2. Exploratory Data Analysis

2.1 Data Summaries

2.1.1 Introduction

Standard summaries `mean()`, `median()` and `var()` are available for summarizing data. The first two take individual variables as arguments, and the argument for `var()` can be either a single variable or a data matrix. If the latter then a complete variance-covariance matrix is returned. `summary()` will return the minimum, 1st quartile, median, 3rd quartile and maximum, together with the mean. The first five of these are the (0,0.25,0.5,0.75,1) quantiles and can be produced by `quantile()`. This can also be used to produce any arbitrary quantiles by including a vector of the required probabilities:

```
> attach(hills)
> summary(dist)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   4.500   6.000   7.529   8.000  28.000
> quantile(dist)
  0%  25%  50%  75% 100%
 2.0  4.5  6.0  8.0 28.0
> quantile(dist,c(0.25,0.33,0.4,0.8))
25% 33% 40% 80%
4.5 5.0 5.3 9.6
```

Note the use of `c(0.25,0.33,0.4,0.8)` to ***concatenate (=join together)*** the numbers into a vector.

[The quantiles are obtained by linear interpolation in the ordered sample]

However, these summaries (especially `mean()` and `var()`)are ***sensitive to outliers***, i.e. they are not ***robust***.

2.1.2 Robust Summaries

```
> data(chem)

> chem
 [1] 2.90 3.10 3.40 3.40 3.70 3.70 2.80 2.50 2.40 2.40
                                           2.70 2.20
[13] 5.28 3.37 3.03 3.03 28.95 3.77 3.40 2.20 3.50
                                     3.60 3.70 3.70
```

The data above are values of 24 determinations of copper in ppm in wholemeal flour. *The* `[1]` *and* `[13]` *indicate that these lines begin with the* 1^{st} *and* 13^{th} *element of the object* `chem`.

Note the very large value `28.95`. It is an ***outlier***.

```
> mean(chem)
[1] 4.280
```

The value of the mean is highly influenced by this outlier (it is larger than all but two of the observations).

The sample mean $\bar{x} \to \pm\infty$ if any data value $x_i \to \pm\infty$, whereas the median is hardly affected if any single value of tends to $\pm\infty$.

In fact, the median will not be affected until **50%** of the data are grossly contaminated.

The median is ***resistant to gross errors***, but the mean is not.

The median has a ***breakdown point of 50%***, the mean has a breakdown point of 0%.

A more robust estimate of location is a trimmed mean, i.e. the mean when a percentage of the largest and smallest observations are *trimmed* away from the sample. Specifically, an α-trimmed mean is the mean of the sample after removal of the upper and lower $100\times\alpha$% portions of the sample, i.e. of the middle 1-2α part of the distribution.

Example (chemical data above):

```
> mean(chem, trim=0.01)
[1] 4.280417
> mean(chem, trim=0.04)
[1] 4.280417
> mean(chem, trim=0.05)
[1] 3.253636
> mean(chem, trim=0.1)
[1] 3.205
```

Questions:

1. What breakdown point does an α-trimmed mean have?
2. Which observations have been trimmed and why in the four calculations above?
3. What will an 0.5-trimmed mean give?

Other robust estimators of location are ***M-estimators***, see e.g. Venables & Ripley, (1999), but these are not implemented as standard in **R** [yet] but are available in S-plus.

Robust estimators of scale:

Consider the following estimators of scale

1. s, where $s^2 = \frac{1}{n-1}\sum(x_i - \bar{x})^2$
2. $\tilde{\sigma} = \sqrt{\frac{\pi}{2}}\frac{1}{n}\sum|x_i - \bar{x}|$
3. IQR=$0.741\times(x_{[3n/4]} - x_{[n/4]})$

   (Inter-Quartile Range)
4. MAD=$1.4826\times$median$\{|x_i - \text{median}(x_j)|\}$

   (Median Absolute Deviation)

All of these are [approximately] unbiased estimators of σ (or their squares of σ^2) if the $x_i \sim N(\mu,\sigma^2)$.

Questions:

1. How resistant are these to outliers?, i.e. what are their breakdown points?
2. How can these be calculated in R using functions `mad()`, `sum()`, `mean()`, `median()`, `abs()`?
3. Do we need to use the function `mad()`?

Relative Efficiency: This measures what price is paid in using a robust estimator instead of an alternative one. The relative efficiency of two estimators $\tilde{\theta}$ and $\hat{\theta}$ is RE($\tilde{\theta}$;$\hat{\theta}$)=(variance of $\hat{\theta}$)/(variance of $\tilde{\theta}$) where the variances are calculated for the particular distribution that the sample comes from (***assuming we know what this is***). This will probably depend on the sample size n and we can consider the Asymptotic Relative efficiency as $n\to\infty$

e.g. for Normal data, (1) ARE($\tilde{\sigma}^2$;s^2)=88%, (2) ARE(MAD;s)=37% and ARE(median;mean)=64%.

We can interpret these as saying that for Normal data we need roughly only 37% of the sample size to estimate σ with s to achieve the same precision of estimation as we would have with MAD. This does not look attractive — it is a high price to pay for protection against outliers.

However, these calculations are based on the sample **really** coming from a Normal distribution.

If the data come from a student t-distribution on 5 d.f., t_5, the ARE(median; mean)=96% (not 64%)

If the data come from a Normal distribution with ε% contamination from a Normal with the same mean but 3 times the standard deviation, i.e. from $(1-\varepsilon)N(\mu,\sigma^2)+\varepsilon N(\mu,9\sigma^2)$ then the table of ARE($\tilde{\sigma}^2$;s^2) values is

ε(%)	**ARE($\tilde{\sigma}^2$;s^2)**
0	87.6%
0.1	94.8%
0.2	101.2%
1	144%
5	204%

Thus we can see that $\tilde{\sigma}^2$ is ***robust to model deviation***, i.e. if the data do not come from the Normal model that we have assumed but instead from a slightly different model then this estimator provides good protection.

As well as robust data summaries (and implicitly estimators) we can consider methods of more general statistical analysis that are ***robust*** or ***resistant*** to ***model deviation*** and ***data contamination***.

