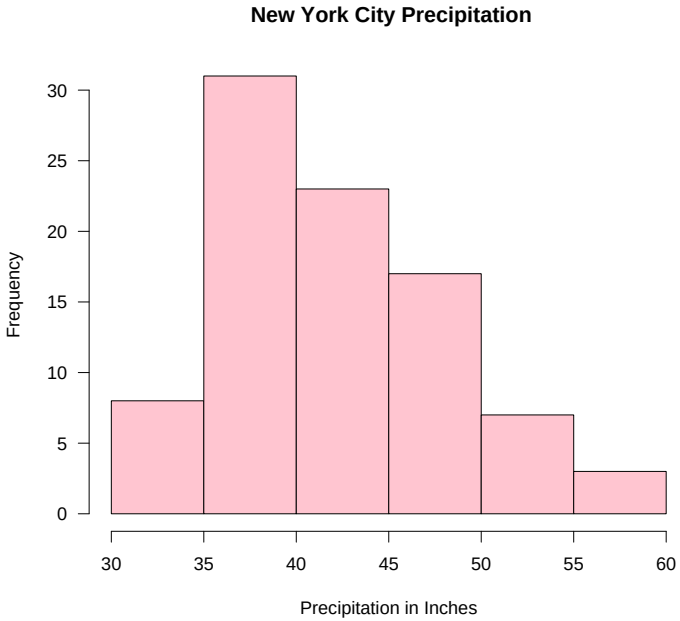
- The R function which draws histograms is called `hist`.
- The `hist` function can draw either frequency or relative frequency histograms and gives full control over cell choice.
- The simplest use of `hist` produces a frequency histogram with a default choice of cells.
- The function chooses approximately $\log_2 n$ cells which cover the range of the data and whose end-points fall at "nice" values.

Example: Simple Histograms

Here is the simplest possible example of drawing a histogram with R.

```
> hist(rain.nyc, col = hcl(0),
       main = "New York City Precipitation",
       xlab = "Precipitation in Inches" )
```

This draws a histogram with the default cell choice and with the bars coloured pink.


New York City Precipitation
Frequency
0
5
10
15
20
25
30
30
35
40
45
50
55
60
Precipitation in Inches

Example: Simple Histograms

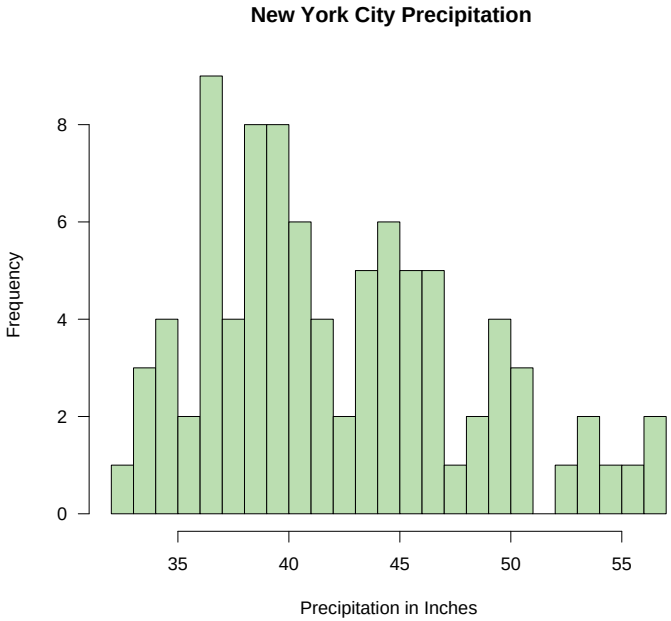
Here are two examples of drawing histograms with R.

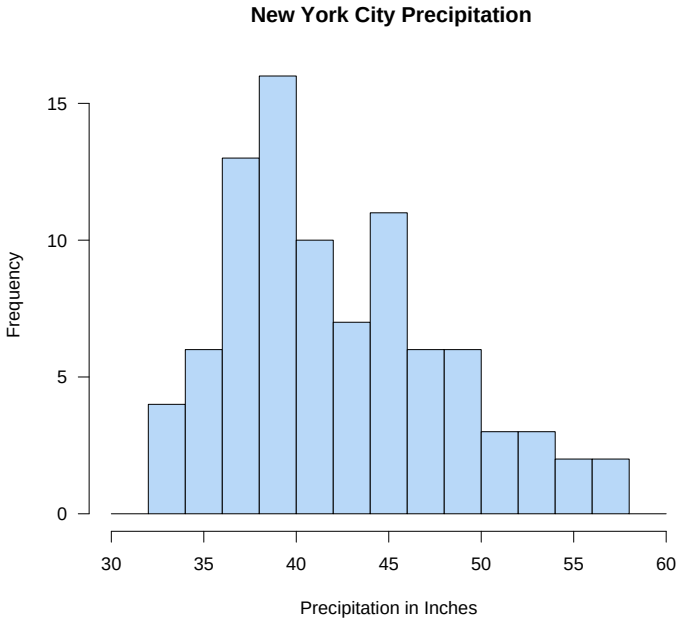
1. A request for approximately 20 bars.

```
> hist(rain.nyc, breaks = 20,
       col = hcl(120),
       main = "New York City Precipitation",
       xlab = "Precipitation in Inches" )
```

2. Explicit setting of the cell breakpoints.

```
> hist(rain.nyc, breaks = seq(30, 60, by = 2),
       col = hcl(240),
       main = "New York City Precipitation",
       xlab = "Precipitation in Inches")
```


New York City Precipitation
Frequency
0
2
4
6
8
35
40
45
50
55
Precipitation in Inches


New York City Precipitation
Frequency
0
5
10
15
30
35
40
45
50
55
60
Precipitation in Inches

