University of Washington

STATISTICS

Elements of Statistical Methods

Data (Ch 7)

Fritz Scholz

Spring Quarter 2010

May 5, 2010

Data

Data consists of the actual numbers observed in a random experiment.

If such an experiment gives rise to a random variable X, its observed value is usually indicated by writing it as a lower case x or the specific number, say 1.

If you toss a coin and `Heads` is observed, our random variable X
(taking on the values 1 and 0 with probabilities p and $1-p$)
is distributed as $X \sim \text{Bernoulli}(p)$. In this case the observed value is $x = 1$.

To learn more about an experiment and the distribution of its associated r.v. X we replicate the experiment a fixed number of times.

Replication means that the experiment is repeated under the same conditions and that the repetitions can be considered to be independent.

Replicated Experiments

The replicated experiment just described is a random experiment in itself.

We observe a random vector $(X_1, \ldots, X_n)$ with $X_i \sim P$ and $X_1, \ldots, X_n$ independent.

The probability distribution P and the independence of the X_i allow us to write

$$P(X_1 \in A_1, \ldots, X_n \in A_n) = P_X(X_1 \in A_1) \cdot \ldots \cdot P_X(X_n \in A_n)$$

for all $A_i \in \mathcal{C}$.

We express this by writing $X_1, \ldots, X_n \sim P$ or $X_1, \ldots, X_n$ i.i.d. $\sim P$,

where i.i.d. is short for independent and identically distributed.

The set of observed values of $X_1, \ldots, X_n$ is denoted by $\vec{x} = \{x_1, \ldots, x_n\}$.

This set is also called a sample.

To emphasize observation order we also use the sample vector $x = (x_1, \ldots, x_n)$.

The Basic Problem of Statistics

We can model experiments with the mathematical abstraction of a probability space and associated random variables or random vectors.

We will never be completely certain about the distributional nature of the random variables involved.

By observing samples we hope to get some idea about certain aspects of the underlying probability model.

Is it a reasonable model?

Can we approximately say something about its mean or standard deviation?

This segment introduces descriptive techniques for examining data or samples.

Simulation

A very powerful technique is to learn how samples and certain derived statistics behave under simulated sampling from various probability models.

In a class room setting this was not easy or practical until recently.

R provides many useful tools that assist in such simulations, with much ease.

```
> SampleSpace <- c(1,2,3,4,5,6)
> sample(x=SampleSpace,size=20,replace=T)
 [1] 1 6 4 1 5 2 6 4 5 3 4 6 1 4 5 6 6 4 2 3
 > sample(x=SampleSpace,size=20,replace=T)
 [1] 5 3 2 3 1 1 3 3 4 6 1 5 1 3 5 1 6 6 4 5
```

Note the different results when repeating the command `sample(...)`.

rnorm

The previous slide gave an example of sampling from a finite population $(1,2,3,4,5,6)$.

For most distributions in R there is a command that simulates a random sample from such a distribution, i.e., generates a sample vector $(x_1,\ldots,x_n)$ as observed from $X_1,\ldots,X_n \sim P$.

For example, to generate a sample of size $n = 5$ from $\mathcal{N}(2,3)$ execute

```
> rnorm(5,mean=2,sd=sqrt(3))
[1]  1.4872959  2.7720840 -0.1097585  0.4419888  2.8753901
```

Note the correspondence of `dnorm, pnorm, qnorm, rnorm`.

Similar quadruple functions exist for the other distributions in R, e.g.,

`dunif, punif, qunif, runif`.

Random Number Generation

The previous examples of simulating random experiments use a random number generator (RNG) algorithm.

Being algorithmic such a random number generator is actually deterministic.

It generates a very, very long sequence of numbers in the inteval $(0, 1)$ and then it repeats itself.

However, the sequence of these numbers has the appearance of being independent uniform random numbers from $(0, 1)$.

There are several versions of such generators, but we will take the default version.

All other r.v.'s can be simulated from uniform (0,1) random numbers.

Setting the Seed of an RNG

While it is desirable to get new samples in each simulation

(unless we have exhausted the RNG, which is unlikely),

sometimes one wants to get the same sample.

When debugging an analysis using random numbers, one may want to make sure that any changes in analysis results are due to a change in analysis and not due a change in the random sample used.

Or one wants to make sure that all students use the same sample in their analysis, without having to type or import the data.

Such repeat behavior of an RNG can be obtained in R

by using the function `set.seed`.

Examples of Setting the Seed

Note and understand the differences or identical behavior in the following

```
> rnorm(5)
[1]  0.5219445  2.1372648  1.6340132  0.4252533 -0.1603633
> rnorm(5)
[1] -0.45036275 -0.01862231  0.83056581  1.29583131 -0.84968317
> set.seed(321)
> rnorm(5)
[1]  1.7049032 -0.7120386 -0.2779849 -0.1196490 -0.1239606
>  set.seed(321)
> rnorm(5)
[1]  1.7049032 -0.7120386 -0.2779849 -0.1196490 -0.1239606
> rnorm(5)
[1]  0.2681838  0.7268415  0.2331354  0.3391139 -0.5519147
```

Another Example

```
> set.seed(321)
> rnorm(5)
[1]  1.7049032 -0.7120386 -0.2779849 -0.1196490 -0.1239606
> rnorm(5)
[1]  0.2681838  0.7268415  0.2331354  0.3391139 -0.5519147
> set.seed(321)
> rnorm(5)
[1]  1.7049032 -0.7120386 -0.2779849 -0.1196490 -0.1239606
> rnorm(5)
[1]  0.2681838  0.7268415  0.2331354  0.3391139 -0.5519147
```

The Plug-In Principle

Previously we introduced a structure for describing random experiments, using probability spaces, random variables, their distributions, and some of their characteristics (mean, median, standard deviation, iqr, etc.).

Now we will capitalize on this structure by viewing any given sample as a sample space with an empirical probability measure defined on it.

In doing so, we construct random variables, their distributions and characteristics in complete parallel with the original experiment giving rise to $\vec{x} = \{x_1, \ldots, x_n\}$.

This action is very useful and is sometimes referred to as the plug-in principle.

Aside from the parallelism this principle is supported by the Fundamental Theorem of Statistics, to be discussed later.

The Empirical Probability Distibution

Definition: Let $\vec{x} = \{x_1, \ldots, x_n\}$ be a sample. The empirical probability measure, associated with $\vec{x}$ and denoted by $\hat{P}_n$, is the discrete probability distribution which assigns probability $1/n$ to $x_i, i = 1, \ldots, n$, i.e., $\hat{P}_n(\{x_i\}) = 1/n$.

The corresponding empirical sample space is $S_n = \vec{x}$, which may contain duplicates.

Suppose a fair die is rolled $n = 20$ times, resulting in

$$\vec{x} = \{1,6,3,2,2,3,5,3,6,4,3,2,5,3,2,2,3,2,4,2\}$$

x_i	$\#\{x_i\}$	$\hat{P}_{20}(\{x_i\})$
1	1	0.05
2	7	0.35
3	6	0.30
4	2	0.10
5	2	0.10
6	2	0.10

The Table shows the empirical probability distribution.

The fact that it differs from P is an expression of sampling variation.

Using `table`

```
> x <- c(1,6,3,2,2,3,5,3,6,4,3,2,5,3,2,2,3,2,4,2)
> x
 [1] 1 6 3 2 2 3 5 3 6 4 3 2 5 3 2 2 3 2 4 2
> xtab <- table(x)
> xtab
x
1 2 3 4 5 6
1 7 6 2 2 2
> names(xtab)
[1] "1" "2" "3" "4" "5" "6"
> as.numeric(xtab)
[1] 1 7 6 2 2 2
> as.numeric(names(xtab))
[1] 1 2 3 4 5 6
```

$\hat{P}_n$ is a Probability Distribution

Let $\mathcal{C}_n$ be the collection of all subsets of S_n. For any set $A \in \mathcal{C}_n$ we define

$$\hat{P}_n(A) = \frac{\#\{x_i \in A\}}{n}$$

and it satisfies all our axioms for a probability measure.

Here A, like S_n, may contain duplicate values, and they are counted as such.

We can define a random variable $X : S_n \longrightarrow R$ by $X(x_i) = x_i$.

The cdf of this X (w.r.t. $\hat{P}_n$) is

$$\hat{F}_n(y) = \hat{P}_n(X \leq y) = \frac{\#\{x_i \leq y\}}{n}$$

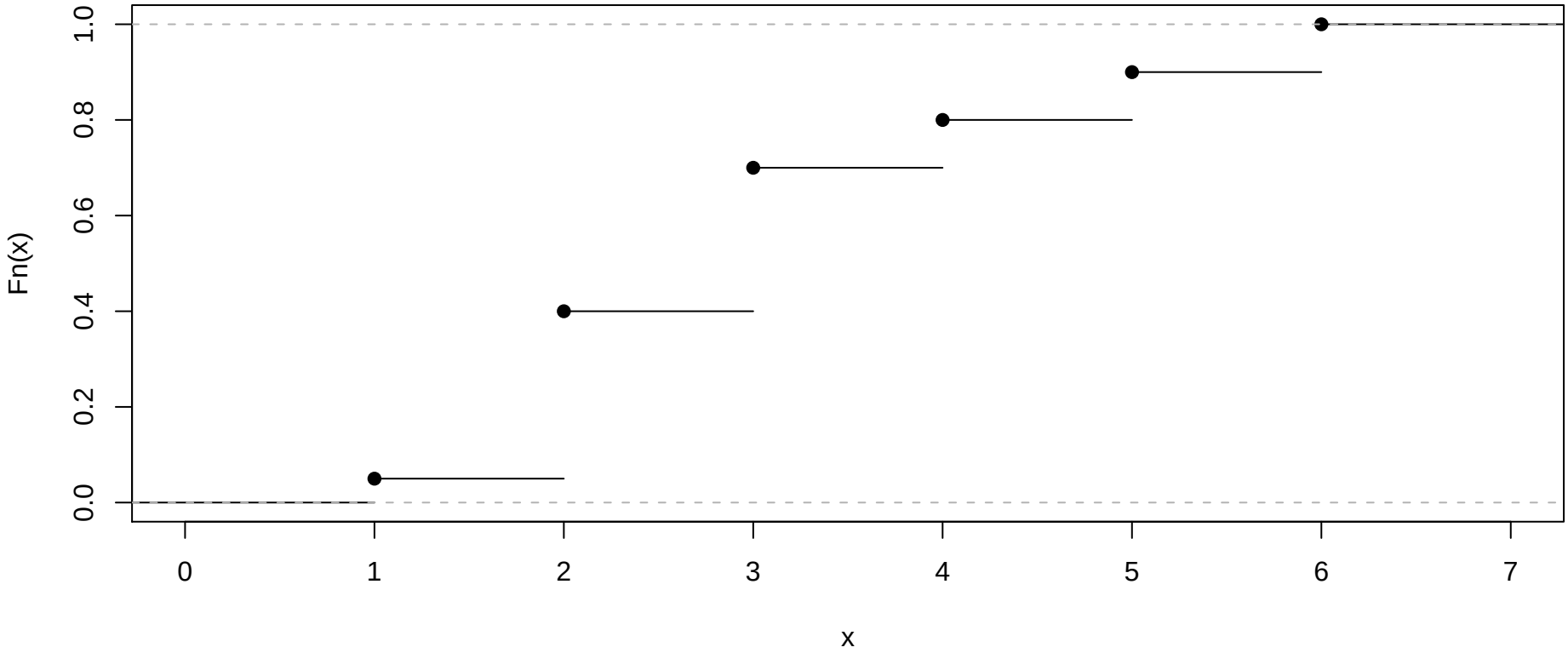
It is called the empirical cdf of the sample.

Note: the above X is different from the X w.r.t. P, which has cdf $F(y) = P(X \leq y)$.

The context should make clear which X is meant. Sometimes $X^* : S_n \longrightarrow R$

is used in place of the above X. $\hat{F}_n(y) = \hat{P}_n(X^* \leq y) = \#\{x_i \leq y\}/n$, the same.

Empirical CDF (ECDF) Plot



Using our previous `x <- c(1,6,3,2,2,3,5,3,6,4,3,2,5,3,2,2,3,2,4,2)` and the command `plot.ecdf(x,main="")` produced the above plot. Note that the vertical axis is labeled $F_n(x)$, rather than $\hat{F}_n(x)$, a common alternative notation.

Plug-In Estimate of the Mean

Definition: The plug-in estimate of $\mu = EX$, denoted by $\hat{\mu}_n$,

is the mean of the ECDF:

$$\hat{\mu}_n = \sum_{i=1}^{n} x_i \cdot \frac{1}{n} = \frac{1}{n} \sum_{i=1}^{n} x_i = \bar{x}_n$$

This quantity is also called the sample mean or sample average.

For the fair die we have

$$\mu = EX = 1 \cdot \frac{1}{6} + 2 \cdot \frac{1}{6} + \ldots + 6 \cdot \frac{1}{6} = \frac{1+2+\ldots+6}{6} = 3.5$$

For our previous sample `x` we get

$$\hat{\mu}_{20} = \bar{x}_{20} = 1 \cdot \frac{1}{20} + 2 \cdot \frac{7}{20} + 3 \cdot \frac{6}{20} + 4 \cdot \frac{2}{20} + 5 \cdot \frac{2}{20} + 6 \cdot \frac{2}{20} = \texttt{mean(x)} = 3.15$$

Note: $\hat{\mu}_{20} \neq \mu$ an example of sampling variation.

Plug-In Estimate of the Variance

Definition: The plug-in estimate of σ^2, denoted $\widehat{\sigma}^2_n$, is the variance of the ecdf

$$\widehat{\sigma}^2_n = \sum_{i=1}^{n} (x_i - \hat{\mu}_n)^2 \cdot \frac{1}{n} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{\mu}_n)^2 = \frac{1}{n} \sum_{i=1}^{n} x_i^2 - \left(\frac{1}{n} \sum_{i=1}^{n} x_i \right)^2 = \frac{1}{n} \sum_{i=1}^{n} x_i^2 - \hat{\mu}_n^2$$

We do not call $\widehat{\sigma}^2_n$ the sample variance. That term is reserved for

$$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \hat{\mu}_n)^2 = \texttt{var(x)} \qquad \text{note that} \qquad \widehat{\sigma}^2_n = s^2 \cdot \frac{n-1}{n}$$

For a fair die we have

$$\sigma^2 = EX^2 - (EX)^2 = \sum_{i=1}^{6} i^2 \cdot \frac{1}{6} - 3.5^2 = \frac{91}{6} - \left(\frac{7}{2} \right)^2 = \frac{35}{12} = 2.916667$$

For our example of 20 rolls of a die we get

$$\begin{aligned} \widehat{\sigma}^2_{20} &= \left(1^1 \cdot 0.05 + 2^2 \cdot 0.35 + 3^2 \cdot 0.30 + \ldots + 6^2 \cdot 0.10 \right) - 3.15^2 \\ &= \texttt{var(x)} * 19/20 = 1.9275 = \texttt{mean((x - mean(x))}^2\texttt{)} \end{aligned}$$

again $\widehat{\sigma}^2_{20} \neq \sigma^2$, again an example of sampling variation!

