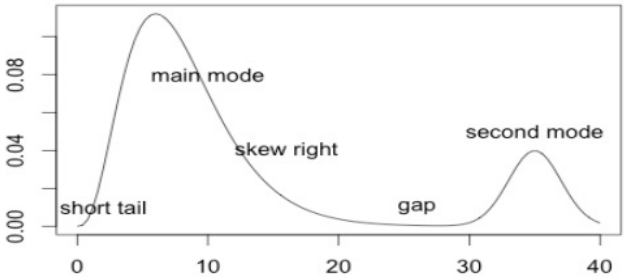
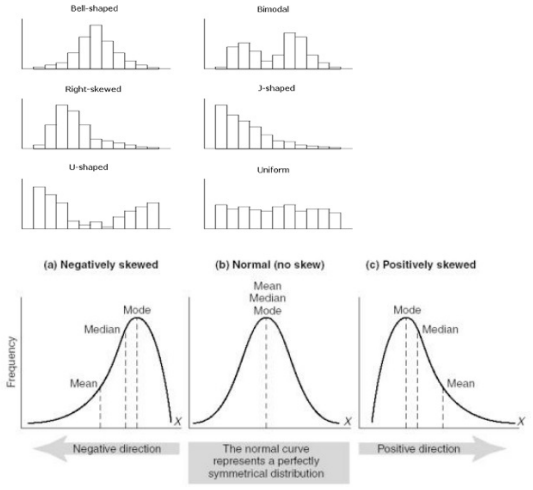
Recall



- **Distribution:** Pattern of values for a variable.
- **Mode:** High density region.
- **Long tail:** Many observations far from centre.
- **Symmetry/Skewness:** Distribution of values to the left and right of the centre.
- **Gaps:** Places where there are no observations.
- **Outliers:** Unusually large or small values that fall well beyond overall pattern of data.

Recall



Source: www.condor.depaul.edu

Histograms

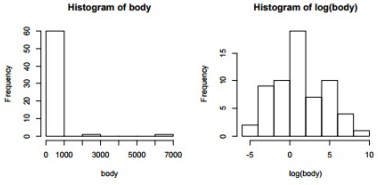
Description

The generic function `hist` computes a histogram of the given data values. If `plot = TRUE`, the resulting object of class `"histogram"` is plotted by `plot.histogram`, before it is returned.

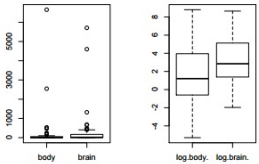
Usage

```
hist(x, ...)

## Default S3 method:
hist(x, breaks = "Sturges",
     freq = NULL, probability = !freq,
     include.lowest = TRUE, right = TRUE,
     density = NULL, angle = 45, col = NULL, border
= NULL,
     main = paste("Histogram of" , xname),
     xlim = range(breaks), ylim = NULL,
     xlab = xname, ylab,
```

Histograms And Transformations



Data transformation can be very helpful for both data visualisation and model fit. A logarithmic scale (as used here) is very common in biological applications.




```
> hist(body)
> hist(log(body))
> boxplot(data.frame(body,brain))
> boxplot(data.frame(log(body),log(brain)))
```

Boxplots

Description

Produce box-and-whisker plot(s) of the given (grouped) values.

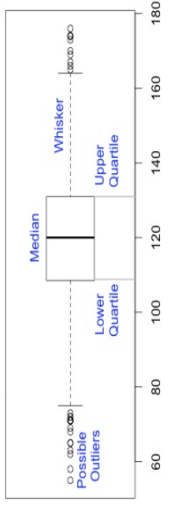
Usage

```
boxplot(x, ...)

## S3 method for class 'formula'
boxplot(formula, data = NULL, ..., subset, na.action = NULL)

## Default S3 method:
boxplot(x, ..., range = 1.5, width = NULL, varwidth = FALSE,
        notch = FALSE, outline = TRUE, names, plot = TRUE,
        border = par("fg"), col = NULL, log = "",
        pars = list(boxwex = 0.8, staplewex = 0.5, outwex = 0.5),
        horizontal = FALSE, add = FALSE, at = NULL)
```