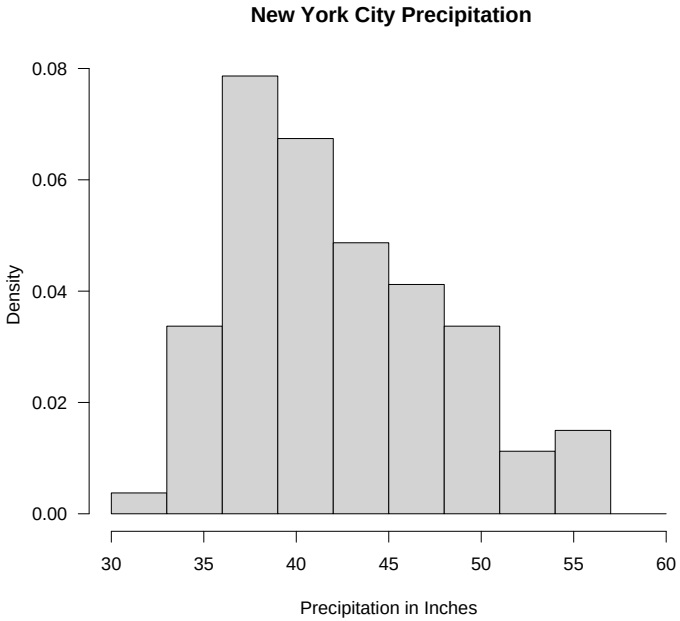
Example: Histogram Options

Optional arguments can be used to customise histograms.

```
> hist(rain.nyc, breaks = seq(30, 60, by=3),
       prob = TRUE, las = 1, col = "lightgray",
       main = "New York City Precipitation",
       xlab = "Precipitation in Inches")
```

The following options are used here.

1. `prob = TRUE` makes this a *relative frequency* histogram.
2. `col = "gray"` colours the bars gray.
3. `las = 1` rotates the y axis tick labels.

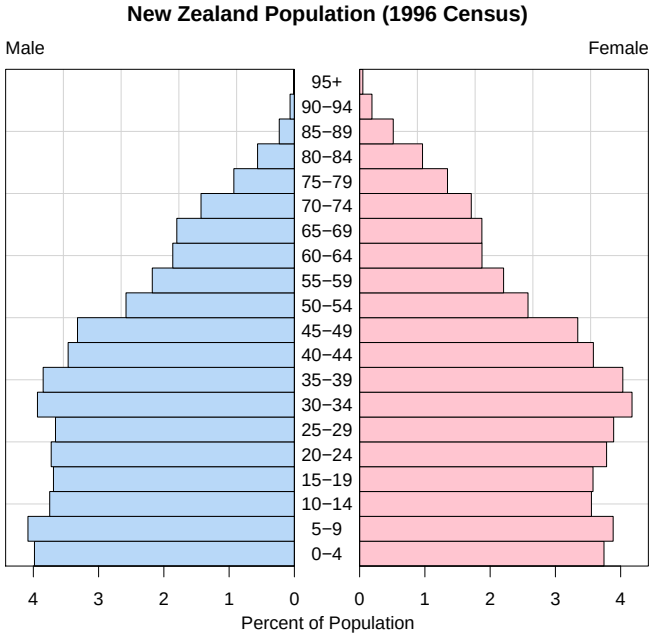

New York City Precipitation
0.08
0.06
0.04
0.02
0.00
Density
30
35
40
45
50
55
60
Precipitation in Inches

Histograms and Perception

- Information in histograms is conveyed by the heights of the bar tops.
- Because the bars all have a common baseline, the encoding is based on “position on a common scale.”
- Histograms convery their message using the best possible encoding method.

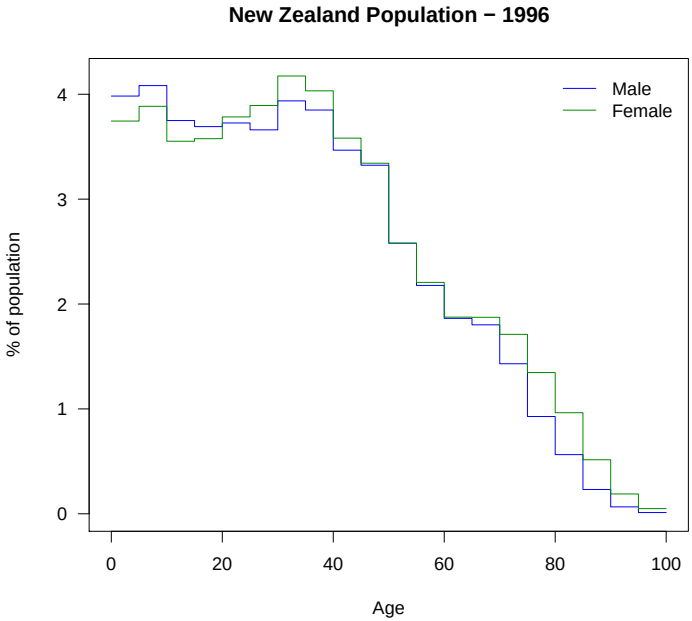
Comparison Using Histograms

- Sometimes it is useful to compare the distribution of the values in two or more sets of observations.
- There are a number of ways in which it is possible to make such a comparison.
- One common method is to use “back to back” histograms.
- This is often used to examine the structure of populations broken down by age and gender.
- These are referred to as “population pyramids.”


