# 1

# Learning from data, and tools for the task

*Chapter summary*

We begin by illustrating the interplay between questions driven by scientific curiosity and the use of data in seeking the answers to such questions. Graphs provide a useful window through which meaning can be extracted from data. Numeric summary statistics and probability distributions provide a form of quantitative scaffolding for models of random as well as nonrandom variation. Simple regression models foreshadow the issues that arise in the more complex models considered later in the book. Frequentist and Bayesian approaches to statistical inference are touched upon, the latter primarily using the Bayes Factor as a summary statistic which moves beyond the limited perspective that p-values offer. Resampling methods, where the one available dataset is used to provide an empirical substitute for a theoretical distribution, are also introduced. Remaining topics are of a more general nature. Section 1.9 will discuss the use of RStudio and other such tools for organizing and managing work. Section 1.10 will include a discussion on the important perspective that replication studies provide, for experimental studies, on the interplay between statistical analysis and scientific practice. The checks provided by independent replication at another time and place are an indispensable complement to statistical analysis. Chapter 2 will extend the discussion of this chapter to consider a wider class of models, methods, and model diagnostics.

*A note on terminology — variables, factors, and more!*

Much of data analysis is concerned with the statistical modeling of relationships or associations that can be gleaned from data, with a mathematical formula used to specify the model. There is an example at the beginning of Section 1.1.6.

The word *variable* will be used when data values are numeric. These include counts, as for example in `count` from the `DAAG::ACF1` data frame which has numbers of aberrant lesions in the lining of a rat's colon. The term *factor* will be used when values are on a categorical scale. Thus, in the data frame `DAAG::kiwishade`, `yield` is a variable with values such as 101.11, and `block` is a factor with levels `east`, `north`, and `west`. A factor may also represent values on an ordinal scale. Thus the factor `tint` in the data frame `DAAG::tinting` has ordered levels `no`, `lo`, and `hi`.

Continuous measurements can be further classified as having either an *interval*

scale or a *ratio* scale. Variables defined on an interval scale can take positive or negative values, and differences in the data values meaningful. Variables defined on a ratio scale are usually positive only, so quotients are more meaningful.

## 1.1 Questions, and data that may point to answers

Accounts of observed phenomena become part of established science once we know the circumstances under which they will recur. This process is relatively straightforward when applied to the study of regular events, such as a solar eclipse or the ocean tide levels in the Bay of Fundy in eastern Canada. Mathematical models that are based on sound physical principles can provide very accurate predictions for such events. Not everything is so readily predictable. How effective is a particular vaccine in preventing Covid-19-associated hospitalizations? How fast will a wildfire spread through a region with known topography and vegetation under given wind, temperature, and moisture conditions? Data from a suitable experiment or series of experiments may be able to go at least part of the way towards providing an answer. Thus, results from prescribed burns in designated forest stands where all relevant variables have been measured can provide a starting point for assessing the rate of spread of surface fires.

Or it may be necessary to rely on whatever data are already available. How effective are airbags in reducing the risk of death in car accidents? Data on car accidents in the United States over the period 1997-2002 are available. While careful and critical analyses of these data can help answer the question, caveats apply when the interest is in effectiveness at a later time and in another country. There have been important advances in the subsequent two decades in airbag design, manufacture, and systems that control deployment.

In Canada, there is a tendency for car passengers to use seatbelts at a higher rate than in the USA, so that efficacy assessments based on the American data have to be tempered when applied to the Canadian experience. The decision on which of the available datasets is best designed to provide an answer, and the choice of model, have called for careful and critical assessment. The help pages `?DAAG::nassCDS` and `?gamclass::FARS` provide further commentary. There is a strong interplay between the questions that can reasonably asked, and the data that are available or can be collected. Keep in mind, also, that different questions, asked of the same data, may demand different analyses.

### *1.1.1 A sample is a window into the wider population*

The population comprises all the data that might have been. The sample is the data that we have. Subjects for a sample to be surveyed should be selected randomly. In a clinical trial, it is important to randomly allocate subjects to different treatment groups.

Suppose, for example, that names on an electoral roll are numbered from 1 to

9384. The following uses the function `sample()` to obtain a random sample of 12 individuals:

```
## For the sequence below, precede with set.seed(3676)
sample(1:9384, 12, replace=FALSE) # NB: `replace=FALSE` is the default
```

```
 [1] 2263 9264 4490 8441 1868 3073 5430   19 1305 2908 5947  915
```

The numbers are the numerical labels for the 12 individuals who are included in the sample. The task is then to find them! The option `replace=FALSE` gives a *without replacement* sample, i.e., it ensures that no one is included more than once.

A more realistic example might be the selection of 1200 individuals, perhaps for purposes of conducting an opinion poll, from names numbered 1 to 19,384, on an electoral roll. Suitable code is:

```
chosen1200 <- sample(1:19384, 1200, replace=FALSE)
```

The following randomly assigns 10 plants (labeled from 1 to 10, inclusive) to one of two equal sized groups, control and treatment:

```
## For the sequence below, precede with set.seed(366)
split (sample(seq(1:10)), rep(c("Control","Treatment"), 5))
```

```
$Control
[1]  5  7  1 10  4

$Treatment
[1] 8 6 3 2 9
```

```
# sample(1:10) gives a random re-arrangement (permutation) of 1, 2, ..., 10
```

This assigns plants 3, 5, 10, 2, and 7 to the control group. This mechanism avoids any unwitting preference for placing healthier-looking plants in the treatment group.

The simple independent random sampling scheme can be modified or extended in ways that take account of structure in the data, with random sampling remaining a part of the data selection process.

### *Cluster sampling*

Cluster sampling is one of many probability-based variants on simple random sampling. See Barnett (2002). The function `sample()` can be used as before, but now the numbers from which a selection is made correspond to clusters. For example, households or localities may be selected, with multiple individuals from each. Standard inferential methods then require adaptation to account for the fact that it is the clusters that are independent, not the individuals within the clusters. Donner and Klar (2000) describe methods that are designed for use in health research.

### **A note on with-replacement samples*

For data that can be treated as a random sample from the population, one way to get an idea of the extent to which it may be affected by random variation is to take with-replacement random samples from the one available sample, and to do this

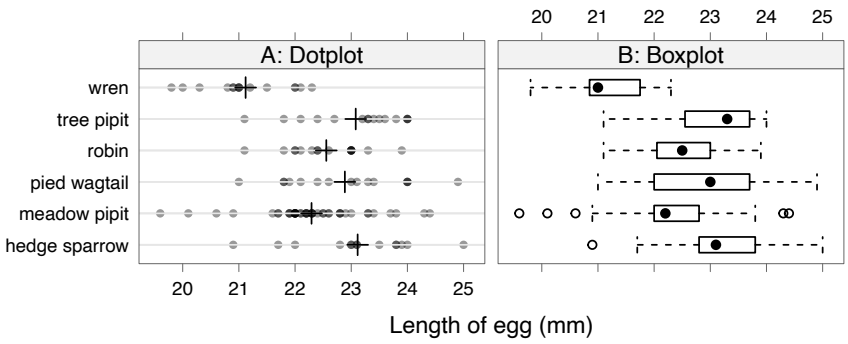



Figure 1.1 Dotplot (Panel A) and boxplot (Panel B) displays of cuckoo egg lengths. In Panel A, points that overlap have a more intense color. Means are shown as +. The boxes in Panel B take in the central 50% of the data, from 25% of the way through the data to 75% of the way through. The dot marks the median, Data are from Latter (1902).

repeatedly. The distribution that results can be an empirical substitute for the use of a theoretical distribution as a basis for inference.

We can randomly sample from the set $\{1, 2, \ldots, 10\}$, allowing repeats, thus:

```
sample(1:10, replace=TRUE)
```

```
 [1]  1  3  7  5  5 10  3  3  2  9
```

```
## sample(1:10, replace=FALSE) returns a random permutation of 1,2,...10
```

With-replacement sampling is the basis of *bootstrap* sampling.The effect is that of repeating each value an infinite number of times, and then taking a without replacement sample. Subsection 1.8.3 will demonstrate its use for calculating confidence intervals, as a way to reduce reliance on theoretical assumptions. Subsections 1.8.3 and 1.8.4 will demonstrate the methodology.

### *1.1.2 Formulating the scientific question*

Questions should be structured with a view both to the intended use of results, and to the limits of what the available data allow. Predictions of numbers in hospital from Covid-19 two weeks into the future do not demand the same level of scientific understanding or detailed data as needed to judge who among those infected are most likely to require hospitalization.

*Example: a question about cuckoo eggs*

*Cuculus canorus* is one of several species of cuckoos that lay eggs in the nests of other birds. The eggs are then unwittingly adopted and hatched by the hosts. Latter (1902) collected the data in `DAAG::cuckoos` and shown in Figure 1.1 in order to investigate claims in Newton and Gadow (1896, p. 123) that the cuckoo eggs tend to match the eggs of the host bird in size, shape and color. Panel A is a dotplot

Table 1.1 *Mean lengths of cuckoo eggs, compared with mean lengths of eggs laid by the host bird species. The table combines information from the two* DAAG *data frames* `cuckoos` *and* `cuckoohosts`.

| Host species | Meadow pipit | Hedge sparrow | Robin | Wagtails | Tree pipit | Wren | Yellow ammer |
|---|---|---|---|---|---|---|---|
| Length (cuckoo) | 22.3 (45) | 23.1 (14) | 22.5 (16) | 22.6 (26) | 23.1 (15) | 21.1 (15) | 22.6 (9) |
| Length (host) | 19.7 (74) | 20.0 (26) | 20.2 (57) | 19.9 (16) | 20 (27) | 17.7 (-) | 21.6 (32) |

(Numbers in parentheses are numbers of eggs)

display of the raw data. Panel B is the more summary boxplot form of display (to be discussed further in Section 1.1.5) that is designed to give a rough indication of how variation between groups compares with variation within groups.[1]

Table 1.1 adds information that suggests a relationship between the size of the host bird's eggs and the size of the cuckoo eggs that were laid in that nest. Observe that apart from several outlying egg lengths in the meadow pipit nests, the length variability within each host species' nest is fairly uniform.

In the paper (Latter, 1902) that supplied the cuckoo egg data of Figure 1.1 and Table 1.1, the interest was in whether cuckoos do in fact match the eggs that they lay to the host eggs, and if so in assessing which features match and to what extent.

Uniquely among the birds listed, the architecture of wren nests makes it impossible for the host birds to see the cuckoo's eggs, and the cuckoo's eggs do not match the wren's eggs in color. For the other species the color does mostly match. Latter concluded that the claim in Newton and Gadow (1896) is correct, that the eggs that cuckoos lay tend to match the eggs of the host bird in ways that will make it difficult for hosts to distinguish their own eggs from the cuckoo eggs.

Issues with the data in Table 1.1 and Figure 1.1 are:

- The cuckoo eggs and the host eggs are from different nests, collected over the course of several investigations. Data on the host eggs are from various sources.
- The host egg lengths for the wren are indicative lengths, from Gordon (1894).

There is thus a risk of biases, different for the different sources of data, that limit the inferences that can be drawn. How large, then, relative to statistical variation, is the difference between wrens and other species? Would it require an implausibly large bias to explain the difference? A more formal comparison between lengths for the different species based on an appropriate statistical model will be a useful aid to informed judgment.

Stripped down code for Figure 1.1 is:

```
library(latticeExtra)    # Lattice package will be loaded and attached also
cuckoos <- DAAG::cuckoos
## Panel A: Dotplot without species means added
dotplot(species ~ length, data=cuckoos)  ## `species ~ length` is a 'formula'
## Panel B: Box and whisker plot
bwplot(species ~ length, data=cuckoos)
## The following shows Panel A, including species means & other tweaks
av <- with(cuckoos, aggregate(length, list(species=species), FUN=mean))
```

[1] The code at the end of this section can be used to generate the separate graphs.

```
dotplot(species ~ length, data=cuckoos, alpha=0.4, xlab="Length of egg (mm)") +
  as.layer(dotplot(species ~ x, pch=3, cex=1.4, col="black", data=av))
  # Use `+` to indicate that more (another 'layer') is to be added.
  # With `alpha=0.4`, 40% is the point color with 60% background color
  # `pch=3`: Plot character 3 is '+'; `cex=1.4`: Default char size X 1.4
```

### *1.1.3 Planning for a statistical analysis*

First steps in any coordinated scientific endeavor must include clear identification of the question of interest, followed by careful planning. Consultation with subject matter specialists, as well as with specialists in statistical aspects of study design, will help avoid obvious mistakes in any of the steps: designing the study, collecting and/or collating data, carrying out analyses, and interpreting results.

If new data are to be acquired, one must decide if a designed experiment is feasible. In human or animal experimentation, such as in clinical trials to test a new drug therapy, ethics are an immediate concern. Data from experiments appear throughout this text – examples are the data on the tinting of car windows that is used for Figure 7.8 in Section 7.5, and the kiwifruit shading data that is discussed in Subsection 1.3.2. Such data can, if the experiment has been well-designed with a view to answering the questions of interest, give reliable results. Always, the question must be asked: "How widely do the results generalize?" For example, we might be interested in knowing to what extent the results for the kiwifruit shading conditions be generalized to other locations with different soil types and weather conditions.

#### *Understand the data*

Most standard elementary statistical methods assume that sample values were all chosen independently and with equal probability from the relevant population. If the data were from an observational study, such as in the cuckoo eggs example of Subsection 1.1.2, special care is required to consider what biases may have been induced by the method of data collection, and to ensure that they do not not lead to incorrect conclusions.

Temporal and spatial dependence are common forms of departure from independence, often leading to more complicated analyses. Data points originating from points that are close together in time and/or space are often more similar. Tests and graphical checks for dependence are necessarily designed to detect specific forms of dependence. Their effectiveness relies on recognizing forms of dependence that can be expected in the specific context.

If the data were acquired earlier and for a different purpose, details of the circumstances that surrounded the data collection are especially important. Were they from a designed experiment? If so, how was the randomization carried out? What factors were controlled? Was there a hierarchical structure to the data, such as would occur in a survey of students, randomly selected from classes, which are themselves randomly selected from schools, and so on. If the data were collected as part of an observational study, such as in the cuckoos example of Subsection

1.1.2, special care is required to ensure that hidden biases induced by the method of data collection do not lead to incorrect conclusions. Biases are likely when data are obtained from 'convenience' samples that have the appearance of surveys but which are really poorly designed observational studies. Online voluntary surveys are of this type. Similar biases can arise in experimental studies if care is not taken. For example, an agricultural experimenter may pick one plant from each of several parts of a plot. If the choice is not made according to an appropriate randomization mechanism, a preference bias can easily be introduced.

Nonresponse, so that responses are missing for some respondents, is endemic in most types of sample survey data. Or responses may be incomplete, with answers not provided to some questions. Dietary studies based on the self-reports of participants are prone to measurement error biases. With experimental data on crop or fruit yields, results may be missing for some plots because of natural disturbances caused by animals or harsh weather. One ignores the issue at a certain risk, but treating the problem is nontrivial, and the analyst is advised to determine as well as possible the nature of the missingness. It can be tempting to simply replace a missing height value for a male adult in a dataset by the average of the other male heights. Such a *single imputation* strategy will readily creates unwanted biases. Males that are of smaller than average weight and chest measurement are likely to be of smaller than average height. *Multiple imputation* is a generic name for methodologies that, by matching incomplete observations as closely as possible to other observations on the variables for which values are available, aim to fill in the gaps.

*Causal inference*

With data from carefully designed experiments, it is often possible to infer causal relationships. Perhaps the most serious danger is that the results will be generalized beyond the limits imposed by the experimental conditions.

Observational data, or data from experiments where there have been failures in design or execution, is another matter. Correlations do not directly indicate causation. A and B may be correlated because A drives B, or because B drives A, or because A and B change together, in concert with a third variable. For inferring causation, other sources of evidence and understanding must come into play.

*What was measured? Is it the relevant measure?*

The `DAAG::science` and `DAAG::socsupport` data frames are both from surveys. The former concerns student attitudes towards science in Australian private and public school systems. The latter concerns social and emotional support resources as they might relate to psychological depression in a sample of individuals.

In either case it is necessary to ask: "What was measured?" This question is itself amenable to experimental investigation. For the dataset `science`, what did students understand by 'science'? Was science, for them, a way to gain and test knowledge of the world? Or was it a body of knowledge? Or, more likely, was it a label for their experience of science laboratory classes (interesting sights, smells

and sounds perhaps) and field trips? Answers to other questions included in the survey shed some limited light.

In the `socsupport` dataset, an important variable is the Beck Depression Inventory or `BDI`, which is based on a 21-question multiple choice self-report. It is the outcome of a rigorous process of development and testing. Since its first publication in 1961, it has been extensively used, critiqued, and modified. Its results have been well validated, at least for populations on which it has been tested. It has become a standard psychological measure of depression (see, e.g., Streiner et al., 2014).

For therapies that are designed to prolong life, what is the relevant measure? Is it survival time from diagnosis? Or is a measure that takes account of quality of life over that time more appropriate. Two such measures are "Disability Adjusted Life Years" (DALYs) and "Quality Adjusted Life Years" (QALYs). Quality of life may differ greatly between the therapies that are compared.

*Use relevant prior information in the planning stages*

Information from the analysis of earlier data may be invaluable both for the design of data collection for the new study and for planning data analysis. When prior data are not available, a pilot study involving several experimental runs can sometimes provide such information.

Graphical and other checks are needed to identify obvious mistakes and/or quirks in the data. Graphs that draw attention to inadequacies may be suggestive of remedies. For example, they may indicate a need to numerically transform the data, such as by taking a logarithm or square root, in order to more accurately meet the assumptions underlying a more formal analysis. At the same time, one should keep in mind the risk that use of the data to influence the analysis may bias results.

*Subject area knowledge and judgments*

Data analysis results must be interpreted against a background of subject area knowledge and judgment. Some use of qualitative judgment is inevitable, relating to such matters as the weight that can be placed on claimed subject area knowledge, the measurements that are taken, the details of study design, the analysis choices, and the interpretation of analysis results. These, while they should be as informed as possible, involve elements of qualitative judgment. A well-designed study will often lead to results that challenge the insights and understandings that underpinned the planning.

*The importance of clear communication*

When there are effective lines of communication, the complementary skills of a data analyst and a subject matter expert can result in effective and insightful analyses. When unclear about the question of interest, or about some feature of the data, analysts should be careful not to appear to know more than is really the case. The subject matter specialist may be so immersed in the details of their problem that, without clear signals to the contrary, they may assume similar knowledge on the part of the analyst.

*Data-based selection of comparisons*

In carefully designed studies where subjects have been assigned to different groups, with each group receiving a different treatment, comparisons of outcomes between the various groups, and of subgroups within those groups (e.g. female/male, old/young) will be of interest. Among what may be many possible comparisons, the comparisons that will be considered should be specified in advance. Prior data, if available, can provide guidance. Any investigation of other comparisons may be undertaken as an exploratory investigation, a preliminary to the next study.

Data-based selection of one or two comparisons from a much larger number is not appropriate, since large biases may be introduced. Alternatively, there must be allowance for such selection in the assessment of model accuracy. The issues here are non-trivial, and we defer further discussion until later.

*Models must be fit for their intended use*

Statistical models must, along with the data upon which they rely, be applied according to their intended use. Architects and engineers have in the past relied heavily on scale models for giving a sense of important features of a planned building. For checking routes through the building, for the plumbing as well as for humans, such models can be very useful. They will not give much insight on how buildings in earthquake prone regions are likely to respond to a major earthquake — a lively concern in Wellington, New Zealand, where the first author now lives. For that purpose, engineers use mathematical equations that are designed to reflect the relevant physical processes. The credibility of predictions will strongly depend on the accuracy with which the models can be shown to represent those processes.

### *1.1.4 Results that withstand thorough and informed challenge*

Statistical models aim to give real world descriptions that are adequate for the purposes for which the model will be used. What checks will give confidence that a model will do the task asked of it? As argued in Tukey (1997), there must be exposure to diverse challenges that can build (or destroy!) confidence in model-based inferences. We should trust those results that have withstood thorough and informed challenge.

A large part of our task in this text is to suggest effective forms of challenge. Specific types of challenge may include:

- For experiments, carefully check and critique the design.
- Look into what is known of the processes that generated the data, and consider critically how this may affect its use and the reliance placed on it. Are there possible or likely biases?
- Look for inadequacies in laboratory procedure.
- Use all relevant graphical or other summary checks to critique the model that underpins the analysis.
- Where possible, check the performance of the model on test data that reflects

the manner of use of results. (If for example predictions are made that will be applied a year into the future, check how predictions made a year ahead panned out for historical data.)
- For experimental data, have the work replicated independently by another research group, from generation of data through to analysis.

In areas where the nature of the work requires cooperation between scientists with a wide range of skills, and where data is shared, researchers provide checks on each other. For important aspects of the work, the most effective critiques are likely to come from fellow researchers rather than from referees who are inevitably more remote from the details of what has been done. Failures of scientific processes are a greater risk where scientists work as individuals or in small groups with limited outside checks.

There are commonalities with the issues of legal and medical decision making that receive extensive attention in Kahneman et al. (2021, p.372), on the benefits of 'averaging', i.e., using the perspectives of multiple judges as a basis for decision making when sentencing, the authors comment:

> The advantage of averaging is further enhanced when judges have diverse skills and complementary judgment patterns.

Also needed is a high level of shared understanding.

For observational data, the challenges that are appropriate will depend strongly on the nature of the claims made as a result of any analysis. Dangers of over-interpretation and/or misinterpretation of results gleaned from observational data will be exemplified later in the text.

### *1.1.5 Using graphs to make sense of data*

Ideas of *Exploratory Data Analysis* (EDA), as formalized by John W. Tukey, have been a strong influence in the development of many of the forms of graphical display that are now in wide use. See Hoaglin (2003). A key concern is that the data should as far as possible speak for itself, prior to or as part of a formal analysis.

A use of graphics that is broadly in an EDA tradition continues to develop and evolve. The best modern statistical software makes a strong connection between data analysis and graphics, combining the computer's ability to crunch numbers and present graphs with the that of a trained human eye to detect pattern. Statistical theory has an important role in suggesting forms of display that may be helpful and interpretable.

#### *Graphical comparisons*

Figure 1.1 was a graphical comparison between the lengths of cuckoo eggs that had been laid in the nests of different host species. The boxes that give boxplots their name focus attention on quartiles of the data, i..e., the three points on the axis that split the data into four equal parts. The lower end of the box marks the first quartile, the dot marks the median, and the upper end of the box marks the third

quartile. Points that lie out beyond the 'whiskers' are plotted individually, and are candidates to be considered outliers. The widths of the boxes will of course vary randomly, leading in some cases to the flagging of points that should not be treated as extreme. The narrow box may largely account for the 5 values that are flagged for meadow pipit.

Figure 1.1 strongly suggested that eggs planted in wrens' nests were substantially smaller than eggs planted in other birds' nests. The upper quartile (75% point) for eggs in wren's nests lies below all the lower quartiles for other eggs.

### *1.1.6 Formal model-based comparison*

For comparing lengths between species in the cuckoo eggs data, we use the model:

$$\text{Egg length} = \text{Mean for species} + \text{Random variation}$$

The means in the dataset `cuckoos` are:

```
av <- with(cuckoos, aggregate(length, list(species=species), FUN=mean))
setNames(round(av[["x"]],2), abbreviate(av[["species"]],10))
```

```
hedgsparrw meadowpipt piedwagtal      robin tree pipit       wren
     23.11      22.29      22.89      22.56      23.08      21.12
```

The model postulates that the length of a cuckoo egg found in a given nest depends in some way on the host species. There are likely to be additional factors that have not been observed but which also influence the egg length. The variation due to these unobserved factors is aggregated into one term which is referred to as statistical error or random variation. Where none of these observed factors predominate and their effects add, a normal distribution will often be effective as a model for the random variation.

The species means are estimated from the data and are called *fitted values*. The differences between the data values and those means are called *residuals*. For example, suppose $\ell_i$ is the length of the $i$th egg in the nest of a wren, and $\bar{\ell}$ is the average of all eggs in the wrens' nests. Then the $i$th residual for this group is

$$e_i = \ell_i - \bar{\ell}.$$

The `scale()` function provides a convenient way to calculate such residuals; its usage below *centers* the data by subtracting the average from each data point. Thus, the residuals for the wren length model are:

```
with(cuckoos, scale(length[species=="wren"], scale=FALSE))[,1]
```

```
 [1] -1.32  0.98  0.38 -0.22  0.88 -0.12  1.18 -0.12 -0.82 -0.22  0.88
[12] -1.12 -0.32  0.08 -0.12
```

Is the variability different for different species? The boxes in Figure 1.1, with endpoints set for each species to contain the central 50% of the data, hint that variation may be greater for the Pied Wagtail than for other species. (The box widths equal the inter-quartile-range, or IQR. See further, Subsection 1.3.4.)

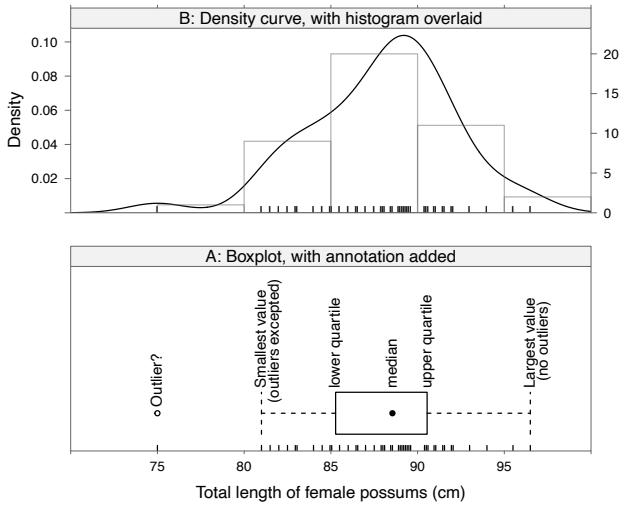



Figure 1.2 Panel A shows a boxplot, with annotation that explains boxplot features. Panel B shows a density plot, with a histogram overlaid. Histogram frequencies are shown on the right axis of Panel B. In both panels, the individual data points appear as a "rug" along the lower side of the bounding box. Where necessary, they have been moved slightly apart to avoid overlap.

## 1.2 Graphical tools for data exploration

In this section, we illustrate basic approaches to the graphical exploration of data. Three R static graphics systems enjoy wide use. These are: *base* (or 'traditional') graphics using `plot()` and associated commands, *lattice* which offers more stylized types of graphs, and *ggplot2* whose rich array of features comes at the cost of extra graphics language complexity.

Later chapters will make extensive use both of *base* graphics and of *lattice* graphics, resorting to *ggplot2* on those occasions when features are needed that are not readily available in the other packages. Some lattice graphs will be printed in a style (use a *theme*) akin to the default ggplot2 style. Section A.5 has further details.