2.2 Graphical Summaries

2.2.1 Stem-and-leaf plots

Examples:

(1) Scottish hill race data

```
> data(hills)
> dist
 [1] 2.5 6.0 6.0 7.5 8.0 8.0 16.0 6.0 5.0 6.0 28.0
                                5.0 9.5 6.0 4.5
[16] 10.0 14.0 3.0 4.5 5.5 3.0 3.5 6.0 2.0 3.0 4.0
                                6.0 5.0 6.5 5.0
[31] 10.0 6.0 18.0 4.5 20.0
> stem(dist)
The decimal point is 1 digit(s) to the right of the |
  0 | 2333344
  0 | 55555556666666667888
  1 | 0004
  1 | 68
  2 | 0
  2 | 8
```

(2) Durations and intervals between eruptions of Old Faithful.

```
> data(geyser)
> summary(geyser)
    waiting          duration
 Min.   : 43.00   Min.   :0.8333
 1st Qu.: 59.00   1st Qu.:2.0000
 Median : 76.00   Median :4.0000
 Mean   : 72.31   Mean   :3.4608
 3rd Qu.: 83.00   3rd Qu.:4.3833
 Max.   :108.00   Max.   :5.4500
```


```
> stem(duration)
```

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

  8 | 3
 10 |
 12 |
 14 |
 16 | 223370023357778
 18 | 000222233333355577788800223333333555557778
 20 | 00000000000000000000000223578023578
 22 | 0278
 24 | 7807
 26 | 05
 28 | 373
 30 | 00
 32 | 583
 34 | 523
 36 | 00235
 38 | 0277802377
 40 | 00000000000000000000000000000000000000000000000000000023780233355777
 42 | 00222222355557778802333557788888
 44 | 00222255555557777800000233888
 46 | 00002225577778800033357778
 48 | 0033782277788
 50 | 30
 52 | 7
 54 | 5
```

```
> stem(waiting)
```

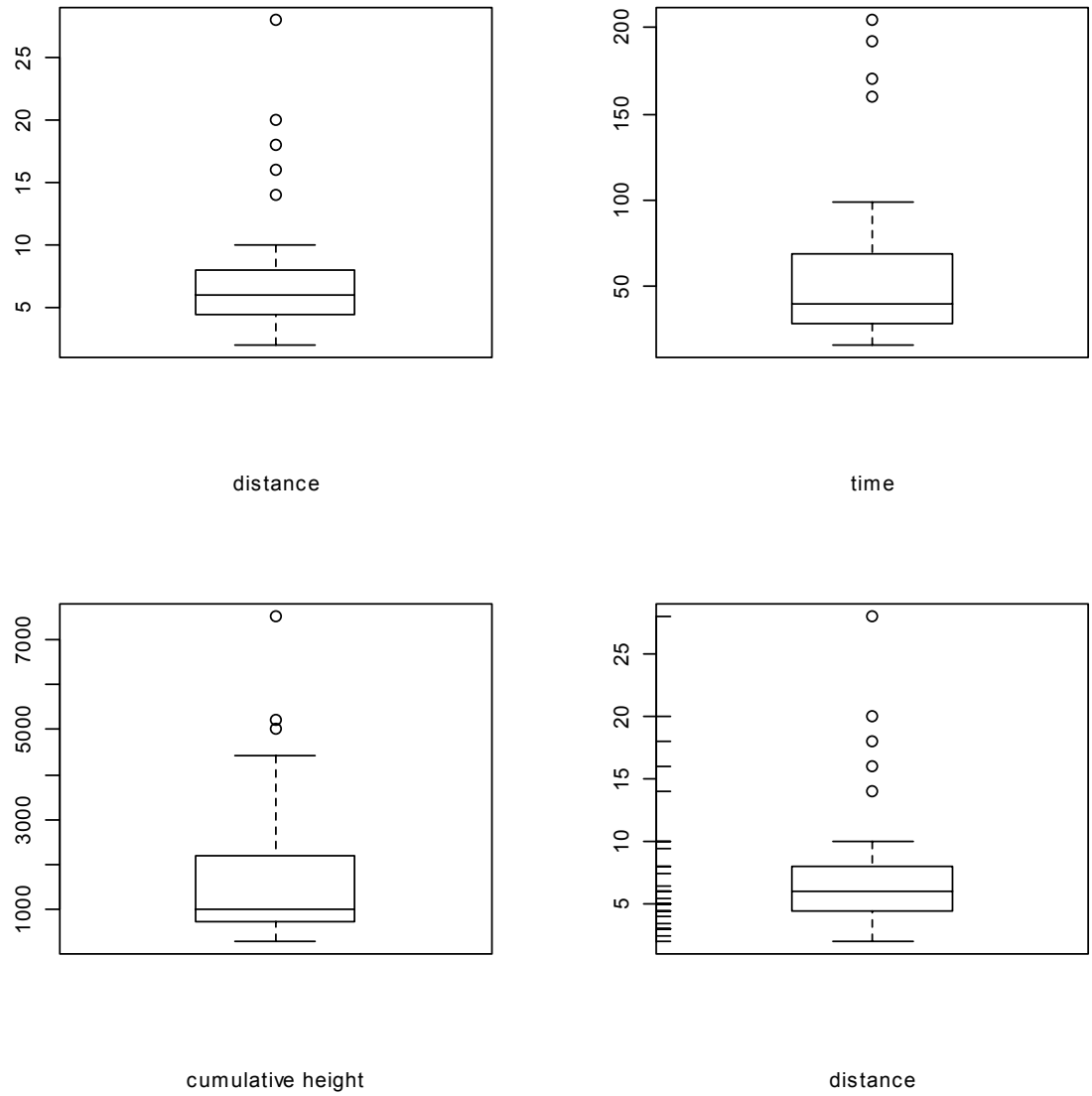
```
The decimal point is 1 digit(s) to the right of the |