New Zealand Population (1996 Census)
Male
Female
95+
90−94
85−89
80−84
75−79
70−74
65−69
60−64
55−59
50−54
45−49
40−44
35−39
30−34
25−29
20−24
15−19
10−14
5−9
0−4
4
3
2
1
0
0
1
2
3
4
Percent of Population

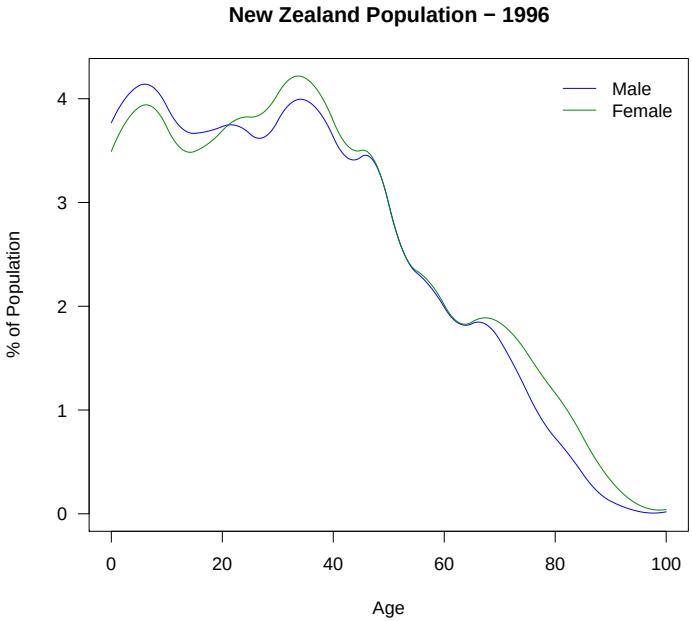
Back to Back Histograms and Perception

- Comparisons within either the “male” or “female” sides of this graph are made on a “common scale.”
- Comparisons between the male and female sides of the graph must be made using length, which does not work as well as position on a common scale.
- A better way of making this comparison is to superimpose the two histograms.
- Since it is only the bar tops which are important, they are the only thing which needs to be drawn.


New Zealand Population − 1996
Male
Female
% of population
0
1
2
3
4
0
20
40
60
80
100
Age

Smoothed Histograms

- The discontinuous nature of histograms creates visual clutter in the previous plot.
- It can be useful to produce a smoothed version of the plot.
- This can be done as follows:
  - Integrate the histogram to obtain a distribution function (this is just a cumulative sum).
  - Fit a spline curve through the points of the distribution function.
  - Differentiate the distribution function to obtain a density.

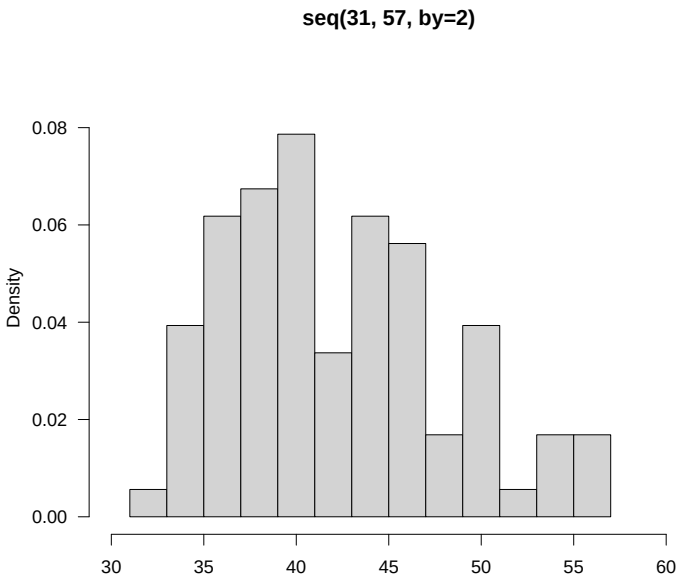

New Zealand Population − 1996
Male
Female
% of Population
0
1
2
3
4
0
20
40
60
80
100
Age

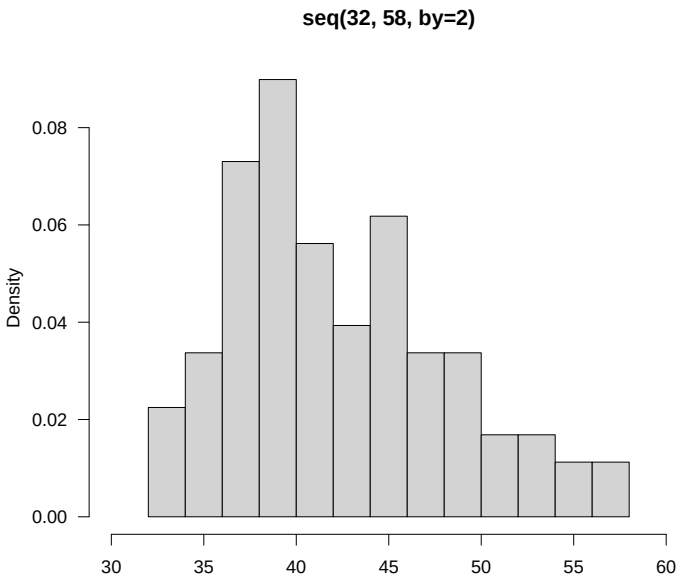
Superposition and Perception

- Superimposing one histogram on another works well because comparisons both within and between distributions are made on a common scale.
- The separate histograms provide a good way of examining the distribution of values in each sample.
- Comparison of two (or more) distributions is easy.

The Effect of Cell Choice

- Histograms are very sensitive to the choice of cell boundaries.
- We can illustrate this by drawing a histogram for the NYC precipitation with two different choices of cells.
  - `seq(31, 57, by = 2)`
  - `seq(32, 58, by = 2)`
- These different choices of cell boundaries produce quite different looking histograms.


seq(31, 57, by=2)
Density
0.08
0.06
0.04
0.02
0.00
30
35
40
45
50
55
60


seq(32, 58, by=2)
Density
0.00
0.02
0.04
0.06
0.08
30
35
40
45
50
55
60

The Inherent Instability of Histograms

- The shape of a histogram depends on the particular set of histogram cells chosen to draw it.
- This suggests that there is a fundamental instability at the heart of its construction.
- To illustrate this we'll look at a slightly different way of drawing histograms.
- For an ordinary histogram, the height of each histogram bar provides a measure of the density of data values within the bar.
- This notion of data density is very useful and worth generalising.