Plug-In Estimates of Quantiles

Definition: The plug-in estimate of the population α-quantile is the α-quantile of the ecdf, also called the sample α-quantile. In particular, the sample median is the median of the ecdf. The sample interquartile range= interquartile range of the ecdf.

Simulate a sample of size $n = 20$ from the Uniform(1,5) distribution, with median 3.

```
> set.seed(27) # sets the seed so that you get the same results
> x<-sort(runif(20,min=1,max=5)) # sorts the simulated sample
> options(width=60) # controls the display width
> x # displays the sorted sample
 [1] 1.009801 1.184266 1.289994 1.335030 1.362751 1.548376
 [7] 1.767636 1.889102 2.316925 2.606593 3.010988 3.375997
[13] 3.464381 3.953290 4.145181 4.398470 4.495480 4.505791
[19] 4.887001 4.934419
```

As sample median we should take the average of the middle two values

`2.606593, 3.010988`, i.e., `(x[10]+x[11])/2 = 2.808791 = median(x)`.

Plug-In Estimate of the IQR

Here we have to determine the 0.25- and 0.75-sample quantiles.

We run into the ambiguity that any value in the interval $[\mathrm{x}[5], \mathrm{x}[6]]$ $= [1.362751, 1.548376]$ qualifies as 0.25-sample quantile and any number in $[\mathrm{x}[15], \mathrm{x}[16]] = [4.145181, 4.398470]$ qualifies as 0.75-sample quantile.

One could take the respective interval midpoints $(\mathrm{x}[5] + \mathrm{x}[6])/2 = 1.455564$ and $(\mathrm{x}[15] + \mathrm{x}[16])/2 = 4.271825$, and their difference as our sample iqr:

$$\widehat{\mathrm{iqr}}_{20} = 4.271825 - 1.455564 = 2.81626 \neq \mathrm{iqr}(X) = 4 - 2 = 2$$

Again an example of sampling variability.

No consensus in the statistical community on how best to resolve the ambiguity.

The Quantile Function `quantile`

The R function `quantile` exemplifies the multitude of possibilities.

```
> quantile(x,prob=c(.25,.75))
     25%      75%
1.501970 4.208503 # iqr = 4.208503-1.501970 = 2.706533
# same as quantile(x,prob=c(.25,.75),type=7) default
quantile(x,prob=c(.25,.75),type=2)
     25%      75%
1.455564 4.271825
```

that latter, using `type=2`, yields our average ambiguity resolution.

R offers 9 different options $1, 2, \ldots, 9$ for the `type` input variable to `quantile`.

My suggestion is not to worry, go with the default. Differences in large samples will be small and in small samples the sampling variability is so large that larger differences don't matter.

`quantile` Function for $n = 21$

When $n = 21$ the 0.25- and 0.75-quantiles should be uniquely defined as the 6^{th} and 16^{th} ordered sample values, since $\frac{5}{21} < 0.25 < \frac{6}{21}$ and $\frac{15}{21} < 0.75 < \frac{16}{21}$.

`type=1, type=2, type=7` give the expected answer, the others don't. (Try it.)

```
> set.seed(627)
> xx <- sort(runif(21))
> c(xx[6],xx[16])
[1] 0.1611664 0.8112966
> quantile(xx,prob=c(.25,.75),type=1)
      25%       75%
0.1611664 0.8112966
> quantile(xx,prob=c(.25,.75),type=2)
      25%       75%
0.1611664 0.8112966
> quantile(xx,prob=c(.25,.75),type=7) # the default
      25%       75%
0.1611664 0.8112966
```

The `summary` Function

```
> summary.out <- summary(x) # stores the result in summary.out
> summary.out
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.010   1.502   2.809   2.874   4.209   4.934
> q1q3 <- as.vector(c(summary.out[2],summary.out[5]))
> q1q3[2]-q1q3[1] # sample iqr
  2.707
```

Note that `summary` uses the `type=7` default in `quantile`.

The `as.vector` construct strips away distracting labels. Try it without.

An `iqr` Function

Often it is useful to turn several lines of commands into a function

```
> iqr <- function(x,type=7){
+ q <- as.vector(quantile(x,probs=c(.25,.75),type=type))
+ return(q[2]-q[1])
+ }
> iqr(x)
[1] 2.706533
> iqr(x,2)
[1] 2.816262
```

Creating `iqr` Function in Text Editor

You can also edit a text file, say iqr.txt, with the following content

```
iqr <- function(x,type=7){
  q <- as.vector(quantile(x,probs=c(.25,.75),type=type))
  return(q[2]-q[1])
}
```

and then import it into your R session, started from the same directory that contains `iqr.txt`, via

```
> iqr0 <- dget("iqr.txt")
> iqr0(x)
[1] 2.706533
> iqr0(x,2)
[1] 2.816262
```

Graphical Procedures

While sample mean, median, and other plug-in estimates have their place,
a powerful supplement or alternative is to examine samples graphically.

R provides a wide array of graphical procedures which produce high quality plots.
These can be customized and annotated in many ways.

Such plots can be saved in various formats:

Postscript, pdf, png, jpeg, bmp, tiff and others.

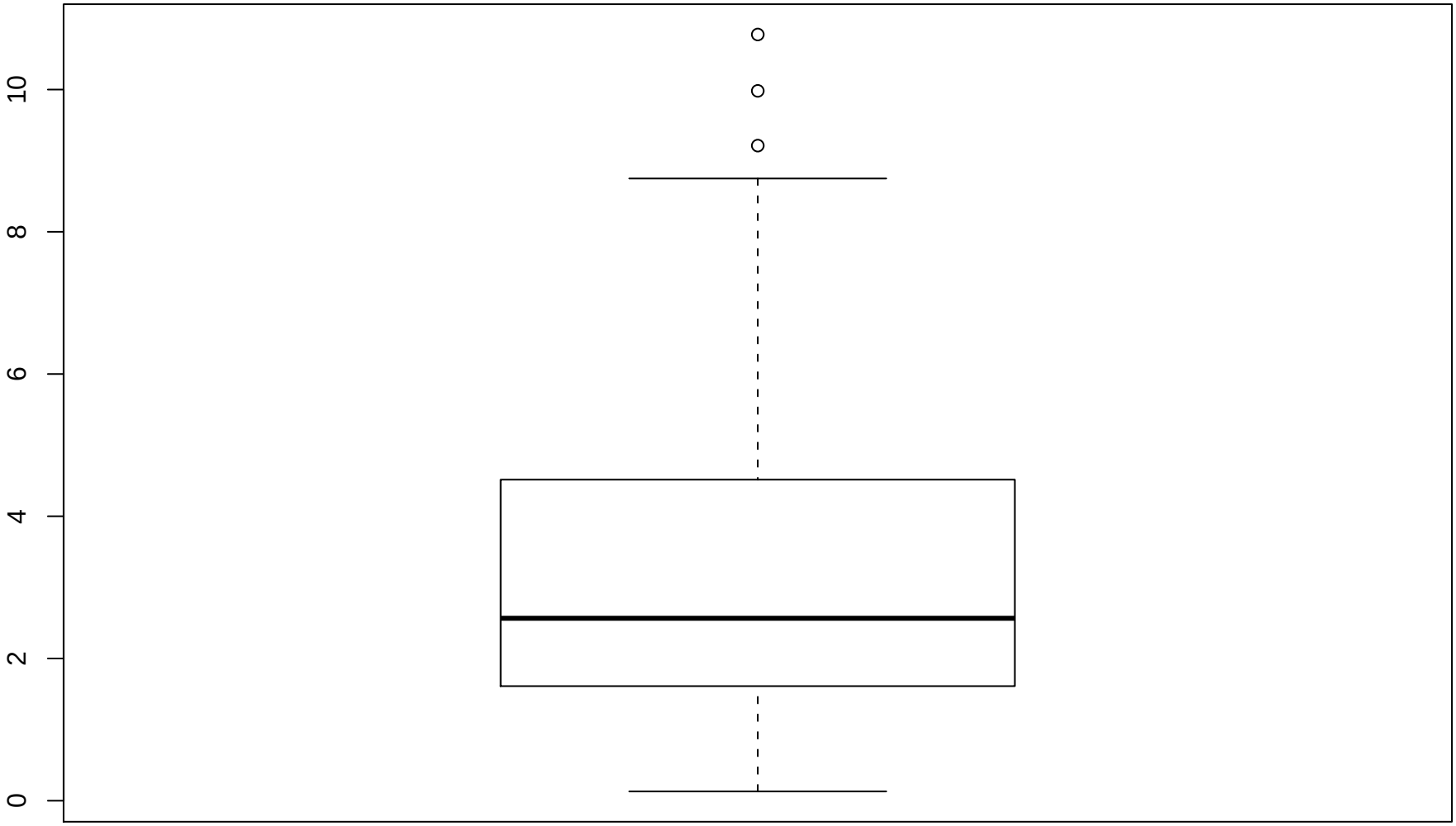
In Windows you can save plots to the clipboard to be used for inclusion in
Word or other documents.

All my illustrations are done in R.

$\chi^2(3)$ Density

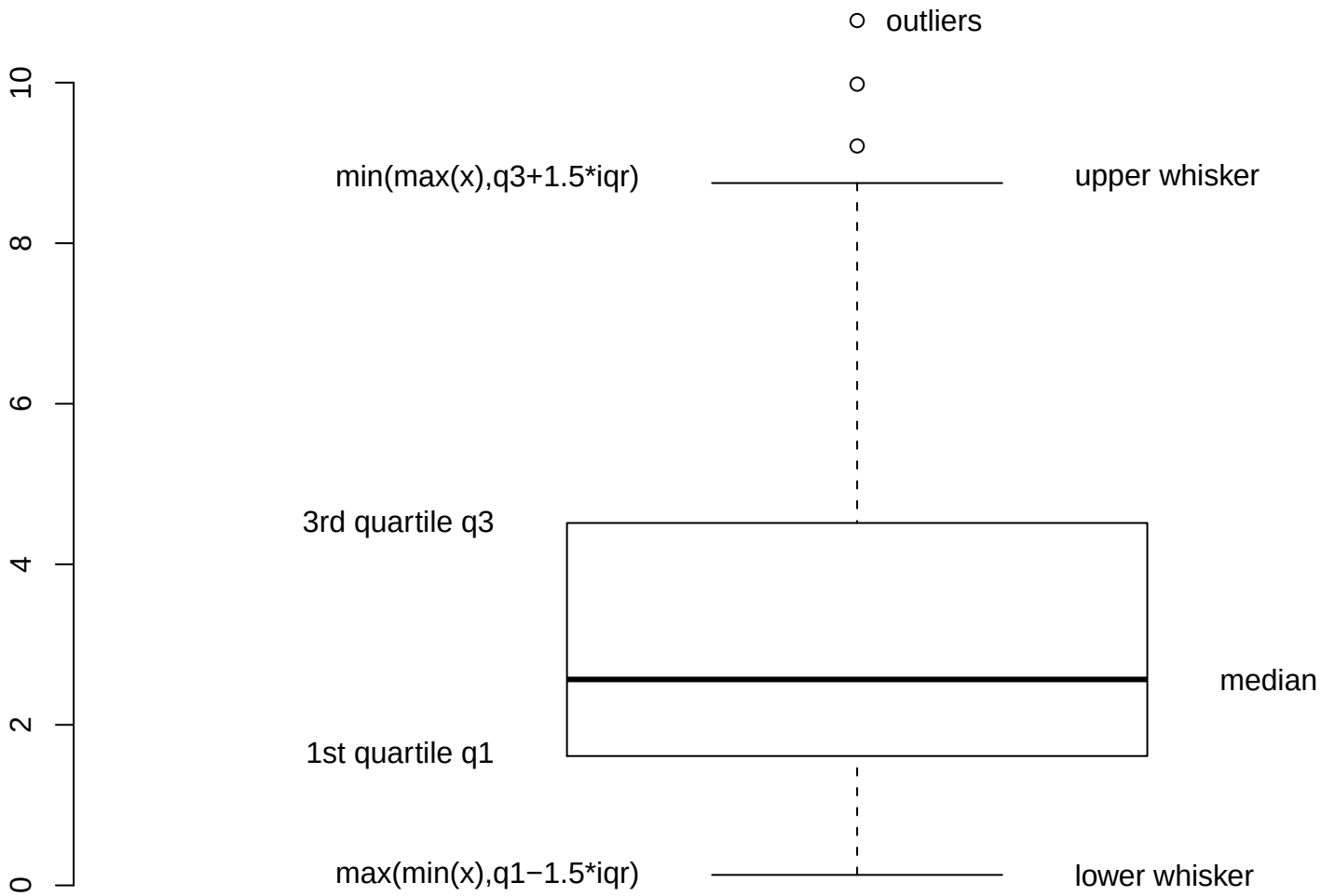
We obtain a sample: `set.seed(267); y <- rchisq(100,3); boxplot(y).`

Box Plots



The box plot gives a quick visual impression of the essential sample features.

Box Plots Explained



The sampled distribution appears not to be symmetric.

Box Plots for Normal Samples

$$\text{iqr} = \texttt{qnorm}(.75) - \texttt{qnorm}(.25) = 1.348980,$$

$$\Longrightarrow \quad \texttt{qnorm}(.75) + 1.5 * 1.348980 = 2.69796$$

for the upper standard normal population whisker.

$$P(|Z| > 2.69796) = 2 * \texttt{pnorm}(-2.69796) = 0.006976582$$

We would expect 7 out 1000 observations to fall beyond the whiskers,

i.e., to have 7 outliers in a sample of size $n = 1000$.

This holds for the standard normal population

but also for any other $\mathcal{N}(\mu, \sigma^2)$ population.