### *1.2.1 Displays of a single variable*

A basic form of display for a single numeric variable is the dotplot, which plots the individual data points along a number line or single axis. The boxplot provides a coarser summary of *univariate* data. The histogram and density curve offer more fine-grained alternatives.

Figure 1.2A shows a boxplot of total lengths of females in the `possum` dataset, with annotation added that explains the interpretation of boxplot features. Figure 1.2B shows a density curve, with a histogram overlaid, for the same data. Both panels contain rug plots which are essentially dotplots consisting of vertical bars added along the lower edge.

One data point lies outside the boxplot 'whiskers' to the left, and is flagged as a

possible outlier. An outlier is a point that is determined to be far from the main body of the data. Under the default criterion, about 1% of normally distributed data would be judged as outlying.

A histogram is a crude form of density estimate. A smooth density estimate is, often, a better alternative. The height of the density curve at any point is an estimate of the proportion of sample values per unit interval, locally at that point. Both histograms and density curves involve an element of subjective choice. Histograms require the choice of breakpoints, while density estimates require the choice of a bandwidth parameter that controls the amount of smoothing. In both cases, the software has default choices that should be used with care.

Code for a slightly simplified version of Figure 1.2B is:

```
fossum <− subset(DAAG::possum, sex=="f")
densityplot(~totlngth, plot.points=TRUE, pch="|", data=fossum) +
  layer_(panel.histogram(x, type="density", breaks=c(75,80,85,90,95,100)))
```

*Comparing univariate displays across factor levels*

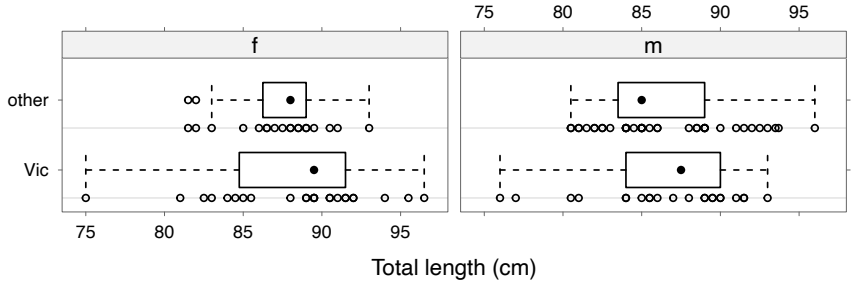



Figure 1.3 Total lengths of possums, by `sex` and (within panels) by geographical location (Victorian or other).

Univariate summaries can be broken down by one or more factors between and/or within panels. Figure 1.3 overlays dotplots on boxplots of the distributions of Australian possum lengths, broken down by `sex` and (within panels) by geographical region (Victoria or other).

```
## Create boxplot graph object --- Simplified code
gph <− bwplot(Pop~totlngth | sex, data=possum)
## plot graph, with dotplot distribution of points below boxplots
gph + latticeExtra::layer(panel.dotplot(x, unclass(y)−0.4))
```

The normal distribution is not necessarily the appropriate reference. Points may be identified as outliers because the distribution is skew (usually, with a tail to the right). Any needed action will depend on the context, requiring the user to exercise good judgement. Subsection 1.2.8 will comment in more detail.

### 1.2.2 Patterns in univariate time series

Figure 1.4 shows time plots of historical deaths from measles in London. (Here, 'measles' includes both what is nowadays called measles and the closely related rubella or German measles.)

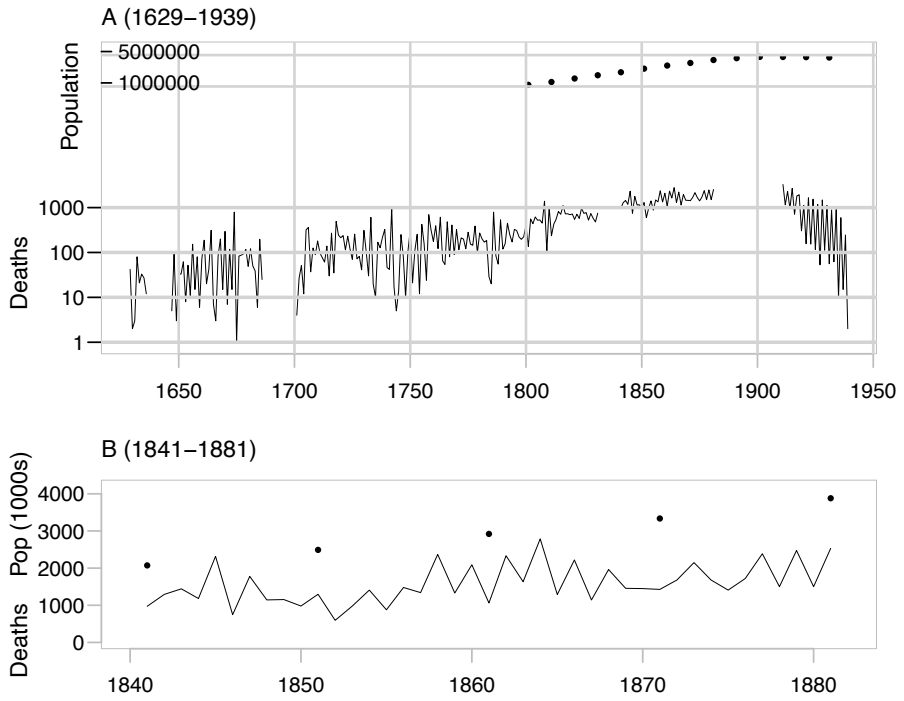



Figure 1.4 The two panels provide different insights into data on mortality from measles, in London over 1629-1939. Panel A uses a logarithmic scale to show the numbers of deaths from measles in London for the period from 1629 through 1939 (black curve). The black dots show, for the period 1800 to 1939 the London population in 1000s. Panel B shows, on the linear scale (black curve), the subset of the measles data for the period 1840 through 1882 together with the London population (in thousands, black dots).

Panel A uses a logarithmic vertical scale while Panel B uses a linear scale and takes advantage of the fact that annual deaths from measles were of the order of one in 500 of the population. Thus, deaths in thousands and population in half millions can be shown on the same scale.

Panel A shows broad trends over time, but is of no use for identifying changes on the time-scale of a year or two. In Panel B, the lines that show such changes are, mostly, at an angle that is in the approximate range of 20° to 70° from the horizontal. A sawtooth pattern is evident, indicating that years in which there were many deaths were often followed by years in which there were fewer deaths. To obtain this level of detail for the whole period from 1629 until 1939, multiple panels would be necessary.

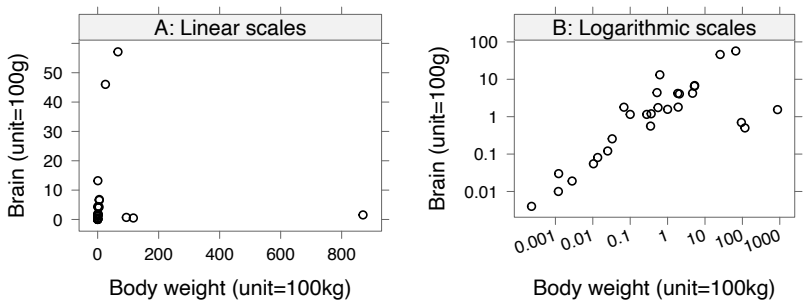



Figure 1.5 Brain weight versus body weight, for 28 animals that vary greatly in size. Panel A has untransformed scales, while Panel B has logarithmic scales, on both axes.

Simplified code is:

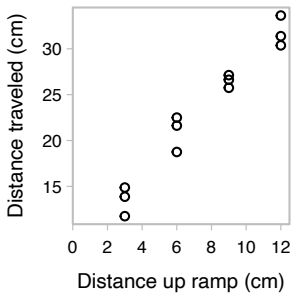
```
measles <− DAAG::measles
## Panel A
plot(log10(measles), xlab="", ylim=log10 (c(1,5000*540)),
     ylab=" Deaths; Population", yaxt="n")
ytiks1 <− c(1, 10, 100, 1000); ytiks2 <− c(1000000, 5000000)
## London population in thousands
londonpop <−ts(c(1088,1258,1504,1778,2073,2491,2921,3336,3881,
  4266,4563,4541,4498,4408), start=1801, end=1931, deltat=10)
points(log10(londonpop*600), pch=16, cex=.5)
abline(h=log10(ytiks1), lty = 2, col = "gray", lwd = 2)
abline(h=log10(ytiks2*0.5), lty = 2, col = "gray", lwd = 2)
axis(2, at=log10(ytiks1), labels=paste(ytiks1), lwd=0, lwd.ticks=1)
axis(2, at=log10(ytiks2*0.5), labels=paste(ytiks2), tcl=0.3,
     hadj=0, lwd=0, lwd.ticks=1)
## Panel B
plot(window(measles, start=1840, end=1882), ylim=c (0, 4600), yaxt="n")
points(londonpop, pch=16, cex=0.5)
axis(2, at=(0:4)* 1000, labels=paste(0:4), las=2)
```

For details of the data, and commentary, see Guy (1882), Stocks (1942), and Senn (2003) where interest was in the comparison with smallpox mortality. The population estimates (`londonpop`) are from Mitchell (1988).

### *1.2.3 Visualizing relationships between pairs of variables*

Patterns and relationships linking multiple variables are a primary focus of data analysis. The following example is concerned with the relationship between two variables and illustrates an important question that often arises: what is the appropriate scale?

Figures 1.5A and B plot brain weight (g) against body weight (kg), for 28 animals. Panel A indicates that the distributions of data values are highly positively skew, on both axes, but is otherwise unhelpful. Panel B's logarithmic scales spread points out more evenly, and the graph tells a clearer story. Note that, on both axes, tick




| Starting point | Distance traveled | | |
|---|---|---|---|
| 3 | 31.38 | 30.38 | 33.63 |
| 6 | 26.63 | 25.75 | 27.13 |
| 9 | 18.75 | 22.50 | 21.63 |
| 12 | 13.88 | 11.75 | 14.88 |

Figure 1.6 Distance traveled (*distance.traveled*) by model car, as a function of starting point (*starting.point*), up a 20° ramp.

marks are separated by an amount that, when translated back from log(weight) to weight, differ by a factor of 100. The argument `aspect="iso"` has ensured that these correspond to the same physical distance on both axes of the graph. Code is:

```
## Untransformed vs log transformed scales
Animals <− MASS::Animals
asp <− with(Animals, sapply(list(log(brain/100), log(body/100)),
                            function(x) diff(range(x)))) |> (\(d)d[1]/d[2])()
xlab <− "Body weight (unit=100kg)"; ylab <− "Brain (unit=100g)"
gphA <− xyplot(I(brain/100) ~ I(body/100), data=Animals, aspect=asp,
               xlab=xlab, ylab=ylab)
gphB <− xyplot(log(brain/100) ~ log(body/100), data=MASS::Animals,  # Panel B
               aspect='iso',  xlab=xlab, ylab=ylab)
labx <− 10^c((−3):3); laby <− 10^c((−2):2)
gphB <− update(gphB, scales=list(x=list(at=log(labx), labels=labx, rot=20),
                                 y=list(at=log(laby),  labels=laby)))
```

A logarithmic scale is appropriate for quantities that change multiplicatively. Thus, if cells in a growing organism divide and produce new cells at a constant rate, then the total number of cells changes multiplicatively, resulting in what is termed exponential growth. Large organisms may similarly increase in a given time interval by the same approximate fraction as smaller organisms. Growth rate on a natural logarithmic scale ($\log_e$) equals the relative growth rate.

Anyone who works with real data — biologists, economists, physical scientists — will do well to make themselves comfortable with the use and interpretation of logarithmic scales. See Subsection 2.5.6 for a brief discussion of other commonly used transformations.

### *1.2.4 Response lines (and/or curves)*

The data shown to the right of Figure 1.6, and plotted in the figure, were generated by releasing a model car three times at each of four different distances (`starting.point`) up a 20° ramp. The experimenter recorded distances traveled from the bottom of the ramp across a concrete floor. Response curve analysis, using regression, is appropriate. It would be a mistake to treat the four starting points as factor levels in a one-way analysis.

For these data, the physics suggests the likely form of response. Where no such help is available, careful examination of the graph, followed by systematic examination of plausible forms of response, may suggest a suitable form of response curve.

### 1.2.5* Multiple variables and times

Overlaying plots of several time series (sequences of measurements taken at regular intervals) might seem appropriate for making direct comparisons. However, this approach will only work if the scales are comparable for the different series.

Figures 1.7A and B show alternative views of labor force numbers (thousands), for various regions of Canada, at quarterly intervals over the 24-month period from January 1995 to December 1996. Over this time, Canada was emerging from a deep economic recession. The ranges of values, for each of the six regions, are:

```
## Apply function range to columns of data frame jobs (DAAG)
sapply(DAAG::jobs, range)  ## NB: `BC` = British Columbia
```

```
       BC Alberta Prairies Ontario Quebec Atlantic  Date
[1,] 1737    1366      973    5212   3167      941 95.00
[2,] 1840    1436      999    5360   3257      968 96.92
```

With a logarithmic scale, as in Figure 1.7A, similar changes on the scale correspond to similar proportional changes. The regions have been taken in order of the number of workers in December 1996 (or, in fact, at any other time). This ensures that the order of the labels in the key matches the positioning of the points for the different regions. Code that has been used to create and update the graphics object `basicGphA`, then updating it to obtain the labeling on the $x$- and $y$-axes is:

```
## Panel A: Basic plot; all series in a single panel; use log y-scale
formRegions <- Ontario+Quebec+BC+Alberta+Prairies+Atlantic ~ Date
basicGphA <-
  xyplot(formRegions, outer=FALSE, data=DAAG::jobs, type="l", xlab="",
         ylab="Number of workers", scales=list(y=list(log="e")),
         auto.key=list(space="right",  lines=TRUE, points=FALSE))
  ## `outer=FALSE`: plot all columns in one panel
## Create improved x- and y-axis tick labels; will update to use
datelabpos <- seq(from=95, by=0.5, length=5)
datelabs <- format(seq(from=as.Date("1Jan1995", format="%d%b%Y"),
                       by="6 month", length=5), "%b%y")
## Now create $y$-labels that have numbers, with log values underneath
ylabposA <- exp(pretty(log(unlist(DAAG::jobs[,-7])), 5))
gphA <- update(basicGphA, scales=list(x=list(at=datelabpos, labels=datelabs),
               y=list(at=ylabposA, labels=ylabelsA)))
```

Because the labor forces in the various regions do not have similar sizes, it is impossible to discern any differences among the regions from this plot. Plotting on the logarithmic scale was not enough on its own. Figure 1.7B, where the six different panels use different *slices* of the same logarithmic scale, is an informative alternative. Simplified code is:

```
## Panel B: Separate panels (`outer=TRUE`); sliced log scale
basicGphB <-
```

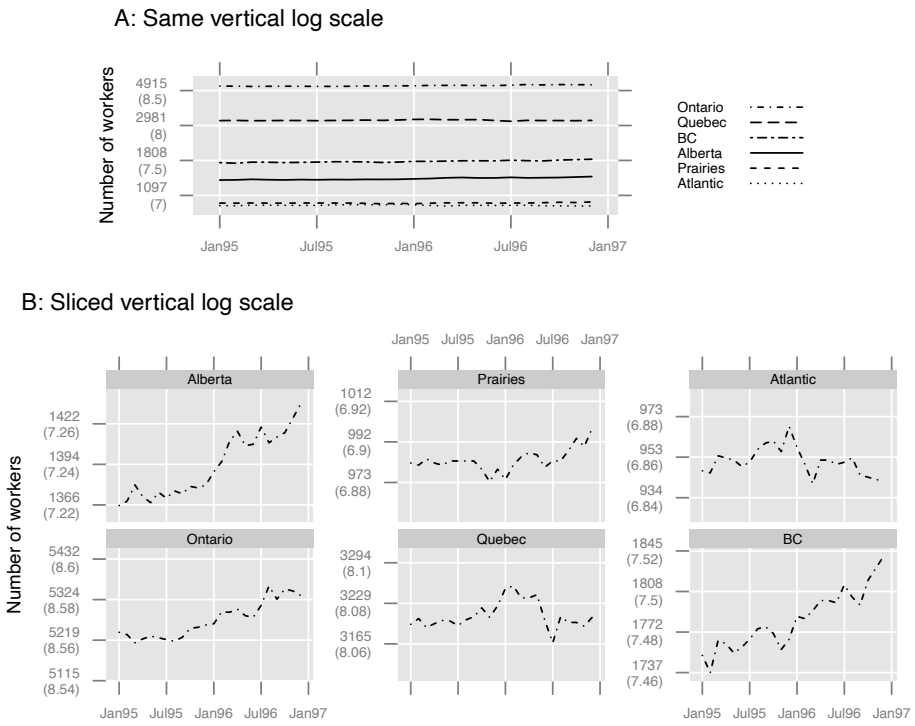



Figure 1.7 Data are labor force numbers (thousands) for various regions of Canada, at quarterly intervals over 1995-1996. Panel A uses the same logarithmic $y$-scale for all regions. Panel B shows the same data, but now with separate ('sliced') logarithmic $y$-scales on which the same percentage increase, e.g., by 1%, corresponds to the same distance on the scale, for all plots. Distances between ticks are 0.02 on the $\log_e$ scale, i.e., a change of close to 2%.

```
xyplot(formRegions, data=DAAG::jobs, outer=TRUE, type="l", layout=c(3,2),
       xlab="", ylab="Number of workers",
       scales=list(y=list(relation="sliced", log=TRUE)))
```

Use of `outer=TRUE`, causes separate columns (regions) to be plotted on separate panels. As before, equal distances on the scale correspond to equal relative changes. It is now clear that Alberta and BC experienced the fastest job growth and that there was little or no job growth in Quebec and the Atlantic region.

The following are the changes in numbers employed, in each of Alberta and BC, from January 1995 to December 1996. The changes are shown in actual numbers, and on scales of $\log_2$, $\log_e$ and $\log_{10}$. Figure 1.8 shows this graphically.

| | | Increase | | |
|---|---|---|---|---|
| | Rel. change | $\log_2$ | $\log_e$ | $\log_{10}$ |
| Alberta (1366 to 1466; increase=70) | 1.051 | 0.072 | 0.050 | 0.022 |
| BC (1752 to 1840; increase=88) | 1.050 | 0.070 | 0.049 | 0.021 |

From the beginning of 1995 and the end of 1996, the increase of 70 in Alberta

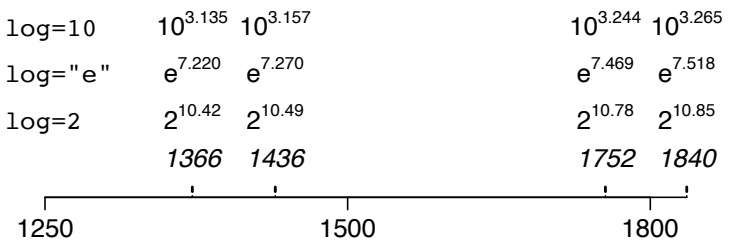



Figure 1.8 Labeling of the values for Alberta (1366, 1436) and BC (1752, 1840), with alternative logarithmic scale choices.

from 1366 to 1436 is by a factor of $1436/1366 \simeq 1.051$). For BC, an increase by 88 from 1752 to 1840 is by a factor of 1.050. The proper comparison is not between the absolute increases, but between very nearly identical multipliers of 1.051 and 1.050.

Even better than using a logarithmic $y$-scale, particularly if ready comprehension is important, would be to standardize the labor force numbers by dividing, e.g., by the respective number of persons aged 15 years and over at that time. Scales would then be directly comparable. (The `plot` method for time series could then suitably be used to plot the data as a multivariate time series. See `?plot.ts`.)

### *1.2.6* Labeling technicalities*

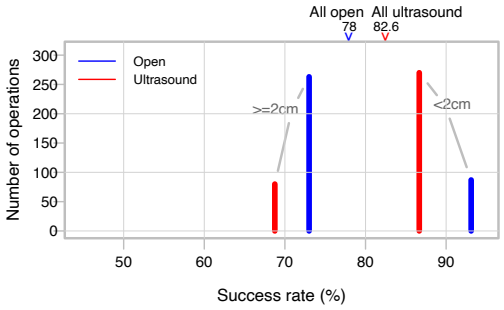
For lattice functions, the arguments `log=2` or `log="e"` or `log=10` are available. The latter two scales are referred to as natural and common log scales, respectively. These use the relevant logarithmic axis labeling, as in Figure 1.8, for axis labels. In base graphics, with one of the arguments `log="x"` or `log="y"` or `log="xy"`, the default is to label the specified axis or axes in the original units.

An alternative, both for traditional and lattice graphics, is to enter the log-transformed values, using whatever base is preferred (`2` or `"e"` or `10`), into the graphics formula. Unless other tick labels are manually entered, ticks will be automatically transformed to the correct scale.

Note again the reason for placing $y$-axis tick marks a distance 0.02 apart on the $\log_e$ linear scale used in Figure 1.7. On the $\log_e$ scale a change of 0.02 is very nearly a 2% change.

### *1.2.7 Graphical displays for categorical data*

Figure 1.9 illustrates the possible hazards of adding values in a multi-way table over one of its margins. Data are from a study (Charig, 1986) that compared the use of open surgery for kidney stones with a method that made a small incision and used ultrasound to destroy the stone. Stones were classified by diameter: either at least 2 cm or less than 2 cm. For each subject, the outcome was assessed as successful ('yes') or unsuccessful ('no').




```
Success
yes no Total %Yes
Method     Size
open       <2cm    81  6    87 93.1
>=2cm  192 71   262 73.0
ultrasound <2cm   234 36   280 86.7
>=2cm   55 25    80 68.8

Add over Size
open              273 77   350 78.0
ultrasound        289 61   350 82.6
```

Figure 1.9 Outcomes are for two different types of surgery for kidney stones. The overall (apparent) success rates (78% for open surgery as against 83% for ultrasound) favor ultrasound. The success rate for each size of stone separately favors, in each case, open surgery.

If we consider small stones and large stones separately, it appears that surgery is more successful than ultrasound. The blue vertical bar Figure 1.9 is in each case to the right of the corresponding red vertical bar. The overall counts, which favor ultrasound, are thus misleading. For open surgery, the larger number of operations for large stones (263 large, 87 small) weights the overall success rate towards the low overall success rate for large stones. For ultrasound surgery (red bars), the weighting (80 large, 280 small) is towards the high success rate for small stones. This is an example of the phenomenon called the Simpson or Yule-Simpson paradox. (See also Subsection 2.1.2.)

Note that without additional information, the results are not interpretable from a medical standpoint. Different surgeons will have preferred different surgery types, and the prior condition of patients will have affected the choice of surgery type. The consequences of unsuccessful surgery may have been less serious for ultrasound than for open surgery.

The table `stones`, shown to the right of Figure 1.9, has three `margins` — `Success`, `Method`, and `Size`. The table `margin12` that results from adding over `Size` retains the first two of these. Code used is:

```
stones <- array(c(81,6,234,36,192,71,55,25),  dim=c(2,2,2),
                dimnames=list(Success=c("yes","no"),
                Method=c("open","ultrasound"), Size=c("<2cm", "≥2cm")))
margin12 <- margin.table(stones, margin=1:2)
```

Mosaic plots are an alternative type of display that can be obtained using either `mosaicplot()` from base graphics or `vcd::mosaic()`. Figure 1.9 makes the point of interest for the kidney stone surgery data more simply and directly.

### *1.2.8 What to look for in plots*

We now note points to keep in mind when examining data.

*Outliers*

Outliers are points that appear, or are judged, isolated from the main body of the data. Such points, whether errors or genuine values, can indicate departure from model assumptions, and may distort any model that is fitted..

Boxplots, and the normal quantile-quantile plot that will be discussed in Subsection 1.4.3, are useful for highlighting outliers in one dimension. Scatterplots may highlight outliers in two dimensions. Some outliers will, however, be apparent only in three or more dimensions.

*Asymmetry of the distribution*

Positive skewness (a tail to the right) is a common form of departure from normality. The largest values are widely dispersed, and values near the minimum are likely to be bunched up together. Provided that all values are greater than zero, a logarithmic transformation typically makes such a distribution more symmetric. Negative skewness (a tail to the left) is less common. Severe skewness is typically a more serious problem for the validity of analysis results than other types of non-normality.

If values of a variable that takes positive values range by a factor of more than 10:1 then, depending on the application area context, positive skewness is to be expected. A logarithmic transformation should be considered.

*Changes in variability*

Boxplots and histograms readily convey an impression of the extent of variability or scatter in the data. Side by side boxplots, such as in Figure 1.1B, or dotplots such as in Figure 1.1A, allow rough comparisons of the variability across different samples or treatment groups. They provide a visual check on the assumption, common in many statistical models, that variability is constant across treatment groups.

It is easy to over-interpret such plots. Statistical theory offers useful and necessary warnings about the potential for such over-interpretation. (The variability in a sample, typically measured by the variance, is itself highly variable under repeated sampling. Measures of variability will be discussed in Subsection 1.3.3.)

When variability increases as data values increase, the logarithmic transformation will often help. Constant relative variability on the original scale becomes constant absolute variability on a logarithmic scale.

*Clustering*

Clusters in scatterplots may suggest features of the data that may or may not have been expected. Upon proceeding to a formal analysis, any clustering must be taken into account. Do the clusters correspond to different values of some relevant variable? Outliers are a special form of clustering.

*Nonlinearity*

Where it seems clear that one or more relationships are nonlinear, a transformation may make it possible to model the relevant effects as linear. Where none of the

common standard transformations meets requirements, methodology is available that will fit quite general nonlinear curves. See Subsection 4.4.2.

If there is a theory that suggests the form of model, then this is a good starting point. Available theory may, however, incorporate various approximations, and the data may tell a story that does not altogether match the available theory. The data, unless they are flawed, have the final say!

*Time trends in the data*

It is common to find time trends that are associated with order of data collection. It can be enlightening to plot residuals, or other quantities, against time. Patterns of increase or decrease are common and are readily recognized, but one should also be alert to the possibility of seasonality or periodic behavior.

## 1.3 Data Summary

Data summaries may: (1) be of interest in themselves; (2) give insight into aspects of data structure that may affect further analysis; (3) be used as data for further analysis. In case (3), it is necessary to ensure that important information, relevant to the analysis, is not lost. Before adding counts across the margins of multi-way tables, or otherwise pooling data across different groups, it is important to check the potential for distortions that are artifacts of the way that the data have been summarized. Examples will be given.

If there is no loss of information, use of summary data can allow a helpful simplicity of analysis and interpretation, Do not, however, proceed without careful consideration!