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

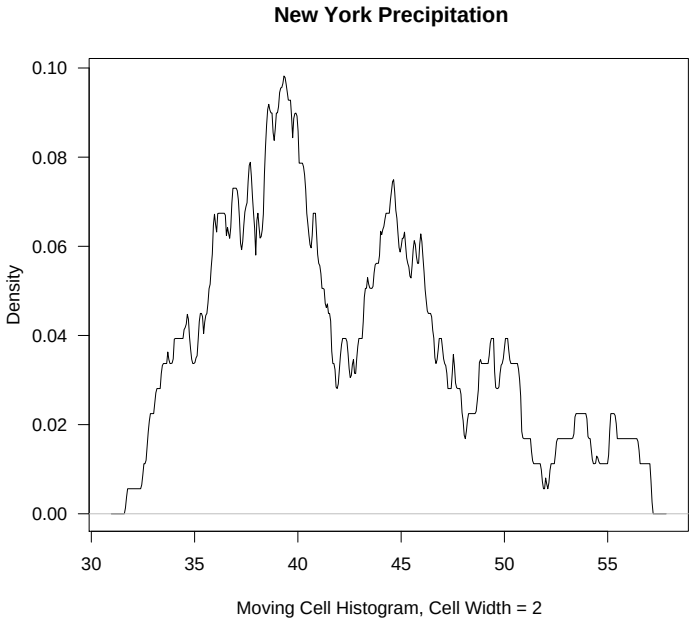
$$h(x) = \frac{k}{n} \times \frac{1}{w}$$

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

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

Moving Cell Histograms

- We can use a single histogram cell, centred at a point x and having width w to estimate the density of data values near x.
- By moving the cell across the range of the data values we will get an estimate of the density of the data points throughout the range of the data.


New York Precipitation
0.10
0.08
0.06
0.04
0.02
0.00
Density
30
35
40
45
50
55
Moving Cell Histogram, Cell Width = 2

Stability

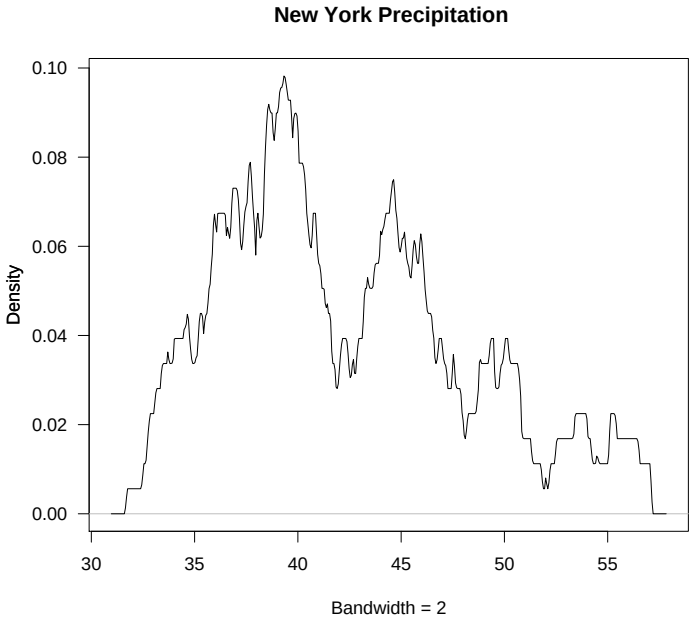
- The basic idea of computing and drawing the density of the data points is a good one.
- It seems, however, that using a sliding histogram cell is not a good way of producing a density estimate.
- This is because there seems to be a good deal of instability in the estimate.
- We will now look at more stable estimates of data density.