`range`

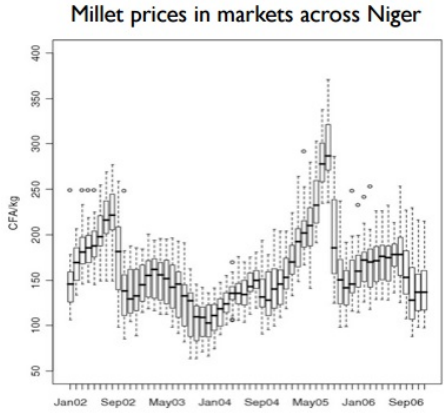
this determines how far the plot whiskers extend out from the box. If range is positive, the whiskers extend to the most extreme data point which is no more than range times the interquartile range from the box. A value of zero causes the whiskers to extend to the data extremes.

Default: `range = 1.5`



Boxplots For Comparing Distributions

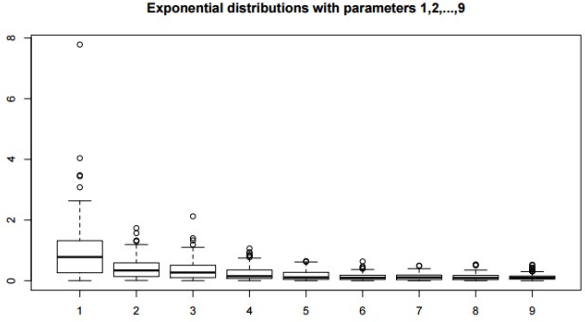
Concise nature of boxplots (median, IQR, min/max, outliers) allow for comparison of great many of them. This facilitates detection of patterns, e.g. temporal trends or dependency on other factors.



Source: www.springerimages.com/Images/LifeSciences/1-10.1007_s12571-010-0065-4-2

Distribution Comparisons With Boxplots

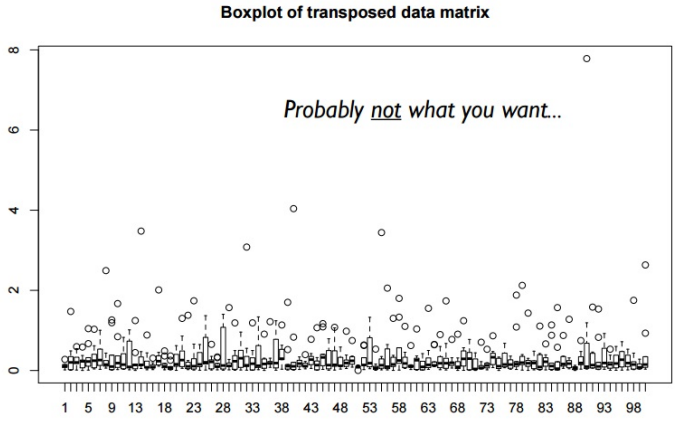
```
> M<-array(0, dim=c(100,9))
> for (n in 1:9){
+ M[,n]<-rexp(100,n)}
> boxplot(M, main="Exponential distributions with
parameters 1,2,...,9")
```



A Note Of Caution

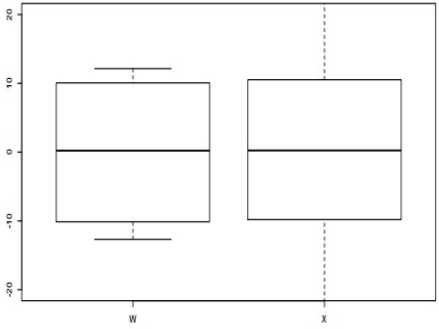
Make sure to take boxplots of columns and not rows!

```
> boxplot(t(M))
```



What You Can See From A Boxplot...

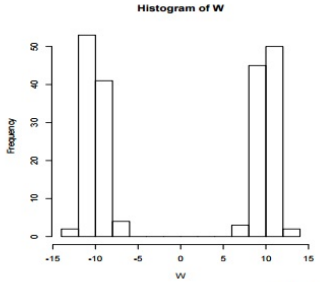
```
> par(mfrow=c(1,1))
> boxplot(data.frame(W,X),ylim=c(-20,20))
```

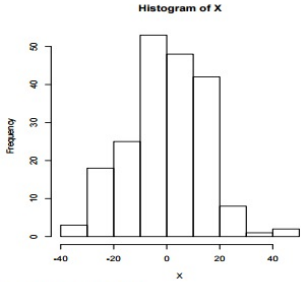


W and X have about the same median and interquartiles. X seems to have more spread overall and longer tails.