  4 | 3
  4 | 577888889999999
  5 | 00000000000011111222223333333444444444
  5 | 5556677777777788888999
  6 | 0000001112222234
  6 | 5555555668889999
  7 | 01111122222233333344444444
  7 | 55555555566666666777777777788888888888888888899999999999
  8 | 00000000000001111111111122222222333333444444444444
  8 | 555555566666777777777777788888889999999
  9 | 0011222333333334
  9 | 668
 10 |
 10 | 8
```

Comments: Quick, easy, no data are lost — actual values are retained

2.2.2 Boxplots

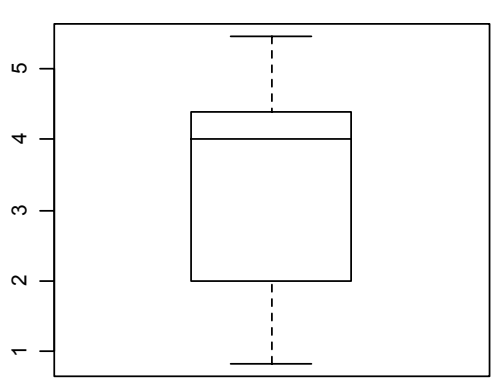
Examples:

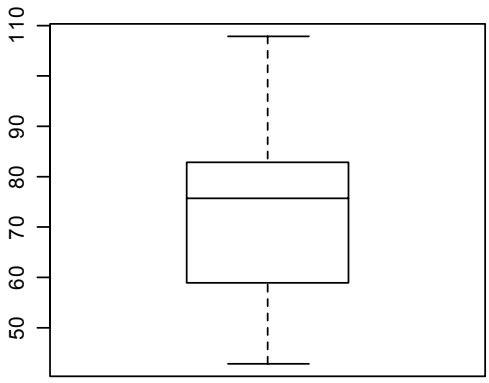
```
> data(hills)
> par(mfrow=c(2,2))
> boxplot(dist,sub="distance")
> boxplot(time,sub="time")
> boxplot(climb,sub="cumulative height")
> boxplot(dist,sub="distance")
> rug(dist,side=2)
>
```

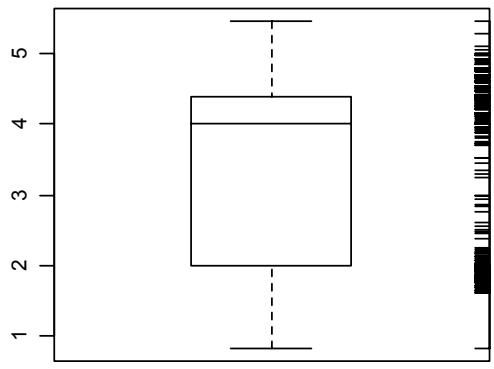


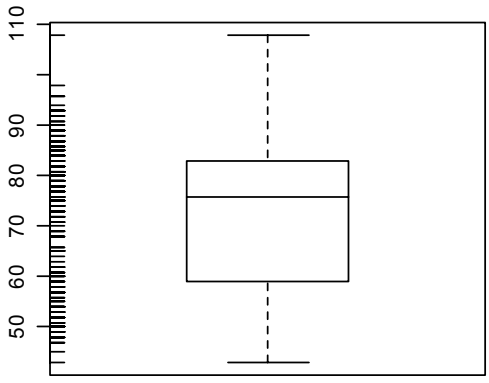
Note use of **`par(mfrow=c(2,2))`** and **`rug()`**

```
> data(geyser)
> boxplot(duration,sub="duration")
> boxplot(waiting,sub="waiting time")
> boxplot(duration,sub="duration")
> rug(duration,side=4)
> boxplot(waiting,sub="waiting time")
> rug(waiting, side=2)
```





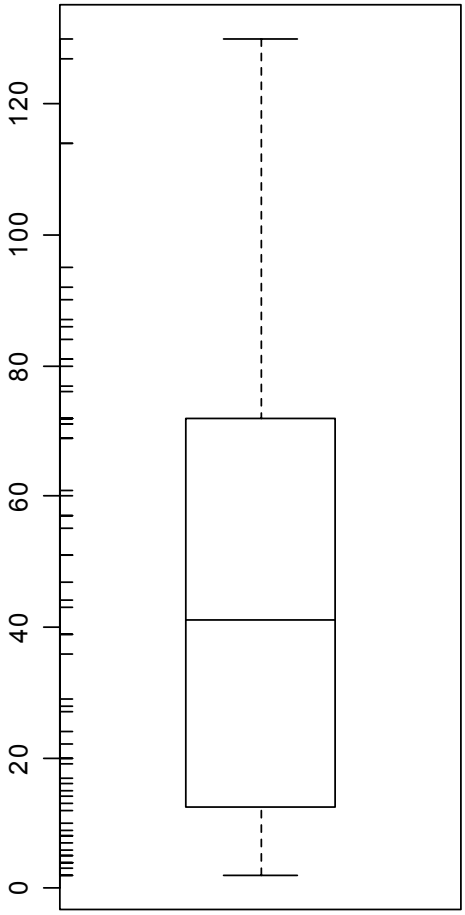


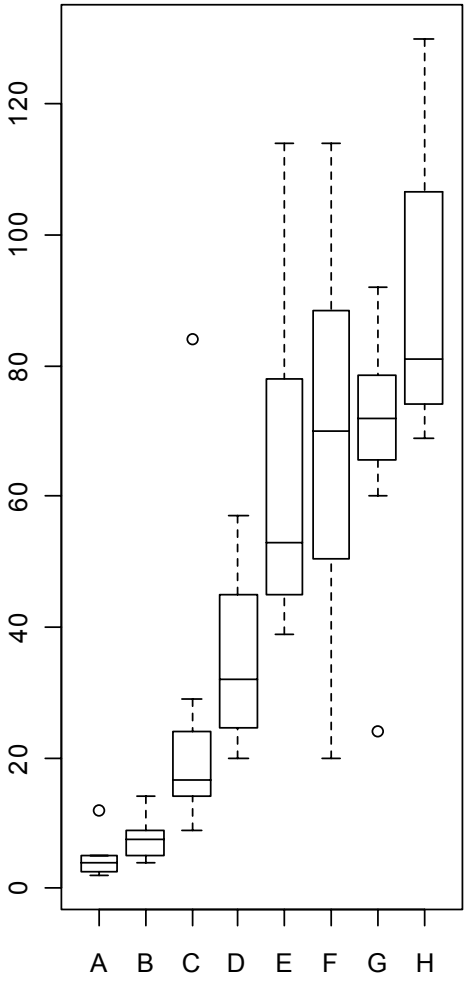


Comments: quick summaries for data but may miss gross features, e.g. bimodality, though addition of a *rug-plot* can help. However, most useful for plotting several related data sets for comparison, see example below.

Example: OrchardSpray data give decrease in counts on bees in response to 8 levels of sulphur treatment. The experiment was performed as an 8×8 Latin Square with row and column positions. Here we ignore the Latin Square structure and treat the data as one-way classification example.