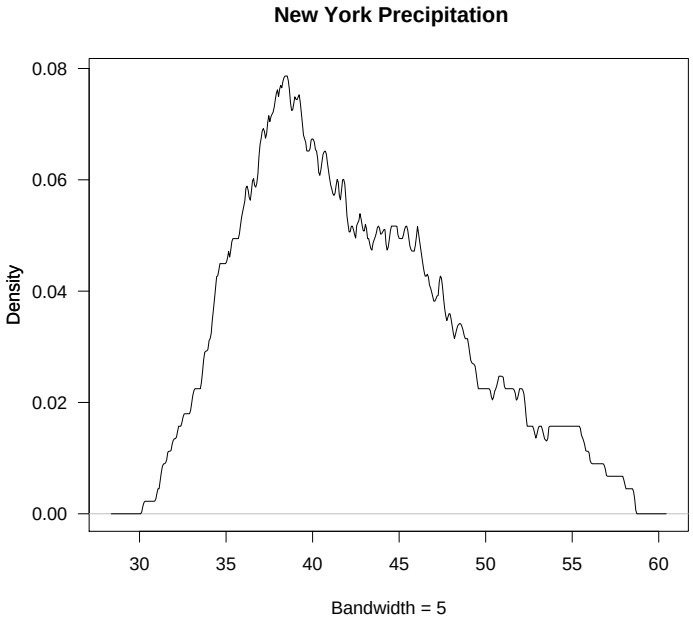
Terminology

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

Notes

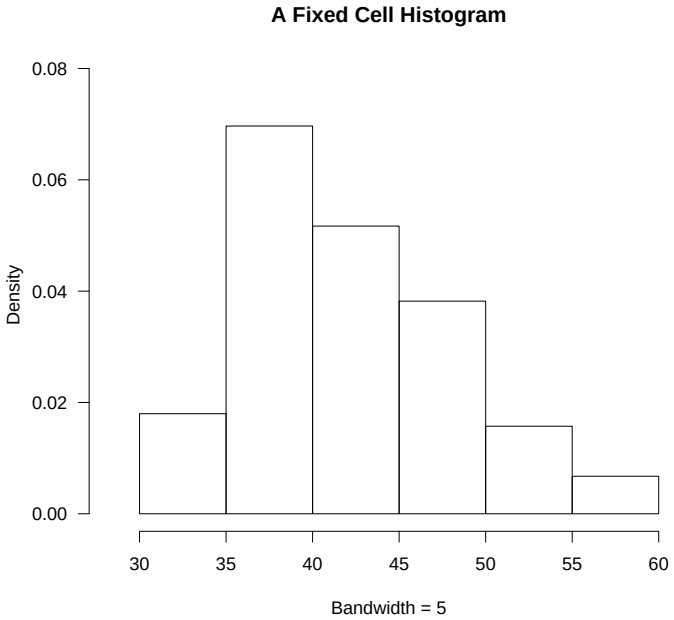
- $h(x)$ is defined for every x value.
- The area under $h(x)$ is 1.

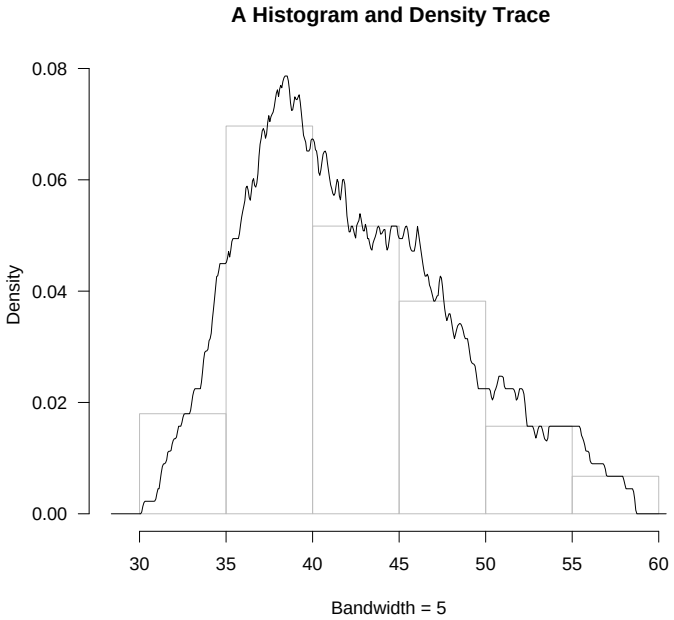

New York Precipitation
Density
0.10
0.08
0.06
0.04
0.02
0.00
30
35
40
45
50
55
Bandwidth = 2

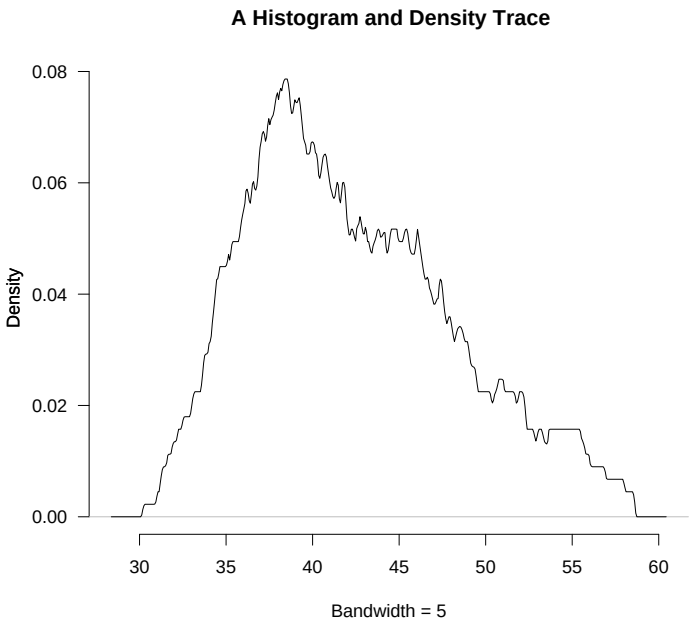

New York Precipitation
Density
0.08
0.06
0.04
0.02
0.00
30
35
40
45
50
55
60
Bandwidth = 5

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.


A Fixed Cell Histogram
0.08
0.06
0.04
0.02
0.00
Density
30
35
40
45
50
55
60
Bandwidth = 5


A Histogram and Density Trace
Density
0.08
0.06
0.04
0.02
0.00
30
35
40
45
50
55
60
Bandwidth = 5


A Histogram and Density Trace
Density
0.08
0.06
0.04
0.02
0.00
30
35
40
45
50
55
60
Bandwidth = 5

Lack of Smoothness

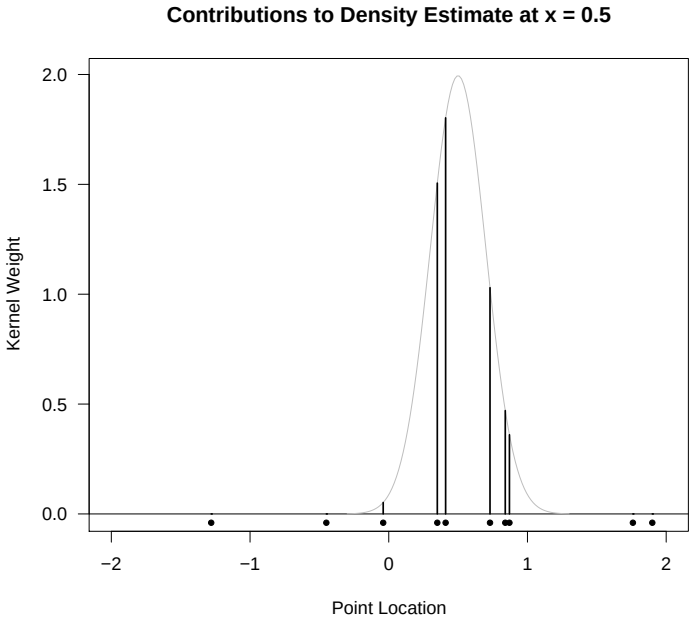
- Histogram density estimates have a very rough appearance.
- This is because points enter and leave the window (histogram cell) suddenly and this causes jumps in $h(x)$.
- When a point is within a distance $w/2$ of x, it contributes an amount $1/nw$ to the value of $h(x)$.
- When it is a greater distance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