### *1.3.1 Counts*

The data frame `DAAG::nswpsid1` is from a study (Lalonde, 1986) that compared two groups of individuals with a history of unemployment problems – one an 'untreated' control group and the other a 'treatment' group whose members were exposed to a labor training program. The data include measures that can be used for checks on whether the two groups were, aside from exposure (or not) to the training program, otherwise plausibly similar. The following compares the relative numbers between who had completed high school (`nodeg = 0`) and those who had not (`nodeg = 1`).

```
## Table of counts example: data frame nswpsid1 (DAAG)
## Specify `useNA="ifany"` to ensure that any NAs are tabulated
tab <- with(DAAG::nswpsid1, table(trt, nodeg, useNA="ifany"))
dimnames(tab) <- list(trt=c("none", "training"), educ = c("completed", "dropout"))
tab
```

```
          educ
trt        completed dropout
  none          1730     760
  training        80     217
```

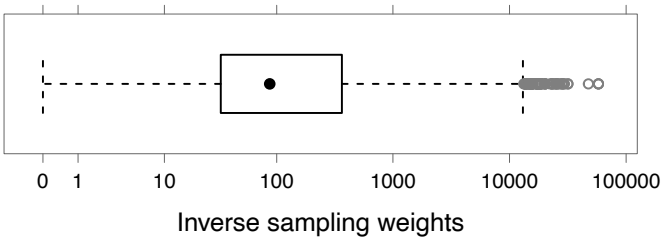



Figure 1.10 Boxplot showing `weights` (inverse sampling fractions), in the dataset `DAAG::nassCDS`. A `log(weight+1)` scale): has been used.

The training group has a much higher proportion of dropouts. Similar comparisons are required for other factors, variables, and combinations of two factors or variables. The data will be investigated further in Section 9.7.1.

*Tabulation that accounts for frequencies or weights – the `xtabs()` function*

Each year the National Highway Traffic Safety Administration in the United States uses a stratified random sampling method to collect data from all police-reported collisions in which there is an injury to people or property and where at least one vehicle is towed. Sampling fractions differ according to class of accident. The subset in `DAAG::nassCDS` is restricted to front-seat occupants.[2]

Factors whose effect warrant investigation include: `airbag` (was an airbag fitted?), `seatbelt` (was a seatbelt used?), and `dvcat` (a force of impact measure). The column *weight* (*national inflation factor*) holds the inverses of the sampling fraction estimates. The less accurate estimates that come where the sampling fraction is small have to be given an accordingly greater weight in the calculation of overall estimates, in order to fairly represent the population. Very large weights, for some classes of accident, will exaggerate the effect, both of any mistakes in data collection, and of deviations from the prescribed (and relatively complex) sampling scheme. The following contrasts numbers in the sample with estimated total numbers of collisions, obtained by applying the sampling weights:

```
sampNum <- table(nassCDS$dead)
popNum <- as.vector(xtabs(weight ~ dead, data=nassCDS))
rbind(Sample=sampNum, "Total number"=round(popNum,1))
```

```
                alive  dead
Sample          25037  1180
Total number 12067937 65595
```

Use of `xtabs()` to classify the estimated population numbers (in 1000s) by airbag use, and adding the marginal death rates per 1000 to the table, gives:

```
nassCDS <- DAAG::nassCDS
```

[2] It holds a subset of the columns from a corrected version of the data analyzed in Meyer and Finney (2005). See also Farmer (2005) and Meyer (2006). More complete data are available from one of the web pages noted on the help page for `nassCDS`.

```
Atab <- xtabs(weight ~ airbag + dead, data=nassCDS)/1000
## Define a function that calculates Deaths per 1000
DeadPer1000 <- function(x)1000*x[2]/sum(x)
Atabm <- ftable(addmargins(Atab, margin=2, FUN=DeadPer1000))
print(Atabm, digits=2, method="compact", big.mark=",")
```

```
airbag | dead    alive    dead DeadPer1000
none           5,445.2    39.7          7.2
airbag         6,622.7    25.9          3.9
```

This might suggest that the fitting of an airbag substantially reduces the risk of mortality. Consider however:

```
SAtab <- xtabs(weight ~ seatbelt + airbag + dead, data=nassCDS)
## SAtab <- addmargins(SAtab, margin=3, FUN=list(Total=sum))  ## Gdet Totals
SAtabf <- ftable(addmargins(SAtab, margin=3, FUN=DeadPer1000), col.vars=3)
print(SAtabf, digits=2, method="compact", big.mark=",")
```

```
seatbelt airbag | dead        alive         dead DeadPer1000
none     none           1,342,021.9    24,066.7         17.6
         airbag           871,875.4    13,759.9         15.5
belted   none           4,103,224.0    15,609.4          3.8
         airbag         5,750,815.6    12,159.2          2.1
```

The `Total` column gives the weights that are, effectively, applied to the values in the `DeadPer1000` column when the raw numbers are added over the *seatbelt* margin. In the earlier table (*Atab*), the results for *airbag=none* were mildly skewed ( 4119:1366) to those for *belted*. Results with airbags were strongly skewed ( 5763:886) to those for *seatbelt=none*. Hence, adding over the *seatbelt* margin gave a spuriously large advantage to the presence of an airbag.

The reader may wish to try an analysis that accounts, additionally, for estimated force of impact (*dvcat*):

```
FSAtab <- xtabs(weight ~ dvcat + seatbelt + airbag + dead, data=nassCDS)
FSAtabf <- ftable(addmargins(FSAtab, margin=4, FUN=DeadPer1000), col.vars=3:4)
print(FSAtabf, digits=1)
```

There is no consistent pattern in the difference between *"none"* and *"airbag"*.

Further terms, including the age of vehicle and the age of driver, demand consideration. The estimated effect of *airbag*, or of any factor other than `seatbelt`, varies depending on what further terms are included in the model. Seatbelts have such a large effect that their contribution stands out irrespective of what other terms appear in the model. These data, tabulated as above, have too many uncertainties and potential sources of bias to give reliable answers.

A better starting point for investigation are the data from the Fatality Analysis Recording System (FARS). The `gamclass::FARS` dataset has data for the years 1998 to 2010. This has, in principle at least, a complete set of records for the more limited class of accidents where there was at least one fatality.

Farmer (2005) used the FARS data for an analysis, limited to cars without passenger airbags, that used front seat passenger mortality as a standard against which to compare driver mortality. In the absence of any effect from airbags, the ratio of driver mortality to passenger mortality should be the same, irrespective of whether

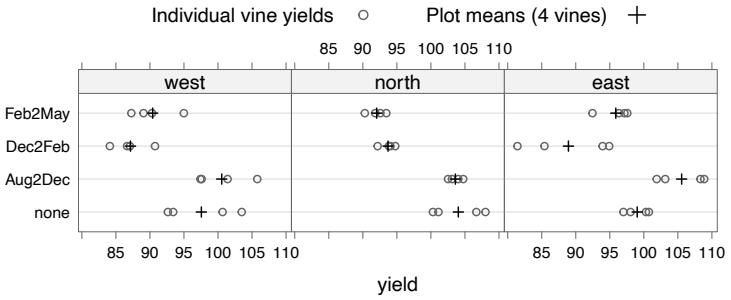



Figure 1.11 Individual yields and plot-level mean yields of kiwifruit (in kg) for each of four treatments (season) and blocks (exposure).

there was a driver airbag. Farmer found a ratio of driver fatalities to passenger fatalities that was 11% lower in the cars with driver airbags. Factors that have a large effect on the absolute risk can be expected to have a much smaller effect on the relative risk.

In addition to the functions discussed, note the function `gmodels::CrossTable()`, which offers a choice of SPSS-like and SAS-like output formats.

### *1.3.2 Summaries of information from data frames*

The data frame `DAAG::kiwishade` has yield measurements from 48 kiwifruit vines. Plots, made up of 4 vines each, were the experimental units. Figure 1.11 plots both the aggregated means and the individual vine results.

The 12 plots were divided into three blocks of four plots each. One block of four was north-facing, a second block west-facing, and a third block east-facing. (Because the trial was conducted in the Southern hemisphere, there is no south-facing block.) Shading treatments were applied to whole plots, i.e., to groups of four vines, with each treatment occurring once per block. The shading treatments were applied either from August to December, December to February, February to May, or not at all. For more details of the experiment, look ahead to Figure 7.5.

As treatments were applied to whole plots, a focus on the individual vines exaggerates the extent of information that is available, in each block, for comparing treatments. To gain an accurate impression of the strength of the evidence, focus on the means, represented by `+`. The code is given as a footnote.[3] The code includes

[3]
```
## Individual vine yields, with means by block and treatment overlaid
kiwishade <- DAAG::kiwishade
kiwishade$block <- factor(kiwishade$block, levels=c("west","north","east"))
keyset <- list(space="top", columns=2,
text=list(c("Individual vine yields", "Plot means (4 vines)")),
points=list(pch=c(1,3), cex=c(1,1.35), col=c("gray40","black")))
panelfun <- function(x,y,...){panel.dotplot(x,y, pch=1, ...)
av <- sapply(split(x,y),mean); ypos <- unique(y)
lpoints(ypos~av, pch=3, col="black")}
dotplot(shade~yield | block, data=kiwishade, col="gray40", aspect=0.65,
       panel=panelfun, key=keyset, layout=c(3,1))
## Note that parameter settings were given both in the calls
## to the panel functions and in the list supplied to key.
```

a user-defined panel function to take means, for each combination of block and shading treatment, "on the fly". Code that creates the means separately from the graph, with the first line of output following, is:

```
## mean yield by block by shade: data frame kiwishade (DAAG)
kiwimeans <- with(DAAG::kiwishade,
                  aggregate(yield, by=list(block, shade), mean))
names(kiwimeans) <- c("block","shade","meanyield")
head(kiwimeans, 4)  # First 4 rows
```

```
  block   shade meanyield
1  east    none     99.03
2 north    none    104.03
3  west    none     97.56
4  east Aug2Dec    105.56
```

The `aggregate()` function splits the data frame according to the specified combinations of factor levels, and then applies a specified function to each of the resulting subgroups.

Should the analysis then use the aggregated data? The form of analysis of variance that will be applied to these data in Subsection 7.4.1 will give, for treatment comparison purposes, the same results as an analysis based directly on the plot means, tacitly assuming that the mean is the appropriate form of summary. If there were occasional highly aberrant values, use of medians might be preferable. Use of aggregated data gives the analyst more control over which summary statistic to use.

*The benefits of data summary – dengue status example*

Hales et al. (2002) examined the implications of climate change projections for the worldwide distribution of dengue, a mosquito-borne disease that is a risk in hot and humid regions. Dengue status, i.e., information on whether dengue had been reported during 1965–1973, was available for 2000 administrative regions. Climate information was available on a much finer grid of about 80 000 pixels at 0.5° latitude and longitude resolution. Should the analysis work with summary data for 2000 administrative regions, or with the much larger dataset that has one row for each of the 80 000 pixels?[4] The following are reasons that might suggest working with the summary data:

- Use of the values at the pixel level to calculate summary climate statistics prior to the analysis, by administrative region, gives the user control over the choice of statistical summary statistic. If for example values for some pixels are extreme relative to other pixels in the administrative region, medians may be more appropriate than means. The mean will give the same weight to sparsely populated cold mountainous locations as to highly populated hot and humid locations on nearby plains.

[4] Working with spatio-temporal data aggregated over different subregions can require highly complex analysis procedures. Lee et al. (2017) describes a Canadian bladder cancer study where data from postal code regions was merged in a nontrivial manner with data from census regions.

- Correlation between nearby observations, though still substantial, will be less of an issue for the dataset in which each row is an administrative region. Points that repeat essentially identical information are a problem for the interpretation of plots and can be a problem for the analysis. Regions that are close together tend to have similar climates and the same dengue status.
- Analysis and data screening are simpler with the reduced dataset. Scatterplots are less likely to degenerate into a dense mass of black ink.

The mean and the median are the most commonly used measures of central value, among several alternatives. (In fact, the paper used the disaggregated data.)

### 1.3.3 Measures of variation

The *standard deviation* (often represented by the symbol $\sigma$) is a standard form of summary of variability (or *spread*) of sample or population values. Given a random sample $x_1, x_2, \ldots, x_n$, $\sigma$ can be estimated by the sample standard deviation:

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}.$$

For $s$ to be an accurate estimator for $\sigma$, the sample must be large. In R, `sd()` can be used to calculate $s$.

The square of the standard deviation, termed the *variance*, is widely used in formal inference. The standard deviation lends itself to easier interpretation because it is in the same units as the original measurements.

*Cuckoo eggs example*

Consider the cuckoo eggs data from Subsection 1.1.2. The standard deviations for each group, with numbers of eggs shown in parentheses, are:

```
hedgsparrow meadowpipit piedwagtail       robin  tree pipit        wren
  1.05 (14)   0.92 (45)   1.07 (15)   0.68 (16)   0.88 (15)   0.75 (15)
```

The variability in egg length is smallest when the robin is the host. Note however that the numbers are all, except for meadow pipit, 14 or 15 or 16. The standard deviations are subject to large statistical uncertainty.

The footnote has code for the calculations used to create the table.[5]

*Degrees of freedom*

The denominator $n-1$ in the formula used to calculate $s$ is the number of degrees of freedom remaining after estimating the mean. With one data point, the sum of squares about the mean is zero, the degrees of freedom are zero, and no estimate of variability is possible.

In later chapters, standard deviation calculations will be based on the variation that remains after fitting a model (most simply, a line), to the data. Degrees of freedom are reduced by 1 for each model parameter that is estimated.

[5]
```
## SD of length, by species: data frame cuckoos (DAAG)
z <- with(cuckoos, sapply(split(length,species), function(x)c(sd(x),length(x))))
print(setNames(paste0(round(z[1,],2)," (",z[2,],")"),
```

### 1.3.4 Inter-quartile range and median absolute deviation

In a boxplot, as in Figure 1.2 in Section 1.1.5, the box spans the range between the lower and upper quartiles. The inter-quartile range is the width of the box. For data that are close to normally distributed $s \approx 0.75H$.

The median absolute deviation (MAD) is the median of the absolute deviations from the median. When multiplied by the value 1.4286, this statistic provides an approximately unbiased estimator for the standard deviation of a normally distributed population. The calculation is carried out by the function `mad()`. The value of the multiplier is set by the parameter `constant`. The MAD is a more robust estimate of the standard deviation than that based on the inter-quartile range meaning that if a normally distributed sample were to be contaminated by a few outlying erroneous observations, the MAD is least affected. See the Exercises for a simple demonstration of this.

### 1.3.5 A pooled standard deviation estimate

Suppose independent random samples have been taken from each of two populations that have the same standard deviation $\sigma$. If the respective sample sizes are $n_1$ and $n_2$, then the number of degrees of freedom for estimating the (common) variance is $n_1 + n_2 - 2$, after estimation of the possibly different population means. The pooled standard deviation estimator for $\sigma$ is:

$$s_p = \sqrt{\frac{\sum(x - \bar{x})^2 + \sum(y - \bar{y})^2}{n_1 + n_2 - 2}}.$$

Diagnostic checks that the common variance assumption is plausible, and/or transformations that will assist in making variances more homogeneous, can be crucial. A logarithmic transformation will often be effective in making variances more homogeneous.

*Elastic bands example*

A set of 21 elastic bands was randomly divided into two groups of 10 and 11. The amount of stretch under a weight of 1.35 kg was immediately measured for bands in the first group. The other bands were immersed in hot water at 65°C for four minutes, then left at air temperature for ten minutes, with their stretch then measured under a weight of 1.35 kg. The means and standard deviations for the two groups were:

```
sapply(DAAG::two65, function(x) c(Mean=mean(x), sd=sd(x))) |> round(2)
```

```
     heated ambient
Mean 253.50  244.09
sd     9.92   11.73
```

The pooled standard deviation estimate is $s = 10.91$, with 19 $(= 10 + 11 - 2)$ degrees of freedom. Since the separate standard deviations ($s_1 = 9.92$; $s_2 = 11.73$) are similar, the pooled standard deviation estimate is an acceptable summary of the variation in the data.

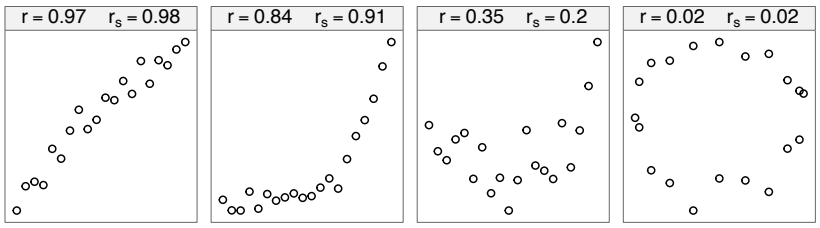



Figure 1.12 Different relationships between $y$ and $x$. The Pearson (linear) correlation is $r$, while the Spearman rank correlation is $r_h$.

### 1.3.6 Effect size

Effect sizes and effect size estimates offer a standardized measure of a difference that is of interest, relative to variation in the population or in the data. The *effect size* that will be discussed here is that known as Cohen's $d$, obtained by dividing the effect of interest by its standard deviation. For the elastic bands data listed above, the estimated effect size is:

```
setNames(diff(c(ambient=244.1, heated=253.5))/c(sd=10.91), "Effect size")
```

```
Effect size
     0.8616
```

If a drug (e.g., designed to reduce blood pressure) makes on average a very small difference relative to variation in a relevant defined population, its effect will be hard to demonstrate in clinical trials. Its usefulness for individual patients will be too much a matter of chance to justify its use in medical practice.

For more on effect measures that may be defined for use in a range of contexts, including with correlations, see Ben-Shachar et al. (2020), and especially the vignette that can be displayed by typing:

```
vignette('effsize', package='effsize')
```

### 1.3.7 Correlation

The Pearson or product–moment correlation is a summary measure of linear relationship. Calculation of a correlation should always be accompanied by a check that the relevant scatterplot shows a linear relationship. It can be helpful to add a smooth trend line. Separate distributions of the two variables should be roughly normal, or at least not highly skew. If the relationship between the variables appears to be nonlinear or the separate distributions are asymmetric, a Spearman rank correlation may be more appropriate.

The four panels in Figure 1.12 show four different types of relationship. In the first panel, the Pearson correlation is close is to 1.0, and the evident nonlinearity is not of much consequence. The magnitude of the correlation $r$, or of the squared correlation $r^2$, does not of itself indicate whether the underlying fit is adequate.

For the second panel, the Pearson correlation is $r = 0.84$, while the Spearman correlation is $r_h = 0.91$ and better captures the strength of the relationship. A linear fit is clearly inadequate.

The `cor()` function provides both measures. In addition, or as an alternative to the function `cor()`, note `cor.test()`. This returns a confidence interval and a test for no association. Subsection 2.2.4 describes the underlying assumptions.

A further possibility is to specify `method="kendall"` when `cor.test()` is called, giving the Kendall correlation. This is applicable, for example, where the same individuals are assessed by two different judges, and is related to the probability that the two judges will assign the same ranking to an individual.

Ways in which correlations may mislead are:

- There may be a subgroup structure in the data. If, for example, random samples are taken from each of a number of villages and the data are pooled, then any overall correlation at the level of individuals may reflect a correlation between village averages or a correlation between individuals within villages, or some of each. The two correlations may not be the same, or may even go in different directions. See Cox and Wermuth (1996).
- Any correlation between a constituent and a total amount is likely to be, in part at least, a mathematical artifact. In a study of an anti-hypertensive drug that hopes to determine whether the change $y - x$ is larger for those with higher initial blood pressure $x$. If $x$ and $y$ have similar variances then $y - x$ will have a negative correlation with $x$, whatever the influence of $x$.

While a correlation coefficient provides a single number summary of the relationship between $x$ and $y$, regression methods offer a richer framework for the examination of such relationships.

## 1.4 Distributions: quantifying uncertainty

The models that will be used in later chapters will typically have both deterministic (or *signal*) and random (or *noise*) components. The simplest type of model takes the form:

$$y = \mu + \varepsilon$$

where $\mu$ is a constant, and $\varepsilon$ is the random component. In the models that will be discussed in this section, $\mu$ will be the population mean, alternatively termed the *expected value*. More generally, and this will be a major focus in later chapters, $\mu$ can be replaced by a function of one or more variables and/or factors.

The random component is sometimes referred to as *error*. Both *noise* and *error* are technical terms. Use of the word *error* does not imply that there have been mistakes in the collection of the data, though mistakes can of course contribute to the variability.

The R *stats* package has many functions that return theoretical values of statistics for specific distributions. See `?Distributions` for details. The CRAN *Distributions* task view gives details of contributed R packages that extend the range of possibilities.

For each distribution, there are four functions, with names whose first letter is, respectively, `d` (**d**ensity), `p` (cumulative **p**robability), `q` (**q**uantile), and `r` (generate a **r**andom sample).

### 1.4.1 Discrete distributions

The usual starting points for discussing discrete distributions are the binomial and the Poisson. Examples for both now follow.

*Binomial:* Functions are `dbinom()`, `pbinom()`, and `qbinom()`, `rbinom()`
Values are 0, 1, 2, ..., $n$, where the argument `size` specifies $n$, and the argument `prob` specifies the probability $\pi$. The name Bernoulli is used for the special case when `size` $n = 1$. A Bernoulli random variable takes the value 1, with probability $\pi$ and 0, with probability $1 - \pi$. For example, if a fair 6-sided die is tossed once, the number of times a '2' appears is Bernoulli with $\pi = 1/6$.

A binomial random variable with `size` $n > 1$ is the sum of $n$ independent Bernoulli variables. For an example, suppose a sample of 10 items is taken from an assembly line that produces 15% defective items, on average. The probabilities of 0, 1, 2, ..., 10 defectives are (rounded to 3 decimal places):

```
## dbinom(0:10, size=10, prob=0.15)
```

```
    0     1     2     3     4     5     6     7     8     9    10
0.197 0.347 0.276 0.130 0.040 0.008 0.001 0.000 0.000 0.000 0.000
```

The probability of observing 4 or fewer defectives in a sample of size 10 is:

```
pbinom(q=4, size=10, prob=0.15)
```

```
[1] 0.9901
```

The function `qbinom()` goes in the other direction, from cumulative probabilities to numbers of events. It generates *quantiles*, a generalization of the more familiar term *percentiles*. To calculate a 70th percentile of the distribution of the number of defectives in a sample of 10, with Pr[defective=0.15], type:

```
qbinom(p = 0.70, size = 10, prob = 0.15)
```

```
[1] 2
```

```
## Check that this lies between the two cumulative probabilities:
## pbinom(q = 1:2, size=10, prob=0.15)
```

The following generates a random sample of 15 values from a binomial distribution with `p`=0.5 and `size`=4.

```
rbinom(15, size=4, p=0.5)
```

```
 [1] 0 2 3 3 3 2 2 3 1 2 2 1 3 1 2
```

*The Poisson distribution:* `dpois()`, `ppois()`, `qpois()`, `rpois()`
Values are 0, 1, 2, ..., where the argument `lambda` specifies the theoretical mean.

The Poisson distribution is often used to model the number of events that occur in a certain time interval, or the numbers of defects observed in for example manufactured products. The distribution has a single parameter $\lambda$ (the Greek letter 'lambda') which coincides with the mean or expected value.

As an example, consider a population of raisin buns for which there are an average of 3 raisins per bun, i.e., $\lambda = 3$. The possible numbers of raisins are 0, 1, 2, .... Under the Poisson model, which assumes that raisins appear independently in different buns, probabilities for numbers of raisins in a bun are:

```
## dpois(x = 0:8, lambda = 3)
```

```
     0      1      2      3      4      5      6      7      8
0.0498 0.1494 0.2240 0.2240 0.1680 0.1008 0.0504 0.0216 0.0081
```

The functions `ppois()`, `qpois()`, and `rpois()` can be used in exactly the same way as binomial family functions. Thus, in the raisin buns example, the probability of more than 8 raisins (equals 0.0038) can be calculated in any of the following ways:

```
1 − ppois(q = 8, lambda = 3)
ppois(q=8, lambda=3, lower.tail=FALSE)  ## Alternative
1−sum(dpois(x = 0:8, lambda = 3))       ## Another alternative
```

The following simulates numbers of raisins in 20 raisin buns, where the expected number of raisins per bun is 3:

```
raisins <− rpois(20, 3)
raisins
```

```
 [1] 4 2 3 3 1 4 3 0 6 3 3 2 4 2 5 4 3 1 2 2
```

Note that the average of these values is 57/20 which is, not accidentally, near the expected value 3.

In the practical situation, raisins may tend to stick together, or the mixing may not be even through the baking mixture. Something more sophisticated than a simple Poisson model may be required.

*Initializing the random number generator*

In a number of our earlier examples, we have seen calls to the function `set.seed()`. The seed for the random number generator is stored in the workspace, in a hidden variable (`.Random.seed`) that changes whenever there has been a call to the random number generator. Where it is required to repeat the same sequence on successive occasions, the function `set.seed()` can be used to set an initial seed and thus ensure the same sequence. The following uses `set.seed()` to make the call to `rbinom(10, size=1, p=0.5)` thus reproducible:

```
set.seed(23286)  # Use to reproduce the sample below
rbinom(15, size=1, p=0.5)
```

```
 [1] 0 0 0 0 1 0 1 0 1 1 1 1 1 0 0
```

When the workspace is saved, `.Random.seed` is stored in the workspace. When the workspace is loaded again, the value stored in the workspace will be restored. Any new simulations will then be independent of those prior to saving the workspace.

*Means and variances*

*Binomial:* In a sample of 10 manufactured items from a population where 15% are defective, we expect to see 1.5 defectives on average. More generally, the *expected value* or *mean* of a binomial random variable with `size` $= n$ and probability `prob` $= \pi$ is $n\pi$. The variance of a binomial random variable is $n\pi(1-\pi)$.

For the number of defectives in our sample of 10 items with $\pi = 0.15$, the variance is $10 \times 0.15 \times 0.85 \simeq 1.275$.

In practice, the probability of a defective may change from one sample item to the next, perhaps because of unevenness in the quality of the raw material. Thus, there may be a tendency for defects to cluster together. This can lead to a distribution that has a larger variance relative to the mean than predicted by the binomial distribution.

*Poisson:* The variance of a Poisson random variable is equal to its mean, often denoted $\lambda$. Thus if the number of raisins in a bun is Poisson with mean $\lambda = 3$, the variance is also 3.

### *1.4.2 Continuous distributions*

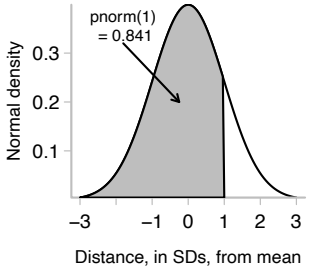
Models for measurement data usually take the form of a *continuous* distribution. The probability that a measurement takes a particular value is then 0. Instead, the distribution of a continuous random variable is modeled by its density function. The area under the density curve between $x = a$ and $x = b$ gives the probability that the random variable lies between those limits. The total area under the density curve is 1.