```
> data(OrchardSprays)

> attach(OrchardSprays)

> summary(OrchardSprays)
    decrease          rowpos          colpos       treatment
 Min.   :  2.00   Min.   :1.00   Min.   :1.00   H       : 8
 1st Qu.: 12.75   1st Qu.:2.75   1st Qu.:2.75   G       : 8
 Median : 41.00   Median :4.50   Median :4.50   F       : 8
 Mean   : 45.42   Mean   :4.50   Mean   :4.50   E       : 8
 3rd Qu.: 72.00   3rd Qu.:6.25   3rd Qu.:6.25   D       : 8
 Max.   :130.00   Max.   :8.00   Max.   :8.00   C       : 8

                                                (Other):16

> par(mfrow=c(1,2))

> boxplot(decrease, sub="decrease in counts")

> rug(decrease,side=2)

> boxplot(decrease~treatment)

>
```

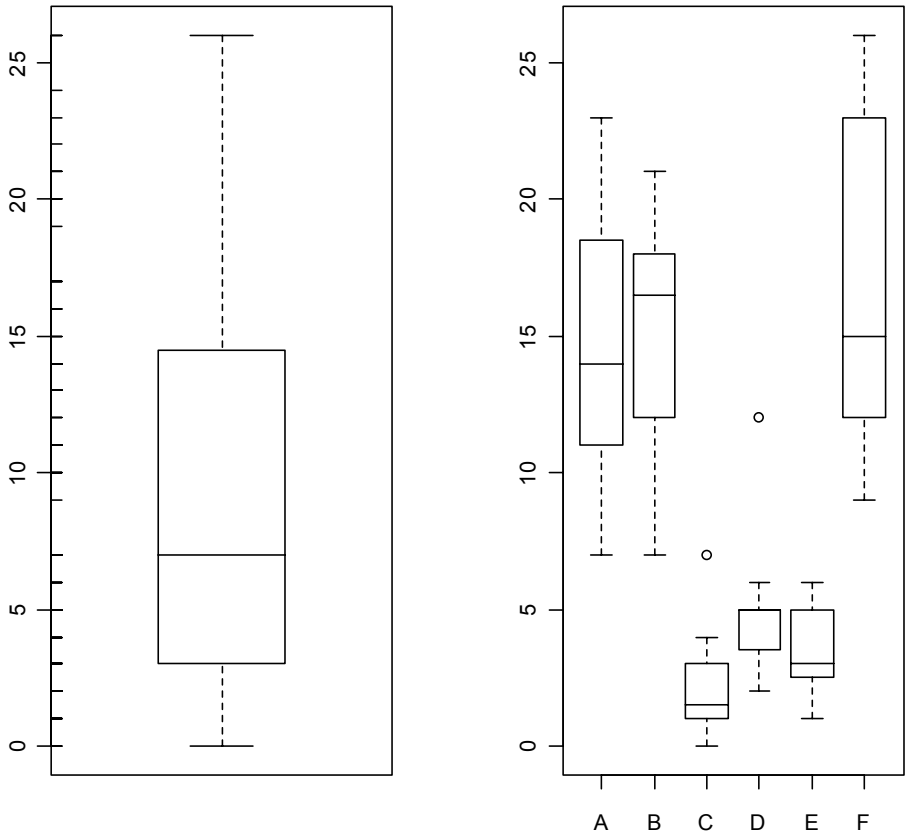



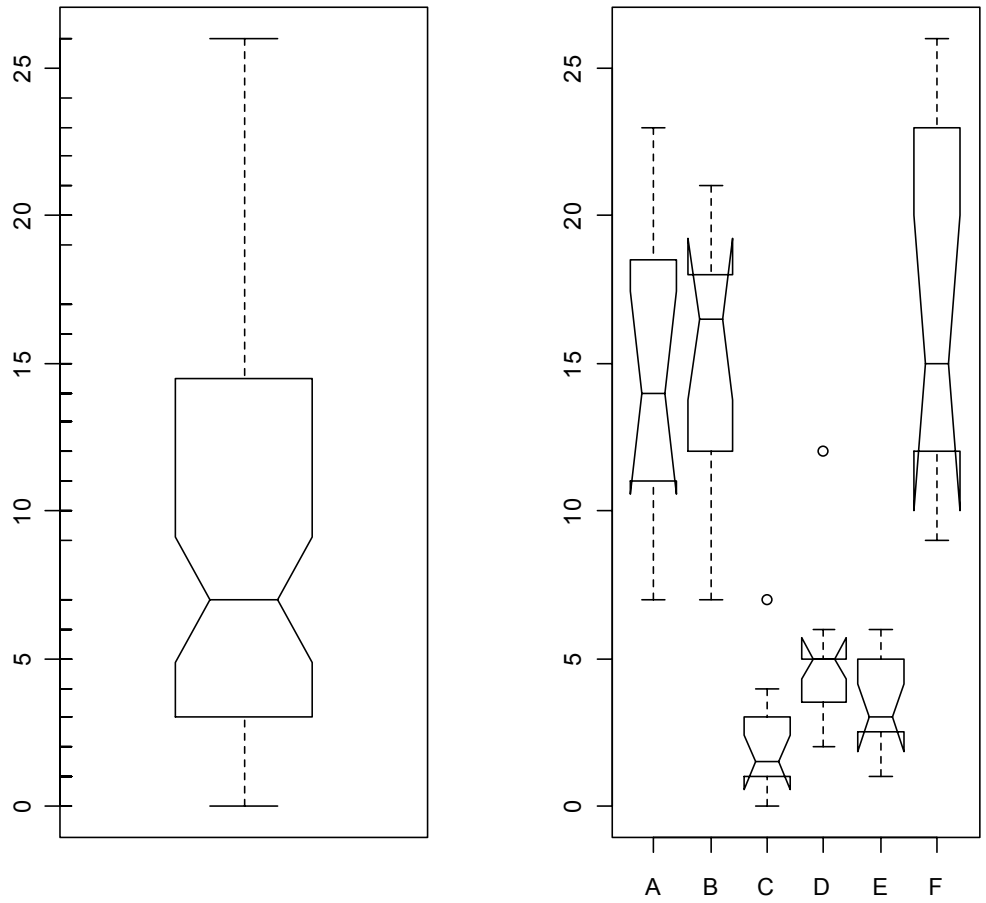


decrease in counts

Example: InsectSprays, similar data to above with six treatments. Note use of the ***logical parameter*** `notch` in the second two plots. This indicates a 'sort of confidence interval' for the median, in the sense that if two notches do not overlap then the medians of those samples are 'significantly different' at the 5% level.