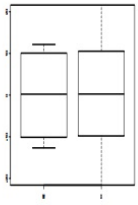
...And What You Can't

```
> hist(W)
> hist(X)
```





The same data W, X!



How the data sets were created:

```
> U<-rnorm(100,-10,1)
> V<-rnorm(100,10,1)
> W<-c(U,V)
> X<-rnorm(200,0,1.4*sd(W))
```

Quantile Function

Another graphical method for comparing two probability distributions is by plotting their quantiles against each other. The corresponding plot is called the **Q–Q (quantile-quantile) plot**. To describe this plot we define the **quantile function**. Let X be a random variable with cumulative distribution function F:

$$F(x) = P(X \leq x) \ (x \in \mathbb{R}).$$

Then the quantile function is defined as

$$Q(p) = \inf\{x \in \mathbb{R} | p \leq F(x)\}$$

i.e., $Q(p)$ is the smallest value x such that the probability that $X \leq x$ is p.

Quantile Function In R

Description

The generic function `quantile` produces sample quantiles corresponding to the given probabilities. The smallest observation corresponds to a probability of 0 and the largest to a probability of 1.

Usage

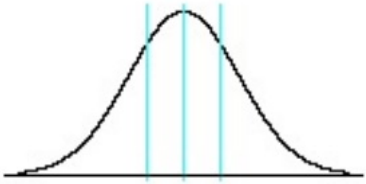
```
quantile(x, ...)

## Default S3 method:
quantile(x, probs = seq(0, 1, 0.25), na.rm = FALSE,
         names = TRUE, type = 7, ...)
```

```
> X<-rexp(100)
> quantile(X)
         0%         25%         50%         75%
100%
0.01551301 0.31147708 0.72407041 1.39531874
5.47955673
> quantile(X, c(0.05,0.95))
        5%        95%
0.1005241 3.1334639
```

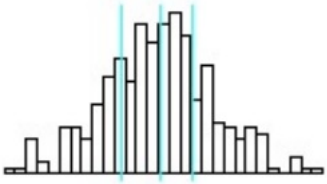
Refined Comparison Of Distributions

Compare quantiles.