In the latter case the whiskers are at $\mu \pm 2.69796\,\sigma$ and

$$P(|X - \mu| > 2.69796\,\sigma) = P(|Z| > 2.69796) = 0.006976582$$

Comparative Box Plots of Several Samples

Aside from judging whether samples come from symmetric distributions

or have an inordinate number of outliers,

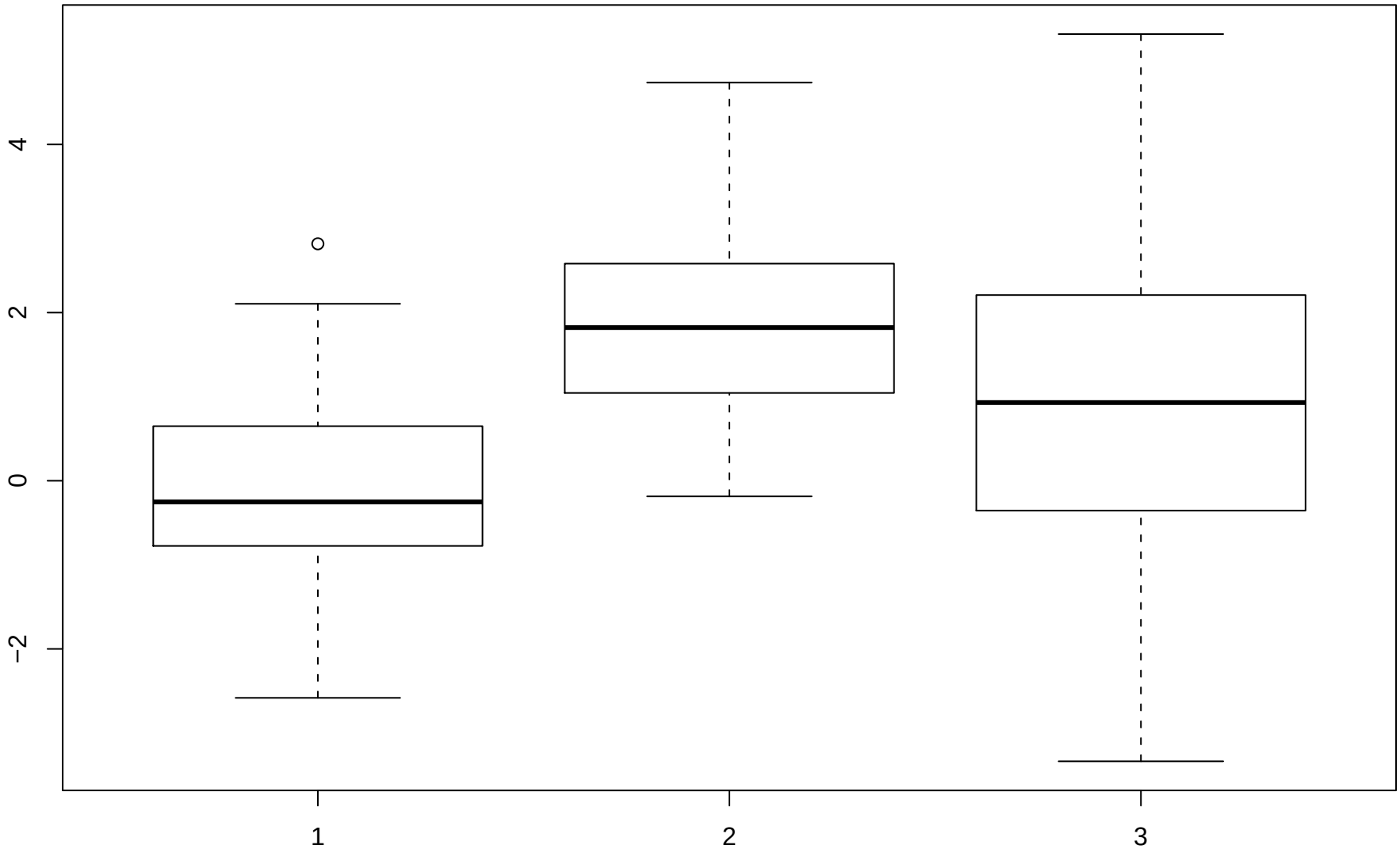
box plots are even more useful in comparing several samples.

The samples don't have to be of equal size.

Let us create three normal random samples and compare them via box plots

```
> set.seed(345)
> z1 <- rnorm(100)
> z2 <- rnorm(100,mean=2,sd=1)
> z3 <- rnorm(100,mean=1,sd=2)
> boxplot(z1,z2,z3)
```

Comparative Box Plots



The box plots clearly point out the differences w.r.t. the sampled populations.

Normal Probability Plots

The normal distribution is often used to model many experimental observations.

We will give an explanation for this later.

It is thus important to have a diagnostic normal model check.

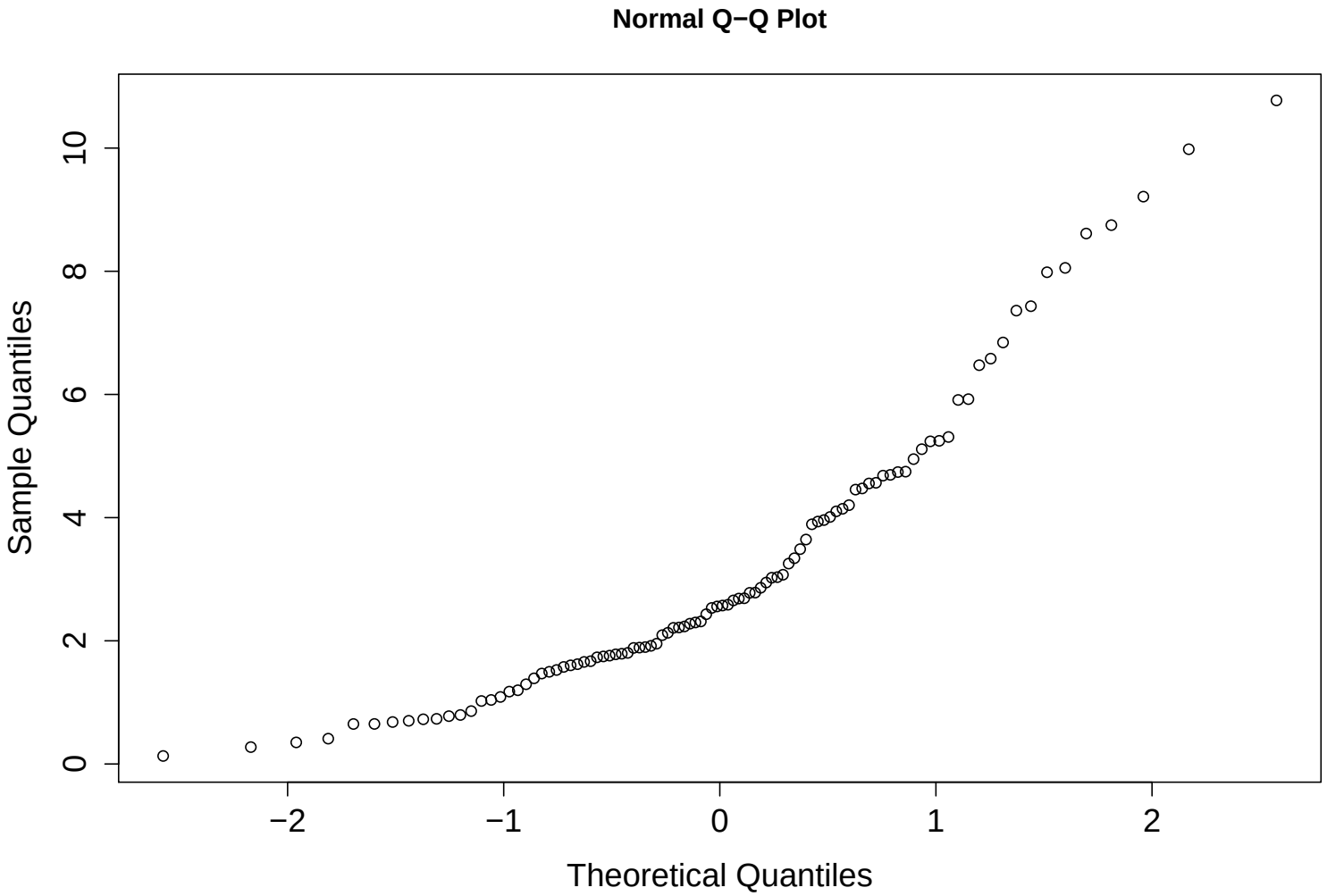
`qqnorm(x)`, applied to a sample vector `x`, gives an informal graphical diagnostic.

For normal samples we expect a roughly linear pattern.

To judge such resulting plots appropriately requires some experience.

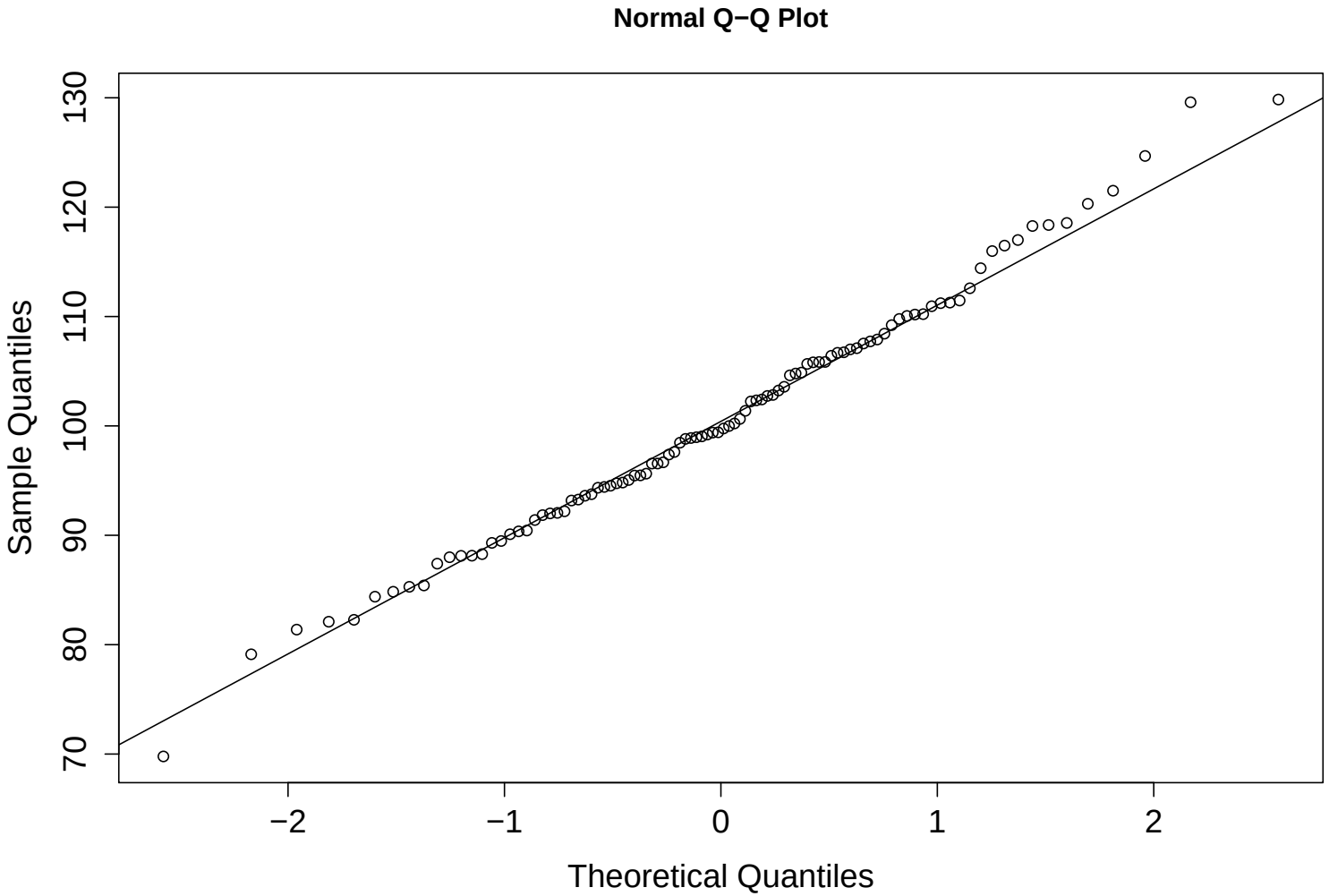
Such experience is very easy to get in R.

Normal Probability Plot for $\chi^2(3)$ Sample



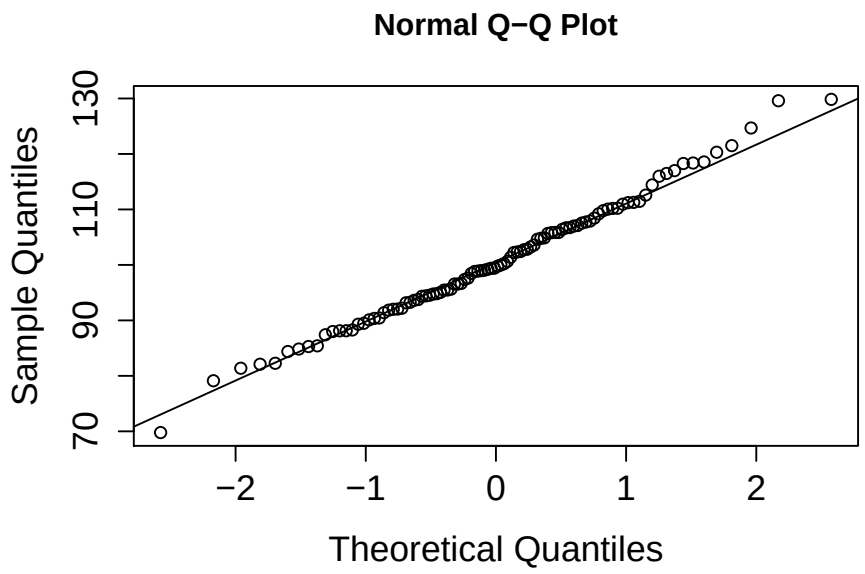
The point pattern shows a strong curved behavior (not normal!).