```
> data(InsectSprays)
> attach(InsectSprays)
> boxplot(count)
> rug(count, side=2)
> boxplot(count~spray)
> boxplot(count,notch=TRUE)
> rug(count,side=2)
> boxplot(count~spray,notch=TRUE)
```





Note that the notches may be bigger than the boxes e.g. for spray F, this is likely to happen with small amounts of data.

Question: Why, in this example, is the rug plot not very informative?

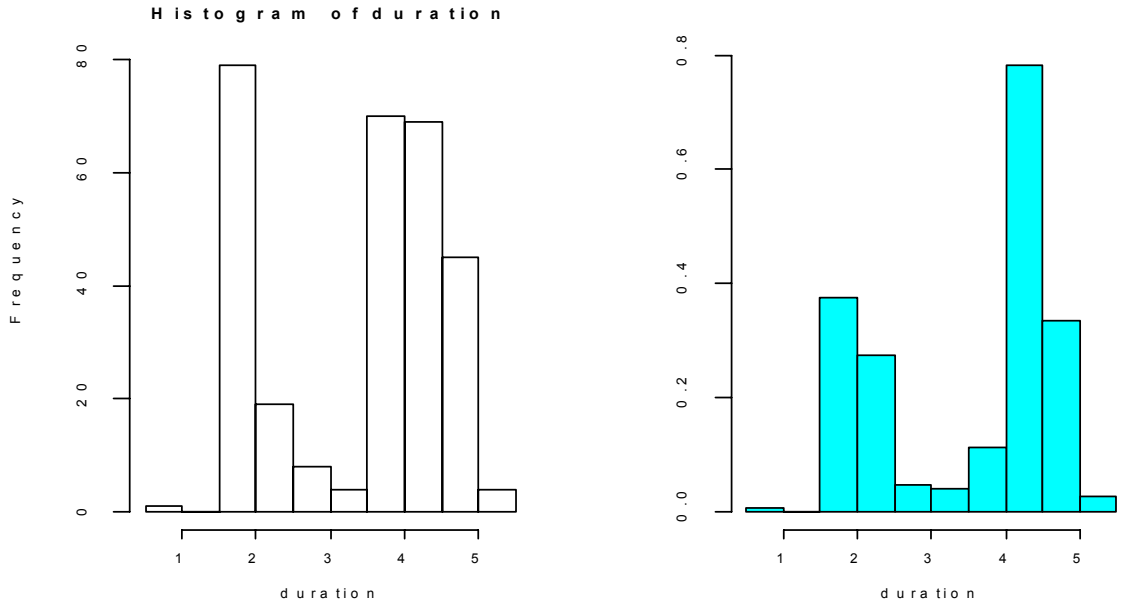
2.2.3 Histograms and density estimation.

2.2.3.1 Histograms

Histograms provide a very simple *density estimate* of the data.

Two functions are useful for drawing histograms, `hist()`(shownon the left below) and `truehist()` (on the right). The first comes from the base library of **R** and by default plots frequencies vertically, the second comes from the MASS library of Venables & Ripley and plots *relative frequencies* vertically,so the total area under the histogram in the second one is 1. Both take many optional arguments controlling the bin width, the number of bins, the class boundaries and it is possible to use unequal bin widths. Type `help(truehist)` to find out more.

```
> data(geyser)
> hist(duration)
> truehist(duration)
```



Question: why are these different (e.g. in range 1.5 to 2.5)?

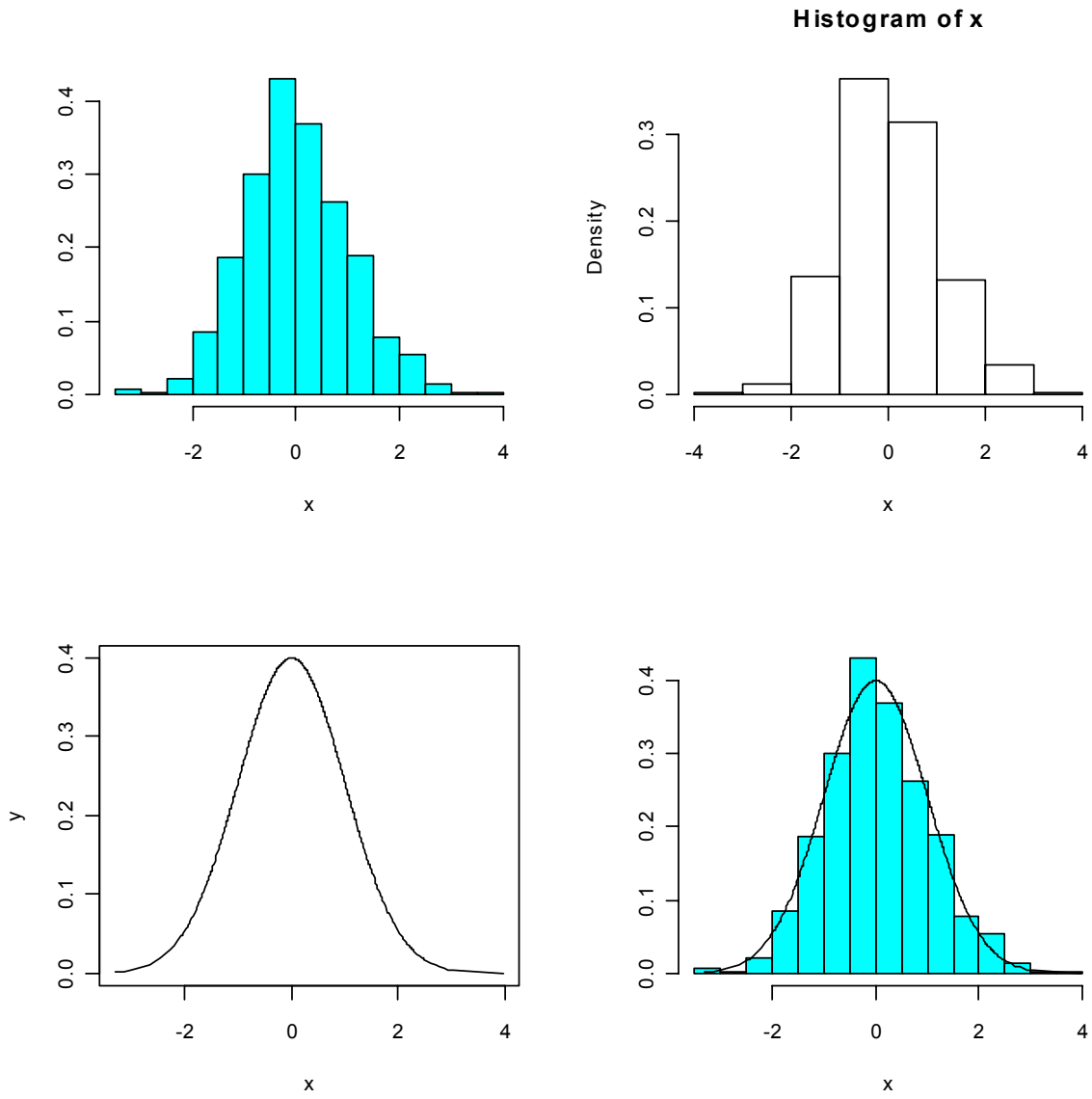
If we think of the data as coming from some density f(.) [i.e. that the data are observations of a random variable with probability density function f(.)] then for any value of x the histogram gives an estimate of f(x),

Specifically, if the class intervals are $c_0, c_1, \ldots, c_k$ and x is in interval (c_i, c_{i+1}) then the histogram estimate of f(x) is $\tilde{f}(x) = \frac{\#(x; c_i \le x < c_{i+1})}{n(c_{i+1} - c_i)}$

If the number of points is large then this will provide quite a good estimate of the true density, but it will depend on the number of bins and the starting values. It is possible to make choices of these based on measures of optimality for sampling from specific distributions, resulting in rules such as *Sturges' formula:* $h=range(x)/(\log_2(n)+1)$ to give the bin width *h* for a sample of *n* observations. Another is *Scott's formula* which gives $h=3.5s(n^{-1/3})$ where *s* is the sample standard deviation or [better] a robust estimate of standard deviation.

The **R** code below produces a histogram of a random sample taken from N(0,1), with superimposed the 'true' density.