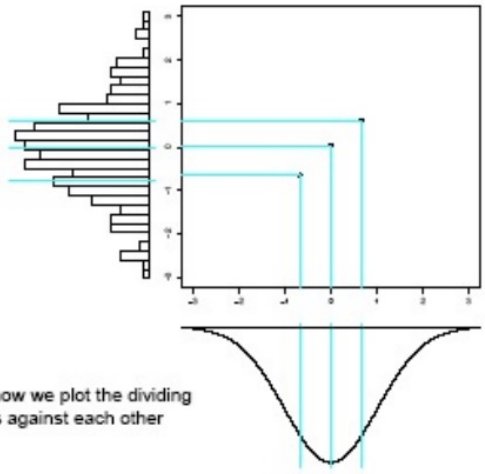
Quartiles normal distribution



Quartiles other distribution

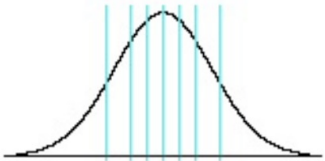


Comparing Quartiles

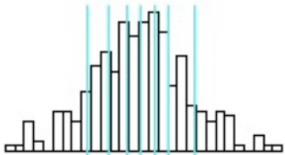


And now we plot the dividing points against each other

Finer Intervals



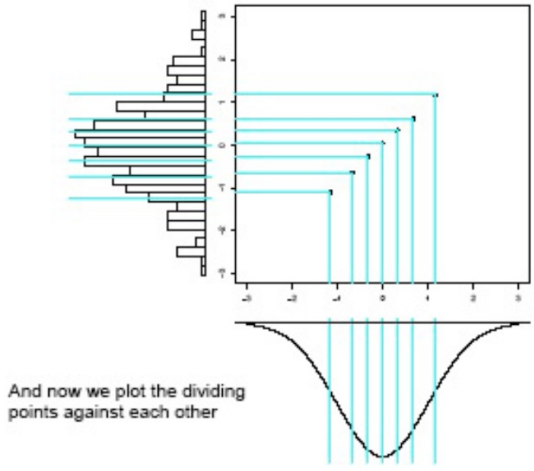
normal distribution



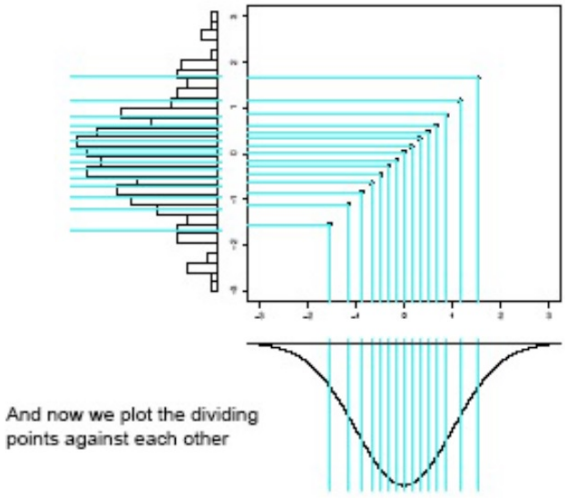
other distribution

8 intervals on the x-axis such that the corresponding areas under the curve are equal

More Comparing



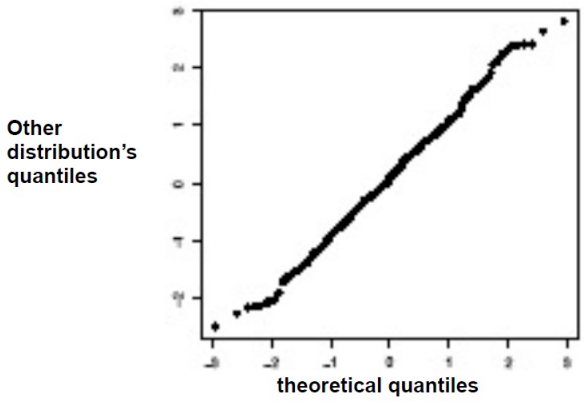
Finer Finer Intervals



And now we plot the dividing points against each other

Q-Q Plot

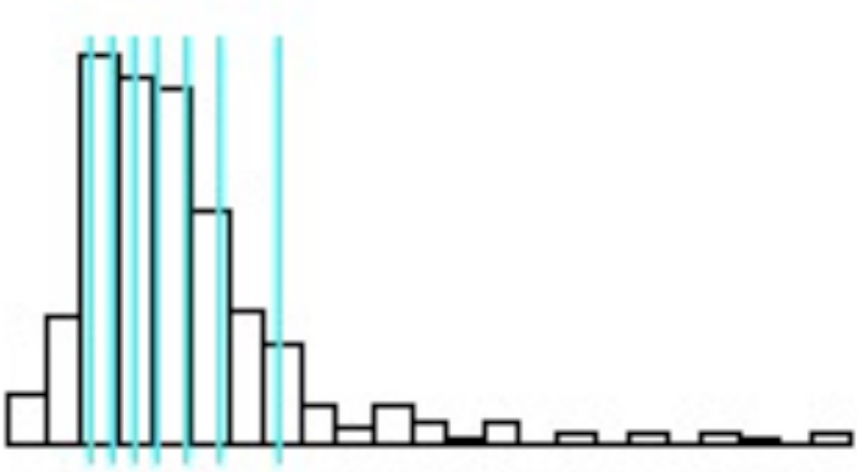
Eventually we get the normalised quantile-quantile plot (Q-Q plot):



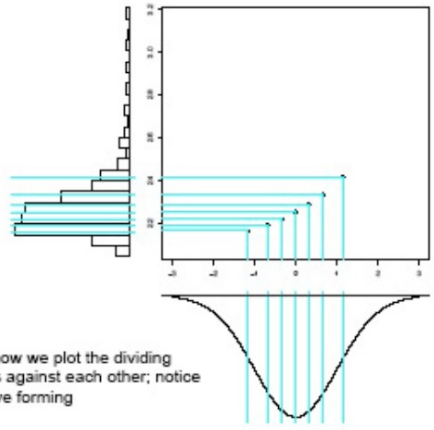
The closer these points are to a straight line, the more plausibly normal the distribution.