*The normal distribution* : The *normal*, or Gaussian, distribution, which has the bell-shaped density curve pictured in Figure 1.13, is widely used to model continuous measurement data. A transformation may be required, as in Figure 1.5 in Subsection 1.2.3, for the normal model to be useful. The density (height) of the curve is given as a function of the distance from the mean.

The density curve shown in Figure 1.13 is for a *standard* normal distribution that has mean 0 and standard deviation 1. Replacing each value $z$ in a population of standard normal variates by $\mu + \sigma z$ changes the mean to $\mu$ and the variance to $\sigma$.

Code to plot the normal density function is:

```
## Plot the normal density, in the range -3 to 3
z <- pretty(c(-3,3), 30)   # Find ~30 equally spaced points
ht <- dnorm(z)             # Equivalent to dnorm(z, mean=0, sd=1)
plot(z, ht, type="l", xlab="Normal variate", ylab="Density", yaxs="i")
# yaxs="i" locates the axes at the limits of the data
```




Calculations using `pnorm()`

```
                        Prob
pnorm(0)               0.500
pnorm(1)               0.841
pnorm(-1.96)           0.025
pnorm(1.96)            0.975
pnorm(1.96, mean=2)    0.484
pnorm(1.96, sd=2)      0.836
```

Figure 1.13 A plot of the normal density. The horizontal axis is labeled in standard deviations (SDs) distance from the mean. The area of the shaded region is the probability that a normal random variable has a value less than one standard deviation above the mean.

Functions for calculations relating to the normal distributions are `dnorm()` (used for plotting the density curve in Figure 1.13), `pnorm()`, `qnorm()`, and `rnorm()`. The function `pnorm()` calculates the cumulative probability, i.e., the area under the curve up to the specified ordinate or $x$-value. Thus, the probability that a normal variate with mean 0 and standard deviation 1 is less than 1.0 is `pnorm(1)` = 0.841. This corresponds to the area of the shaded region in Figure 1.13. To obtain the probability in the upper tail, supply the argument `lower.tail=FALSE`.

The function `qnorm()` computes normal quantiles. Thus, the 90th percentile is:

```
qnorm(.9)          # 90th percentile; mean=0 and SD=1
```

```
[1] 1.282
```

The footnote has additional examples.[6]

*Other continuous distributions:* A simple model is the *uniform distribution*, for which an observation is equally likely to take any value in a given interval. The probability density is constant over that interval.

The *exponential distribution* gives high probability density to positive values lying near 0, with the density decaying exponentially as the values increase. It is the simplest of a class of distributions that have been used to model times between arrivals of customers to a queue. The exponential is a special case of the chi-squared distribution which arises, for example, when checking for dependence between row and column numbers in contingency tables.

### *Different ways to represent distributions*

As noted in Section 1.1.5, the boxplot defaults are set so that 1% of values that are drawn at random from a normal distribution will on average be flagged as possible outliers. If the distribution is not symmetric, more than 1% of points are likely to

[6]
```
## Additional examples:
setNames(qnorm(c(.5,.841,.975)), nm=c(.5,.841,.975))
qnorm(c(.1,.2,.3))   # -1.282 -0.842 -0.524  (10th, 20th and 30th percentiles)
qnorm(.1, mean=100, sd=10)  # 87.2 (10th percentile, mean=100, SD=10)
```

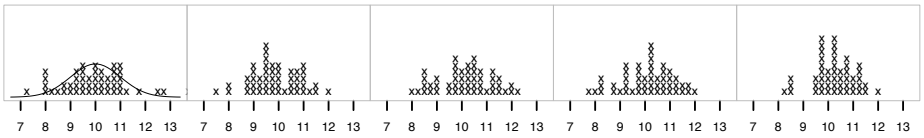



Figure 1.14 Each panel shows a simulated distribution of 50 values from a normal distribution with mean = 10 and sd = 1. The underlying theoretical normal curve is overlaid on the far left panel.

lie outside the whiskers, at the lower end if the distribution is left-skewed or at the upper end if the distribution is right-skewed. If the distribution is symmetric, but "heavy-tailed", then we would expect more than 1% of the values to plot beyond the boxplot whiskers with no preference for either end.

*Generating simulated samples from the normal and other continuous distributions*

The function `rnorm()` generates random samples of values from the normal distribution. To generate 10 random values from a standard normal distribution, we type:

```
options(digits=2)  # Suggest number of digits to display
rnorm(10)          # 10 random values from the normal distribution
```

```
 [1]  0.42  1.15 -1.40  0.18  0.21 -1.13 -0.69  0.29  0.23 -0.21
```

Figure 1.14 demonstrates the use of simulation to indicate the extent of sample-to-sample variation in histogram summaries of the data, when five independent random samples of 50 values are taken from a normal distribution.[7]

Calculations for other distributions, for example `runif()` to generate uniform random numbers, or `rexp()` to generate exponential random numbers, follow the same pattern.

```
runif(n = 20, min=0, max=1) # 20 numbers, uniform distn on (0, 1)
rexp(n=10, rate=3)          # 10 numbers, exponential, mean 1/3.
```

Exercises at the end of this chapter explore further possibilities.

Histograms are not a good basis for deciding whether sample values are consistent with a normal distribution. A more effective tool is the normal quantile-quantile plot.

### 1.4.3 Graphical checks for normality

In a normal quantile-quantile plot the sorted data values are plotted against the expected ordered values for a normal distribution. Thus for data from a normal distribution, the points should scatter about a straight line.

The `DAAG::pair65` dataset has data from an experiment that tested the effect of heat on the stretchiness of elastic bands. Following an initial check for 'stretchiness',

[7]
```
## The following gives conventional histogram representations:
set.seed (21)        # Use to reproduce the data in the figure
df <- data.frame(x=rnorm(250), gp=rep(1:5, rep(50,5)))
lattice::histogram(~x|gp, data=df, layout=c(5,1))
```

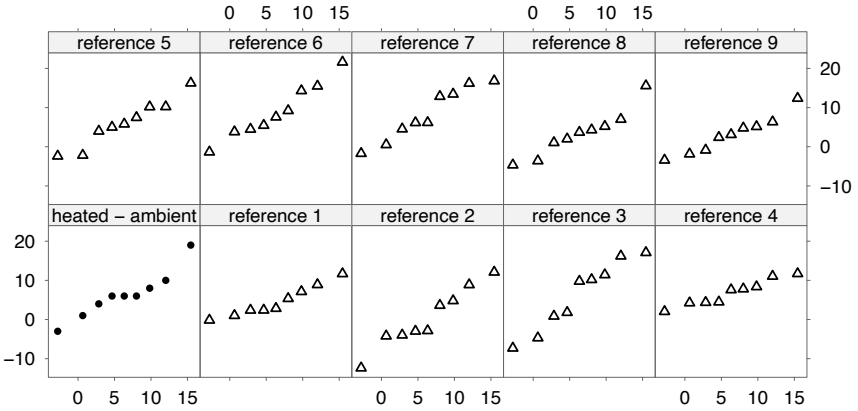



Figure 1.15 The lower left panel is the normal quantile-quantile plot for heated–ambient differences. Remaining panels show normal quantile-quantile plots for samples of nine numbers from a normal distribution.

the bands were arranged into nine pairs, such that the two members of a pair appeared similarly 'stretchy'. One member of each pair, chosen at random, was placed in hot water (60–65 °C) for four minutes, while the other was left at ambient temperature. After a wait of about ten minutes, the amounts of stretch, under a 1.35 kg weight, were recorded. The following were the amounts of stretch, and the differences, for each pair:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| heated | 244 | 255 | 253 | 254 | 251 | 269 | 248 | 252 | 292 |
| ambient | 225 | 247 | 249 | 253 | 245 | 259 | 242 | 255 | 286 |
| heated-ambient | 19 | 8 | 4 | 1 | 6 | 10 | 6 | -3 | 6 |

The normal quantile-quantile plot for these differences is in the lower left panel of Figure 1.15. The other seven plots are for samples (all of size 9) of simulated random normal values. As judged against these plots, the distribution of the sample differences appears consistent with normality. The code is:

```
## Normal quantile-quantile plot for heated-ambient differences,
## compared with plots for random normal samples of the same size
plt <- with(DAAG::pair65, DAAG::qreference(heated-ambient, nrep=10, nrows=2))
update(plt, scales=list(tck=0.4), xlab="")
```

Displays in the style of Figure 1.15 help to calibrate the eye, giving a sense of the nature and extent of departures from linearity that can be expected in random normal samples of the specified size, here 9. The process should be repeated several times. With a sample size of just 9, large departures from a linear pattern will be needed to provide convincing evidence of non-normality.

The base graphics function `qqnorm()` may be used to obtain such plots one at a time. Specify, e.g., `qqnorm(rnorm(9))`.

The methodology extends to allow a comparison of ordered sample values with expected ordered values for any distribution that is of interest. See `?qqplot`.

In practice exact normality is unlikely, and is for most inferential purpose not

required. Concern arises when there are gross departures from normality, such as skewed or heavy-tailed data. In small samples (e.g., of the order of 10 or less), large departures from normality, of an extent that affect the validity of results, will frequently go undetected. It is typically necessary to rely on sources of evidence that are external to the data, including where possible previous experience with similar data.

### 1.4.4 Population parameters and sample statistics

*Parameters*, such as the mean ($\mu$) or standard deviation ($\sigma$), numerically summarize various aspects of a population. Such parameters are usually unknown and are estimated using *statistics* that are calculated for a sample (which should be a random sample) from the population. Thus the sample mean is used to estimate the population mean, and the sample standard deviation estimates the population standard deviation.

Other commonly used statistics are the proportion, variance, median, the quartiles, the extremes, the slope of a regression line, and the correlation coefficient. Each may be used as an estimate of the corresponding population parameter.

To what extent is it possible to use the one sample that we have as the basis for some wider generalization? If another sample were taken, would it very likely give a similar result? The sampling distribution plays a central role in assessing the extent to which results can be generalized beyond the one available sample.

The sampling distribution of the mean, and extensions to the sampling distributions of regression coefficients, underpin a large part of the methodology described in this text. They allow the extension of results that apply to random samples from normal populations, for use with populations that show various types of deviations from normality. Is is then important to assess, in any particular case, the extent to which the normal distribution result can be trusted? Recourse to simulation allows a value of the statistic of interest to be calculated from each of a multiple repeated random samples from the relevant distribution. The collection of simulated sample statistics can be summarized by a distribution called the *sampling distribution* of the statistic.

The information on the sampling distributions of the statistics of interest offers an *empirical*, i.e., sampling based, alternative to the direct use of statistical theory. It can be used when theoretical results are not available or are of uncertain relevance. Here, it will be used to investigate what the relevant theoretical result – the Central Limit Theorem – may mean in practice.

In practice, even if the main part of the population distribution appears symmetric, there will often be occasional aberrant values. Such aberrant values do, perhaps fortunately, work in a conservative direction – they reduce the chances that genuine differences will be detected. A take-home message is that, especially in small samples, the probabilities and quantiles can be quite imprecise. They are rough guides, intended to assist in making a judgment.

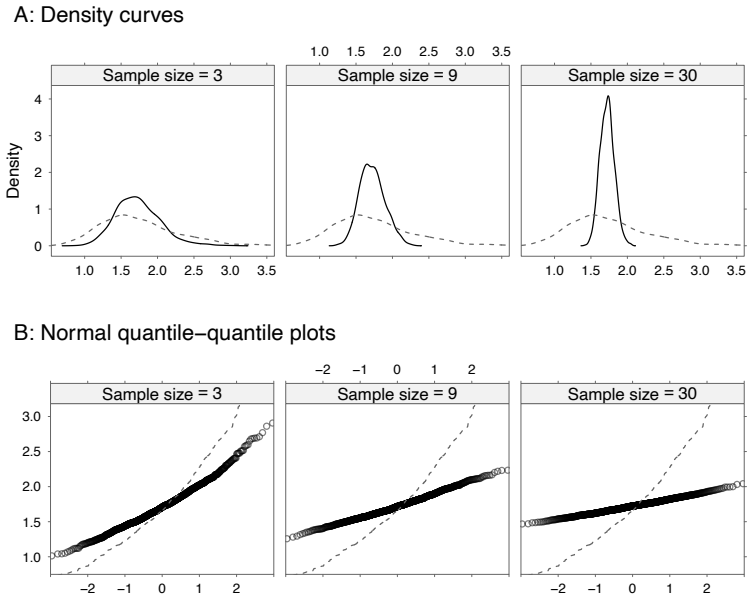



Figure 1.16 Data values are from simulations of the sampling distribution of the mean, for a mildly skew distribution. Panel A shows density curves, while Panel B shows normal quantile-quantile plots. The plot for the population, repeated in each panel, is shown as a dashed curve. Simulated sampling distributions, each from 1000 simulations, are shown as solid curves. The three panels show the plots for samples of respective sizes 3, 9 and 30.

### *The sampling distribution of the mean*

The standard deviation of the sampling distribution of the mean is termed the *standard error of the mean* (SEM). If the population mean is $\mu$ and the standard deviation is $\sigma$, then given independent random samples

$$\text{SEM} = \frac{\sigma}{\sqrt{n}}$$

A consequence of the "Central Limit Theorem" is that for a statistic such as a mean or a regression slope, an averaged value may have a sampling distribution that is close to normal, even if the underlying population distribution is clearly not normal.[8]

In practice, if the sampled population has a distribution that is approximately normal, the average of samples of size $n = 3$ or $n = 4$ can usually, for most practical purposes be treated as coming from a normal distribution. Skewed or heavy-tailed distributions will require larger (perhaps much larger) samples than distributions that have lighter tails than the normal.

Figure 1.16 shows density curves and normal quantile-quantile plots for simulated

[8] More precisely, the distribution of the sample mean approximates the normal distribution with increasing accuracy, as the sample size increases, assuming values are independent, and the population standard deviation is finite. There are similar results for many other sample statistics.

sampling distributions of the mean (sample sizes 3, 9, and 30), for sample values from a mildly skewed distribution. As the sample size increases from $n = 3$ to $n = 9$ to $n = 30$, the density curves become more nearly symmetric with a decreasing standard deviation, while the normal quantile-quantile plots become more nearly linear, with a reduced slope. The reductions in slope in Panel B reflect the reduced SEMs, from $\frac{\sigma}{\sqrt{3}}$ to $\frac{\sigma}{\sqrt{9}}$ to $\frac{\sigma}{\sqrt{30}}$. Even for a sample size of 3, much of the skewness has gone.

Code for the plots in Figure 1.16A is:

```
library ( lattice )
## Generate n sample values; skew population
sampfun = function(n) exp(rnorm(n, mean = 0.5, sd = 0.3))
gph <- DAAG::sampdist(sampsize = c(3, 9, 30), seed = 23, nsamp = 1000,
  FUN = mean, sampvals=sampfun, plot.type = "density")
```

For Figure 1.16B, replace `plot.type="density"` in the call to `DAAG::sampdist()` with `plot.type="qq"`. The skewness of the population can be increased by increasing `sd` in the call to `sampfun()`. For example, try `sd = 0.8`.

### * *The sampling distribution of s*

It is usually simplest to work with the sampling distribution of:

$$\frac{(n-1)s^2}{\sigma^2}.$$

Under the independently and identically distributed (iid) normal assumption, this quantity has a chi-squared distribution with $n-1$ degrees of freedom, independently of $\bar{x}$. Its distribution is close to normal for very large $n$. Exercise 20 in Subsection 2.12 investigates transformations that improve the approximation to normality.

For non-normal data, the distribution of $\frac{(n-1)s^2}{\sigma^2}$ will differ from the chi-squared. Simulation can be a useful mechanism for investigating the extent of the difference that can be expected for specific patterns of departure from normality.

### * *The standard error of the median*

For data from a normal distribution, the standard error of the median can be calculated using is

$$\text{SE}_{\text{median}} = \sqrt{\frac{\pi}{2}}\frac{s}{\sqrt{n}} \approx 1.25\frac{s}{\sqrt{n}}.$$

This quantity is about 25% greater than the standard error of the mean. For the `cuckoos` data, median length of eggs in nests of wrens is 21.0mm, with standard error of the median equal to 0.244.

The median is often employed when the distribution is positively skewed. Then the median is less than the mean, and the standard error formula given above is not applicable.

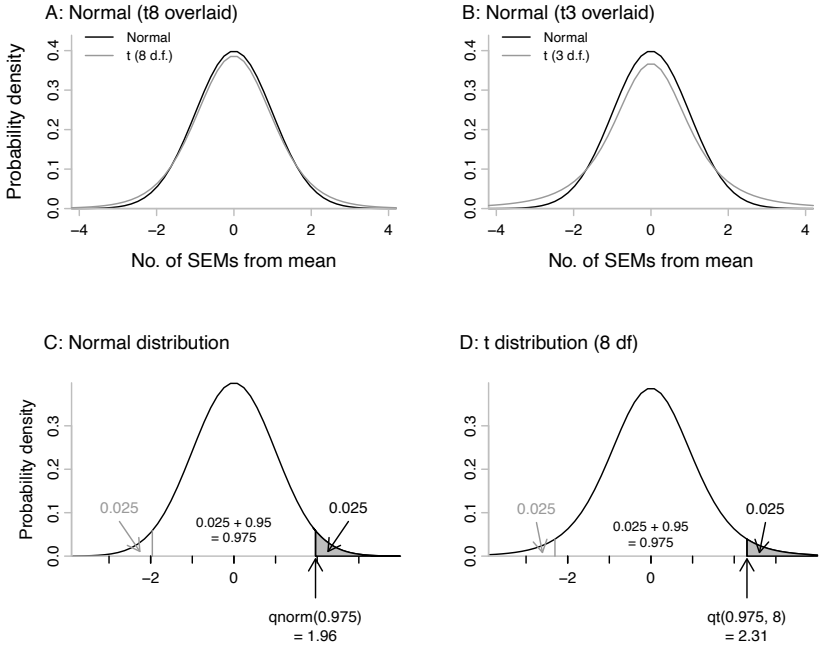



Figure 1.17 Panels A and B overlay the density for a normal distribution with the density for a $t$-distribution, in Panel A $t$ with 8 degrees of freedom, and in Panel B $t$ with 3 degrees of freedom. Panels C and D show the endpoints of symmetrically placed regions that enclose 95% of the probability, in Panel C for a normal distribution, and in Panel D for $t$ with 8 degrees of freedom. In each panel, the upper 2.5% of the area under the curve is shaded in gray.

*Simulation in learning and research*

In statistical theory and in practice, simulation is widely used, when analytical results are not available. It can be a useful tool in teaching and learning. The R package *animation* (Xie and Cheng, 2008) has a variety of simulations that are intended for use in teaching or self-instruction.

### *1.4.5 The $t$-distribution*

Above, it was noted that, under Central Limit Theorem Conditions, the sampling distribution of the mean could be approximated by normal distributions with standard deviations given thus:

One-sample case: The standard deviation of $\bar{x}$ is termed *standard error of the mean* (SEM), and equals $\frac{\sigma}{\sqrt{n}}$.

Two-sample case: The standard deviation of $\bar{x_1} - \bar{x_2}$ is termed the *standard error of difference* (SED). Assuming a common variance, it equals $\sigma(\frac{1}{n_1} + \frac{1}{n_2})$

In the usual case where $\sigma$ has to be replaced by an estimate $s$, the relevant stan-

Table 1.2 *Comparison of normal distribution endpoints (multipliers for the SEM) with the corresponding t-distribution endpoints on 8 degrees of freedom.*

| Probability enclosed between limits | Cumulative probability | Number of SEMs | |
|---|---|---|---|
| | | Normal distribution | *t*-Distribution (8 df) |
| 68.3% | 84.1% | 1.0 | 1.07 |
| 95% | 97.5% | 1.96 | 2.31 |
| 99% | 99.5% | 2.58 | 3.36 |
| 99.9% | 99.95% | 3.29 | 5.04 |

dardized differences are:

$$\text{One-sample:} \quad t = \frac{(\bar{x} - \mu)}{SEM} \text{ where } SEM = \frac{s}{\sqrt{n}} \tag{1.1}$$

$$\text{Two-sample:} \quad t = \frac{\bar{x_1} - \bar{x_2} - \mu}{SED} \text{ where } SED = s\sqrt{\frac{1}{n_1} + \frac{1}{n_2}} \tag{1.2}$$

The statistic $t$ expresses the difference of interest in standard error units. In the one-sample case the sampling distribution is $t$ with $n-1$ degrees of freedom. In the two-sample case, assuming a common variance, degrees of freedom are $n_1 + n_2 - 2$. Figures 1.17 A and B show the density curve for a normal distribution overlaid with those for $t$-distributions with 8 and 3 degrees of freedom respectively. Replacing $\sigma$ by $s$ introduces a source of uncertainty additional to that in $\bar{x}$, an uncertainty that is larger as the degrees of freedom for $s$ are smaller.

Figures 1.17 C and D compare area under the curve calculations for a $t$-distribution with those for a normal distribution. The difference is most obvious in the tails. In the terminology of Subsection 1.4.2, the $t$-distribution is heavy-tailed – heavier for smaller than for larger degrees of freedom. The 'limiting' distribution as degrees of freedom increase is the standard normal.

Changing from a normal distribution to a $t$-distribution with 8 degrees of freedom leads to a small change, from 1.0 to 1.07, in the $t$-distribution quantile for enclosing the central 68.3% of the area. There is a substantial difference, an increase from 1.96 to 2.31, for enclosing 95% of the area. The $t$-distribution underpins a wide range of statistical analysis approaches, both in frequentist and in Bayesian methodology. It will feature extensively in the next chapter.

Table 1.2 compares the multipliers for a normal distribution with those for an 8 degrees of freedom $t$-distribution, for several different choices of area under the curve. Examples of the calculations required are:

```
qnorm(c(0.975,0.995), mean=0)    # normal distribution
```

```
[1] 2.0 2.6
```

```
qt(c(0.975, 0.995), df=8)        # t-distribution with 8 d.f.
```

```
[1] 2.3 3.4
```

The sampling distribution of the $t$ statistic will be the starting point for the confidence interval and hypothesis testing approaches of Section 1.6.

### *1.4.6 The likelihood, and maximum likelihood estimation*

Consider the model

$$y_i = \mu + \epsilon_i,\ i = 1, 2, \ldots, n$$

where $\mu$ is an unknown constant, and where the errors $\epsilon$ are assumed to be independent and normally distributed with mean 0 and variance $\sigma^2$. More generally, and this will be the main focus of interest in the later text, the elements $\mu_i$ of $\mu$ may themselves be functions of explanatory variables and/or factors.

The probability density for the $i$th $y$-value is normal with mean $\mu$ and variance $\sigma^2$. Because of the independence assumption, the probability density of the entire sample of $y$'s is simply the product of these normal densities. This product, when viewed as a function of $\mu$ and $\sigma$, has the name *likelihood*. The *maximum likelihood estimates* are the values of $\mu$ and $\sigma$ which maximize this function. A calculus argument can be used to show that the estimates are $\bar{y}$ and $s\sqrt{(n-1)/n}$. Thus, the usual estimator of the standard deviation differs slightly from the maximum likelihood estimator. The denominator in the usual variance estimate is the number of degrees of freedom ($n-1$ in this case), while it is $n$ for the maximum likelihood estimate. This difference is negligible in large samples.

In practice, it is usually most convenient to work with the log-likelihood, rather than with the likelihood. Instead of multiplying densities to obtain the likelihood, the logarithms of the densities are added to obtain the log-likelihood Maximizing on the log scale leads to exactly the same estimates as on the original scale.

Where a model $m_1$ with maximum likelihood $L_1$ is obtained from a model $m_0$ with maximum likelihood $L_0$ by fitting an additional term or terms, the log-likelihood ratio $\log(\lambda_{lr}) = \log(\frac{L_1}{L_0})$ may be used to compare them. As $L_1$ has more parameters to estimate, $L_1 \geq L_0$.

Exact expressions for the distribution of the log-likelihood ratio $\lambda_{lr}$ are not in general available. Where the sample size $n$ is large enough, use can be made of the result that in the large sample limit, $2\log(\lambda_{lr}) = 2(\log(L1) - \log(L_0)$ is distributed as chi-squared with degrees of freedom equal to the number of additional parameters estimated in the more complex model.

## 1.5 Simple forms of regression model

In this introductory account, the primary focus of attention will be models for data that can be displayed as a scatterplot. By convention, the $x$-variable, plotted on the horizontal axis, has the role of explanatory or predictor variable. The $y$-variable, by convention plotted on the vertical axis, has the role of response or outcome variable. Many of the issues that arise for these simple regression models are fundamental to all regression methods. Size and shape data, discussed in Section 2.5.8, is one of a number of applications that raise their own specific issues.

Scrutiny of the scatterplot should precede regression calculations. It can be useful to compare the fitted line with a fitted smooth curve such as will be a major focus in Chapter 4, as a help in judging whether a straight line is appropriate.

### 1.5.1 Line or curve?

The data shown in Figure 1.18 is from an experiment where different weights of roller were rolled over different parts of a lawn and the depression noted (data are from Stewart et al., 1988). The data are;

```
roller <- DAAG::roller
t(cbind(roller, "depression/weight ratio"=round(roller[,2]/roller[,1],2)))
```

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| weight | 1.9 | 3.10 | 3.3 | 4.8 | 5.3 | 6.1 | 6.4 | 7.6 | 9.8 | 12 |
| depression | 2.0 | 1.00 | 5.0 | 5.0 | 20.0 | 20.0 | 23.0 | 10.0 | 30.0 | 25 |
| depression/weight ratio | 1.1 | 0.32 | 1.5 | 1.0 | 3.8 | 3.3 | 3.6 | 1.3 | 3.1 | 2 |

Write $(x_i, y_i)$ $(i = 1, 2, \ldots 10)$ for the 10 (`weight`, `depression`) pairs. Three possible models, all modeling the dependent value as the sum of a deterministic or *signal* component and a *noise* component $\varepsilon_i$ for each $(x_i, y_i)$ pair, are:

1. $\dfrac{y_i}{x_i} = \mu + \varepsilon_i$ (the signal is a ratio that is constant across observations).
2. $y_i = \beta x_i + \varepsilon_i$ (the "line through the origin" model).
3. $y_i = \alpha + \beta x_i + \varepsilon_i$ (allows an arbitrary line).