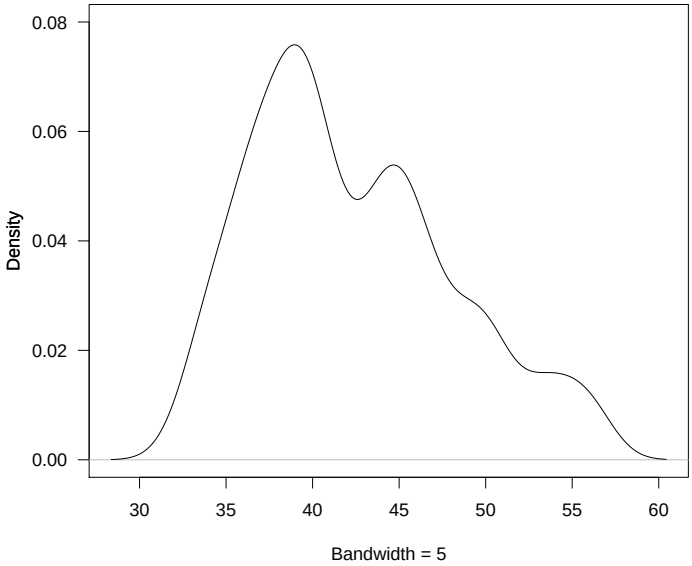
Kernel Density Estimates I

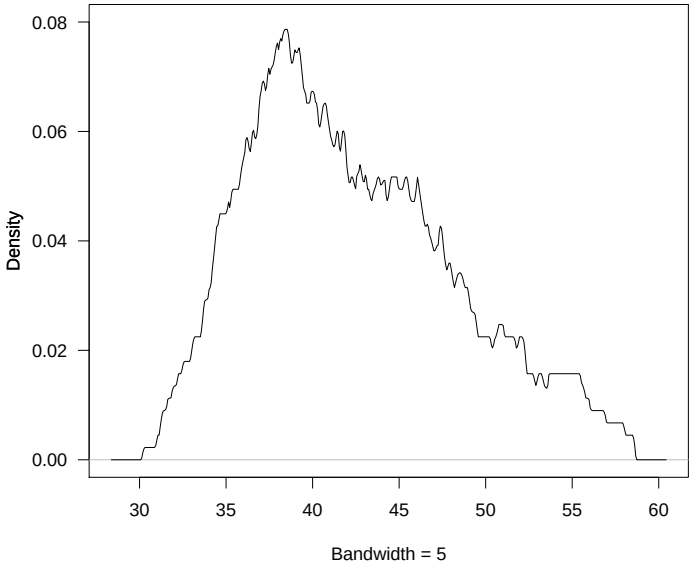
- It is possible to make density traces smoother by changing the way points make a contribution to $h(x)$.
- Smooth density estimates work by making the contribution a point makes to $h(x)$ depend on its distance to x. A small distance means a large contribution and vice versa.

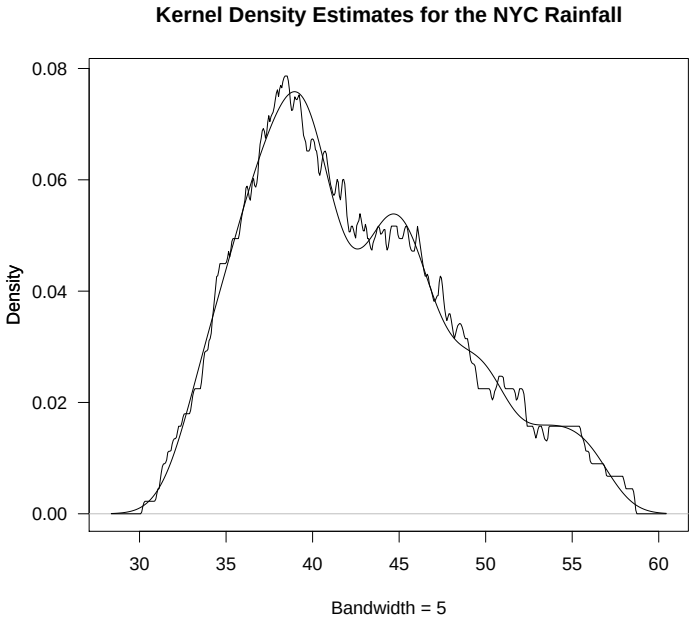
Kernel Density Estimates II

- 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.


Contributions to Density Estimate at x = 0.5
Kernel Weight
2.0
1.5
1.0
0.5
0.0
−2
−1
0
1
2
Point Location


A Gaussian Kernel Density Estimate for the NYC Rainfall
Density
0.08
0.06
0.04
0.02
0.00
30
35
40
45
50
55
60
Bandwidth = 5


A Rectangular Kernel Density Estimate for the NYC Rainfall
Density
0.08
0.06
0.04
0.02
0.00
30
35
40
45
50
55
60
Bandwidth = 5


Kernel Density Estimates for the NYC Rainfall
Density
0.08
0.06
0.04
0.02
0.00
30
35
40
45
50
55
60
Bandwidth = 5

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.
- The bandwith is the width of the support of a rectangular kernel with the same standard deviation as the given kernel.
- Estimates with the same bandwith perform roughly the same amount of smoothing, even if they have different kernels.