Q-Q Plot Example

What does the Q-Q plot for this positively-skewed distribution look like?

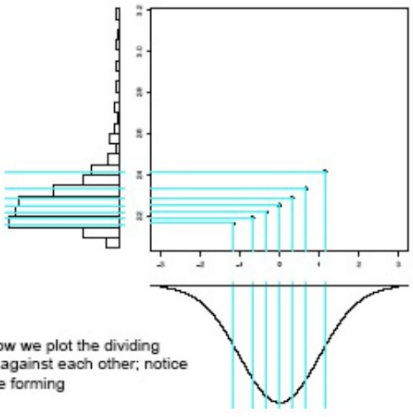


Q-Q Plot Example Cont.

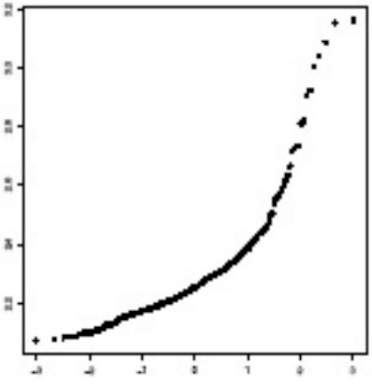


And now we plot the dividing points against each other; notice a curve forming

Q-Q Plot Example Cont.



And now we plot the dividing points against each other; notice a curve forming



theoretical quantiles

The qqnorm Function

Description

qqnorm is a generic function the default method of which produces a normal QQ plot of the values in y. qqline adds a line to a "theoretical", by default normal, quantile-quantile plot which passes through the probs quantiles, by default the first and third quartiles.

qqplot produces a QQ plot of two datasets.

Graphical parameters may be given as arguments to qqnorm, qqplot and qqline.

Arguments

x
The first sample for qqplot.

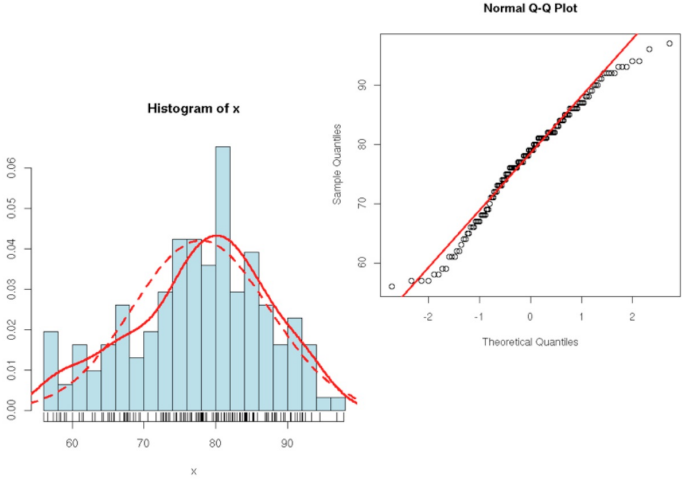
y
The second or only data sample.

xlab, ylab, main
plot labels. The xlab and ylab refer to the y and x axes respectively if datax = TRUE.

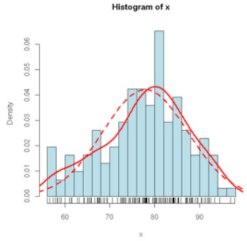
plot.it
logical. Should the result be plotted?

...

Use Of `qqline` Function: Example



(How I Drew The Last Slide)




```
data(airquality)
x <- airquality[,4]
hist(x, probability=TRUE, breaks=20,
col="light blue")
rug(jitter(x, 5))
points(density(x), type='l', lwd=3,
col='red')
f <- function(t) {
  dnorm(t, mean=mean(x), sd=sd(x) )
}
curve(f, add=T, col="red", lwd=3,
lty=2)
```

```
x <- airquality[,4]
qqnorm(x)
qqline(x,
       col="red", lwd=3)
```

Leptokurtosis Example