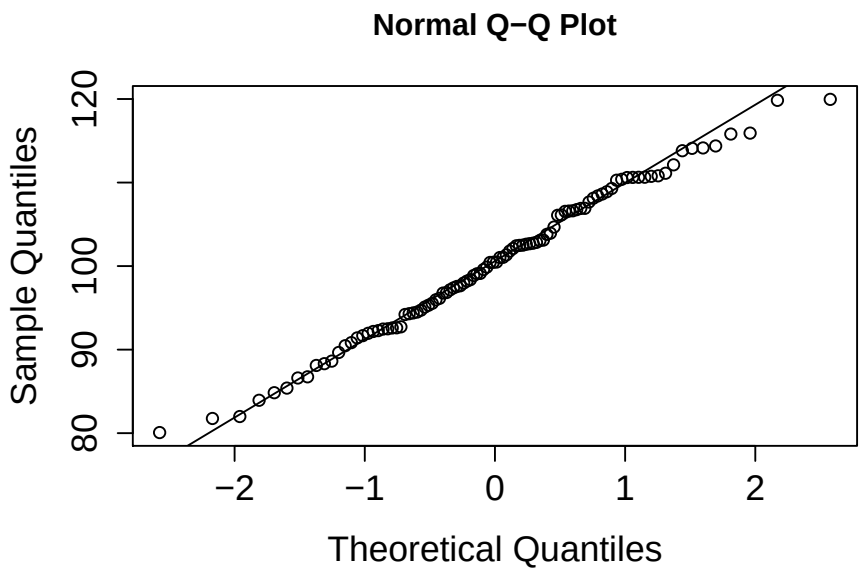
Normal Probability Plot for $\mathcal{N}(100, 10^2)$ Sample

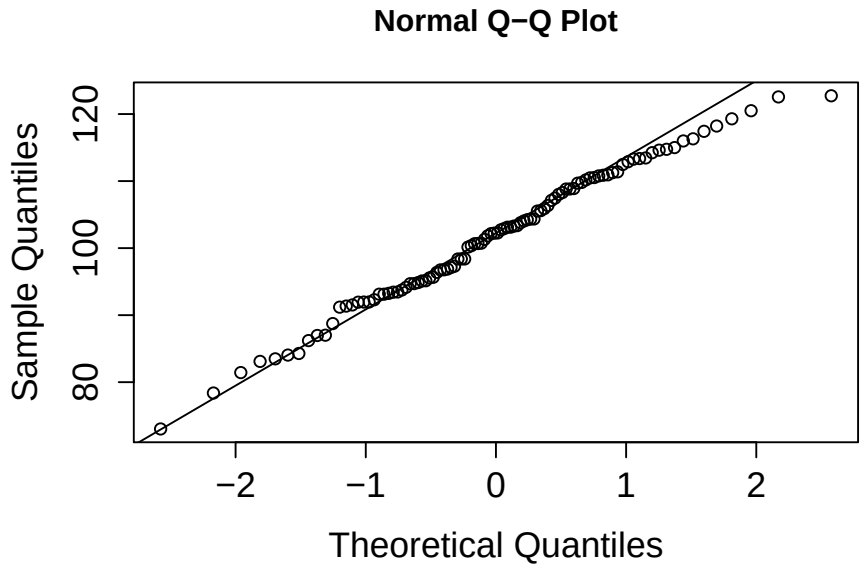


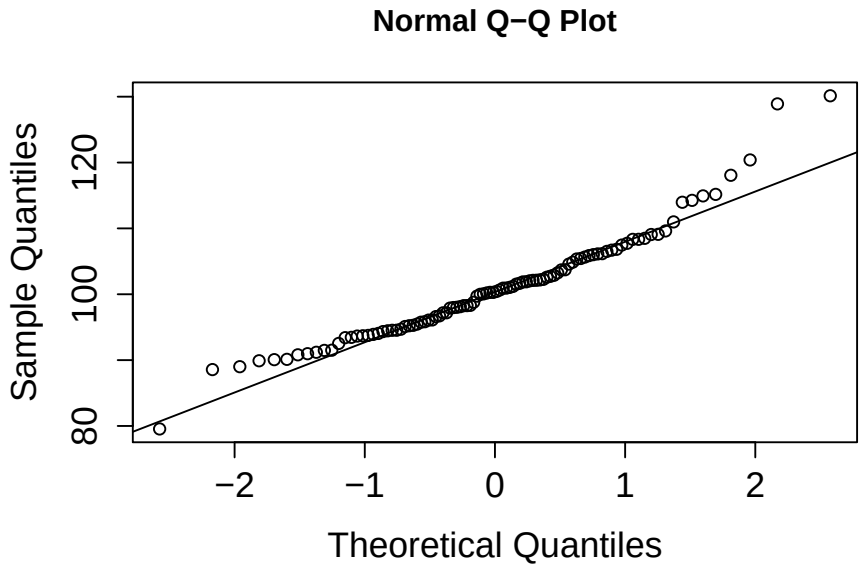
The point pattern looks fairly linear (i.e., normal!). The deviations from the line appear to be sampling variation. The line was added by `qqline(x)`.

Several Normal Probability Plots for $\mathcal{N}(100, 10^2)$ Samples









Code for Previous Slide

The previous slide should give some appreciation of what deviations from linearity to expect from truly normal samples of size $n = 100$.

```
par(mfrow=c(2,2)) # sets up a 2 by 2 set of plots on one page
set.seed(267) # not needed, but gives results as shown
for(i in 1:4){ # start of a loop for making 4 plots
   x <- rnorm(100,100,10) # generating a normal sample of size 100
   qqnorm(x,cex=1,cex.axis=1.4,cex.lab=1.4)
   # makes normal probability plot for x
   qqline(x) # adds a fitted line to the plot, to judge linearity
}
```

The label Normal Q-Q Plot on top of the plots stands for quantile-quantile plot.

But since each α-quantile is associated with a probability $\alpha \in (0, 1)$,

such plots are also called normal probability plots.

The Idea of the Normal Probability Plot

When discussing quantiles of $X \sim \mathcal{N}(\mu, \sigma^2)$ we pointed out their simple relationship to quantiles of $Z \sim \mathcal{N}(0,1)$, namely

$$q(X;\alpha) = \mu + \sigma\, q(Z;\alpha) \quad \text{for any } \alpha \in (0,1)$$

i.e., $q(X;\alpha)$ is a linear function of $q(Z;\alpha)$, with intercept μ and slope σ.

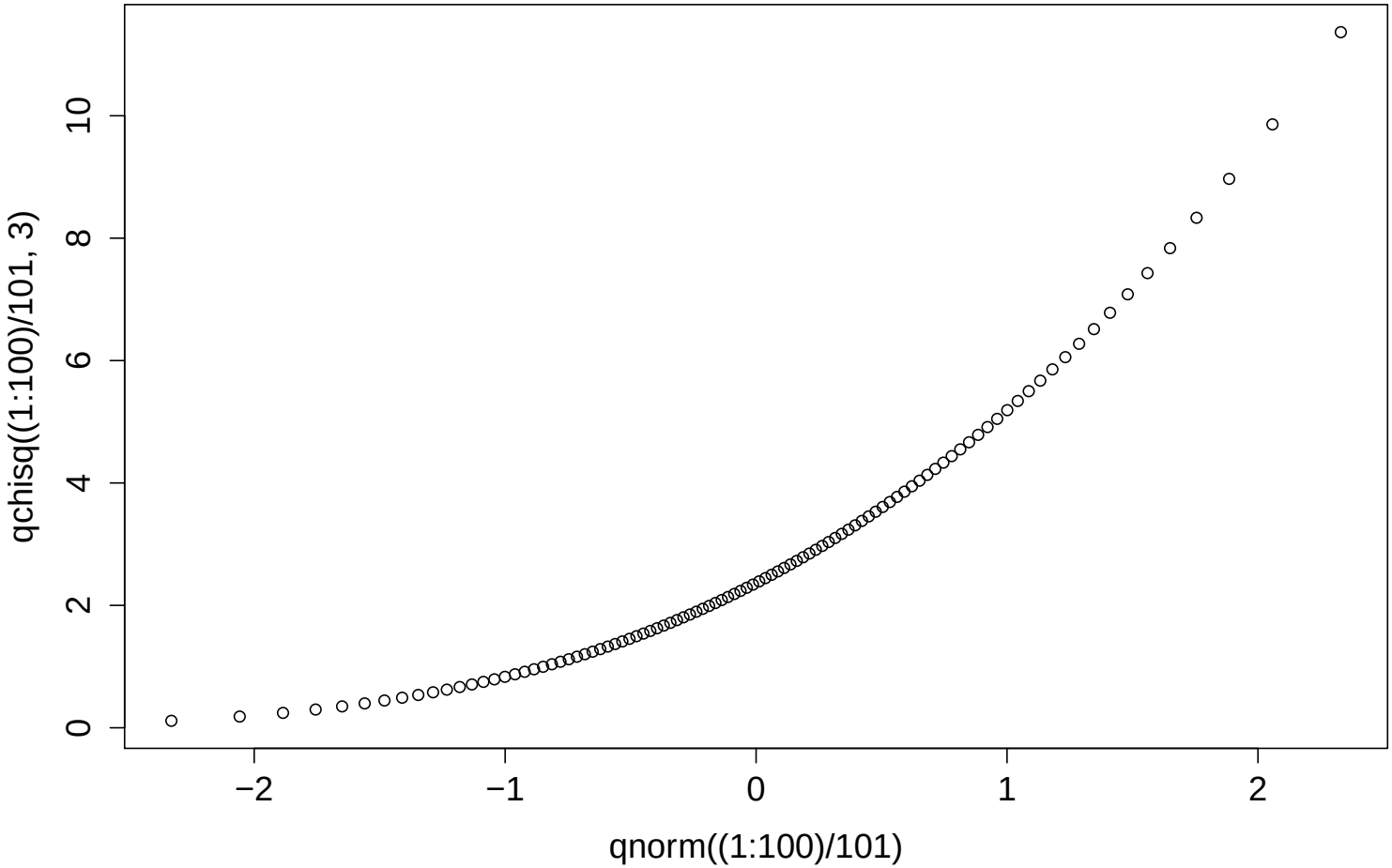
If $\tilde{q}(X;\alpha)$ is the α-quantile of any distribution, we can examine the normality of that distribution by plotting $\tilde{q}(X;\alpha)$ against $q(Z;\alpha)$ for a lot of α values.

We could check it for the n equally spaced α values $\alpha_i = i/(n+1), i = 1,2,\ldots,n$.

The next slide does this in a comparison of $\chi^2(3)$ quantiles with the standard normal quantiles $q(Z;\alpha)$, i.e., we issue the command

`plot(qnorm((1:100)/101),qchisq((1:100)/101,3))` and get

Normal Probability Plot for $\chi^2(3)$ Distribution



The point pattern does not look linear, i.e., $\chi^2(3) \neq \mathcal{N}$.

In fact, it looks similar to the normal probability plot for our $\chi^2(3)$ sample.

The Idea of the Normal Probability Plot

The previous slide suggests how to examine samples for normality.

Rather than using quantiles of the unknown sampled population

we use the quantiles for the ecdf of the sample of size n.

Which α-quantiles should we use?

Popular choices, both leading to the ordered sample values $x_{(1)} < x_{(2)} < \ldots < x_{(n)}$ as the sample α_i-quantiles, $i = 1, \ldots, n$, are

$$\alpha_i = \frac{i}{n+1}, \quad i = 1, \ldots, n \qquad \text{or} \qquad \alpha_i = \frac{i-.5}{n}, \quad i = 1, \ldots, n$$

since in both cases

$$\hat{P}_n(X < x_{(i)}) = \frac{i-1}{n} \leq \alpha_i \qquad \text{and} \qquad \hat{P}_n(X \leq x_{(i)}) = \frac{i}{n} \geq \alpha_i \quad \text{for } i = 1, \ldots, n$$

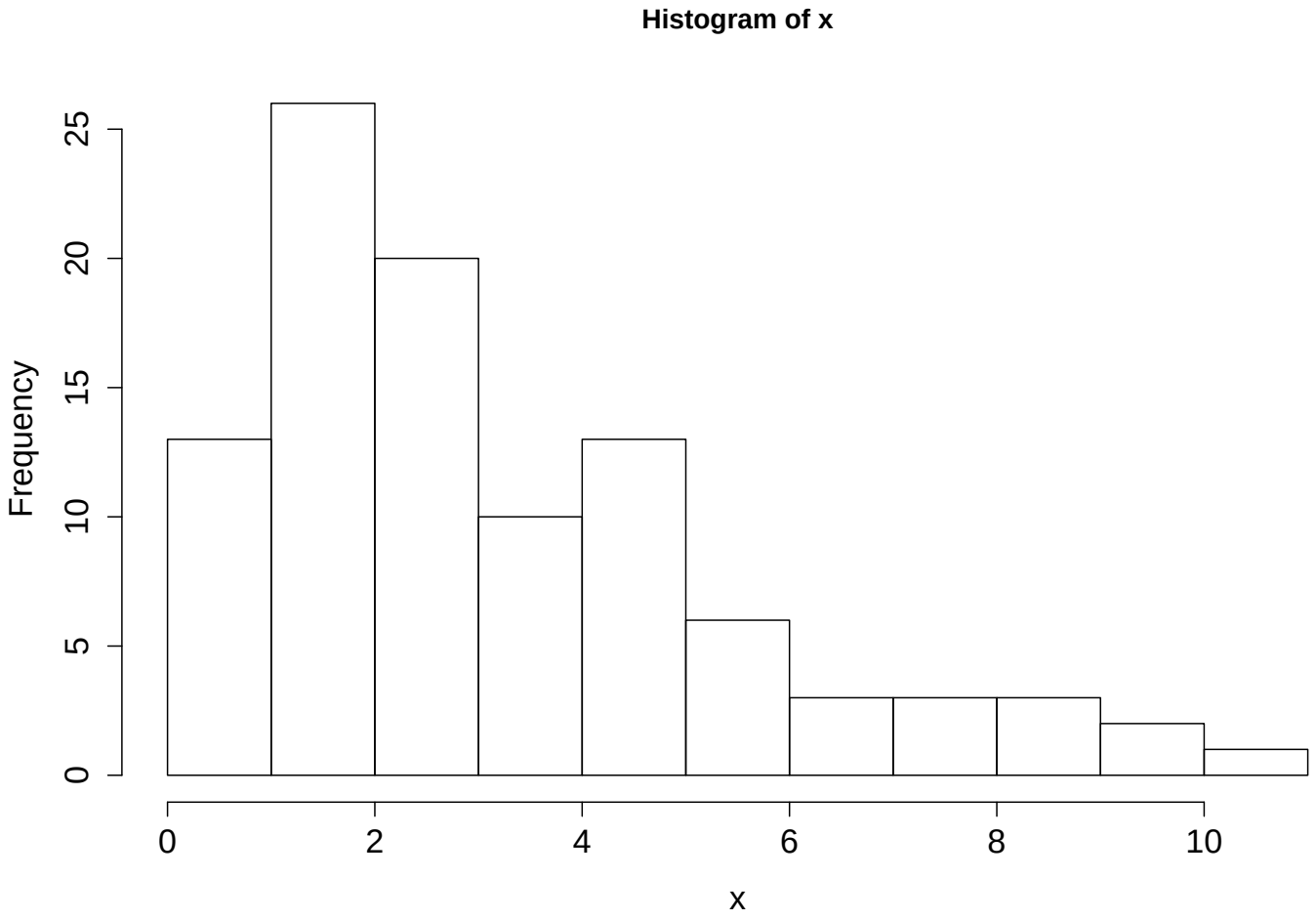
verifying $x_{(1)} < x_{(2)} < \ldots < x_{(n)}$ as the α_i-quantiles.

Thus plot these $x_{(i)}$ against $q(Z; \alpha_i) = \texttt{qnorm(i/(n+1))}$ or $= \texttt{qnorm((i-.5)/n)}$, for $i = 1, \ldots, n$. $\quad \Longrightarrow$ `qqnorm(x)`.

Histograms

Another popular method for examining sample features is the histogram.

It is obtained for our previous $\chi^2(3)$ sample by `hist(x)`.



Comments on Histograms

The previous slide showed a histogram where the abscissa was divided into equal size adjacent intervals (also called buckets).