```
> x<- rnorm(1000)
> x<- sort(x)
> y<- exp(-x*x/2)/sqrt(2*pi)
> truehist(x)
> hist(x,probability=TRUE)
> plot(x,y,type='l')
> truehist(x)
> lines(x,y,type='l')
```

[Asides: Note the use of the **assign operator** <- which assigns names to objects. Note also the use of `lines()` to add lines to an existing plot (the most recent one), just as `rug()` does.]

2.2.3.2 Kernel Density Estimates

Definition: If we have data $x_1, x_2, \ldots x_n$ which are observations of a density f(.) and if K(.) is any probability density function then the ***Kernel Density Estimate*** of f(x), with kernel K(.) and bandwidth b is given by

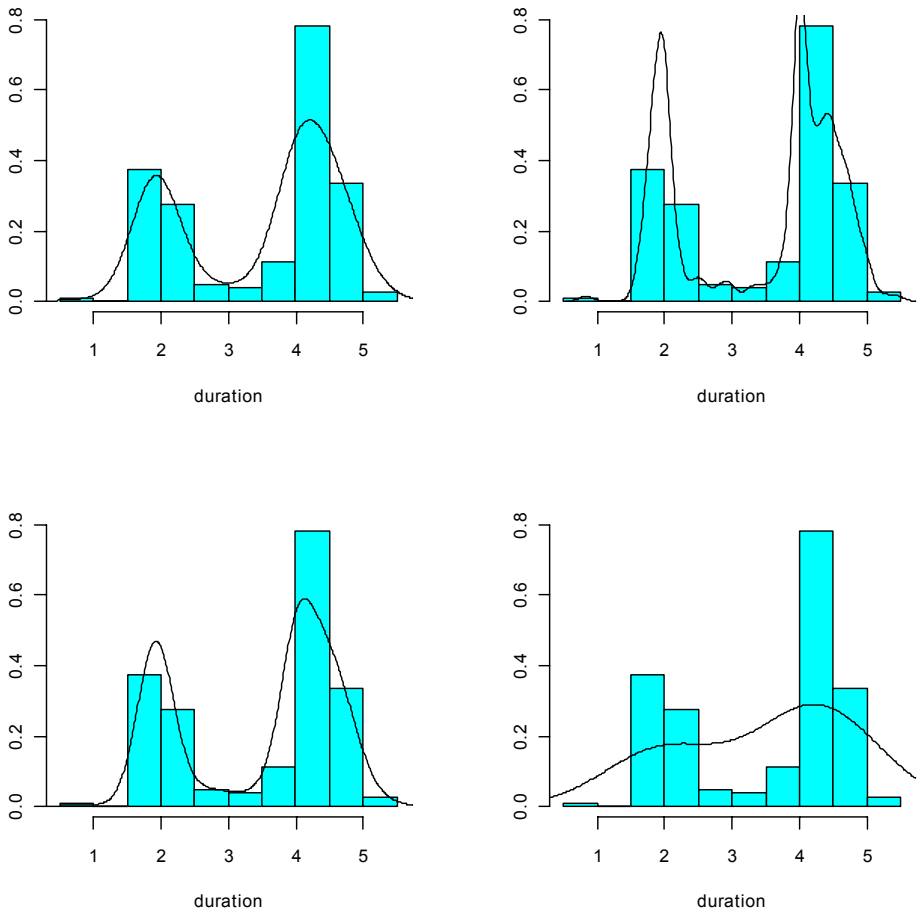
$$\hat{f}(x) = \frac{1}{nb}\sum_{j=1}^{n} K\left(\frac{x - x_j}{b}\right)$$

[It is easy to check that this is a genuine probability density provided that K(.) is, i.e. $\hat{f}(x) \geq 0$ for all x and $\int f(t)dt = 1$]

The ***smoothing parameter*** or bandwidth b is open to choice and is similar to the bin width in histograms. If b is small then the kernel estimate is very rough, if it is large then the estimate is smooth. Similar arguments to choosing the bin width for histograms can be used to show that the best bandwidth is proportional to $n^{-1/5}$ with the constant of proportionality dependent both on the kernel used and on the underlying distribution (which you are trying to estimate of course).

A common choice of kernel function is the standard Normal or Gaussian, i.e. $f(x) = (2\pi)^{-\frac{1}{2}} \exp(-\frac{1}{2}x^2)$ but other choices are available (e.g. rectangular, triangular and Epanechnikov) and there are various theoretical results available for choosing them.