R Functions

- The R function `density` computes density estimates.
- A better option is to use the R "dtrace" library which is 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, or added to an existing plot with the `lines` function.

R Examples

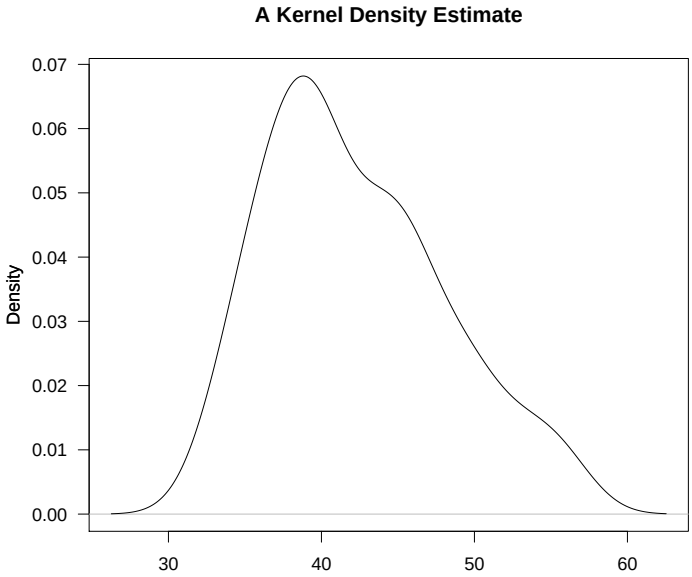
It is simple to construct density plots using R.

Long hand . . .

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

Or equivalently . . .

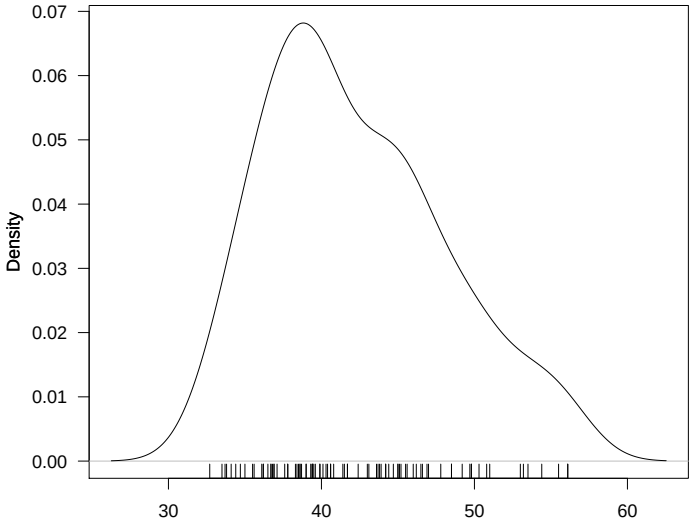
```
> plot(dtrace(rain.nyc))
> title(main = "A Kernel Density Estimate")
```


A Kernel Density Estimate
Density
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
30
40
50
60

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)
```


0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
Density
30
40
50
60

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.

We'll illustrate this with data on the time between erruptions for the old-faithful geyser in Yellowstone National Park, Wyoming, USA.

The variables in the data set can be accessed as follows:

```
> attach(faithful)
```

Bandwidth for the Geyser Eruptions

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

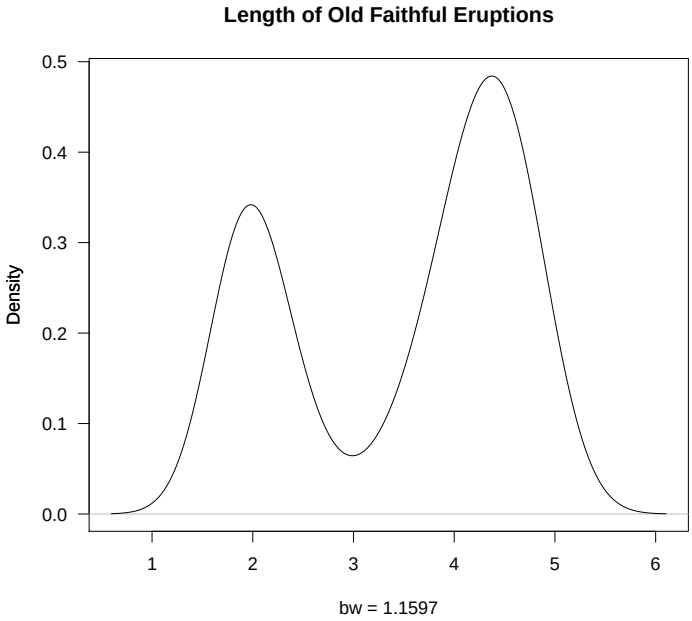
```
> d = dtrace(eruptions)
> d$bw
[1] 1.159702
```

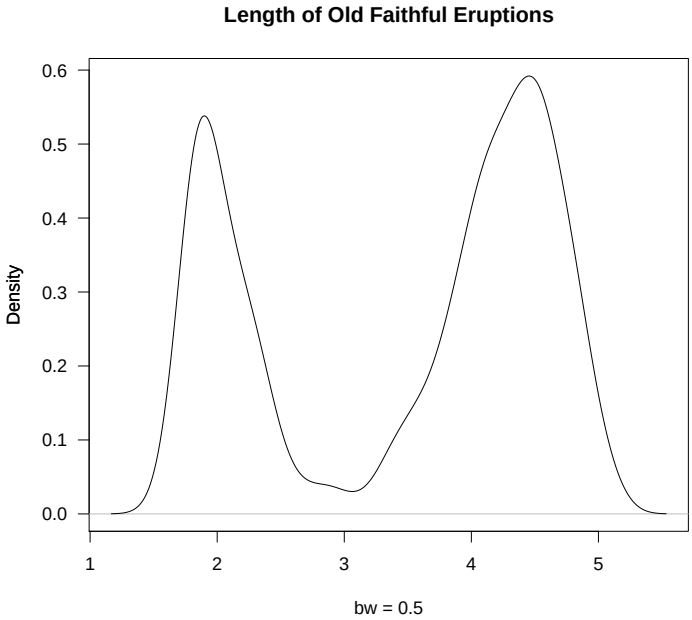
Plots for this bandwidth can be produced as follows.

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

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

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

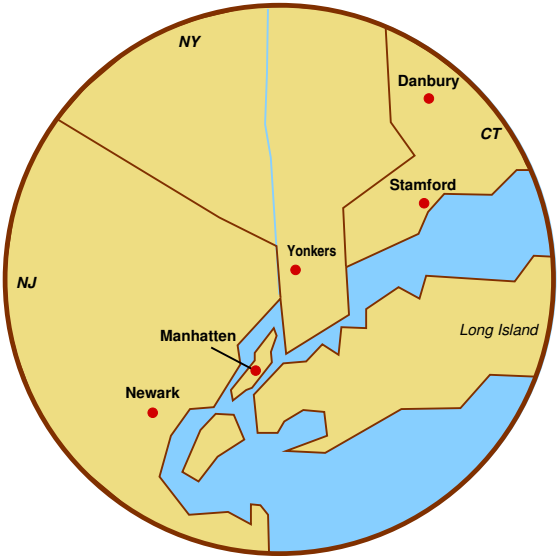

Length of Old Faithful Eruptions
Density
0.5
0.4
0.3
0.2
0.1
0.0
1
2
3
4
5
6
bw = 1.1597


Length of Old Faithful Eruptions
Density
0.6
0.5
0.4
0.3
0.2
0.1
0.0
1
2
3
4
5
bw = 0.5

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).

The New York Metropolitan Area



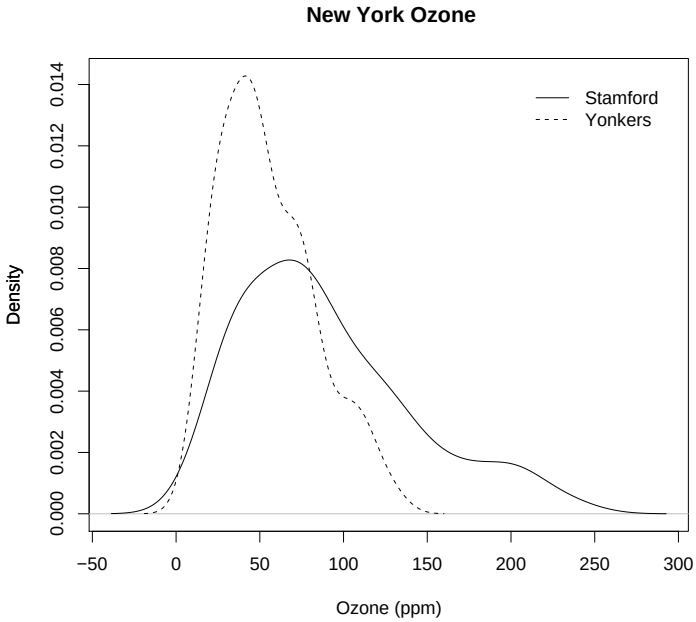
Graphical Comparison Using Density Traces

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.

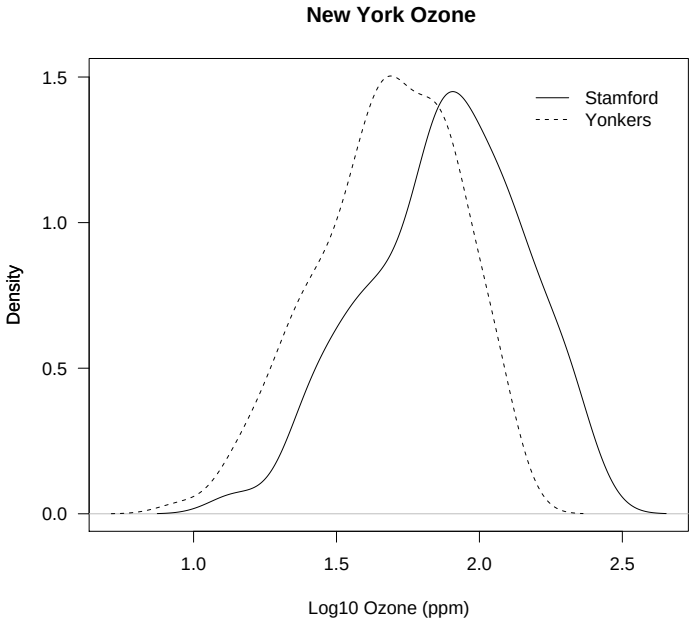
```
> d = dtrace(list(Stamford = stamford,
                  Yonkers = yonkers))
> plot(d, lty = c("solid", "dashed"),
        main = "New York Ozone",
        xlab = "Ozone (ppm)", las = 0)
```


New York Ozone
Stamford
Yonkers
Density
0.000
0.002
0.004
0.006
0.008
0.010
0.012
0.014
−50
0
50
100
150
200
250
300
Ozone (ppm)

Data Transformation

- The previous plot indicates that the ozone concentrations in Stamford are a multiple of those in Yonkers (about 1.5 to 2 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)")
```


New York Ozone
Stamford
Yonkers
Density
1.5
1.0
0.5
0.0
1.0
1.5
2.0
2.5
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 approaching double those of Yonkers.