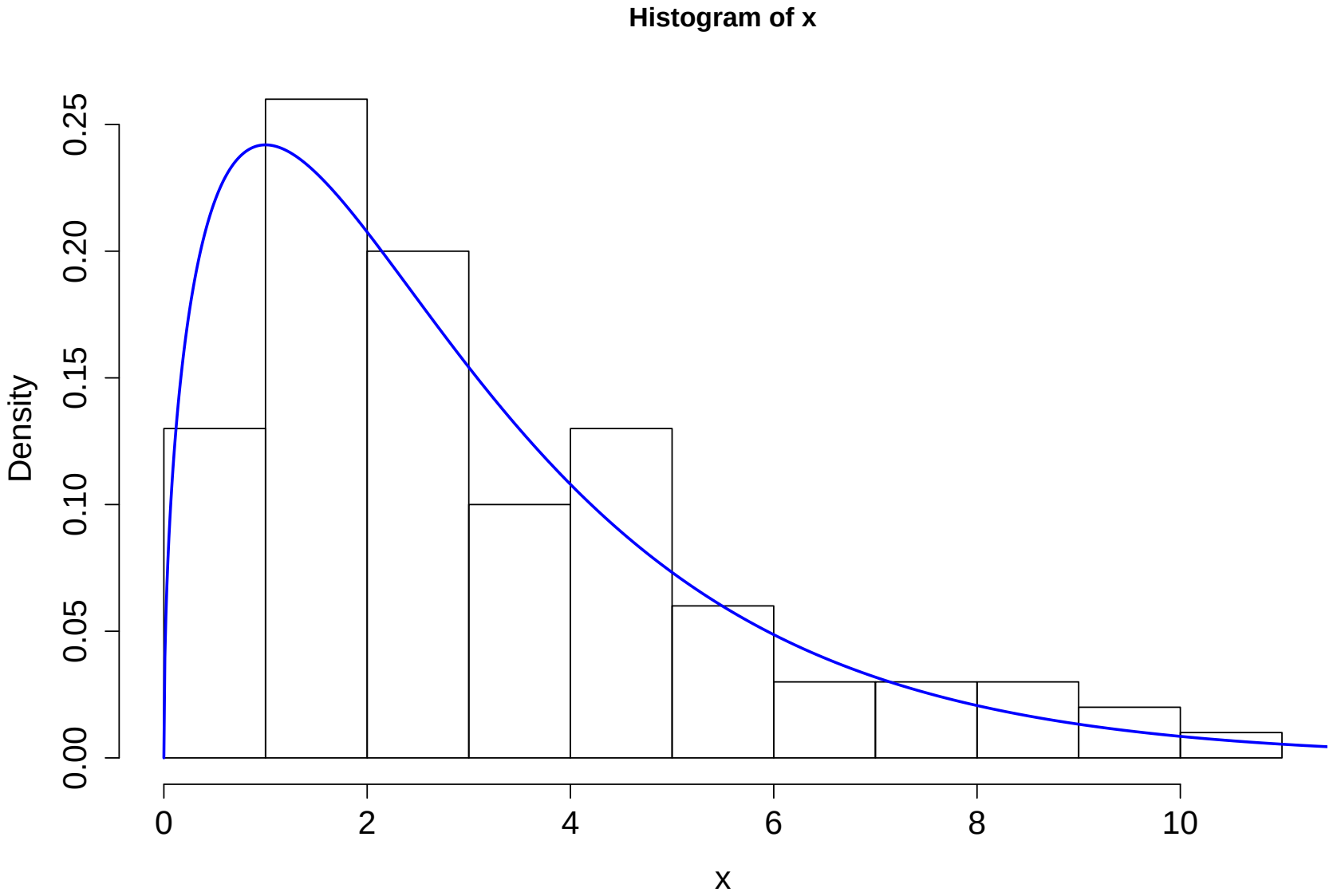
The height of the bar over each interval represents the frequency count of sample values within that interval.

Using `hist(x,probability=T)` makes the bar area proportional to the count, with total areas adding to one. Next slide shows superimposed sampled population.

The number and positioning of the intervals can change the histogram appearance.

See the documentation on `hist` for all its capabilities.

Histogram & Population Comparison



Code for Previous Slide

```
set.seed(267) # not necessary, but makes the plot exactly as shown
x <- rchisq(100,3) # generates sample of size n=100
                   # from chi-square(3) distribution
hist(x,prob=T,cex=1.4,cex.axis=1.4,cex.lab=1.4) # plots histogram
z <- seq(0,13,.01) # vector from 0 to 13 in increments of .01
fz <- dchisq(z,3) # vector of corresponding density values
lines(z,fz,col="blue",lwd=2) # adds a curve with (z,fz) coordinates
```

Note that above we used `prob=T` instead of `probability=T`.

Functions in R typically interpret shortened input arguments correctly when no ambiguity can arise.

Kernel Density Estimates

For samples from a continuous distribution with density $f(x)$,

is it possible to get an estimate $\hat{f}_n(x)$?

This could serve as a counterpart to the ecdf, which places probability $1/n$ at each of the sample values $x_i, i = 1, \ldots, n$?

After all, $\hat{F}_n(x)$ is a discrete distribution and does not capture the fact that a continuous distribution puts zero probability on any countable set of points.

Kernel density estimates replace the point probability $1/n$ at x_i with a density centered at x_i multiplied by $1/n$, i.e.,

$$\hat{f}_n(x) = \sum_{i=1}^{n} \frac{1}{n} \frac{1}{h} K\left(\frac{x - x_i}{h}\right)$$ where $\frac{1}{h} K\left(\frac{x - x_i}{h}\right)$ is a density, with area 1 under it

$\hat{f}_n(x) \geq 0$ & it has total area 1 under it, since each summand contributes area $1/n$.

Given K and h and a sample $\vec{x} = \{x_1, \ldots, x_n\}$, it is very easy to calculate $\hat{f}_n(x)$.

Choices for the Kernel $K()$

Typically one chooses as K a density symmetric around the origin,

for example, the density of $\mathcal{N}(0,1)$ or Uniform$(-0.5, 0.5)$.

The quantity $h > 0$ in

$$\frac{1}{h}K\left(\frac{x-x_i}{h}\right) \quad \text{e.g.} \quad = \frac{1}{h\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left[\frac{x-x_i}{h}\right]^2\right)$$

provides a scaling of the density, expressing spread.

Small h means the density is highly concentrated near $x - x_i = 0$, i.e., near $x = x_i$.

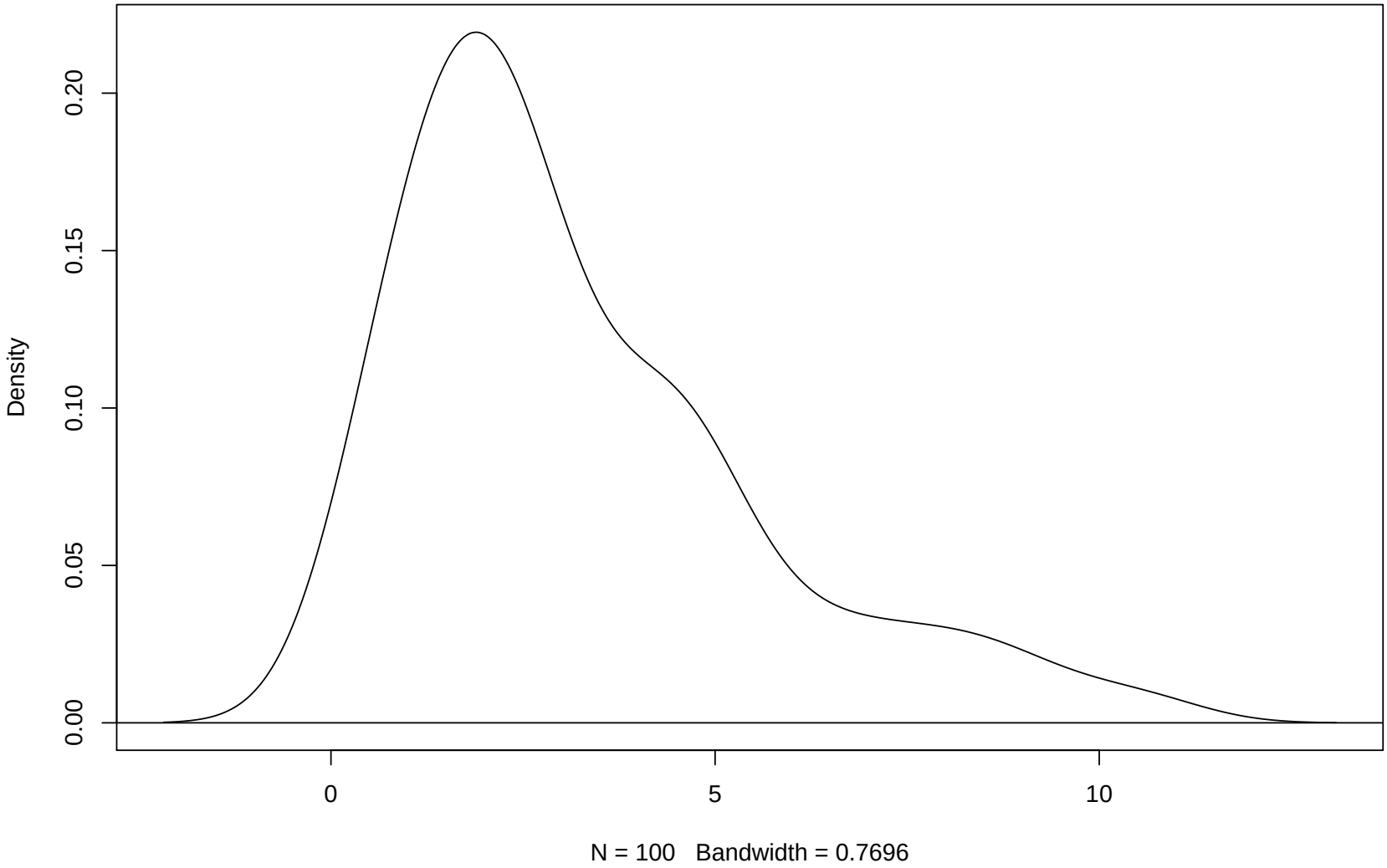
Large values of h indicate that the density is widely spread but still centered on x_i.

How to choose the correct h depends very much on the sample size.

Large n allow small h (more detail) and small n require larger h (broader brush).

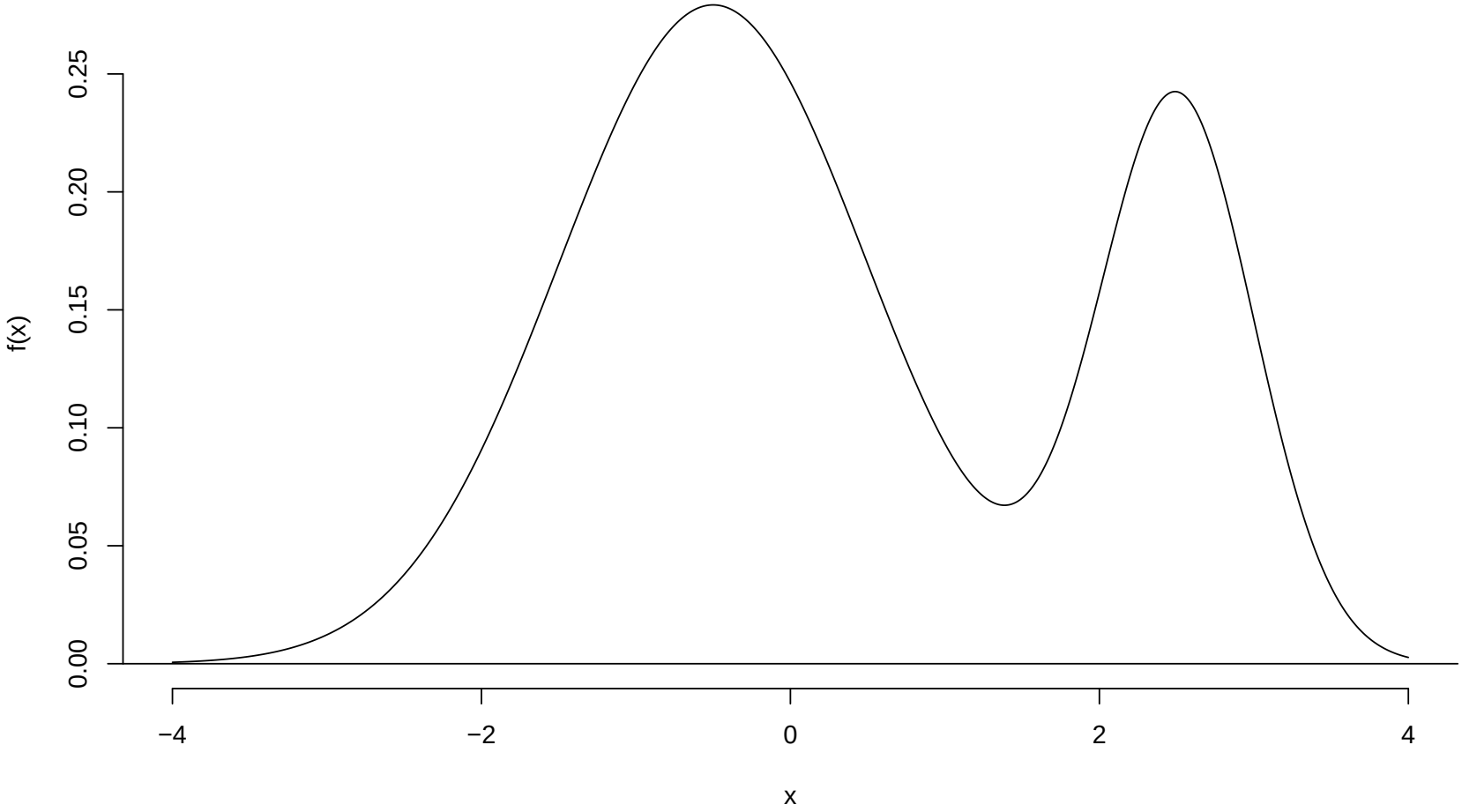
The choice is not trivial and we leave that to the default method used by R.