Model 3, which fitted a line using the linear modeling function `lm()`, gave the fitted line in Figure 1.18A. This used a least squares approach to calculate the intercept and slope, i.e., minimize

$$\sum_{i=1}^{10} \varepsilon_i^2$$

The strict requirement for calculation of standard error and $p$-value information is the *iid normal* assumption that the $\varepsilon_i$ are independently and identically distributed as normal variables with mean 0 and variance $\sigma^2$. Equivalently, it is assumed that, given $x_i$, the responses $y_i$ are sampled independently from a normal distribution with mean $\alpha + \beta x_i$. The output from `lm()` includes an estimate of the variance $\sigma^2$.

Independence implies that the size and sign (negative or positive) of one element give no information on the likely size and sign of any other element. It implies that elements are uncorrelated. With different assumptions (e.g., a sequential correlation between successive data points), the standard errors will be different.

*Using models to predict*

Interest may be in the rate of increase of depression with increasing roller weight. For models 2 and 3 above, the slope of the line ($\beta$) is then the focus of interest. Alternatively, or additionally, the aim may be the prediction of values for new data. Different models may serve different purposes.

A data-based assessment of how results may generalize to other lawns would require data from multiple lawns, then using a modeling approach that (such as will be discussed in Chapter 7) that accounts for between-lawn variation as well as for the within-lawn variation on which the present data gives limited information. The mechanical properties of the soil would usefully feature in such a model.

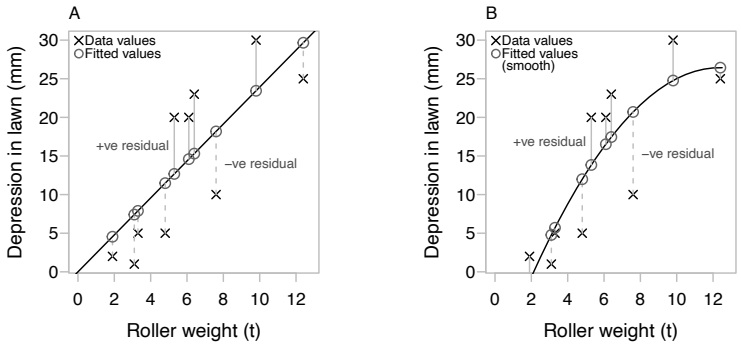



Figure 1.18 In Panel A, a line has been fitted, while Panel B has used the function `lowess()` to fit a smooth curve. Residuals (the 'rough') appear as vertical lines. Positive residuals are solid lines, while negative residuals are dashed. For code used to create the figures, see the web page for the book.

*Which model is best — line or curve?*

The fitting of a curve, as in 1.18B, can help in judging whether a line, as in Figure 1.18A, really is appropriate. Here there is just one point that seems to be causing the line, and the fitted curve, to bend down. As might be expected, a formal statistical analysis suggests that the curve in Figure 1.18B is an 'over-fit'. See further Exercise 2 in Chapter 4.

### *1.5.2 Fitting models – the model formula*

The same model formula `depression ~ weight` can be used both with the function `lm()` to specify the fitting of a line to the data of Figure 1.18, and with the function `plot()` to specify the $y$- and $x$-variables for a plot:

```
## Fit line - by default, this fits intercept & slope.
roller.lm <- lm(depression ~ weight, data=DAAG::roller)
## Compare with the code used to plot the data
plot(depression ~ weight, data=DAAG::roller)
## Add the fitted line to the plot
abline( roller.lm )
```

In the formula, `weight` is the *predictor* or *explanatory* variable, while `depression` is the *response*.[9]

*Model objects*

The model object, saved above as `roller.lm`, is a list. The element names give clues on what the elements contain:

```
roller.lm <- lm(depression ~ weight, data=DAAG::roller)
names(roller.lm)      # Get names of list elements
```

[9] `## For a model that omits the intercept term, specify`
`lm(depression ~ 0 + weight, data=roller)  # Or, if preferred, replace `0` by `-1``

```
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "xlevels"       "call"          "terms"         "model"
```

Most information that is commonly required from model objects can be obtained by the use of an *extractor* function. Note in particular:

```
coef( roller.lm )          # Extract coefficients
summary(roller.lm)         # Extract model summary information
coef(summary(roller.lm))   # Extract coefficients and SEs
fitted ( roller.lm )       # Extract fitted values
predict ( roller.lm )      # Predictions for existing or new data, with SE
                           # or confidence interval information if required.
resid ( roller.lm )        # Extract residuals
```

Information in the list object `roller.lm` can be accessed directly, thus:

```
roller.lm $coef            # An alternative is roller.lm[["coef"]]
```

Use of an extractor function, when available, is preferable.

The default summary information for the `roller.lm` model object is:

```
print(summary(roller.lm), digits=3)
```

```

Call:
lm(formula = depression ~ weight, data = DAAG::roller)

Residuals:
   Min     1Q Median     3Q    Max
 -8.18  -5.58  -1.35   5.92   8.02

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    -2.09       4.75   -0.44   0.6723
weight          2.67       0.70    3.81   0.0052

Residual standard error: 6.7 on 8 degrees of freedom
Multiple R^2:  0.644,    Adjusted R^2:   0.6
F-statistic: 14.5 on 1 and 8 DF,  p-value: 0.00518
```

The intercept of the fitted line is $a = -2.09$ (SE $= 4.75$), while the estimated slope is $b = 2.67$ (SE $= .70$). The $p$-value for the slope (testing the null hypothesis that $\beta =$ true slope $= 0$) is small, consistent with the evident linear trend. The $p$-value for the intercept (testing $\alpha = 0$) is 0.67, i.e., the difference from zero may well be random sampling error. Thus, consistently with the intuition that depression should be proportional to weight, the intercept term should be dropped. We leave this as an exercise.

The standard deviation of the noise term, here identified as the residual standard error, is 6.735. We defer comment on $R^2$ and the F-statistic until Subsection 2.5.2.

### *Residual plots*

The residuals allow limited checks on model assumptions. (Here, the dataset is not large enough to allow detection of any except extreme departures.) The function

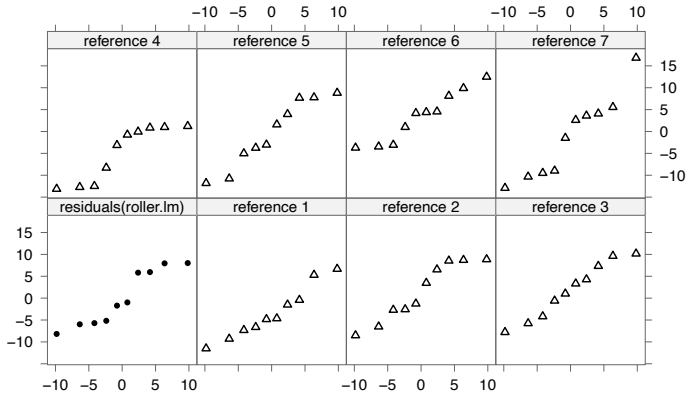



Figure 1.19 The normal quantile-quantile plot for the regression of Figure 1.18A is shown in the lower left panel. Other panels show normal quantile-quantile plots for computer-generated normal data.

`plot()`, used with an `lm` object as argument, has as its default the display of four standard diagnostic plots. Its use will be demonstrated for the example in the next subsection.

The third of the default plots is the normal quantile-quantile plot. As an aid to judging whether this differs from a straight line by more than can be expected as a result of random generation, this is suitably compared with plots for computer-generated random normal data. Figure 1.19 uses the function `DAAG::qreference()` to display 7 such plots alongside the normal quantile-quantile plot for the residuals from the model. Such plots provide the eye with a reference standard.

```
## Normal quantile-quantile plot, plus 7 reference plots
DAAG::qreference(residuals(roller.lm), nrep=8, nrows=2, xlab="")
```

*Simulation of regression data*

It is often useful to repeatedly simulate data from a fitted model, then re-fit to each new set of simulated data. This provides a check on variation under such repeated simulation. The function `simulate()` can be used for this purpose.

Thus, to obtain 10 sets of simulated outcome values for the model that was fitted to the `roller` data, do:

```
roller.lm <− lm(depression ∼ weight, data=DAAG::roller)
roller.sim <− simulate(roller.lm, nsim=20) # 20 simulations
```

The object `roller.sim` is a data frame with one column for each of the 20 sets of simulated values of `depression`. These are obtained from the column of fitted values by adding normal random deviates with mean 0 and residual standard deviation as given by `sigma(roller.lm)`. To visualize this output, try:

```
with(DAAG::roller, matplot(weight, roller.sim , pch=1, ylim=range(depression)))
points(DAAG::roller, pch=16)
```

Table 1.3 *Details are for the first three rows of the model matrix, for calculations of fitted values and residuals, in the straight line fit to the lawn roller data.*

| Model matrix | | | | |
|---|---|---|---|---|
| Multiply by $-2.09$ | Multiply by 2.67 | Add multiplied values to give fitted value $\widehat{y}$ | Observed y | Residual = $y - \widehat{y}$ |
| 1 | 1.9 | $-2.09 + 2.67 \times 1.9 = 2.98$ | 2 | $2 - 2.98$ |
| 1 | 3.1 | $-2.09 + 2.67 \times 3.1 = 6.18$ | 1 | $1 - 6.18$ |
| 1 | 3.3 | $-2.09 + 2.67 \times 3.3 = 6.71$ | 5 | $5 - 6.71$ |
| ... | ... | ... | ... | ... |

### 1.5.3 The model matrix in regression

For the use of `lm()` and related functions, the model matrix has a crucial role.

In straight line regression, the model or $X$ matrix has two columns – a column of 1s and a column that holds values of the explanatory variable $x$. The fitted straight line is model is

$$\begin{aligned}\widehat{y} &= a + bx \\ &= 1 \times a + x \times b\end{aligned}$$

The first two columns of Table 1.3 show (first 3 rows only) the model matrix used in fitting a straight line model to the lawn roller data. To extract the model matrix from the model object, use the extractor function `model.matrix()`, thus:

```
model.matrix(roller.lm)
## Specify coef(roller.lm) to obtain the column multipliers.
```

For each row, fitted values are multiple of the value 1 in the first column (here, to two decimal places, -2.09), another multiple (here 2.67) of the value in the second column, and adding them.

For the simpler (no intercept) model $\widehat{y} = bx$. The model matrix then has only a single column, holding the values of $x$.

*From straight line regression to multiple regression*

Above, we considered a model that had a `weight`$^2$ explanatory term as well as a `weight` term. In principle, there is no limit to the terms that can be added. Much of the content of following chapters will be an exploration of the possibilities afforded by adding explanatory terms that add further columns to the model matrix.

The `DAAG::litters` data frame has observations on brain weight, body weight, and litter size of 20 mice. A model that will be considered in Subsection 3.2.5 is:

```
mouse.lm <- lm(brainwt ~ lsize+bodywt, data=DAAG::litters)
coef(summary(mouse.lm))
```

Are both the explanatory variables `lsize` and `bodywt` contributing to the predictive power of the equation?

## 1.6 Data-based judgments – frequentist, in a Bayesian world

Here, the attention will be on approaches that depend on distributional assumptions. Section 1.8 that follows will be concerned with approaches that are more empirically based.

### *1.6.1 Inference with known prior probabilities*

In daily life we continually update the information on which we rely as better or more complete information has become available. It may be that the old completely supplants the new. Or it may be that the old provides a wider context in which to understand the new information. Bayesian approaches use what is already known or surmised as a context for the analysis and interpretation of new data.

In a medical context, consider a rare disease that occurs in 2 in 1000 in a target population. Suppose that in a person with the disease (a true positive) the disease will be detected with a probability of 0.8 (this is termed the *sensitivity*), while a person who does not have the disease (a true negative) will be detected as negative with a probability of 0.95 (this is termed the *specificity*). Then, in a population of 10000 where 20 have the disease and 9800 do not (a prevalence of 0.002), the results will on average divide up thus:

| | Test $+ve$ | Test $-ve$ | TOTAL | |
|---|---|---|---|---|
| True $+ve$ | 16 | 4 | 20 | *Sensitivity* $= 16/20 = 80\%$ |
| True $-ve$ | 499 | 9481 | 9980 | *Specificity* $= 9481/9900 = 95\%$ |

Thus, among those who test positive, a fraction of only $16/(16{+}499) \simeq 0.03$ will be true positives.

Prior knowledge of the prevalence of the disease in the target population is crucial for judging the risk indicated by the test result. Additionally, accurate estimation of sensitivity and specificity requires large trials, with results that apply to individuals who would meet the entry criteria for the study.

Bayes' theorem gives this result in a manner that does not depend on a specific numerical example, and makes it easy to change the assumptions:

$$\begin{aligned}\Pr[\text{D} \mid \text{test } +\text{ve}] &= \frac{\Pr[\text{test } +\text{ve} \mid \text{D}] \times \Pr[\text{D}]}{\Pr[\textit{test} + \textit{ve}]} \\ &= \frac{\Pr[\textit{test} + \textit{ve} \mid \text{D}] \times \Pr[\text{D}]}{\Pr[\textit{test} + \textit{ve} \mid \text{D}] \times \Pr[\text{D}] + \Pr[\textit{test} + \textit{ve} \mid \neg\text{D}] \times \Pr[\neg\text{D}]}\end{aligned}$$

The following function can be used to calculate the probability that a patient who tests positive has the disease:

```
## `before` is the `prevalence` or `prior`.
after <− function(prevalence, sens, spec){
  prPos <− sens∗prevalence + (1−spec)∗(1−prevalence)
  sens∗prevalence/prPos}
## Compare posterior for a prior of 0.002 with those for 0.02 and 0.2
setNames(round(after(prevalence=c(0.002, 0.02, 0.2), sens=.8, spec=.95), 3),
         c("Prevalence=0.002", "Prevalence=0.02", "Prevalence=0.2"))
```

```
Prevalence=0.002  Prevalence=0.02   Prevalence=0.2
           0.031             0.246            0.800
```

With a prevalence of 0.002 in a population of 10,000, 99.98% of those tested will not have the disease, but will contribute 499 out of the 515 of those who test positive. With a prevalence that equals 0.02, the probability of a positive is much less strongly weighted toward the figure for those who do not have the disease. The finding of a positive has then to be placed in the context of an assessment of the prevalence in the relevant wider population, if necessary using a ballpark estimate. This is especially necessary if the prevalence is very small.

One can think of the probability $\Pr[test + ve \mid \neg \mathrm{D}]$, here equal to 0.05 as a $p$-value for testing the null hypothesis that a patient who does not have a disease will test positive. In this context it is very clear that in order to get a reasonable sense of what $p = 0.05$ might mean, there has to be attention to the best available estimate, or guess at, the prior probability.

*Relating 'incriminating' evidence to the probability of guilt*

Consider the case where there is a police search of a DNA database of 5000 individuals, with the incriminating DNA type found in 1 individual in 1000. The following summarizes the expected result of the police search for a DNA match, optimistically assuming that the person whose DNA was found at the crime scene will be among those netted.

| Not from crime scene | From crime scene |
|---|---|
| 4 (false) positives | 1 true positive |

Odds are then 1:4 or worse that the DNA belongs to the defendant. The DNA match falls well short, on its own, of identifying the defendant as the person whose DNA was found at the crime scene. Writing Pr[I] for Pr[innocence] and Pr[E] for the Pr[evidence], Pr[I | E] is very different from Pr[E | I]. In order to establish a link there has to be other evidence that explains why this particular individual was identified as the suspect, rather than others with the same DNA match. Quoting the 0.001 figure as the probability that the DNA is not from the defendant is a version of what is termed the *prosecutor's fallacy*.

### 1.6.2 Treatment differences that are on a continuous scale

Consider now `sleep` data from a 1905 study that has change in hours of sleep for 10 individuals, for each of two soporific drugs.

```
## Use pipe syntax, introduced in R 4.1.0
sleep <- with(datasets::sleep, extra[group==2] - extra[group==1])
sleep |> (function(x)c(mean = mean(x), SD = sd(x), n=length(x)))() |>
    (function(x)c(x, SEM=x['SD']/sqrt(x['n'])))() |>
    setNames(c("mean","SD","n","SEM")) -> stats
    print(stats, digits=3)
```

```
  mean      SD       n     SEM
 1.580   1.230  10.000   0.389
```

For assessing the difference between the drugs, the relevant statistic is then:

$$t = \frac{\bar{d}}{s/\sqrt{n}} = \frac{6.33}{2.03} = 3.11.$$

The mean is $t = 3.11$ times the SEM. The symbol $t$ is used because the relevant reference for a formal statistical hypothesis test is, under normal distribution assumptions, a $t$-distribution, here with 8 degrees of freedom. We may report: "The mean difference is 1.58 [SEM = 0.389], based on $n = 10$ differences."

Formal hypothesis testing requires the setting out of *null* and *alternative* hypotheses. Taking the population mean to be $\mu$, a common choice of (*point*) null hypothesis is

$$\mathrm{H}_0 : \ \mu = 0 \qquad \text{with alternative hypothesis } \mu \neq 0.$$

The (two-sided) $p$-value is twice the probability that the $t$-statistic is greater than

$$\text{mean/SEM} = 1.580/0.389 = 4.062$$

```
## Sum of tail probabilities
2*pt(1.580/0.389, 9, lower.tail =FALSE)
```

```
[1] 0.0028
```

A standard form of summary statement is: "Based on the sample mean of 1.58, the population mean is greater than zero ($p = 0.0028$)."

A confidence interval offers a different perspective on the result, one that focuses on the estimate and on its accuracy. Assuming that data are from a normal distribution with mean $\mu$, values of $t_9$ will with probability 0.95 lie between -2.262 and 2.262, i.e., in 95% of such samples $\bar{d}$ will lie within a distance 2.262 SEMs of $\mu$. These are the endpoints of what is known as the 95% 'confidence' interval for $\mu$:

$$(1.58 - 2.262 \times 0.389, 1.58 + 2.262 \times 0.389) = (0.7, 2.46)$$

For a 99% confidence interval, replace the multiplier 2.26 by 3.25. This leads to the wider interval $(0.316, 2.84)$.

Code that short-circuits the above calculations, for a 95% confidence interval, is:

```
## 95% CI for mean of heated-ambient: data frame DAAG::pair65
t.test (sleep , conf.level =0.95)
```

*An hypothesis test*

If the 95% confidence interval for the population mean does not contain zero then, using the language of hypothesis testing, the null hypothesis that the population mean is zero will be rejected at a "significance level" of $\alpha = 1 - 0.95 = 0.05$. The value of $p$ that is on the borderline between rejection and non-rejection is termed the critical value, commonly denoted $\alpha$.

Note that $p = 0.00283$ is at the upper end of a range of values, extending from 0 to 0.00283, that under the distributional assumptions made, would all be equally likely if the true mean is zero. With $t = 4.06$, the $p$-value is a $|t| \geq 4.06$ ("as large or larger") probability, not a $|t| = 4.06$ probability, and has to be understood accordingly.

If the only interest was in whether the first drug gave a greater increase than the second drug, the relevant probability is that in the upper tail:

```
pt(4.06, 9, lower.tail =F)
```

```
[1] 0.0014
```

*The $p$-value probability relates to a decision process*

The direct interpretation of a $p$-value as a probability is to a decision process that lumps together all $p$-values that are smaller than a set threshold $\alpha$. Common practice has been to set $\alpha = 0.05$, or to 0.01 or less if greater assurance is required.

When planning an experiment or sampling scheme, in a case where the interest is in a possible difference between two groups, the $p \leq \alpha$ perspective makes a certain sense, with commonly used values are 0.05, or 0.01, or 0.005. Once the results are available, it makes sense to look at the specific $p$-value, and ask what that means. In this text, the Bayes Factors that will be treated in Subsection 1.7.2 will be used as a source of insight.

### 1.6.3 Use of simulation with $p$-values

All a $p$-value offers is an assessment of implications that flow from accepting the null hypothesis. It is important to ask: "How much more likely does the test statistic become under the alternative?" In the extreme case where data is random noise, no given $p$-value is any more likely under the alternative than under the null!

In order to assess the probability under the alternative, assumptions must be made regarding a prior distribution. For any given prior, the probability under the alternative depends, in the one- or two-sample case, on a standardized difference, known as the *effect size*, as noted in Subsection 1.3.6.

Observe that:

- The substantial reduction in variation in the $p$-values returned as effect sizes increase is most noticeable at an effect size of 1.2 for `n=10`, and at 0.8 and 1.2 for `n=40`.
- With an effect size of 0.2, the proportion of $p$-values that are greater than 0.05 changes from 172% at `n=10` to 130% at `n=40`. It remains more likely than not that the $p$-value will be greater than 0.05.
- Again with `eff=0.2`, the lower and upper quartiles shift down from 0.105 and 0.532 at `n=10` to 0.024 and 0.283 at `n=40`. The difference, or *inter-quartile range*, reduces from 0.43 to 0.26. The use of a $p^{0.25}$ scale gives a misleading visual impression of the relative widths of the two boxes.

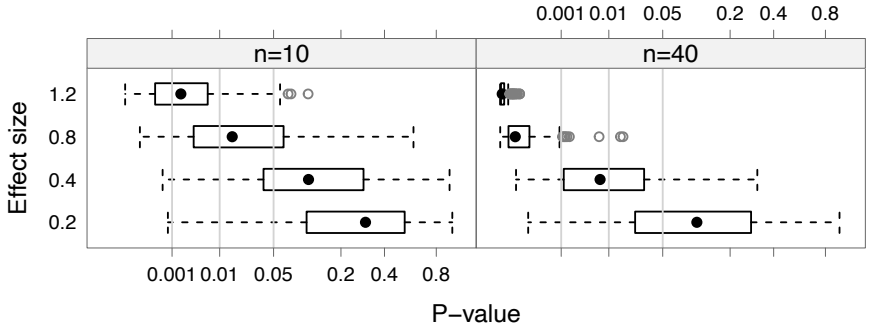



Figure 1.20 Boxplots are for 200 simulated $p$-values for a -sided one-sample $t$-test, for the specified effect sizes `eff` and sample sizes `n`. The $p^{0.25}$ scale on the $x$-axis is used to reduce the extreme asymmetry in the distributions of values that are returned. See 24a for further comment.

The following function was used for the simulations:

```
eff2stat <− function(eff=c(.2,.4,.8,1.2), n=c(10,40), numreps=200,
                     FUN=function(x,N)pt(sqrt(N)∗mean(x)/sd(x), df=N−1,
                                         lower.tail=FALSE)){
  simStat <− function(eff=c(.2,.4,.8,1.2), N=10, nrep=200, FUN){
    num <− N∗nrep∗length(eff)
    array(rnorm(num, mean=eff), dim=c(length(eff),nrep,N)) |>
      apply(2:1, FUN, N=N)
  }
  mat <− matrix(nrow=numreps∗length(eff),ncol=length(n))
  for(j in 1:length(n)) mat[,j] <−
    as.vector(simStat(eff, N=n[j], numreps, FUN=FUN)) ## length(eff)∗numep
  data.frame(effsize=rep(rep(eff, each=numreps), length(n)),
             N=rep(n, each=numreps∗length(eff)), stat=as.vector(mat))
}
```

Code for Figure 1.20 is then:

```
set.seed(31)
df200 <− eff2stat(eff=c(.2,.4,.8,1.2), n=c(10, 40), numreps=200)
labx <− c(0.001,0.01,0.05,0.2,0.4,0.8)
gph <− bwplot(factor(effsize) ∼ I(stat^0.25) | factor(N), data=df200,
              layout=c(2,1), xlab="P−value", ylab="Effect size",
              scales=list(x=list(at=labx^0.25, labels=labx)))
update(gph+latticeExtra::layer(panel.abline(v=labx[1:3]^0.25, col='lightgray')),
       strip=strip.custom(factor.levels=paste0("n=",c(10,40))),
       par.settings=DAAG::DAAGtheme(color=F, col.points="gray50"))
```

The code can readily be adapted to show the equivalent plots for sample effect sizes, for $t$-statistics, and in principle for the Bayes Factors that will be introduced in the next section. See Exercise 24a at the end of the chapter.

### *1.6.4* Power — *minimizing the chance of false positives*

The $p$-value focuses on what can be expected under the null. When designing a new experiment or sampling scheme, it is important to assess the probability of

detecting a 'positive' (i.e., ultimately concluding that the null is false) when the alternative is true. For this purpose it is necessary to specify the size of difference that is of interest. In the one sample and two-sample $t$-tests, this is the Cohen's $d$ measure, defined as the mean difference (one-sample case) or difference in means (two-sample case), divided by the standard deviation.[10]

Given that a decision has been made in advance to regard as *significant* any $p$-value that is less than $\alpha$, the *power* $P_w$ is the probability of detecting an effect of the specified size. (Note that $\beta = 1 - P_w$ has the name "Type II error".)

Suppose for example that the power is $P_w = 0.8$, relative to the $\alpha = 0.05$ cutoff. This is high relative to the standards of much published work. Consider three scenarios, where 300 tests for a drug are divided between true positives and true negatives in one of the ratios 0.2:1, 1:1, 5:1. Results will then, on average, divide up as follows:

```
      True positives False positives
R=0.2 0.05x250=12.5  0.8 x50=40
R=1   0.05x150=7.5   0.8x150=120
R=5   0.05 x50=2.5   0.8x250=160
```

As the prior odds increase from 0, the probability that $p \leq 0.05$ will indicate a real effect will, unless $P_w = 0$, increase from zero. In the extreme case where $P_w = 0$, the probability of a real effect will be be the same (equal to the prior probability) irrespective of the $p$-value.

Power calculations, as implemented in the base R function `power.t.test()` and in the *pwr* package, allow experimenters to determine the size of experiment that can on average be expected to detect, at a specified significance level $\alpha$, a specified Cohen's `d` effect size.

The effect size for which it is reasonable to plan depends on the context. In medicine, the effect size should be large enough to be distinguishable from natural variation in the population. It is useful to check previous trials in related areas of research, being mindful that the refereeing processes involved in publication will in many areas generate substantial positive biases.

*Power calculations – examples*

Examples of power calculations, for two-sided tests with effect size `d=0.5`, are:

```
power.t.test(d=0.5, sig.level=0.05, type="one.sample", power=0.8)
```

```
     One-sample t test power calculation

              n = 33
          delta = 0.5
             sd = 1
      sig.level = 0.05
          power = 0.8
    alternative = two.sided
```

[10] We are assuming a common standard deviation in the case of two samples.

```
pwr1 <- power.t.test(d=0.5, sig.level=0.005, type="one.sample", power=0.8)
pwr2 <- power.t.test(d=0.5, sig.level=0.005, type="two.sample", power=0.8)
## d=0.5, sig.level=0.005, One- and two-sample numbers
c("One sample"=pwr1$n, "Two sample"=pwr2$n)
```

```
One sample Two sample
        57         108
```

Figure 1.20 in Subsection 1.3.6 provided strong indications of the range of $p$-values that could be expected, given a specified effect size and specified sample size. Those who are planning an experiment would do well to plot the equivalent graph that relates the effect size of interest and the planned sample size.