```
> library(MASS)
> data(geyser)
> attach(geyser)
> par(mfrow=c(2,2))
> truehist(duration)
> lines(density(duration))
> truehist(duration)
> lines(density(duration, adjust=0.3))
> truehist(duration)
> lines(density(duration,adjust=0.7))
> truehist(duration)
> lines(density(duration, adjust=2.5))
```



Kernel density estimates of Old Faithful data with default, 0.3×default, 0.7×default and 2.5×default bandwidths.

Comments: Kernel density estimates are an easy and attractive alternative or additional tool to histograms. Although you have to choose the bandwidth, as you do in histograms, they do not depend upon choices of starting values of class intervals nor upon whether you regard the classes as open or closed on the left/right.

A more important reason for considering them is that they can be used in more sophisticated methods, e.g. in problems of testing for mixtures of distributions the minimum value of the bandwidth (b_{crit} say) for which the data is unimodal can be used as a test statistic for bimodality.

Exercises 1

(some of these exercises or for working on during the micro sessions)

1. At various places in the courses notes, questions have been posed. You should think about (and answer!) these questions.

2. If you are not familiar with R (or S-plus) then try repeating some of the sessions given in the notes. In particular, try finding out more about the commands and functions to enhance the analyses and make the displays look better, e.g. with labels, titles etc. Some credit in the assessment will be reserved for the quality of the presentation.

3. Look at the websites for R to look at the various notes on usage of R referred to on page 2.

4. Look again at the data on InsectSprays: Try the following:
```
data(InsectSprays)
attach(InsectSprays)
xcount<- jitter(count)
par(mfrow=c(1,2))
boxplot(count)
rug(count,side=2)
boxplot(xcount)
rug(xcount,side=2)
```
How about putting the rug plot on the boxplot of the actual counts?

5. Try the effect of different bandwidths and kernels on a sample from a normal distribution. (Start by doing`)`)

6. How would you do kernel density estimation for half-normal data? i.e. data obtained by say `x<- abs(rnorm(300)) ?`

Two Dimensional Kernel Density Estimates

Extensions to two dimensions (and more) are straightforward. In **R** they can be calculated using functions `kde2d()`, and displayed using `contour()` and `persp()`.

The two dimensional kernel density estimate is defined by

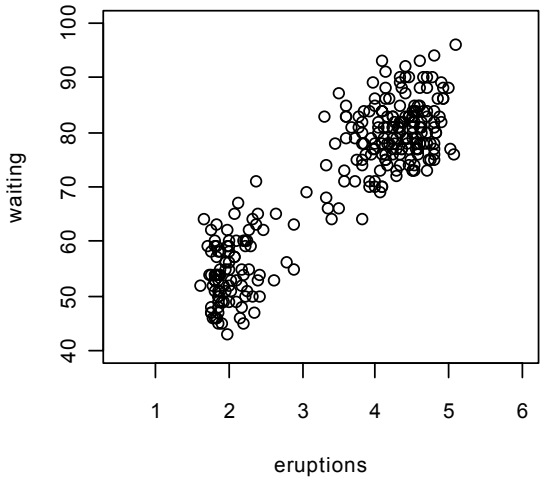
$$\hat{f}(x,y) = \frac{\sum_i \phi((x - x_i)/h_x)\phi((y - y_i)/h_y)}{nh_xh_y}$$ where $\phi(.)$ is a probability density function (e.g. the standard normal) and h_x, h_y are the ***two*** bandwidths.

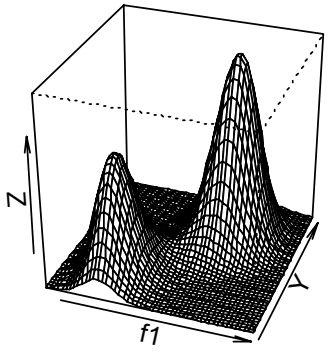
Example: (the data set `faithful` is in the base library and is the same as `geyser` but with a different variable name)

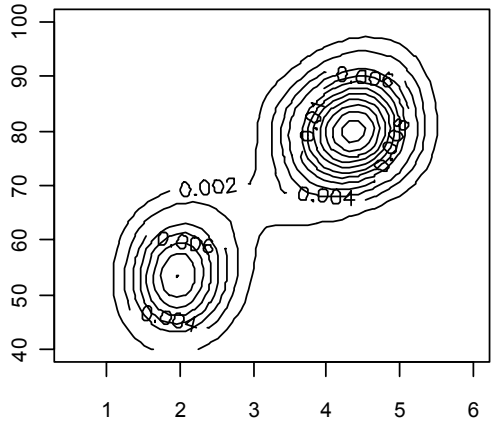
```
> data(faithful)
> attach(faithful)
> summary(faithful)
```

```
   eruptions        waiting
 Min.   :1.600   Min.   :43.0
 1st Qu.:2.163   1st Qu.:58.0
 Median :4.000   Median :76.0
 Mean   :3.488   Mean   :70.9
 3rd Qu.:4.454   3rd Qu.:82.0
 Max.   :5.100   Max.   :96.0
```