For Our Previous $\chi^2(3)$-Sample of $n = 100$



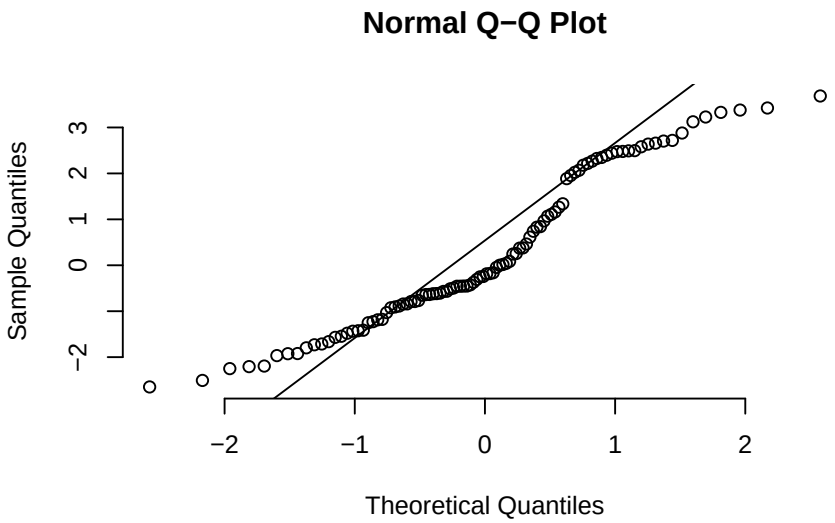
Using: `plot(density(x),main=""); abline(h=0)`

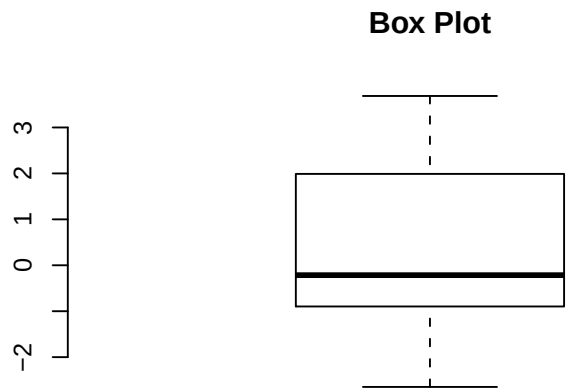
Bimodal Distribution

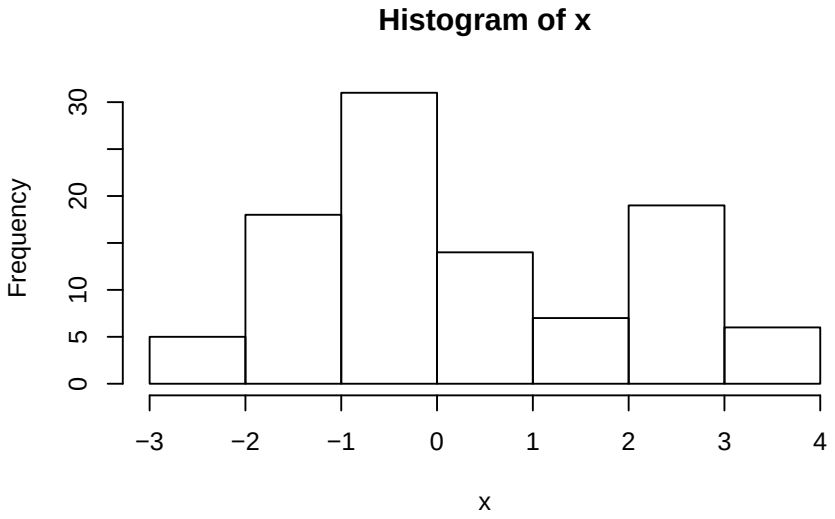


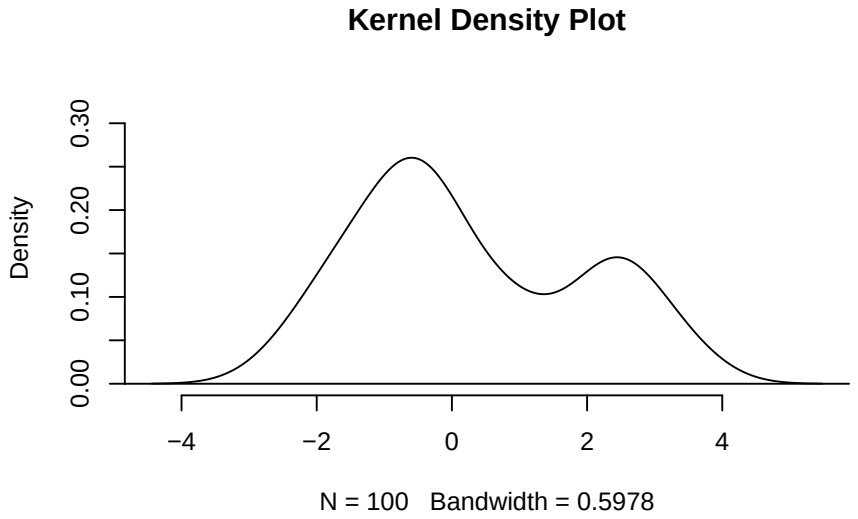
Let us see what happens when sampling $n = 100$ from the above population.

Sample ($n = 100$) from Bimodal Distribution









Comments on Previous Plots

The normal probability plot suggests a nonnormal sampled population (correctly).

`qqline(x)` fits a line to the middle portion (50%) of the sample.

The box plot suggests some skew toward higher values (misses bimodality)

The histogram suggests a bimodal sampled population.

The kernel density estimate also suggests a bimodal sampled population.

Note the vertical gap in the normal probability plot.

It indicates sparseness of data there, i.e., a data density dip and thus bimodality.

To recognize this takes some experience or reasoning.

Bimodality is more obvious in histogram and kernel density plot.

Case Study: Lengths of Forearms

This data set is a typical example of what occupied biometric statistics around 1900.

It was studied by K. Pearson and A. Lee (1903): On the laws of inheritance of man.

I. Inheritance of physical characters. *Biometrika*, 2: 357-462.

Download the data set `forearms.dat` from Trosset's web site:

`http://mypage.iu.edu/~mtrosset/StatInfeR.html`

and place it into the directory `C:\Users\yourname\Stat311\DATA`,

provided this directory path exists in Windows w.r.t. your class work.

Suppose you launched your R session from `C:\Users\yourname\Stat311\R-Ch7`

`> forearms <- scan("../DATA/forearms.dat")` gets data $\longrightarrow$ session.

In Linux the import is the same but your directory setup may be

`/home/yourname/Stat311/DATA` and `/home/yourname/Stat311/R-Ch7`

The `..` in `../DATA` refers to the parent directory `C:\Users\yourname\Stat311`
of the launch directory `C:\Users\yourname\Stat311\R-Ch7` of your R session.
`"../DATA/forearms.dat"` = `"C:/Users/yourname/Stat311/DATA/forearms.dat"`

Continuous Distribution?

```
> table(forearms)
forearms
16.1 16.3 16.4 16.6 16.8 16.9 17.1 17.2 17.3 17.4 17.5 17.6
   1    2    1    1    1    1    4    2    4    2    3    1
17.7 17.8 17.9   18 18.1 18.2 18.3 18.4 18.5 18.6 18.7 18.8
   3    1    3    3    4    2    8    4    9    4    4    5
18.9   19 19.1 19.2 19.3 19.4 19.5 19.6 19.7 19.8 19.9   20
   3    6    5    3    4    5    4    5    3    3    4    3
20.1 20.2 20.3 20.4 20.5 20.6 20.7 20.8 20.9   21 21.4
   2    1    1    3    4    2    1    1    2    1    1
```

We see many observations multiple times (rounding to one decimal).

For samples from continuous distributions we have $P(X_i = X_j) = 0$ for $i \neq j$.

Length measurements should be continuous, not inherently discrete.

We will treat them as continuous and ignore the ties.

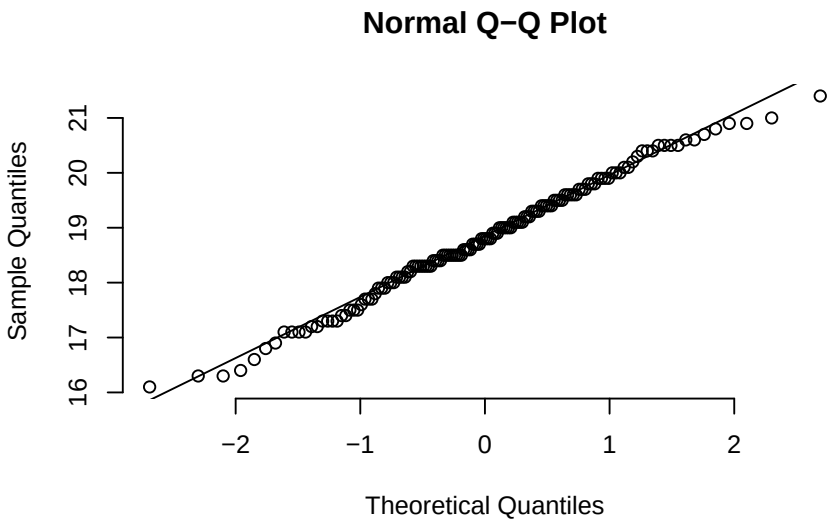
Normality of Forearms?

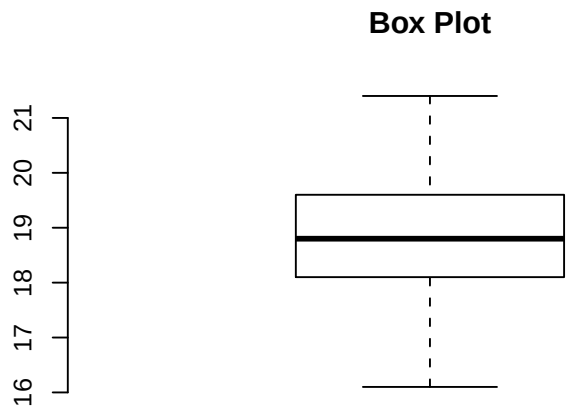
A basic question is whether the data are normally distributed.

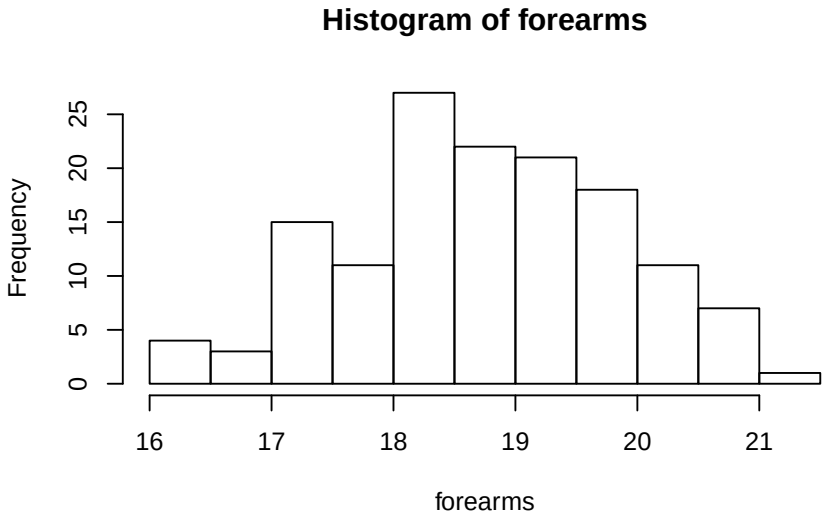
Normality allows simpler analyses and more effectice use of the data.

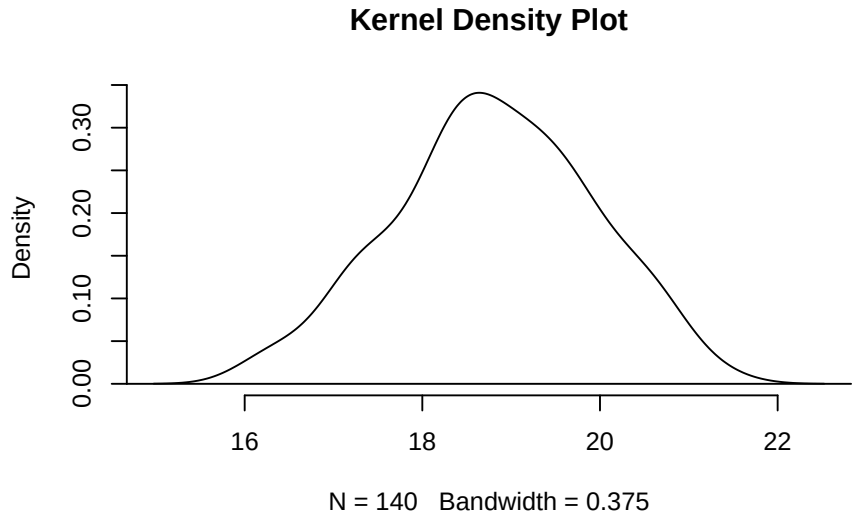
```
par(mfrow=c(2,2)) # sets up 2 by 2 plot page
qqnorm(forearms,axes=F); qqline(forearms)
axis(1); axis(2)
  # adds abscissa and ordinate, since we suppressed axes/box
boxplot(forearms,axes=F,main="Box Plot")
axis(2)
hist(forearms)
plot(density(forearms),main="Kernel Density Plot",axes=F)
axis(1); axis(2)
abline(h=0) # adds horizontal baseline to plot.
```

Lengths of Forearms









Discussion of Plots

Both histogram and kernel density plot indicate a mound shaped distribution.

The box plot indicates good symmetry of quartiles and whiskers w.r.t. median.

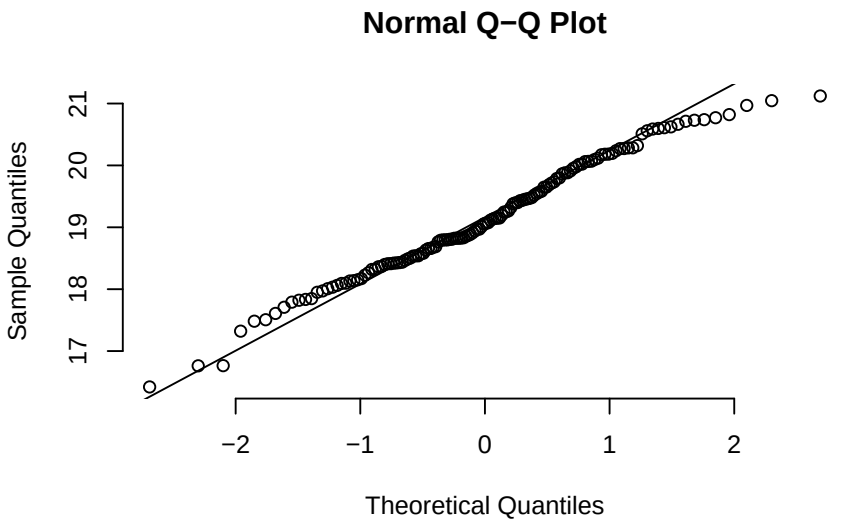
The normal probability plot seems like the best indication of normality.

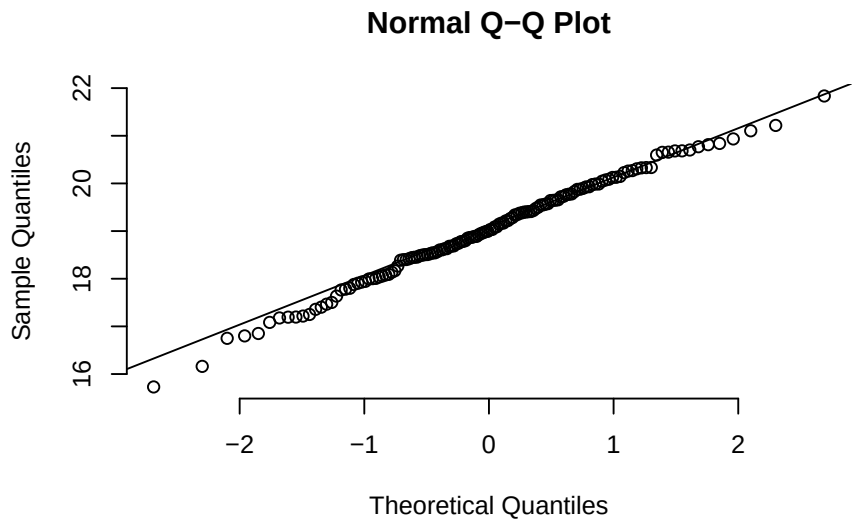
Should we worry about the wiggles at the high and low end?