The simulations are, effectively, experiments. The following shows, based on the simulation results for the combinations shown of effect size (ES) and sample sizes (number of pairs or paired differences), the proportion of simulations where the p-value for the null hypothesis of no effect fell under the specified significance level.:

$\alpha$=0.05

| | n=10 | n=20 | n=40 |
|---|---|---|---|
| ES=0.05 | 0.035 | 0.04 | 0.049 |
| ES=0.2 | 0.082 | 0.134 | 0.234 |
| ES=0.4 | 0.204 | 0.397 | 0.694 |
| ES=0.8 | 0.6162 | 0.9239 | 0.9985 |
| ES=1.2 | 0.9203 | 0.9991 | ∼1.0000 |

$\alpha$=0.005

| | n=10 | n=20 | n=40 |
|---|---|---|---|
| ES=0.05 | 0.004 | 0.005 | 0.006 |
| ES=0.2 | 0.011 | 0.023 | 0.054 |
| ES=0.4 | 0.037 | 0.116 | 0.343 |
| ES=0.8 | 0.2119 | 0.655 | 0.9768 |
| ES=1.2 | 0.5666 | 0.9762 | ∼1.0000 |

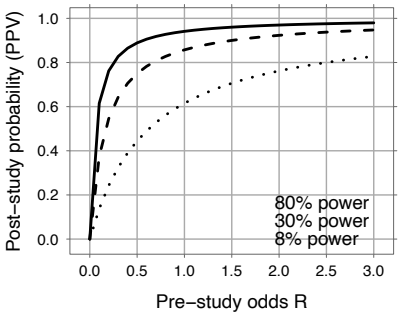
Thus, for `n` = 10 and an effect size of 0.2, close to 8% of 'positives' (where $p \leq 0.05$ is used as the criterion) will be detected as such, as against 5% under the null. Subsection 1.10.2 will discuss estimated effect sizes from replications of published social psychology studies, where a high proportion were in the vicinity of 0.05. For sample sizes of 40 or less, the power is under 0.05, so that more than 19 out of 20 true positives will be missed under an $\alpha = 0.05$ cutoff. The experimental results mainly add noise.

Code for the calculations, for the two different choices of significance level, is:

```
effsize <- c(.05,.2,.4,.8,1.2); npairs <- c(10,20,40)
pwr0.05 <- matrix(nrow=length(effsize), ncol=length(npairs),
                  dimnames=list(paste0('ES=',effsize), paste0('n=',npairs)))
pwr0.005 <- matrix(nrow=length(effsize), ncol=length(npairs),
                  dimnames=list(paste0(effsize), paste0('n=',npairs)))
for(i in 1:length(effsize)) for(j in 1:length(npairs)){
    pwr0.05[i,j] <- power.t.test(n=npairs[j],d=effsize[i], sig.level =0.05,
                                 type='one.sample')$power
    pwr0.005[i,j] <- power.t.test(n=npairs[j],d=effsize[i], sig.level =0.005,
                                  type='one.sample')$power}
tab <- cbind(round(pwr0.05,4), round(pwr0.005,4))
```

### *Positive Predictive Values*

In the example at the start of this subsection, the prior odds R was an actual but unknown ratio of number of drugs with a potentially detectable effect to the number with no effect, giving a prior probability of R/(R+1). The reasoning carries




Example: With R = 1:15 (e.g. 100 true +ves to 1500 true −ves), $\alpha = 0.05$, and $P_w = 0.3$, the posterior odds are:

$$\frac{RP_w}{\alpha} = \frac{0.3}{15 \times 0.05} = 0.4$$

$$\text{PPV} = \frac{0.4}{1 + 0.4} \simeq 0.29$$

Figure 1.21 Post-study probability (PPV)), as a function of the pre-study odds, for different levels of statistical power.

through in the same way if R is a guess at a prior odds ratio, perhaps based on earlier studies.

The effect size (ES) for identifying a detectable effect may be chosen as the smallest change that is of interest. For instance, in an experiment to compare drugs that induce sleep (as in the dataset `datasets::sleep`), a minimum effect size of ES = 30 min might be reasonable.

The (posterior) probability of a genuine effect of the given size is known as the *Positive Predictive Value* or PPV. Examples that will be given shortly should make the idea clear. The *False Discovery Rate* or FDR is related to the PPV thus:

$$\text{FDR} = 1 - \text{PPV}$$

With $\alpha = 0.05$, 5% of tests are expected, under the null hypothesis, to show $p \leq \alpha$. Thus, given that the ratio of positives to negatives is $R : 1$, the odds that an apparent positive will be a true positive is

$$\frac{RP_w}{\alpha} = R\frac{P_w}{\alpha} \tag{1.3}$$

i.e., multiply the prior odds $R$ by $\frac{P_w}{\alpha}$ to get an odds ratio that accounts for the new evidence. Ioannidis (2005) has a modification of Equation 1.3 that allows for bias.

The positive predictive value (PPV), or posterior probability, is then:

$$PPV = \frac{RP_w}{RP_w + \alpha} \tag{1.4}$$

Figure 1.21 illustrates what the formula means in practice.

Based on eight journal articles published between 2006 and 2009, Button et al. (2013) reported an astonishing median power of 0.08 across 461 individual studies of brain volume abnormalities. Results were better, but still not encouraging, in neuroscience more generally. Based on meta-analysis reports published in 2011, Button et al. (2013) found a median power of 0.21 for 730 individual primary studies.

Prinz et al. (2011) reported a success rate of around 30% in their efforts to reproduce the main result in 67 published studies. Two scenarios that, in the absence of other faults, would on average reproduce this approximate success rate are:

- $R = 1{:}15$, $\alpha = 0.05$, $P_w = 0.3$
- $R = 1{:}4$, $\alpha = 0.05$, $P_w = 0.08$

Calculations for the first of these are shown to the right of Figure 1.21. These scenarios ignore the likely contributions of design, data, execution, and presentation faults. Such faults will increase the risk of finding a spurious effect, the risk of failing to detect a genuine effect, and the risk of biases that distort the result.

Small studies with low power compromise the use of $p$-values for any purpose – whether as a screening device, or to confirm an earlier result. Button et al. (2013) note that the magnitude of a true effect, when found, is on average exaggerated. Additionally, small studies may not be conducted with the same care as larger studies that require more careful organization, and the smaller datasets that result are more susceptible to minor changes in the analysis process.

The issues to which Equation 1.4 and Figure 1.21 relate are of central importance for large areas of laboratory science. See in particular Ioannidis (2005). In practical use, for estimating the PPV, such formulae provide only broad guidelines. The estimate of the power $P_w$, and an assessment of what is a reasonable effect size, is often based on pilot study information from a convenience sample and not a random sample, and is susceptible to selection bias. Estimates that are accurate enough to be a good basis for design may be available when a trial is the latest in a series of comparable trials. Gelman and Carlin (2014) argue for the use of information external to the specific study as a basis for choosing the effect size. A literature review will often provide useful leads.

The results given in this subsection rely on setting a threshold $\alpha$ and not distinguishing between $p$-values that lie within a range from 0 to $\alpha$. That makes sense when designing an experiment, but preferably setting $\alpha$ somewhat less than 0.05. Once results are in, it becomes important to take note of the specific $p$-value, and ask what it means.

### *1.6.5 The future for $p$-values*

The critical test for laboratory science is replicability — are other scientists able to replicate the results? $P$-values and other relevant statistics should be seen as a complement to, and not as a substitute for, independent replication. Two quotes from Ronald Fisher, who introduced the use of $p$-values, emphasize the importance of replication:

> Personally, the writer prefers to set a low standard of significance at the 5 per cent point, and ignore entirely all results which fail to reach this level. A scientific fact should be regarded as experimentally established only if a properly designed experiment *rarely* fails to give this level of significance.
> [Fisher (1926)]

Fisher commented, also, that a level of 0.02 or 0.01 might be used if 0.05 did not seem low enough.

> ... no isolated experiment, however significant in itself, can suffice for the experimental demonstration of any natural phenomenon ...
>
> ... we may say that a phenomenon is experimentally demonstrable when we know how to conduct an experiment which will rarely fail to give us a statistically significant result. [Fisher (1935, pp 13–14)]

Independent replication at another time and place is the crucial check, irrespective of the statistical summary measures used It provides checks on the experimental processes, on mistakes in statistical analysis, and on factors that may be local to the place and time of the original experiment. It checks that the published article and supplementary materials are sufficiently accurate and detailed that others can repeat the experiment. Where mistakes have been made, it is unlikely that another research group will repeat the same mistakes.

If both the original study and the replication have been carried out with impeccable care and give clearly different results, identification of the different factor that has been at play may yield new and important insight. The low power of much of what is reported in the literature makes it difficult to be sure that results are "clearly different".

*How small a p-value is needed?*

*P*-values are designed to decrease as the weight of evidence shifts to favor the alternative. They are not absolute measures of the weight of evidence. In a context where positive results are thought to be as likely as not, $p \leq 0.05$ has to be regarded as providing only weak evidence for an effect.

The Bayes factors that will be discussed in the next section will be used alongside *p*-values and can help in assessing what a given *p*-value may mean for the relative probabilities of the null and the alternative.

### *1.6.6 Reporting results*

It is important to report effect sizes. As the sample size increases, the *p*-value for a given estimated effect size will decrease. For large enough sample sizes, a small *p*-value (using whatever value is taken as 'small') will correspond to an effect size that is so small as to be of no consequence. Rather than testing for a point null, it commonly makes much better sense to set a minimum difference $\mu_0$ that is of interest, and test $\mu > \mu_0$ against the null $\mu = \mu_0$.

The quoting of multiple *p*-value or equivalent test results raises awkward questions about what to make of the combined results. In general, significance tests should be closely tied to key points in the paper. Where possible, make a rough assessment of the prior probabilities, at least to the extent of distinguishing between studies with "low prior odds" and those with prior odds that may be of the order of 1:1 or higher. Once it seems clear that an effect is real, the focus of interest should shift to its pattern and magnitude, and to its scientific significance.

It is a poor use of the data to perform tests for each comparison between treatments when the real interest is (or should be) in the overall pattern of response. Where the response depends on a continuous variable, it may be pertinent to ask whether, e.g., the response keeps on rising (falling), or whether it rises (falls) to a maximum (minimum) and then falls (rises).

*Is there an alternative that is more likely?*

Finding $p \leq \alpha$ (commonly with $\alpha = 0.05$) is all very well. The important question is whether there is an alternative that is substantially more likely. Comments in Berkson (1942) highlight the point that $p$-values relate only to what can be expected under the null.

> If an event has occurred, the definitive question is not, "Is this an event which would be rare if the null hypothesis is true?" but "Is there an alternative hypothesis under which the event would be relatively frequent?"

The approaches to be described in the next section address this point very directly.

## 1.7 Information statistics and Bayesian methods with Bayes Factors

Information statistics and Bayes Factors both offer ways to establish a preference for one model over another. Information statistics rely on large sample results, though for the Akaike Information Criterion (AIC) that is in common use, a correction is available that is designed to improve the small sample properties . Bayes Factors rely on the choice of a specific prior. Although a large sample approximation is not involved, the prior is both more important and harder to choose with confidence when sample sizes are small. Unlike the AIC, Bayes Factors have not to date found common use in the statistical mainstream.

### *1.7.1 Information statistics – using likelihoods for model choice*

For using likelihoods to express a preference for one model over another rather than as a basis for a significance test, the likelihood must be adjusted for the number of parameters estimated. Setting $p$ equal to the number of parameters estimated (here, this includes any scale parameters), the Akaike Information Criterion (AIC) is one of two widely used criteria that take the form:

$$kp - 2log(\hat{L}), \quad \text{where, for AIC, } k = 2$$

For the Bayesian Information Criterion (BIC), $k = \log(n)$, where $n$ is the number of observations. In either case, the model that gives the smallest value is preferred. The descriptor 'Bayesian' appears in the name because the choice of $k$ can be motivated by arguments that have a Bayesian connection, albeit in a large sample limit.

The AIC penalty term is designed so that, in the large sample limit with $n$ much larger than $p$, the statistic will select the model with the lowest prediction error. Where $p$ is an overly large fraction of $n$ the criterion is inclined to select overly complex models. For models for which it is available, there is then a strong case

for increasing the amount of the correction, as for the AICc statistic that will be discussed below.

The Bayesian Information Criterion (BIC) sets $k = \log(n)$, thus for $n > 7$ penalizing complex models (many parameters) more strongly. It is designed, in the large sample limit, to select the correct model. As for the AIC, it is important that the number of observations $n$ is much larger than the number $p$ of parameters that are estimated.[11]

For normal theory models the AIC statistic is, in the usual case where the variance has to be estimated:

$$\text{AIC} = n \log\left(\frac{\text{RSS}}{n}\right) + 2p + \text{const}, \quad \text{where} \quad \text{const} = n(1 + log(2\pi)) \tag{1.5}$$

Note again the reason for placing $y$-axis tick marks a distance 0.02 apart on the $\log_e$ linear scale used in Figure 1.7. On the $\log_e$ scale a change of 0.02 is very nearly a 2% change. where RSS is the residual sum of squares and the constant term arises from the assumption of an iid normal distribution for the errors.

Corrections are available, different for different error families that improve the small sample properties of the AIC statistic. For normal theory models, the AICc statistic replaces the AIC penalty term $2p$ by:

$$2\frac{n}{n-p-1}p, \quad \text{i.e., an increase of } 2p\frac{p+1}{n-p-1}$$

The penalty is then larger for smaller values of `n`, with the consequence that the difference in AICc statistics for a `d` degree of freedom increase is smaller for smaller values of `n`. For the comparison between a model with `p` parameters and one with `p+d` parameters, the change in the AICc statistic is less than the change in the AICc statistic by an amount:

$$2(p+d)\frac{n}{n-(p+d)-1} - 2p\frac{n}{n-p-1} - 2d$$

The function `AICcmodavg::AICc()` (Mazerolle, 2020) implements this, as well as corrections for a number of other error families. A vignette that gives an overview of the package has extensive advice on model comparison.[12].

The following checks on the formula just given for the AICc statistic:

```
## Calculations using mouse brain weight data
mouse.lm <- lm(brainwt ~ lsize+bodywt, data=DAAG::litters)
n <- nrow(DAAG::litters)
RSSlogLik <- with(mouse.lm, n*(log(sum(residuals^2)/n)+1+log(2*pi)))
p <- length(coef(mouse.lm))+1  # NB: p=4 (3 coefficients + 1 scale parameter)
k <- 2*n/(n-p-1)
c("AICc" = AICcmodavg::AICc(mouse.lm), fromlogL=k*p-2*logLik(mouse.lm)[1],
  fromFit=k*p + RSSlogLik) |> print(digits=4)
```

```
    AICc fromlogL  fromFit
 -112.9   -112.9   -112.9
```

[11] In more technical language the AIC statistic is designed to be *asymptotically efficient*, where the BIC statistic is designed to be *asymptotically consistent*.

[12] Type `vignette("AICcmodavg", package="AICcmodavg")` to see the vignette

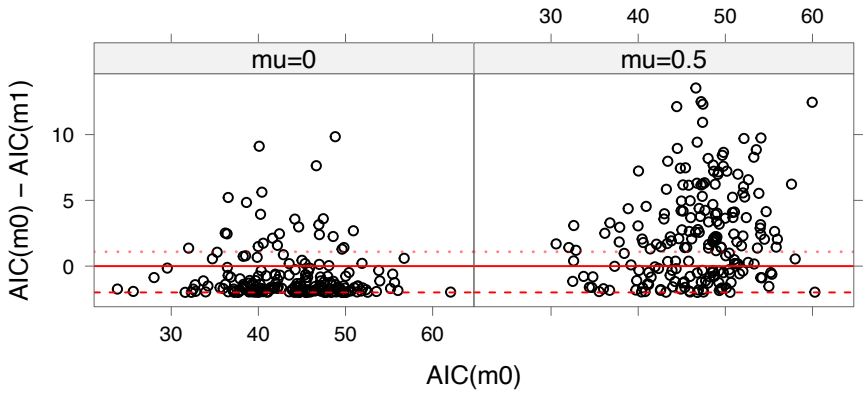



Figure 1.22 Points are from simulations of the sampling properties of the difference in AIC statistics between a model that fits a zero mean and a model that fits a non-zero mean.

Look ahead to Figure 3.17 in Subsection refss:ICstats for a graph that compares the change in penalty terms between the three statistics, for $5 \leq n \leq 35$ from a unit increase in the number of parameters estimated. For the AICc statistic, the cases shown are for an increase from $p$=1 to $p$=2, or from $p$=3 to $p$=4.

*The sampling properties of the difference in AIC statistics*

Figure 1.22 shows results from simulation of the sampling properties of the difference in AIC statistics between a model that fits a zero mean and one that fits a non-zero mean. Panel A is for the case where 15 observations are from a normal distribution with mean 0, while for Panel B the mean was set to $0.5\sigma$. The red solid line is for $y = 0$, while the dashed line is for $y = -2$. Notice that in Panel A, points are relatively dense close to the line $y = -2$, with many more points shifted above this limiting line in Panel B. If the graph is repeated for the AICc statistic, the effect is to shift the solid red horizontal zero line upwards to the fainter dotted red line. Code for Figure 1.22 is:

```
sim0vs1 <− function(mu=0, n=15, ntimes=200){
a0 <− a1 <− numeric(ntimes)
for(i in 1:ntimes){
  y <− rnorm(n, mean=mu, sd=1)
  m0 <− lm(y ~ 0); m1 <− lm(y ~ 1)
  a0[i] <− AIC(m0); a1[i] <− AIC(m1)
}
data.frame(a0=a0, a1=a1, diff01=a0−a1, mu=rep(paste0("mu=",mu)))
}
sim0 <− sim0vs1(mu=0)
sim0.5 <− sim0vs1(mu=0.5)
simboth <− rbind(sim0, sim0.5)
cdiff <− with(list(n=15, p=2), 2*(p+1)*p/(n−(p+1)−1))
xyplot(diff01 ~ a0 | mu, data=simboth, xlab="AIC(m0)", ylab="AIC(m0) − AIC(m1)") +
  latticeExtra :: layer({panel.abline(h=0, col='red');
          panel.abline(h=cdiff, lwd=1.5, lty=3, col='red', alpha=0.5);
          panel.abline(h=−2, lty=2, col='red')})
```

The following are the proportions of samples for which the statistic for 'm1' is smaller than that for 'm0', so that 'm1' is preferred:

```
                              True model is m0 True model is m1
AIC: Proportion choosing m1             0.18             0.66
AICc: Proportion choosing m1            0.32             0.80
```

(Kass and Raftery, 1993) give reasons why AIC is likely to choose overly complex models in the large sample context. When samples are small, even AICc should be used with caution. Checks on distributional assumptions become increasingly less helpful as sample sizes decrease relative to the number of parameters estimated.

### *1.7.2 Bayesian methods with Bayes Factors*

In this text, Bayesian methods are primarily used to complement the use of $p$-values, and to help clarify what they do and do not imply. $P$-values, Bayes factors, and other forms of Bayesian summary, offer different and complementary perspectives on inferences from data.

In a Bayesian approach, the density for the prior is used to weight the density that is fitted to the data, yielding a posterior distribution. There is, in most contexts, an inevitable arbitrariness in the choice of prior. A Bayes Factor is the ratio of the (marginal) likelihood under the alternative to the likelihood under the null. To obtain the odds for the alternative over against the null multiply the Bayes Factor by the prior odds, which in many contexts has likewise to be guessed. Uncertainties in what a $p$-value can be taken to imply are replaced by uncertainties associated with the assumptions made in calculating the Bayes Factor.

Once decisions have been made on the choice of priors, Bayes Factors allow, as $p$-values do not, a probability to be assigned to the null relative to the alternative. The null can in principle be an interval rather than a point.

Here, use will be made of functions in the *BayesFactor* package. The class of priors used places a Cauchy prior (i.e., a $t$-distribution with one degree of freedom) on a difference that is of interest. The default scale parameter is $1/\sqrt{2}$, where a value of 1 corresponds to half the inter-quartile range. The difference made by a different choice of scale parameter is relatively small:

```
pcauchy(1, scale=1)    pcauchy(1, scale=1/sqrt(2))
               0.75                           0.80
```

The family of priors used, commonly referred to as the JZS (Jeffreys Zellner-Siow) family, has been extensively researched. It has the benefit of allowing, for a wide range of models, a numerical approximation to the required integral. See Rouder et al. (2009). It may reasonably be the default in contexts where priors are required that do not overly reflect the judgment of an individual experimenter. It is not at all intuitively obvious how priors should be chosen.

Various upper limits are available in the literature that apply across a wide class of priors that are centered at the null, for distributions that include the Cauchy and the normal. See Subsection 2.9.2. Section 2.9.3 will demonstrate computations that can be adapted for use with other families of priors.

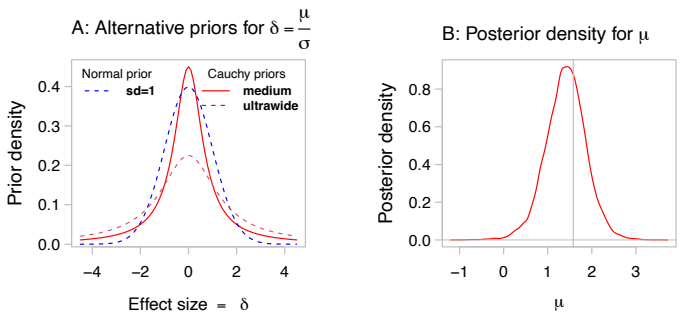



Figure 1.23 Panel A compares alternative Cauchy priors for the standardized effect size $\delta$ (or $\mu$) with a normal distribution. Panel B shows the posterior for the sleep data, obtained using the default prior. A vertical line shows the mean increase in hours of sleep in the sample.

Bayes Factors are compared with two other Bayesian approaches for comparing a null with an alternative in Linde, Tendeiro, et al. (2021). The R package *bayesplay* (Colling, 2021) has several vignettes that provide a step by step introduction to Bayes Factors as implemented in the *BayesFactor* package. Kass and Raftery (1993) give a useful overview that discusses also the AIC and BIC statistics. See also the text "An Introduction to Bayesian Thinking" (Clyde et al., 2022) that is available from `https://statswithr.github.io/book/`.

### *The Cauchy prior with different choices of scale parameter*

Figure 1.23A compares alternative priors that might be used for the `sleep` data of Subsection 1.6.2. Panel B shows a posterior density for $\delta$ was obtained assuming the default `rscale`=$\sqrt{2}/2$ *BayesFactor* prior.

Note that the default is to use a scale parameter that is data dependent, where ideally the user would supply a scale that was based on previous experience with comparable data. The default use of a data dependent scale introduces an uncertainty that becomes an increasing issue for small samples.

```
## Calculate and plot density for default prior - Selected lines of code
x <- seq(from=-4.5, to=4.5, by=0.1)
densMed <- dcauchy(x, scale=sqrt(2)/2)
plot(x, densMed, type='l')
## Panel B
pairedDiffs <- with(datasets::sleep, extra[group==2] - extra[group==1])
ttBF0 <- BayesFactor::ttestBF(pairedDiffs)
## Sample from posterior, and show density plot for mu
simpost <- BayesFactor::posterior(ttBF0, iterations=10000)
plot(density(simpost[,'mu']))
```

### *A thought experiment*

Consider as a thought experiment three alternatives, in a study that tests 300 drugs:

1. 50 have an effect that is in principle detectable, while 250 are inactive. The

Table 1.4 *Posterior odds, as returned by* `BayesFactor::ttest.tstat()`*, when the Bayes factor is multiplied by the respective the prior odds R.*

| | Bayes | Posterior odds | | |
|---|---|---|---|---|
| $p$-value ($t_{19}$) | Factor | R = 0.2 | R = 1 | R = 5 |
| 0.05 (2.1) | 1.39 | 0.28 | 1.39 | 6.9 |
| 0.01 (2.9) | 5.12 | 1.02 | 5.12 | 25.6 |
| 0.005 (3.2) | 9.17 | 1.83 | 9.17 | 45.9 |

(usually unknown) prior odds $R : 1$ are, in this thought experiment, 50:250 = 1:5 or 0.2:1, so that R=0.2, strongly favoring the null hypothesis of no effect.
2. 150 have an effect, and 150 are inactive, so that R=1.
3. 250 have an effect, and 50 are inactive, so that R=5.

Now consider three sets of 20 paired differences, the first resulting in a two-sided $p$-value equal to 0.05, the second 0.01, and the third 0.005. Code that calculates the Bayes Factors that will be used here is:

```
tval <- setNames(qt(1-c(.05,.01,.005)/2, df=19), paste(c(.05,.01,.005)))
bf01 <- setNames(numeric(3), paste(c(.05,.01,.005)))
for(i in 1:3)bf01[i] <- BayesFactor::ttest.tstat(tval[i], n1=20, simple=T)
```

The final three columns of Table 1.4 show the posterior odds from multiplying the relevant Bayes Factors by prior odds of 0.2, 1.0, and 5.0 respectively.

As with $p$-values, it may make more sense to specify the minimum effect size that is of interest, rather than to work with a point null. This is, at the same time, a less important issue for Bayes Factors than for $p$-values. Look ahead to Figure 1.24.

Now compare calculations for the `sleep` data that use the *BayesFactor* package for the three suggested choices of scale factor:

```
pairedDiffs <- with(datasets::sleep, extra[group==2] - extra[group==1])
ttBF0 <- BayesFactor::ttestBF(pairedDiffs)
ttBFwide <- BayesFactor::ttestBF(pairedDiffs, rscale=1)
ttBFultra <- BayesFactor::ttestBF(pairedDiffs, rscale=sqrt(2))
rscales <- c("medium"=sqrt(2)/2, "wide"=1, ultrawide=sqrt(2))
BF3 <- c(as.data.frame(ttBF0)[['bf']], as.data.frame(ttBFwide)[['bf']],
         as.data.frame(ttBFultra)[['bf']])
setNames(round(BF3,2), c("medium", "wide", "ultrawide"))
```

```
   medium      wide ultrawide
       17        18        18
```

Note for comparison the Sellke et al. (2001) upper limit, which applies for a wide range of priors, centered at the null, with densities that tail off in much the same manner as for the normal. With the $p$-value equal to 0, this is:

```
pval <- t.test(pairedDiffs)[['p.value']]
1/(-exp(1)*pval*log(pval))
```

```
[1] 22
```

For degrees of freedom around 10, the value returned by `ttestBF` is not greatly below this lower bound. See Subsection 2.9.2 for further comment.