```
> plot(eruptions,waiting,xlim=c(0.5,6),ylim=c(40,100))
> f1 <- kde2d(eruptions, waiting, n=50, lims=c(0.5,6,40,100))
>  persp(f1, phi=30, theta=20, d=5)
> contour(f1)
```





It is possible to choose the angle of view in the perspective drawing, the levels of contours plotted, put labels on the axes etc, etc, …… .

Choice of bandwidth:

The theoretical optimal choice of bandwidth depends on what the true density is that we are estimating. However, we do not know what this is (which is why we are estimating it). However, we do have an estimate of the density (!!), provided of course we know what the optimal bandwidth is. Can we use this somehow?

Yes, by using ***cross-validation.*** The idea is to leave one observation out and then estimate the density using the Other n–1 observations and compare the estimate with the observation left out in some way. Then we do this again, leaving out the next observation, and then the next. We then choose the bandwidth *b* to make the match as good as possible.

Specifically, in this case we choose b to minimize

$\mathrm{UCV}(b) = \int \hat{f}(x;b)^2 dx - \frac{2}{n}\sum_{i=1}^{n} \hat{f}(x_i;b)_{-i}$ where $\hat{f}(x_i;b)_{-i}$ is the kernel density estimate based on the n–1 observations leaving out x_i .

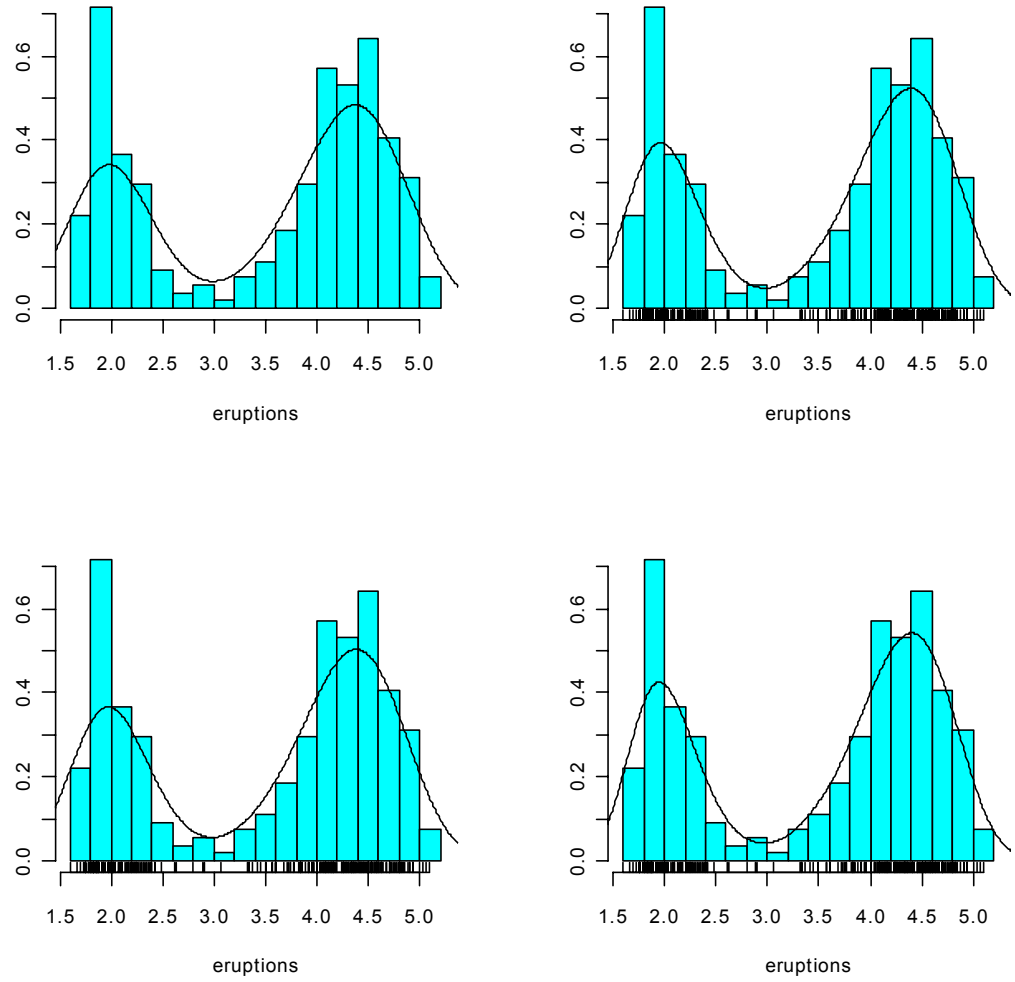
The idea of cross-validation is used in many different contexts in statistical analysis.

Another use of kernel density estimates:

Suppose we want to estimate the median of a set of data (e.g. of the geyser eruptions). Obviously the sample median is a sensible estimate but how do obtain it's standard error? It can be shown that in large samples, the median of a sample from f(.) with true median *m* is asymptotically Normally distributed N(*m,* $1/\{4n[f(m)]^2\}$). So, the standard error depends upon the value of the density at the median. This can be estimated from the kernel density estimate.

Example:

```
> library(MASS)
> data(faithful)
> attach(faithful)
> summary(faithful)

> median(eruptions)
[1] 4

> truehist(eruptions, nbins=15)
> lines(density(eruptions))
> truehist(eruptions, nbins=15)
> lines(density(eruptions,adjust=0.8))
> rug(eruptions)
> truehist(eruptions, nbins=15)
> lines(density(eruptions,adjust=0.9))
> rug(eruptions)
> truehist(eruptions, nbins=15)
> lines(density(eruptions,adjust=0.7))
> rug(eruptions)
> density(eruptions,n=1,from=3.99, to=4.01)$y
[1] 0.3808035
> density(eruptions,n=1,from=3.99, to=4.01,adjust=0.9)$y
[1] 0.3858891
> density(eruptions,n=1,from=3.99, to=4.01,adjust=0.8)$y
[1] 0.3901167
> density(eruptions,n=1,from=3.99, to=4.01,adjust=0.7)$y
[1] 0.3937933
```

Note that we first found that the sample median was 4. Then investigation of the kernel density estimates suggested that the default choice of bandwidth was a little too large, so try a few other values slightly smaller. Then note use of `density` with `n=1`, to ensure only one value calculated, over a range around the sample median and also note the use of `density(.....)$y` to extract the y-coordinate.

Conclusion, $\hat{f}(m) \simeq 0.39$ so standard error of the estimate 4.0 of the median is $(4 \times 272 \times 0.39^2)^{-½} = 0.078$

2.3 Summary

The key ideas introduced here have been problems of

- ways of summarizing and displaying data, perhaps informally
- **robustness** & **resistance** to **model deviation** and **data contamination**
- **kernel density estimates** and their use for a variety of problems
- idea of **cross-validation**

These ideas are especially useful because they allow us to examine assumptions made in statistical analyses and they provide a starting point for developing methods which are not so sensitive to failures in assumptions.