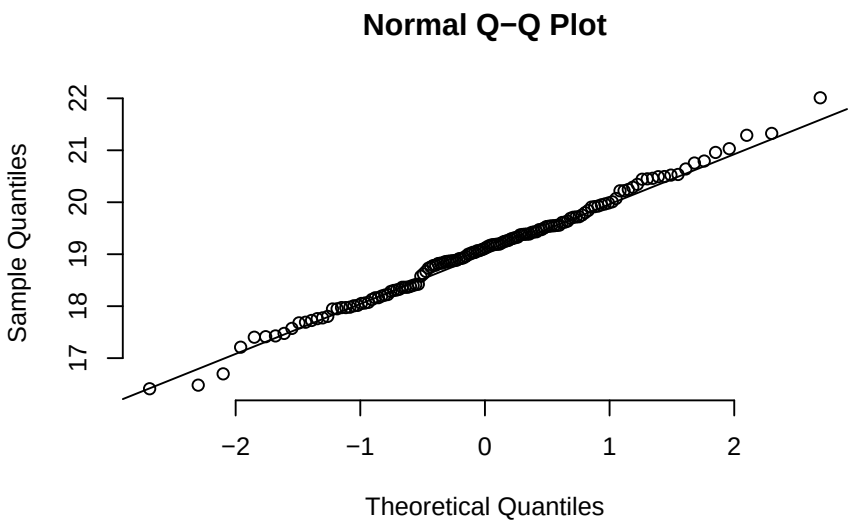
Create comparison normal probability plots for samples of size $n = 140$ from a known normal population, say $\mathcal{N}(19, 1)$.

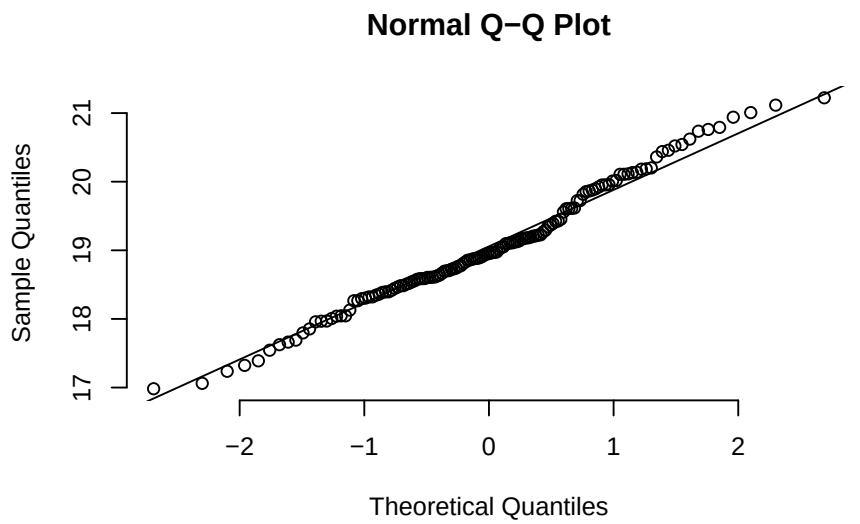
```
par(mfrow=c(2,2))
for(i in 1:4){
  x <- rnorm(140,19,1)
  qqnorm(x,axes=F); qqline(x)
  axis(1); axis(2)
}
```