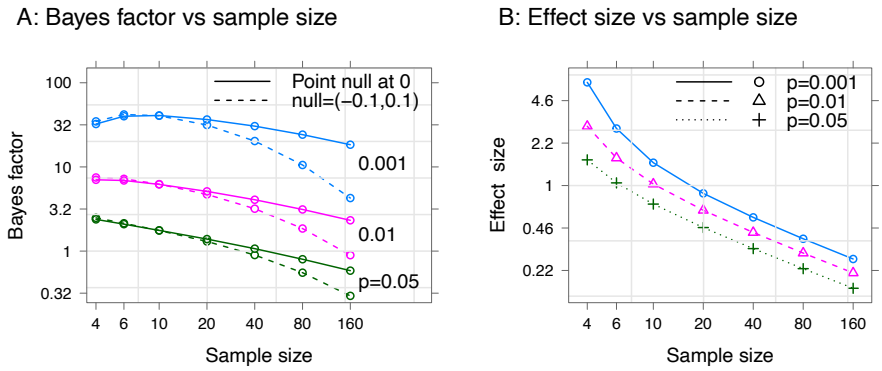



Figure 1.24 Bayes factors are shown that correspond to a given two-sided $p$-value, for a range of sample sizes, calculated using the function `BayesFactor::ttest.tstat()`. Note that calculations for the null (-0.1,0.1) interval case require *BayesFactor* version 0.9.12-4.5 or later. Earlier versions used, for $t$>5, an approximation that failed and gave, for $p = 0.01$ with $n \leq 4$ and for $p = 0.001$ with $n \leq 9$, grossly inflated values.

### *A null interval may make better sense*

A simple change in the code for the one-sample case gives, instead of a comparison with a point null, a comparison between a null interval and its complement. Thus, for the `sleep` data, consider a null interval that extends 45 minutes (0.75hr) either side of 0. The SD is $s = 1.23$, so that 0.75hr gives a standardized difference of $d = 0.75/1.23 = 0.61$.

The first line of the following output compares the interval $-0.61 < d < 0.61$ with 0, while the second line compares the region outside that interval with 0.

```
min45 <− round(0.75/sd(pairedDiffs),2)   ## Use standardized units
ttBFint <− BayesFactor::ttestBF(pairedDiffs, nullInterval=c(−min45,min45))
round(as.data.frame(ttBFint)['bf' ],3)
```

```
                                  bf
Alt., r=0.707  -0.61<d<0.61        5
Alt., r=0.707  !(-0.61<d<0.61)  27
```

```
bf01 <− as.data.frame(ttBFint)[['bf' ]]
```

The ratio $27.39/5.03 = 5.45$ is the Bayes factor that compares an absolute difference $|d| \geq 0.61$ with $|d| < 0.61$.

### *The effect of changing sample size*

Figure 1.24 shows, for three choices of $p$-value and for a range of sample sizes, Bayes factor estimates as returned by `BayesFactor::ttest.tstat()`. Panel B shows the effect sizes — sizes – these are as estimated from a sample that yields the specified $p$-value.

For all three choices of $p$-value, the Bayes factor soon starts to decrease with increasing degrees of freedom. For $p$=0.05 sooner than for smaller $p$-values, a point is

in due course reached where the sample effect size is too small to be of consequence. The Bayes factor reflects this, giving a perspective on the information in the data that is different from that offered by $p$-values.

The somewhat larger Bayes factors at smaller sample sizes, relative to a given $p$-value, come with a caution. Even more than for $p$-values, sampling variation is a greater concern when samples are small.

Exercise 31 looks at the effect of replacing `rscale="medium"` with `rscale="wide"`. Exercise 32 shows results with the JUI family of priors for which the function `statsr:bayes_inference` has provision.[13] In the example considered, the Bayes Factor is smaller.

*Different statistics give different perspectives*

A difference between $p = 0.00001$ and $p = 0.000001$, in a one-sample and one-sided $t$-test with $n$=40, will not ordinarily attract much attention. The corresponding Bayes Factors, calculated using `BayesFactor::ttest.tstat()` and rounded to whole numbers, are 1012 and 8328. While the difference looks impressive, it is readily explainable by statistical variation and/or small departures from assumptions.

```
bf1 <- BayesFactor::ttest.tstat(qt(0.00001, df=40), n1=40, simple=T)
bf2 <- BayesFactor::ttest.tstat(qt(0.000001, df=40), n1=40, simple=T)
rbind("Bayes Factors"=setNames(c(bf1,bf2), c("p=0.00001","p=0.000001")),
  " t-statistics "=c(qt(0.00001, df=40), qt(0.000001, df=40)))
```

```
                p=0.00001 p=0.000001
Bayes Factors      1012.0     8327.8
t-statistics         -4.8       -5.6
```

Various authors (see, e.g. Kass and Raftery, 1995) suggest a scale of evidence akin to the following for Bayes Factors:

| 1 – 3 | 3 – 20 | 20 – 150 | >150 |
|---|---|---|---|
| A bare mention | Positive | Strong | Very strong |

Small Bayes Factors (much less than 1.0) can be interpreted as evidence against the alternative and in favour of the null.

** Technical details of the family of priors used in* BayesFactor

The Cauchy distribution is a $t$-distribution with a single degree of freedom, used as a relatively uninformative prior. It results from assuming a normal distribution for the difference $\delta$, with the variance of $\delta$ distributed as inverse chi-square. The median is its location parameter, while its scale parameter is half the inter-quartile range. The mean and variance are not defined. The Jeffreys distribution has a similar role for the variance of the normal distributions that are assumed both under the null and under the alternative. See `?dcauchy` and `?BAS::Jeffreys`.

[13] Differently from the unit information prior used in Subsection 2.9.2 to motivate the BIC statistic, it is here centered on the null.

## 1.8 Resampling methods for SEs, tests, and confidence intervals

Resampling methods rely on the selection of repeated samples from a 'population' that is constructed from the sample data. They can be an effective recourse when departures from normality

As there are in general too many possible distinct samples to take them all, reliance is on repeated random samples. In this section, we demonstrate permutation and bootstrap methods. We start by demonstrating the use of permutation tests for the equivalent of one-sample and two-sample $t$-tests.

### *1.8.1 The one-sample permutation test*

Consider the paired elastic band data from the data frame `DAAG::pair65`:

```
                1   2   3   4   5   6   7   8   9
heated        244 255 253 254 251 269 248 252 292
ambient       225 247 249 253 245 259 242 255 286
heated-ambient 19   8   4   1   6  10   6  -3   6
```

If the treatment has made no difference, then an outcome of 244 for the heated band and 225 for the ambient band might equally well have been 225 for the heated band and 244 for the ambient band. A difference of 19 becomes a difference of $-19$. The assumption is that each of the $2^9 = 512$ permutations, and thus its associated mean difference, is equally likely. This *exchangability* assumption is weaker than independence. It may be seen as a weak form of independence.

We then locate the mean difference for the data that we observed within this permutation distribution. The $p$-value is the proportion of values that are as large in absolute value as, or larger than, the mean for the data.

The first row of the following table shows absolute values of the 9 differences:

| Difference | 19 | 8 | 4 | 1 | 6 | 10 | 6 | 3 | 6 |
|---|---|---|---|---|---|---|---|---|---|
| | 19 | 8 | 4 | $-1$ | 6 | 10 | 6 | 3 | 6 |
| | 19 | 8 | 4 | 1 | 6 | 10 | 6 | $-3$ | 6 |

In the permutation distribution, these each have an equal probability of taking a positive or a negative sign. There are $2^9$ possibilities, and hence $2^9 = 512$ different values for $\bar{d}$. Rows 1 to 3 of the table show the possibilities that give a mean difference that is as large as or larger than in the actual sample. There are another three possibilities that give a mean difference that is of the same absolute value, but negative. (These three possibilities are obtained by reversing the signs of all elements in rows 1 to 3 of the table.) Hence, $p = 6/512 = 0.0117$.

In general, when the number of pairs is large, the computational demands of an enumeration approach such as has been demonstrated become severe. A suitable recourse is to take repeated random samples from the permutation distribution. The function `DAAG::onetPermutation()` may be used for this.

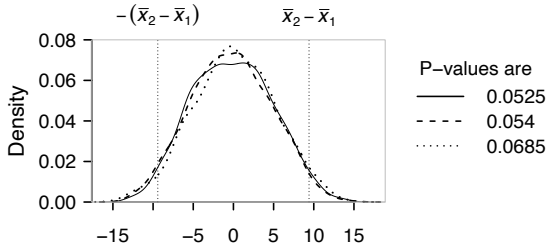



Figure 1.25 Density curves for two samples of 2000 each from the permutation distribution of the difference in means, for the two-sample elastic band data.

### *1.8.2 The two-sample permutation test*

Suppose we have $n_1$ values in one group and $n_2$ in a second, with $n = n_1 + n_2$. The permutation distribution results from taking all possible samples of $n_2$ values from the total of $n$ values. For each such sample, we calculate

mean of the $n_2$ values that are selected – mean of remaining $n_1$ values.

The permutation distribution is the distribution of all such differences of means. We locate the differences of means for the actual samples within this permutation distribution. Thus, consider the data from the dataset `DAAG::two65`. In order to keep computational demands within reasonable bounds, we will take samples of 2000 or more from the permutation distribution:

Ambient: 254 252 239 240 250 256 267 249 259 269 (Mean = 253.5)
Heated: 233 252 237 246 255 244 248 242 217 257 254 (Mean = 244.1)

The proportion of samples (here out of 2000) where the absolute value of the difference of the two resampled means was greater than or equal to that for the observed means can be used as a $p$-value. A much larger sample size than 2000 is not warranted, given the sampling uncertainty inherent in the initial sample sizes of 10 for the ambient bands and 11 for the heated bands.

Figure 1.25 overlays three estimates of the permutation distribution that were obtained by taking, in each instance, 2000 random samples from the permutation distribution. The point where the difference in means falls with respect to the sampled values ($253.5 - 244.1 = 9.4$) has been marked, as has minus this difference. The code used for the first of the density curves was in essence that for the function `DAAG::twotPermutation`, preceded by `set.seed(47)`.

### *1.8.3* Estimating the standard error of the median: bootstrapping*

The bootstrap idea is to treat the one sample that we have, for purposes of estimating the sampling distribution of a sample statistic, as an approximation to the entire population. We take repeated resamples with replacement from the given sample, compute the median for each of the resamples and calculate the standard deviation of all of these medians. Even though the resamples are not genuine new samples, this estimate for the standard error of the median has good statistical

properties. A similar approach works well for estimating the standard error of such statistics as the median, lower and upper quartiles, and correlation. The bootstrap approach contrasts with simulation, sometimes termed the *parametric bootstrap*, where sampling is from a theoretical distribution.

The formula given in Subsection 1.4.4 for the SEM has the same form, irrespective of the distribution, as long as the sample has been chosen randomly. By contrast, the formula for the standard error of the median (Subsection 1.4.4) applies only when data are normally distributed. Use of the bootstrap estimate of the standard error of the median reduces also any need to look for an alternative theoretical distribution that may be a better fit to the data.

A comparison between the bootstrap estimate and the normal theory estimate allows an assessment of the seriousness of any bias that may result from non-normality. We proceed to calculate the bootstrap estimate of the standard error for the median length for the eggs that were in wrens' nests. (The *boot* package (Canty and Ripley, 2021) is needed for all bootstrap examples.) The result will serve as a check on our earlier computation.

Output from R for the lengths of eggs from the wrens' nests is:

```
## Bootstrap estimate of median of wren length: data frame cuckoos
wren <- subset(DAAG::cuckoos, species=="wren")[, "length"]
library(boot)
## First define median.fun(), with two required arguments:
##         data specifies the data vector,
##         indices selects vector elements for each resample
median.fun <- function(data, indices){median(data[indices])}
## Call boot(), with statistic=median.fun, R = # of resamples
set.seed(23)
(wren.boot <- boot(data = wren, statistic = median.fun, R = 4999))
```

```
ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = wren, statistic = median.fun, R = 4999)


Bootstrap Statistics :
    original  bias    std. error
t1*       21   0.054        0.21
```

The original estimate of the median was 21. The bootstrap estimate of the standard error is 0.215, based on 4999 resamples. Compare this with the slightly larger standard error estimate of 0.244 given by the normal theory formula in Section 1.4.4. The bootstrap estimate of the standard error will of course differ somewhat between different runs of the calculation. Also given is an estimate of the bias, i.e., of the tendency to under- or over-estimate the median.

### 1.8.4 Bootstrap estimates of confidence intervals

The usual approach to constructing confidence intervals is based on a statistical theory that relies, in part, on normal distribution assumptions. If the normal assumption is not applicable and an alternative theory is not available, the bootstrap may be helpful. Calculation for the median and for the correlation now follow.

The function `boot.ci()` implements five different methods for using bootstrap estimates of a statistic to calculate confidence intervals. The `perc` (percentile) type is the most the most commonly used. The `bca` type (bias corrected accelerated or $BC_a$) may give a substantial improvement. A sample size of $n = 15$, as in the wren egg length data that will now be used for an example, is too small for the difference to be of consequence. Efron and Tibshirani (1993) describes these methods, together with a theoretical justification for the use of the $BC_a$ method.

*Bootstrap 95% confidence intervals for the median*

We will construct 95% confidence intervals for the median of the cuckoo egg lengths in wrens' nests. Results are given for the $BC_a$ method, as well as for the percentile method that may be preferred for symmetric distributions.

The object `wren.boot` from Subsection 1.8.3 can be used as a starting point The endpoints for the 95% percentile confidence interval are calculated as the 2.5 and 97.5 percentiles of the bootstrap distribution of medians:

```
## Call the function boot.ci() , with boot.out=wren.boot
boot.ci(boot.out=wren.boot, type=c("perc","bca"))
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 4999 bootstrap replicates

CALL :
boot.ci(boot.out = wren.boot, type = c("perc", "bca"))

Intervals :
Level     Percentile            BCa
95%   (21, 22 )   (20, 21 )
Calculations and Intervals on Original Scale
Some BCa intervals may be unstable
```

*The correlation coefficient*

Bootstrap methods do not require bivariate normality. Independence between observations, i.e., between $(x, y)$ pairs, is as important as ever. Note however that a correlation of, e.g., 0.75 for a non-normal distribution may have quite different implications from a correlation of 0.75 when normality assumptions apply.

We will compute 95% confidence intervals for the correlation between `chest` and `belly` for the `possum` data frame. Results are given for $BC_a$ as well as for percentile confidence intervals.

```
## Bootstrap estimate of 95% CI for `cor(chest, belly)`: `DAAG::possum`
corr.fun <- function(data, indices)
  with(data, cor(belly[indices], chest[indices]))
set.seed(29)
```

```
corr.boot <- boot(DAAG::possum, corr.fun, R=9999)
```

```
boot.ci(boot.out = corr.boot, type = c("perc", "bca"))
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 9999 bootstrap replicates

CALL :
boot.ci(boot.out = corr.boot, type = c("perc", "bca"))

Intervals :
Level     Percentile            BCa
95%   ( 0.48,  0.71 )   ( 0.47,  0.71 )
Calculations and Intervals on Original Scale
```

*The bootstrap – parting comments*

Bootstrap methods are not a panacea. They must respect the structure of the data. Any form of dependence in the data must be taken into account. There are contexts where the bootstrap is invalid and will mislead. The bootstrap is unlikely to be satisfactory for statistics, including maximum, minimum and range, that are functions of sample extremes. The bootstrap is usually appropriate for statistics from regression analysis – means, variances, coefficient estimates, and correlation coefficients. It also works reasonably well for medians and quartiles, and other such statistics. See the references in `boot::boot`.

## 1.9 Organizing and managing work, and tools that can assist

Each R session starts in a specific *working directory*, which can be changed using the command `setwd()`, or the equivalent menu item. It can be a challenge to keep track of the resources — R packages, other software, other internet resources, relevant publications, other data — that bear on the task. Work will typically break down into several separate sub-tasks, with occasional need to move between them. Analyses need to be documented, with a description that may become part of a report or paper.

Full advantage should then be taken of tools that can help remove complexity from the analyst's mind or notebook and place it in the external world. The tools now noted can be classified, broadly, as either *Integrated Development Environment*s (IDEs), or *Graphical User Interfaces* (GUIs). R binaries come with a very basic GUI.

As an alternative to the R GUI that is supplied with R binaries, there are several GUIs that provide, also, graphical and analysis abilities. The web page `https://r4stats.com/` lists and compares a number of the possibilities. Look under Articles | Software Reviews | r-gui-comparison.

Especially for novices or infrequent users of R, a GUI interface can be helpful for handling data input, for creating simple graphs, for simple tabulation and summarization, and for fitting standard models. The balance of preference is likely to change in favor of the command line as familiarity with R increases.

Menu and command line modes of use can be mixed. for inspection and/or modification and/or for audit trail purposes. The user can examine the help page for the relevant function(s), modify the code as required, and re-execute it. For ease of presentation, our discussion will usually assume use of the command line, either directly, or from an editor window in a GUI or IDE.

*The RStudio Integrated Development Environment*

Once an R user moves from simple exercises in typing code at the command line to serious work with R, workspace management can be a challenge. For managing work with R, we recommend the highly rated RStudio IDE (go to `www.rstudio.com`). RStudio is designed to organize work by *project*s. Notwithstanding its name, the abilities available in RStudio now extend to working with languages other than R.

The RStudio IDE has extensive features that assist with:

- The organization of work into projects. Moving to a new project will at the same time change the working directory to that for the new project;
- A graphical user interface for handling a wide range of tasks that it is often most convenient to initiate from the menu;
- Maintaining a record of files that have been accessed from RStudio, of help pages accessed, and of plots. The record of files is maintained from one session of a project to the next;
- The editing, maintenance and display of code files;
- Abilities that assist reproducible reporting, using the *knitr* package for processing file formats that include code and code markup that specifies how the code is to be used and processed. Currently, most of the attention is on R Markdown and Sweave. Section 1.9.1 has further details.
- The development and maintenance of R packages.

Click on View | Show Tutorial to be taken to extensive interactive R-related tutorial material, powered by the *learnr* package.

RStudio has very extensive web-based documentation. Within the R main menu panel, click on Help | RStudio Docs to go to links to a wide range of online resources. The RStudio website has, additionally, extensive help on getting started with R.

Among alternatives to RStudio, note the ESS interface to Emacs (`http://ess.r-project.org/`), aimed at advanced users who are comfortable working with the Emacs editor. The R interface is one of several interfaces to different language or statistical package environments.

### *1.9.1 Reproducible reporting — the* knitr *package*

The R package *knitr* has extensive abilities that allow the mixing of text and R code for automatic report generation. There has been heavy reliance on its abilities in the preparation of this text.

The R code chunks are embedded within markup that includes options to control the display of code and of any computer output. When suitably processed, a

document is generated that contains the text, and the specified combination of R code and computer output. The functionality of *knitr* is automatically available, via a GUI interface, to RStudio users.

There are several different types of document where markup code can be used to control how text and other document features will appear after processing for printing. Sweave, used for the present text, is LaTeX with R markup added. R Markdown, which is Markdown with a similar style of markup added, is much simpler to learn and use. It is widely used for preparation of R-related documents and texts. Output alternatives are PDF, or HTML, or Word.

A short R Markdown document that introductory help details can be created from the File menu on the main RStudio panel. Click on New File | R Markdown. This will display a skeleton R Markdown document that can be edited as required, or processed as it stands. Clicking on the **Knit** button will generate a document that includes text as well as output from any embedded R code chunks. Note also the reference material available by clicking on Markdown Quick Reference under Help on the R main menu panel.

## 1.10 The changing environment for data analysis

This is a time when new and rich sources of data abound, with new ways to extract meaning from the available data. The use of open databases has been crucial to progress in such areas as earthquake science, the study of viruses and vaccines, modeling of epidemics, and climate science. In molecular biology, freely accessible databases have played a pivotal role in the technology that allowed a much more effective response to the Covid-19 pandemic than would have been possible two decades earlier. This sharing of data and skills, and use of modern technology, also helps in the critique of what has been published earlier.

There remain too many areas that, because potential gains are less obvious, have not put in place measures that ensure open access to the original data on which reported result are based. Data should, unless there are strong countervailing confidentiality reasons, be available both for other researchers to check and for uses that extend beyond the individual paper. A published result attracts greater confidence if it has been checked out over multiple datasets where a comparable pattern of response would be expected. Multiple datasets may together allow investigation of questions that individual datasets cannot not on their own address. Meta-analyses, or systematic reviews that aim to bring together results of a range of studies that bear on the same issue, can be really effective only if carried out with access to the original data.

### *1.10.1 Models and machine learning*

The models described in this chapter have had, to a varying extent, a theoretical motivation. The tree-based models that are the subject of Chapter 8 have, by contrast, a largely algorithmic motivation. Their use, and checks to examine whether they are serving their intended purpose do, however, involve a modeling of the pro-

cesses involved. The limited assumptions made are important. Tree-based models have been widely used in “machine learning".

*The limits of current machine learning systems*

In an era when new and rich data sources abound, there is more need than ever for tools and approaches that will assist in critical evaluation both of the data and of consequent analyses. Automation of numerical computations makes obvious sense; it frees the analyst to focus on those parts of the exercise that really do require conscious attention. Can machines extend their role beyond this? Can they acquire the skills needed to do the job of a skilled data analyst, as the term *machine learning* (or, more recently, *deep learning*) might seem to suggest?

Machine learning approaches have been very successful in, for example, the creation of automated guidance systems, and in robotics. These are, in key respects, highly automated statistical processing machines. They take large amounts of often noisy data from their sensors, then using that data as a basis for action. Extreme care is needed to avoid or reduce the risk of faults in the data inputs or in the data processing that have an immediate and obvious result. Automated aircraft guidance systems have been, for the most part, extraordinarily successful. Two Boeing 739 Max 8 crashes in 2019, with a total of 346 deaths, tell a story of human failure to take meticulous care in moving such a system to a new context.[14]

When machine learning algorithms extend their reach into social, business, and government decision making, very different types of feedback issues come into play. In contrast to automated guidance systems, there may be little or no direct feedback that can be used to check on data or analysis inadequacies. Consider, in this connection, systems that make judgments on job applicants, on rehiring and promotion, on the risk that prisoners will re-offend, on loan applications, on hedge fund investments, and on much more besides.

Issues of this type, for systems currently in place, are documented at length in O’Neil (2016, "Weapons of Math Destruction"). O’Neil cites the example of a teacher who was fired because of a low score from an automated rating system. The reason was, apparently, that under her tutelage the reading scores of her incoming fifth graders had not progressed from the inflated levels that they had been given, at a feeder school, at the previous year’s end. Automated systems must, to be effective, allow room for the human ability to step back and reflect, to learn from failures, and to correct mistakes. As O’Neil (2016) comments:

> ... it’s not enough to just know how to run a black box algorithm. You actually need to know how and why it works, so that when it doesn’t work, you can adjust.

“Interpretable machine learning” is designed to address such challenges. See for example Rudin et al. (2022).

*Traps in big data analysis*

Primary interest may be in accurate prediction. Or interest may be in the drivers of model predictions. In a study of the effectiveness of seat-belts and airbags reducing

[14] `https://en.wikipedia.org/wiki/Boeing_737_MAX#cite_note-Leap-228`, (2023-2-7).

fatalities in vehicle accidents, the interest is in the factors — seatbelt use, airbag deployment, and other factors that may influence survival. In the various approaches that have been used to predict flu trends, a primary aim has been accurate prediction. The story of Google flu trends highlights the importance of attention to the drivers, even where the chief interest is in prediction.

Google Flu Trends was launched in 2008 and updated in 2009, with the aim of using Google search queries to make accurate and timely prediction of flu outbreaks. The account that now follows is based on the Lazer et al. (2014) article "The Parable of Google Flu: Traps in Big Data Analysis."

In early 2010 the algorithm predicted an outbreak in the mid-Atlantic region of the United States two weeks in advance of official sources. Thereafter, until the publication of estimates ceased in 2013, the system consistently overestimated flu prevalence. In February 2013, the system made headlines by over-predicting doctor visits for influenza-like illness by a factor of more than two. The methodology had large ad hoc elements, and was not taking advantage of time series structure in the data, resulting in a performance that was inferior to that of forward projection methods that used Center for Disease Control data. Google shut down the prediction function in 2015. Lazer et al. (2014) use this history as a warning they term "Big data hubris":

> Big data hubris is the often implicit assumption that big data are a substitute for, rather than a supplement to, traditional data collection and analysis. ... quantity of data does not mean that one can ignore foundational issues of measurement, construct validity and reliability, and dependencies among data.

The insights that informed more conventional approaches can be incorporated into algorithms in the style of Google Flu Trends, as argued in Guo et al. (2021).

*Of mice and machine learning — missing data values*

Missing data, noted earlier in Subsection 1.1.3, are a persistent nuisance to the data analyst. A version of *Multiple Imputation* is the strategy that is usually preferred when some observations are incomplete. The `Amelia` (Honaker et al., 2011) and `mice` (van Buuren and Groothuis-Oudshoorn, 2011) packages both contain a number of functions for this purpose. See further Section 9.8.

A number of machine learning algorithms have been proposed recently to tackle missing data problems as well. Wang et al. (2021) carried out an extensive and carefully conducted simulation comparison of several popular and promising deep learning algorithms with two of the `mice` functions, one based on classification and regression trees, and the other based on random forests. The simulation study was based on repeated removal of swaths of a large survey. The findings of the study indicate that the deep learning algorithms are much faster than the functions in `mice`. Also, for the goal of predicting a particular missing value, the deep learning algorithms possess some advantage. However, the accuracy of repeated-sampling properties of the estimates (i.e. bias and variance) based on `mice` turn out to be vastly superior to those of the deep learning algorithms. Code and data for this study can be found at `https://github.com/zhenhua-wang/MissingData_DL`.

On R resources for multiple imputation, and helpful references, see Sections 9.7 and 9.8. Sangari and Ray (2021) give an overview of studies that compare imputation methods, does a comparison of their own, and has helpful brief summary details of the methods that are compared.

*Humans are not good intuitive statisticians*

While the human mind has remarkable intuitive abilities (consider the ability, without apparent effort, to recognize a familiar face), it is not a good intuitive statistician. Kahneman's ground-breaking book (Kahneman, 2013) on human judgment and decision making argues the case at length. Hence the need for training, and especially for training in forms of critical scrutiny that will help analysts to recognize and learn from their mistakes.

The Yule-Simpson 'paradox', discussed in Subsection 1.2.7, is one of a number of traps for overly simplistic use of data that highlight the limits of untrained human intuition. Smith (2014) gives a number of examples from the public sphere. The traps that such paradoxes set for data analysis results can readily get built into black box automated systems, where there may be no ready way either to discover how the black box reached apparently faulty conclusions, or to get attention to problems that have been identified.