Normal Comparison Plots

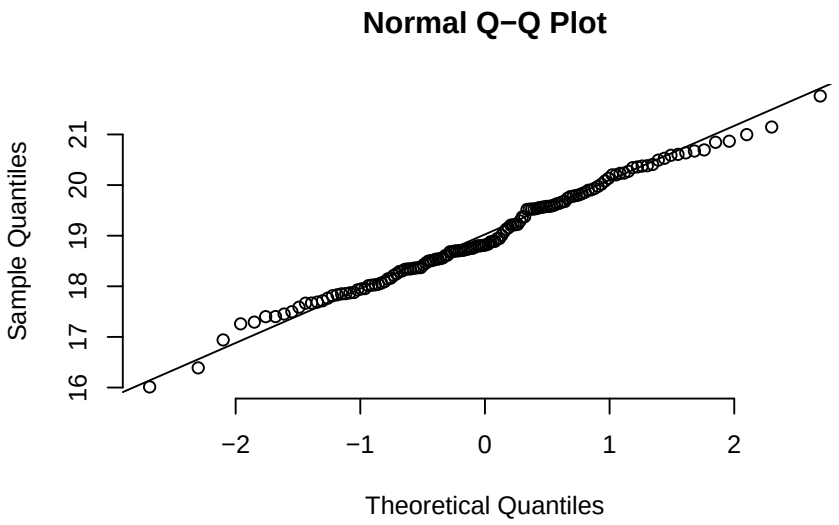


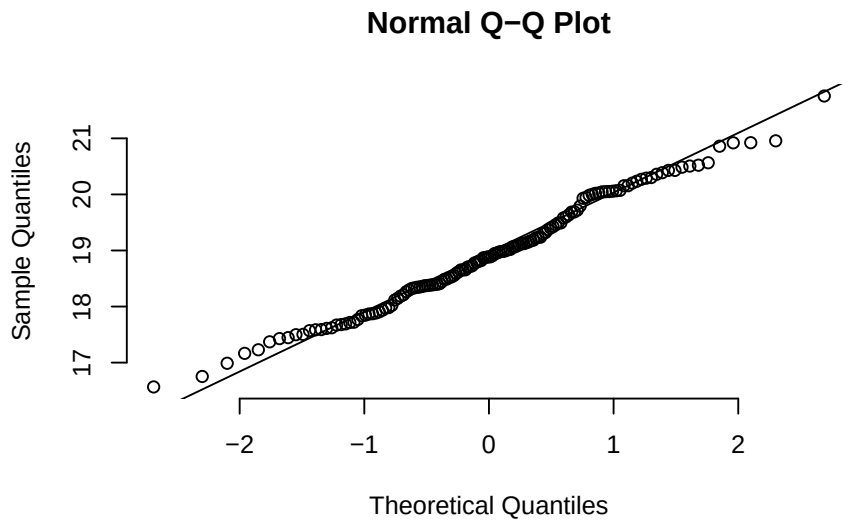


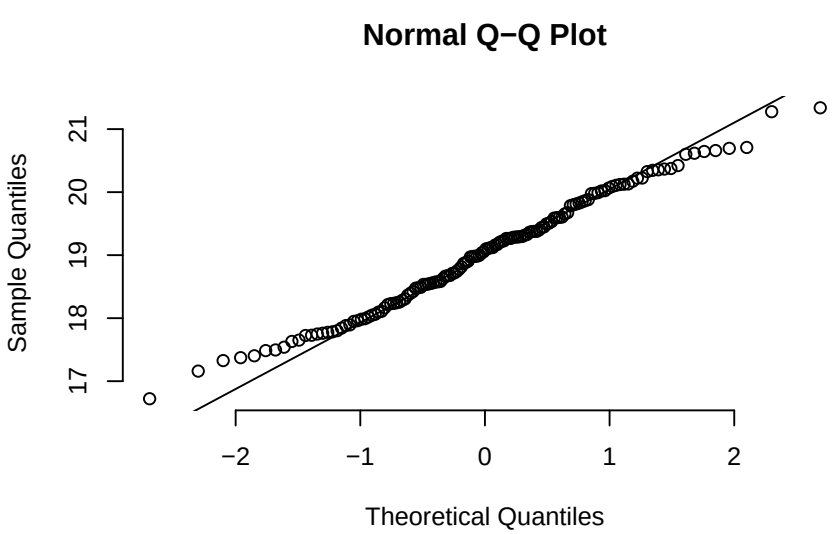


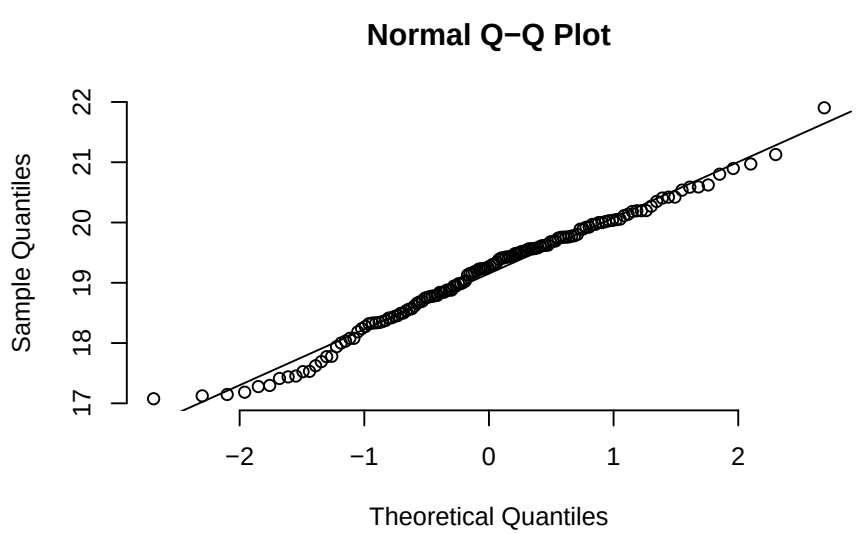


Normal Comparison Plots

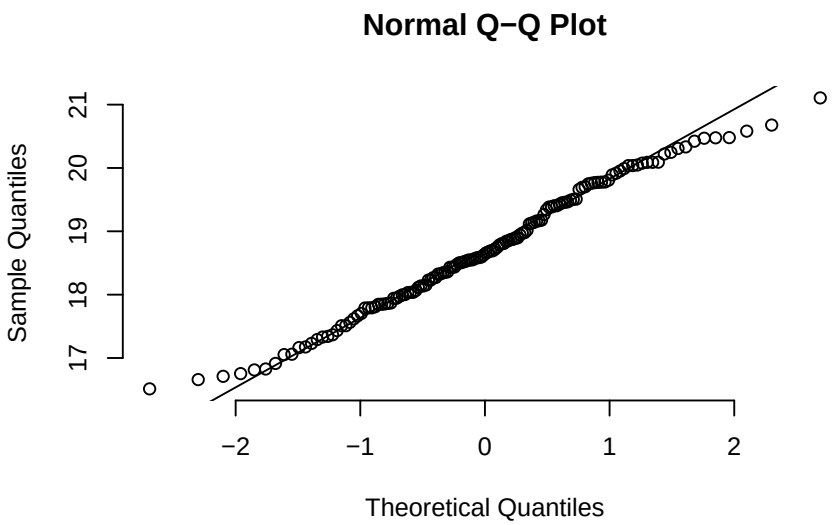


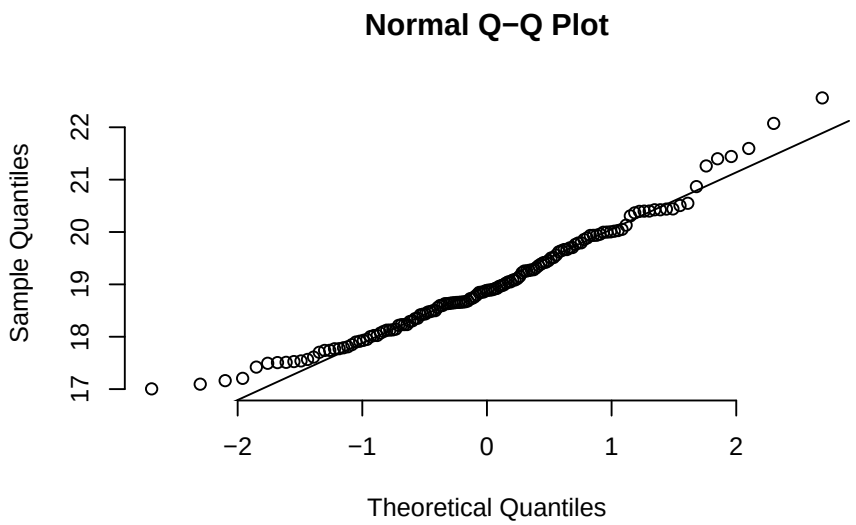


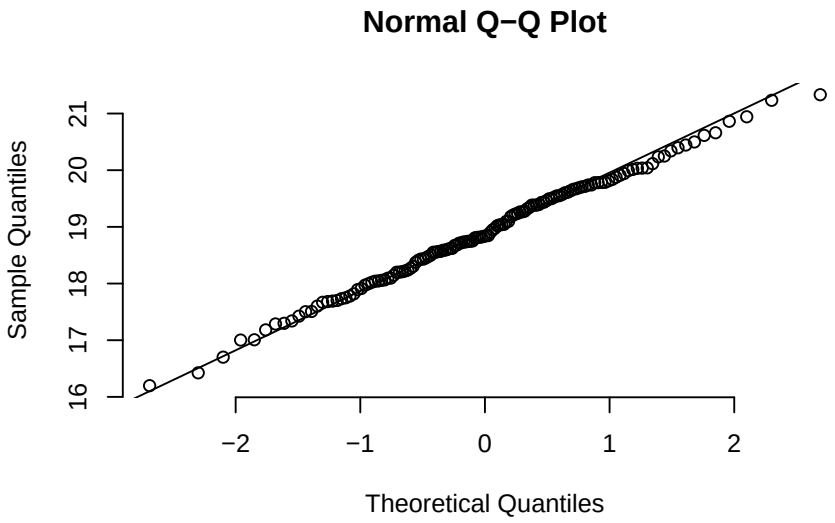


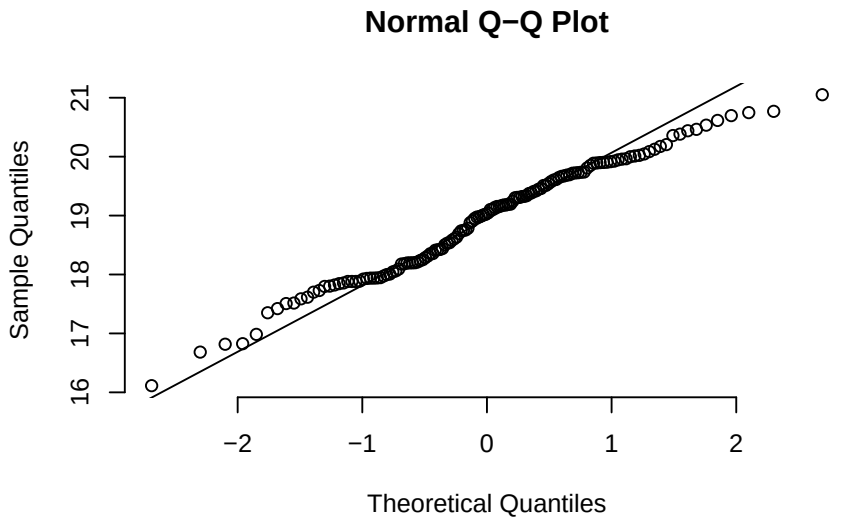


Normal Comparison Plots



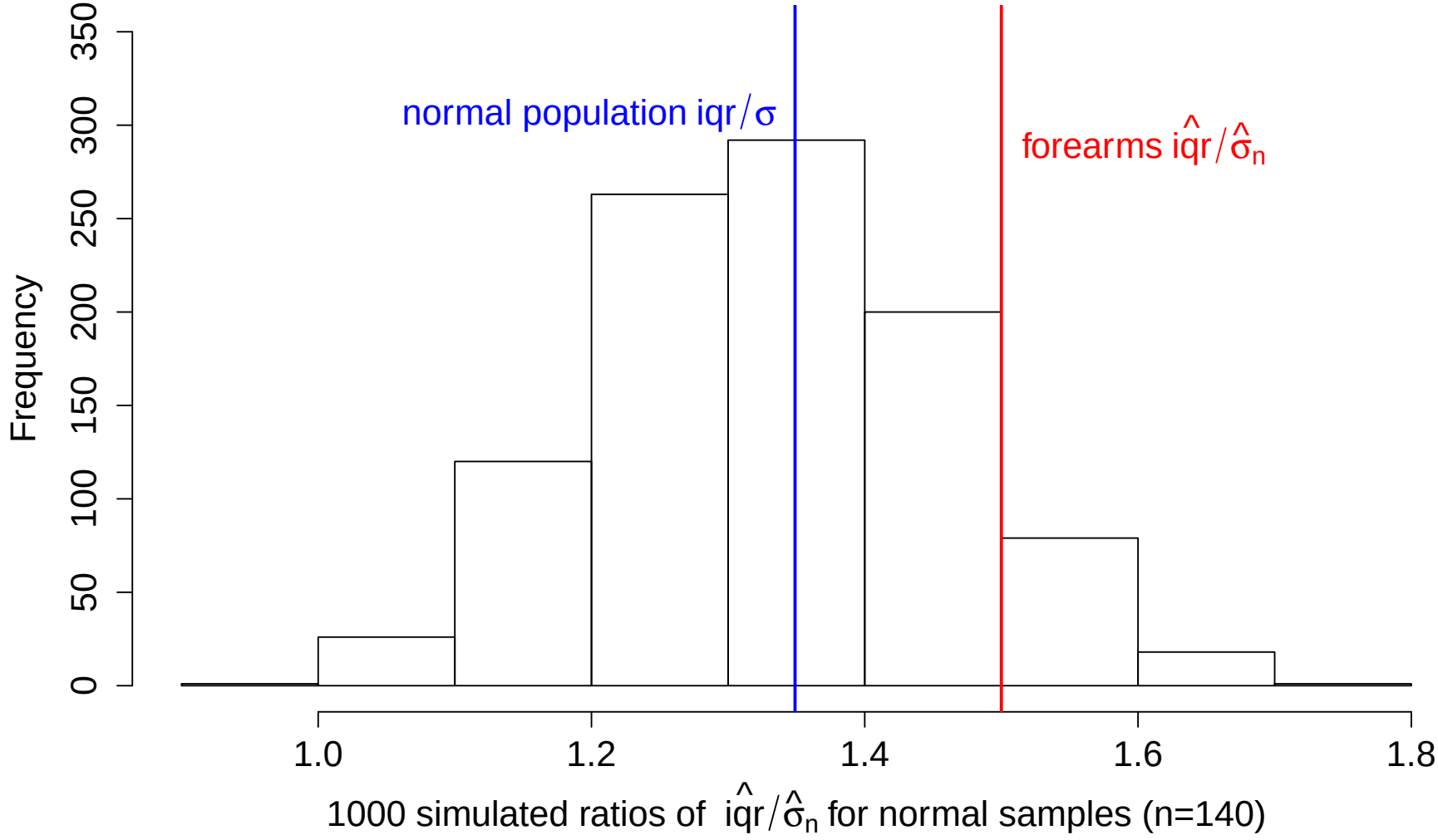






By comparison our forearm normal probability plot looks very good!

$\widehat{\text{iqr}}/\widehat{\sigma}_n$ for Forearms in Normal Sampling Perspective



Comments on $\widehat{\text{iqr}}/\widehat{\sigma}_n$ Perspective

For normal populations we saw that $\text{iqr}/\sigma = 1.34898 \approx 1.35$.

The corresponding sample estimate $\widehat{\text{iqr}}/\widehat{\sigma}_n$ will differ from 1.35 due to sampling variation, even when the samples come from a normal population.

There are other populations for which we have $\text{iqr}/\sigma = 1.35$.

Thus an observed $\widehat{\text{iqr}}/\widehat{\sigma}_n$ reasonably close to 1.35 does not prove normality, but it also does not disprove it.

If the observed $\widehat{\text{iqr}}/\widehat{\sigma}_n$ is sufficiently far from 1.35 compared to normal sampling variation, then we can make a cases against normality.

For our forearms data there is no case against normality based on $\widehat{\text{iqr}}/\widehat{\sigma}_n$.

Transformations

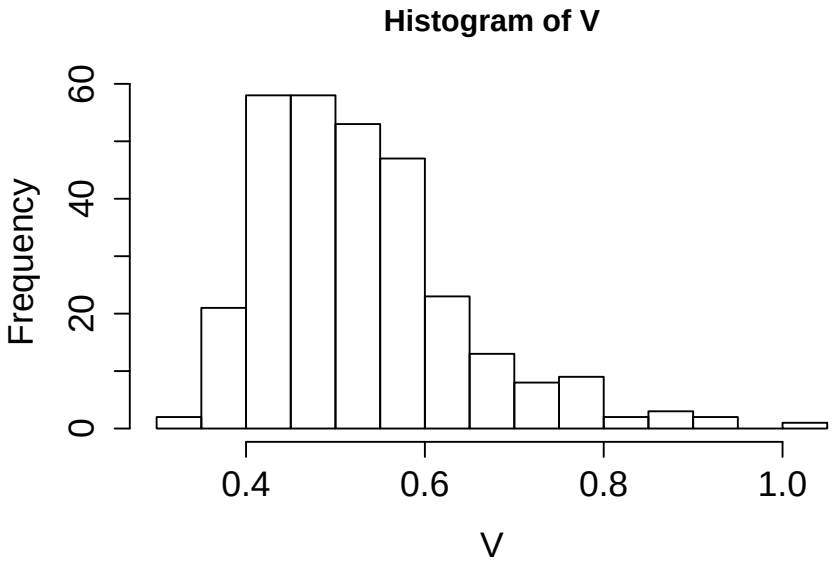
Whether experimental measurements are normally distributed or not often depends on what is being measured.

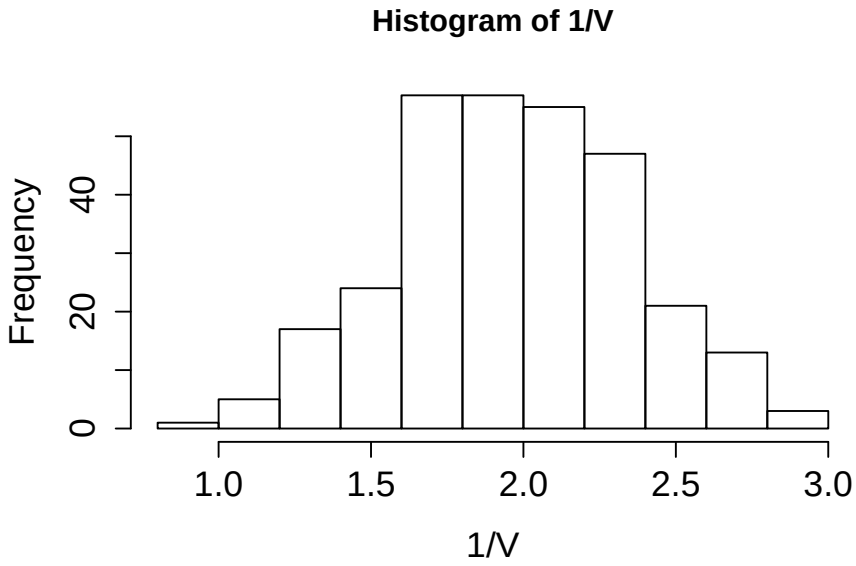
For example, we may measure with what speed a car makes it from A to B. This depends strongly on traffic conditions, driven by random effects. If the distance is D, the time is T, then velocity $V = D/T$. Clearly D is more or less constant, but T can vary quite a bit depending on traffic lights, conditions, etc.

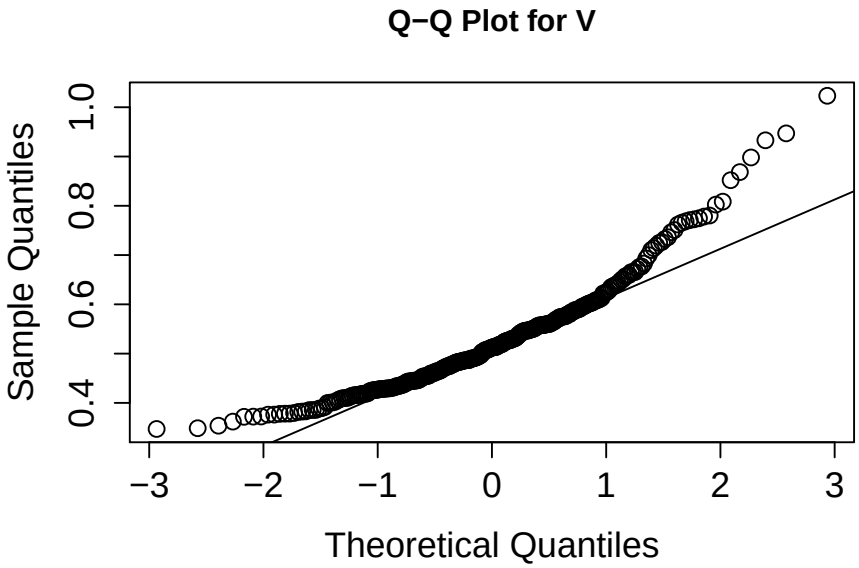
We will see later that a random variable, that can be viewed as a sum of many independent random components, $T = T_1 + \ldots + T_k$, can often be viewed as approximately normally distributed, i.e., $T \approx \mathcal{N}(\mu, \sigma^2)$

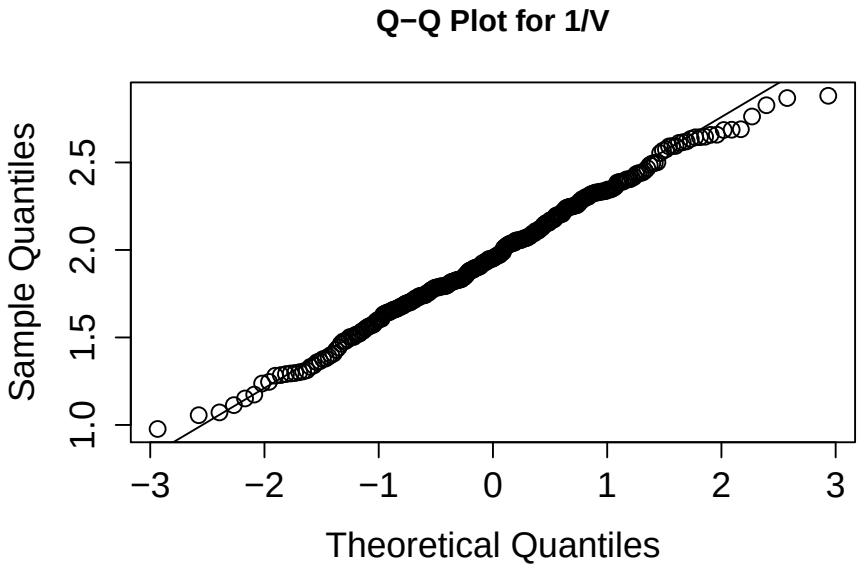
But, as a consequence $V = D/T \not\approx \mathcal{N}(\nu, \tau^2)$ (reciprocal transformation)

Normality of Speed?









Choice of Scale?

A psychologist tests a drug for cognitive function in Alzheimer's patients.

$n = 5$ patients are tested before and after the drug study period.

A_i is the test score after and B_i is the prior baseline score, $i = 1, \ldots, 5$.

How should one compare A_i and B_i? A_i/B_i or $A_i - B_i$?

Should $b_1 = 2$ and $a_1 = 4$ be considered a doubling of cognitive function,

equivalent to $b_2 = 20$ and $a_2 = 40$?

Or should we interpret the 2 point increase from $b_1 = 2$ to $a_1 = 4$

as equivalent to $b_3 = 20$ and $a_3 = 22$?

Fictitious Scores

i	B_i	A_i	$X_i = A_i/B_i$
1	16	32	2.00
2	16	24	1.50
3	16	16	1.00
4	16	12	0.75
5	16	8	0.50

$X_i = 1$ means no effect, $X_i > 1$ means positive effect, $X_i < 1$ means adverse effect.

The average the above X_i scores is `mean(c(2,1.5,1,.75,.5)) =1.15`,

i.e., we have an average positive effect.

If instead we took $Y_i = B_i/A_i = 1/X_i$ as our score, then $Y_i > 1$ is an adverse effect.

The average Y_i score is `mean(1/c(2,1.5,1,.75,.5)) =1.1` (typo in text)

i.e., we have an average adverse effect, using the same data. What is wrong?

The Problem Source and Resolution

Adverse scores correspond to $X_i \in (0,1)$, the positive scores to $X_i \in (1,\infty)$.

The neutral score is $X_i = 1$. The scales are asymmetric around 1.

Also, $(0,1)$ is bounded and $(1,\infty)$ is unbounded.

Viewing these scores on a $\log$-scale

$\log(X_i)$ and $\log(1/X_i) = -\log(X_i)$ are now symmetric around $\log(1) = 0$

```
> mean(log(c(2,1.5,1,.75,.5)))
[1] 0.02355661
> mean(log(1/c(2,1.5,1,.75,.5)))
[1] -0.02355661
```

The $\log$-transformed scores now convey the same message.

Other Transformations

Transformations can be usefully applied when dealing with skewed distributions.

Some popular candidates are: $y = \sqrt{x}, \quad y = \log(x), \quad y = \sqrt[3]{x}, \quad y = 1/x$, etc.

Finding the right transformation is still somewhat of an art form.

Sometimes the correct transformation is simply the inverse (or undoing)
of a hidden transformation that crept into the data reporting process.

The $\log$ transform not only equalized $(0,1)$ and $(1,\infty)$ scales, but also is useful for random variables arising by a multiplicative process, i.e.,

$$X = Y_1 \cdot Y_2 \cdot \ldots \cdot Y_k \quad \Longrightarrow \quad \log(X) = \log(Y_1) + \ldots + \log(Y_k)$$

More on the benefit of that later.

Be mindful of how to report results of a transformed data analysis and also the why.