Disturbingly often, the careful examination of published analyses reveals serious flaws. Over the course of the Covid-19 pandemic, a number of papers have been published which have a focus on matters strong public interest. They attracted critical expert scrutiny and were quickly withdrawn, though not before they had fed viral online media activity, exacerbating a Covid-19 misinformation crisis. An example is a June 2021 paper in *Vaccines*, titled "The Safety of COVID-19 Vaccinations – We Should Rethink the Policy." Among other issues, it treated reports of death from any cause following vaccination, as attributable to COVID-19, and took data on benefits from a study of a different population that examined deaths over a six-week period only, with no adjustment for the different age distributions.[15]

Researchers who wish to focus on the subject-specific aspects of their work may find close attention to the statistical methodology an annoying diversion. There is no good way to escape those challenges. Difficulties arise when, as often, research institutions have not made effective provision for access to high quality statistical advice. Our hope is that this text will be a help along the way.

### *1.10.2 Replicability is the definitive check*

Where a total study is independently repeated, we will use the terms *replicate* and *replicability*. These are used in preference to the terms *reproduce* and *reproducibility* that in some literature relates to the reproducing of the statistical analysis.

Statistical methodology, and scientific processes more generally, have to be justified by their effectiveness in answering questions of interest. Their effectiveness

[15] See `https://www.bmj.com/content/374/bmj.n1726`, (2022-02-24), for commentary.

can and should be open to empirical investigation. The critical test for laboratory science is replicability — are other scientists able to reproduce the results?

### *To what extent is published work replicable? What is the evidence?*

In important areas of laboratory science, worrying evidence has emerged that suggests that a majority of published results are not replicable. In one widely quoted case (Begley and Ellis, 2012), scientists from the bio-pharmaceutical company Amgen who attempted to reproduce 53 'landmark' cancer studies were successful in 6 cases only. The main issues appear to have been with faults in laboratory procedure and in statistical analysis (Begley, 2013). Among other such reports from attempts by industry scientists to reproduce published work see, e.g., Prinz et al. (2011), where results were marginally more encouraging.

How has this happened? Journal editors, and the scientific community at large, have fallen for the seductive notion that the statistical analysis of data, generated at one time and in one laboratory and leading to a suitably small $p$-value, is an effective replacement for the more stringent requirement that other scientists should reproduce the reported results.

### *Some major replication studies*

Concerns raised by the Begley and Ellis paper, and by other reports that point in the same direction, have been the impetus for several systematic attempts to replicate published results. Such studies are important for the light they shed on the interplay between statistical analysis issues (and especially on the role of $p$-values), and issues that relate to the funding, planning, executing and reporting of experimental studies.

Other replication studies, looking at studies more generally, have shown a much higher failure rate. The "Reproducibility: Psychology" project replicated 97 studies, published in 2008 in one of three journals. Using a simplistic $p \leq 0.05$, around 40% of the studies were successfully reproduced. See Open Science Collaboration (2015). Data and R code are available from `https://osf.io/fgjvw/`.

The median Cohen's `d` effect size for the 43 cognitive psychology papers dropped from 1.16 in the original to 0.52 in the replicate. For social psychology (54 papers), the drop was from 0.64 to 0.15. For the 47 studies with an original effect size of 0.5 or less, the median was 0.044, with a mean of 0.072. Unless features of an individual study can be identified that set it apart, an individual researcher who tries to replicate such a study might expect, on average, a difference of similar magnitude from that in the original study. Exercise 17f in Chapter 3 pursues further the analysis of data where the original effect sizes were 0.5 or less.

In an article that is primarily about the replicability of laboratory experiments in economics, Camerer et al. (2016) report on two studies on the replicability of macroeconomic studies, from 1986 and 2006. Only 13% and 23% respectively, of original results were replicable, even when the original data and code were available.

*Replicability in pre-clinical cancer research*

A $1.3 million grant from the Laura and John Arnold Foundation funded an exercise, reported in Rodgers and Collings (2021), that aimed to replicate the 53 "most impactful" pre-clinical cancer biology studies published over 2010-2012. In the end, 50 experiments from 23 papers were repeated.[16] In 92% of the completed experiments, replication effect sizes were smaller than the original, with the median effect size 85% smaller than reported for the original experiment. Barriers to repeating experiments included shortcomings in documentation of the original methodology; failures of transparency in original findings and protocols; failures to share original data, reagents, and other materials; and methodological challenges encountered during the execution of the replication experiments. These challenges meant that 50 only of the planned 193 replication experiments were able to proceed. The challenge to established practices has generated controversy in the research community, and highlighted questions on just what constitutes replication.[17]

*Studies where there may be strong social influences*

Especially where there may be strong social influences at play, it can be important to check the extent to which results generalize to different countries and cultures. The Klein, Ratliff, et al. (2014) study was specifically designed to check the extent to which results from l3 widely quoted "classic and contemporary psychology studies could be reproduced across different samples and settings internationally." Results were obtained, for each of the 13 studies, from 36 independent samples, 25 from the US and 13 from elsewhere, with a total of 6344 subjects participating. Plots showed a relatively similar pattern of between study variation for all 13 studies. Replications were successful for 10 of the 13 studies, weakly successful in one case, and unsuccessful in two cases. The data are available online (at `https://osf.io/wx7ck/?view_only=`). See also Yong (2012).

*The scientific study of scientific processes*

Results from replication studies are important in establishing the extent to which publications (even in very reputable journals) can be relied upon. They provide important evidence, in the areas covered, on the extent of problems with current scientific processes.

Replicability demands can and should go hand in hand with the fostering of informed imaginative exploration and insight. Imaginative exploration provides necessary leads into new areas of work and investigation, but requires checks against allowing an unrestrained imagination to take research down blind alleys.

The issues that these studies raise bear directly on the aims that we have set for this text. Failures in experimental design and in laboratory procedure all too frequently compromise the trustworthiness of the data used for analysis. While

[16] See the web link Errington (2021), and the papers Errington et al. (2021), and Rodgers and Collings (2021).

[17] Nosek and Errington (2020)

experimental design and more general data collection issues are not a particular focus of this text, we do want to emphasize their importance.

The package *ReplicationSuccess* (Held et al., 2021) has functionality that is designed to assist in planning and analyzing replication studies.[18]

A key component of replicability is the reproducibility of the analysis, whether with the data used for the published paper, or with new data. The technology that is now available leaves little excuse for failure to attend both to the maintenance of data records through time, and to reproducible reporting.

*Would lowering the p-value threshold help?*

There have in some quarters been calls to lower the $p$-value significance threshold to $\alpha = 0.005$ (Benjamin et al., 2018). Ioannidis (2018) argues that such a move should be considered only as a temporary recourse, put in place until such time as more durable solutions emerge. A different threshold, perhaps as low as $10^{-6}$, is suggested for observational studies. Especially for large studies, it is important to report estimated effect sizes. An estimated effect size of 0.1 gives, for a one-sample $t$-test with $n$=800, a $p$-value that is a little under 0.005.

Other types of summary statistic, such as Bayes Factors, should be used where appropriate. For Bayes Factors, details of the prior that was assumed would be an essential accompaniment. Given the large element of individual judgment involved in the choice of a prior, it is hard to see how Bayes Factors or other such statistics could readily be a complete replacement for $p$-values. Irrespective of changes that may be made to reporting protocols, there is a clear need for greater attention to statistical literacy as part of research training.

*Peer review at the study planning stage*

Starting in 2013, the Center for Open Science (see `http://cos.io/rr`) has facilitated the submission of registered reports (RRs) for review prior to observing study outcomes, with more than 300 journals now offering this route to publication. As noted on the COS website (`http://cos.io/rr`)

> Manuscripts that survive pre-study peer review receive an in-principle acceptance that will not be revoked based on the outcomes, but only on failings of quality assurance, following through on the registered protocol, or unresolvable problems in reporting clarity or style.

Results that appear in published work provide later researchers with starting points on which to build. They may also have an important role in drawing attention to lines of inquiry that have been pursued earlier and have lead nowhere.

Scheel et al. (2021) describe a comparison between 71 papers that had used the RR mechanism as of November 2018 with with a random sample of 152 hypothesis-testing studies from the standard literature in psychology. The authors found 96% (146/152) of 'positive' results in the standard literature, as opposed to 44% (31/71) for RRs. This suggests strong selection effects in what appears in the published literature.

[18] Enter `vignette('ReplicationSuccess', package='ReplicationSuccess')` to see the extensive commentary in the package vignette.

## 1.11 Further, or supplementary, reading

Extensive R-related tutorial material is available online. The Learn R web page on the website `https://www.r-bloggers.com/` has extensive tutorial content, and includes links to content that is available elsewhere (Galili, 2015). See also `https://r4ds.had.co.nz/`.

Reference was made earlier to the extensive interactive R-related tutorial material, powered by the *learnr* package, that can be accessed by clicking on View | Show Tutorial on the main RStudio menu panel.

Introductions to R include Dalgaard (2008). Braun and Murdoch (2021) is an elementary introduction to the R programming language. More technical and detailed accounts of the R language include: Aphalo (2020), Chambers (2008), Matloff (2011), Wickham (2015), and Wickham (2016).

Kahneman (2013) gives important insight into human propensities for misinterpreting statistical data. O'Neil (2016) has insightful commentary on the nature and limitations of mathematical models, and on the limitations of machine learning technology in current use as a replacement for human judgement. They become, if trained on data that has inherent biases, "weapons of math destruction"! Thaler (2015) is an extended commentary on the mismatch between the decision making processes of the idealized agents of the models of classical economics (he calls these 'Econs') and human agents, with serious implications for economic and social policy.

On planning experimental trials, see Robinson (2000). The detailed discussion, with detailed practical examples, has much to say that is relevant to studies more generally. See also Cox (1958).

Papers that comment on statistical presentation issues, and on deficiencies in the published literature, include Andersen (1990), Maindonald (1992), Wilkinson and Task Force on Statistical Inference (1999), and Allison et al. (2016). On errors in the interpretation of $p$-values, see Greenland et al. (2016), and the extensive list of references given in that paper. Wilkinson and Task Force on Statistical Inference (1999) makes helpful comments on the planning of data analysis, on the role of exploratory data analysis, and more.

On formal hypothesis testing, see Gigerenzer (1998) and Wilkinson and Task Force on Statistical Inference (1999). The misuse of $p$-values has been a strong focus in the debate over reproducibility. Greenland et al. (2016) make the point that $p$-values test the total context in which work has been undertaken. The role of failures in experimental design and execution has not received the same level of attention in the scientific literature as issues that relate to the use and meaning of $p$-values.

Ioannidis (2005) has been a key reference in the debate on reproducibility. Changes are needed in the conduct of major areas of science — in publication, in management, and in reward systems. Nosek, Alter, et al. (2015) and Mogil and Macleod (2017) make detailed proposals. Allison et al. (2016) note common errors. They discuss roadblocks to getting corrections or retractions published. Button et al. (2013) discuss in detail the issues posed by the use of small under-powered studies. They note the huge transformation in the reliability of gene association studies that has

resulted from the collaboration of groups of researchers to increase the sample size and minimize the labor and resource input of any one contributor.

Gigerenzer et al. (1989) discuss the history that has led to differences in styles and approaches to inference in different area of statistical application. Wonnacott and Wonnacott (1990) have an elementary account of Bayesian Methodology. See also Gelman, Carlin, et al. (2003). Gigerenzer (2002) demonstrates the use of Bayesian arguments in several important practical contexts, including AIDS testing and screening for breast cancer. Chapters 3 and later of Ellenberg (2015) are largely occupied with statistical issues, using real-world examples as points of departure. Note, in particular, Part II, titled 'Inference', moving finally to Bayesian inference. Kass and Raftery (1993) remains an informative source of comment on the history of Bayes Factors, and on their use and interpretation.

Books and papers that set out principles of good graphics include Robbins (2012). See also the imaginative uses of R's graphical abilities that are demonstrated in Murrell (2011). Chang (2013) is a helpful resource for *ggplot2*. Bowman et al. (2019) discuss aspects of graphical presentation that are too easily ignored.

## 1.12 Exercises

1. The data frame `DAAG::orings` has details of damage that had occurred in US space shuttle launches prior to the disastrous Challenger launch of January 28, 1986. Observations in rows 1, 2, 4, 11, 13, and 18 were shown in the pre-launch charts used in deciding whether to proceed with the launch, with remaining rows omitted.

   Compare plots of `Total` incidents against `Temperature`: (i) including only the observations shown in the pre-launch charts; and (ii) using all rows of data. What did the full set of data strongly suggest that was less clear from the plot that showed only the selected rows?
2. For the purposes of the exercises that follow, type `?sample` at the R command line, and take a careful look at the help page that appears. Note in particular the arguments `size` and `replace`.
   a. Write out and execute the code required to select a random sample of 25 numbers, *with replacement*, from the numbers from 100 through 200.
   b. Write and execute code that randomly assigns 30 patient labels in such a way that 20 patients are assigned to a treatment group, and 10 patients are to a control group.
   c. Suppose 30 patients are to be assigned randomly to treatment, control and placebo groups. Write and execute code that randomly assigns 10 patient labels to each of the three groups.
3. For the data frame `DAAG::possum`
   a. Use the function `str()` to get information on each of the columns.
   b. Using the function `complete.cases()`, determine the rows in which one or more values is missing. Print those rows. In which columns do the missing values appear?

4. The following plots four different transformations for the columns `brain` and `body` in the `Animals` dataset. What different aspects of the data do these different graphs emphasize? Consider the effect on low values of the variables, as contrasted with the effect on high values.

```
Animals <- MASS::Animals
manyMals <- rbind(Animals, sqrt(Animals), Animals^0.1, log(Animals))
manyMals$transgp <- rep(c("Untransformed", "Square root transform",
  "Power transform, lambda=0.1", "log transform"),
rep(nrow(Animals),4))
manyMals$transgp <- with(manyMals, factor(transgp, levels=unique(transgp)))
lattice::xyplot(brain~body|transgp, data=manyMals,
  scales=list(relation='free'), layout=c(2,2))
```

5. Calculate the following correlations:

```
with(Animals, c(cor(brain,body), cor(brain,body, method="spearman")))
with(Animals, c(cor(log(brain),log(body)),
  cor(log(brain),log(body), method="spearman")))
```

   Comment on the different results. Which is the most appropriate measure?

6. Use the function `abbreviate()` to obtain six-character abbreviations for the row names in the data frame `DAAG::cottonworkers`. Plot `survey1889` against `census1886`, and plot `avwage*survey1889` against `avwage*census1886`, in each case using the six-letter abbreviations to label the points. What indications, different from those of the 1886 survey data, do the 1886 census data give of the contributions of the different classes of worker to the overall wage bill?

7. Plot a histogram of the `earconch` measurements for the `DAAG::possum` data. The distribution should appear *bimodal* (two peaks). This is a simple indication of clustering, possibly due to sex differences. Obtain side by side boxplots of the male and female `earconch` measurements. How do these measurement distributions differ? Can you predict what the corresponding histograms would look like? Plot them to check your answer.

8. For the data frame `DAAG::ais`, draw graphs that show how the values of the hematological measures (red cell count, hemoglobin concentration, hematocrit, white cell count, and plasma ferritin concentration) vary with the sport and sex of the athlete.

9. In the data frame `DAAG::cuckoohosts`, column names with first letter `c` refer to cuckoos, while names starting with `h` refer to hosts. Plot `clength` against `cbreadth`, and `hlength` against `hbreadth`, all on the same graph and using different colors to distinguish points for the cuckoo eggs from points for the host eggs. Join the two points that relate to the same host species with a line. What does a line that is long, relative to other lines, imply? Code that you may wish to use or adapt is:

```
usableDF <- DAAG::cuckoohosts[c(1:6,8),]
nr <- nrow(usableDF)
with(usableDF, {
  plot(c(clength, hlength), c(cbreadth, hbreadth), col=rep(1:2,c(nr,nr)))
  for(i in 1:nr) lines(c(clength[i], hlength[i]), c(cbreadth[i], hbreadth[i]))
  text(hlength, hbreadth, abbreviate(rownames(usableDF),8), pos=c(2,4,2,1,2,4,2))
})
```

10. The following uses a graph to illustrate least-squares estimation of the mean:

```
## Take a random sample of 100 values from the normal distribution
x <− rnorm(100, mean=3, sd=5)
(xbar <− mean(x))
## Plot, against `xbar`, the sum of squared deviations from `xbar`
lsfun <− function(xbar) apply(outer(x, xbar, "−")^2, 2, sum)
curve(lsfun, from=xbar−0.01, to=xbar+0.01)
```

    Write code that repeats the calculations 500 times, storing the sample means in a vector `avs` and the sample medians in a vector `meds`. Create the plot:

```
boxplot(avs, meds, horizontal=T)
```

    Interpret what you see in the light of what Subsection 1.4.4 had to say about the sampling distribution of the median.

11. In the data frame `DAAG::nswdemo`, plot `re78` (1978 income) against `re75` (1975 income). What features of the plot call for attention, if the interest is in finding a relationship?
    a. Restricting attention to observations for which both `re78` and `re75` are nonzero, plot `log(re78)` against `log(re75)`, and fit a trend curve. Additionally, fit a regression line to the plot. Does the regression line accurately describe the relationship. In what respects is it deficient?
    b. Now examine the diagnostic plot that is obtained by using `plot()` with the regression object as parameter. What further light does this shed on the regression line model?
12. The `MASS::galaxies` data frame gives speeds of 82 galaxies (see the help file and the references listed there for more information). Construct a density plot for these data. Is the distribution strongly skewed? Is there evidence of clustering?
13. Figure 1.5B plotted brain weight (units of 100gm) versus body weight (units of 100kg), for 28 animals, using logarithmic scales. Copy the plot and use a ruler or other straight edge to draw a line through the main body of points. Use the ratio of vertical to horizontal distance, between the points where the line intersects the left and right boundaries of the plotting region, to estimate the slope of the line. The slope can be interpreted as the ratio between the relative rate of increase of brain weight, and that for body weight. For a body weight increase of 5% (this counts for this purpose as a small increase), what increase might be expected in brain weight? Compare the line that you have drawn with the regression line for `log(brain)` on `log(body)`.
14. An experimenter intends to arrange experimental plots in four blocks. In each block there are seven plots, one for each of seven treatments. Use the function `sample()` to find four random permutations (i.e. orderings or arrangements) of the numbers 1 to 7 that will be used, one set in each block, to make the assignments of treatments to plots.
15. The following are total numbers of aberrant crypt foci (abnormal growths in the colon) observed in seven rats that had been administered a single dose of the carcinogen azoxymethane and sacrificed after six weeks (thanks to Ranjana Bird, Faculty of Human Ecology, University of Manitoba for use of these data):

```
87 53 72 90 78 85 83
```

Calculate the sample mean and variance. Is the Poisson model appropriate? To investigate how the sample variance and sample mean differ under the Poisson assumption, repeat the following simulation experiment several times:

```
x <- rpois(7, 78.3)
mean(x); var(x)
```

16. The following simulates 100 normal random variates from each of (i) a normal distribution and $t$-distributions with (ii) 4 and (iii) 2 degrees of freedom. Run the code several times, on each occasion counting the number of points that appear out beyond the whiskers in each of the two boxplots.

```
nvals100 <- rnorm(100)
heavytail <- rt(100, df = 4)
veryheavytail <- rt(100, df = 2)
boxplot(nvals100, heavytail, veryheavytail, horizontal=TRUE)
```

Comment on the differences between the three distributions in the number of points that are tagged as potential outliers.

17. Use `t.test()` to test the null hypothesis that the mean is 0 for random samples of 10 values from a normal distribution:

    a. with mean 0 and standard deviation 2;
    b. with mean 1.5 and standard deviation 2.

    Finally, write a function that generates a random sample of $n$ numbers from a normal distribution with mean $\mu$ and standard deviation 1, and returns the $p$-value for the test that the mean is 0.

18. Use the function that was created in Exercise 17 to generate 50 independent $p$-values, all with a sample size $n = 10$ and with mean $\mu = 0$. Use `qqplot()`, with the argument setting `x = qunif(ppoints(50))`, to compare the distribution of the $p$-values with that of a uniform random variable, on the interval [0, 1]. Comment on the plot.

19. The following function draws, when called with `n=1000`, ten boxplots of random samples of 1000 from a normal distribution, and ten boxplots of random samples of 1000 from a $t$-distribution with 7 degrees of freedom:

```
boxdists <- function(n=1000, times=10){
  df <- data.frame(normal=rnorm(n*times), t=rt(n*times, 7),
  sampnum <- rep(1:times, rep(n,times)))
   lattice::bwplot(sampnum ~ normal+t, data=df, outer=TRUE, xlab="",
                   horizontal=T)
}
```

Run the function, first with `n=1000`, and then with `n=200`. Refer back to the Subsection 1.4.2 note on heavy-tailed distributions, and comment on the different numbers of points that are flagged as possible outliers.

20. Run the following code:

```
a <- 1
form <- ~rchisq(1000,1)^a+rchisq(1000,25)^a+rchisq(1000,500)^a
lattice::qqmath(form, scales=list(relation="free"), outer=TRUE)
```

Repeat, first with `a = 0.5`, and then with `a = 0.33`. Which choice of `a` appears to give the best approximation to a normal distribution.

21. The following code generates random normal numbers with a sequential dependence structure:

```
y <- rnorm(51)
ydep <- y1[-1] + y1[-51]
acf(y)       # acf plots `autocorrelation function'(see Chapter 6)
acf(ydep)
```

Repeat this several times. There should be no consistent pattern in the acf plot for different random samples `y`, and a fairly consistent pattern in the acf plot for `ycor` that reflects the correlation that is introduced by adding to each value of `y` the next value in the sequence.

22. Assuming that the variability in egg length in the cuckoo eggs data is the same for all host birds, obtain an estimate of the pooled standard deviation as a way of summarizing this variability. [Hint: Remember to divide the appropriate sums of squares by the number of degrees of freedom remaining after estimating the six different means.]

23. Write a function that simulates simple linear regression data from the model

$$y = 2 + 3x + \varepsilon$$

where the noise terms are independent normal random variables with mean 0 and variance 1.

Using the function, simulate two samples of size 10. Consider two designs: first, assume that the $x$-values are independent uniform variates on the interval $[-1, 1]$; second, assume that half of the $x$-values are -1's, and the remainder are 1's. In each case, compute slope estimates, standard error estimates and estimates of the noise standard deviation. What are the advantages and disadvantages of each type of design?

24. * The following code can be used to obtain and plot, for a specific choice of `eff` and `N`, a set of simulated $p$-values as in Figure 1.20 in Subsection 1.3.6:

```
ptFun <- function(x,N)pt(sqrt(N)*mean(x)/sd(x), df=N-1, lower.tail=FALSE)
simStat <- function(eff=.4, N=10, nrep=200, FUN)
    array(rnorm(n=N*nrep*length(eff), mean=eff), dim=c(length(eff),nrep,N)) |>
      apply(2:1, FUN, N=N)
pval <- simStat(eff=.4, N=10, nrep=200, FUN=ptFun)
# Suggest a power transform that makes the distribution more symmetric
car::powerTransform(pval)   # See Subsection 2.5.6
labx <- c(0.0001, 0.001, 0.005, 0.01, 0.05, 0.1, 0.25)
bwplot(~I(pval^0.2), scales=list(x=list(at=labx^0.2, labels=paste(labx))),
       xlab=expression("P-value (scale is "*p^{0.2}*")") )
```

a. Use the following to compare the distribution from direct use of `simStat()` with that from the data frame `df200` that was created in Subsection 1.3.6:

```
pvalDF <- subset(df200, effsize==0.4 & N==10)$stat
plot(sort(pval^0.2), sort(pvalDF^0.2))
abline(0,1)
```

Compare with the plot of `sort(pval)` against `sort(pvalDF)`. Which plot is more useful, and why?

b. Repeat the calculations using `eff2stat()` with the argument 'FUN' set to calculate two-sided $p$-values. (In the argument `FUN`, replace `mean(x)` by `abs(mean(x))`) and double the value returned by `pt()`.)

c. Repeat with 'FUN' set to simulate sample effect sizes, thus:

```
## Estimated effect sizes: Set `FUN=effFun` in the call to `eff2stat()`
effFun <- function(x,N)mean(x)/sd(x)
  # Try: `labx <- ((-1):6)/2`; `at = log(labx)`; `v = log(labx)
## NB also, Bayes Factors: Set `FUN=BFfun` in the call to `eff2stat()`
BFfun <- function(x,N)BayesFactor::ttest.tstat(sqrt(N)*mean(x)/sd(x), n1=N,
                                               simple=T)
  # A few very large Bayes Factors are likely to dominate the plots
```

d. Create an equivalent of `effFun()` that returns $t$-statistics. For one combination of `eff` and `N`, plot the 100 simulated values.

25. Using the data frame `cars` (*datasets*), plot `dist` (i.e., stopping distance) versus `speed`. Fit and plot the a line. Then try fitting and plotting a quadratic curve. Does the quadratic curve give a useful improvement to the fit? [Readers who have studied the relevant physics might develop a model for the change in stopping distance with speed, and check the data against this model.]

26. The distance that a body, starting at rest, falls under gravity in $t$ seconds is well approximated as $d = \frac{1}{2}gt^2$, where $g \simeq 9.8m.sec^{-2}$. The equation can be modified to take account of the effects of air resistance, which will vary with barometric pressure and other atmospheric conditions. How useful will a time-distance relationship for a human dummy that falls from a height of some thousands of meters be for predicting the time-distance relationship for another dummy, or for a human, falling at another time from a similar height? How is the challenge similar to, and how different from, from, the use of the lawn roller data of Subsection 1.5.1 for such indications as it can provide that are relevant to another time and place? [Humans have very occasionally survived falls from such heights. See `www.greenharbor.com/fffolder/ffresearch.html`]

27. The dataset `MPV::radon` gives percentages of radon released when radon-enriched water was used in showers with different sized orifices. The temperatures were:

```
(degC <- setNames(c(21,30,38,46),paste('rep',1:4)) )
```

```
rep 1 rep 2 rep 3 rep 4
   21    30    38    46
```

a. Use the following code to manipulate the data into the form required and plot the observations against temperature for each orifice diameter:

```
radonC <- tidyr::pivot_longer(MPV::radon, names_to='key',
                              cols=names(degC), values_to='percent')
radonC$temp <- degC[radonC$key]
lattice ::xyplot(percent ~ temp|factor(diameter), data = radonC)
```

(On *tidyr* and other *tidyverse* packages, see Section A.2.5.)

b. What common pattern do you observe in each of the six time plots? What does this tell you about the measurements?

c. A clearer pattern can be obtained by plotting the residuals or scaled residuals. The latter are obtained by subtracting the treatment means and then dividing by the treatment standard deviations, as in

```
matplot(scale(t(MPV::radon[,-1])), type="l", ylab="scaled residuals")
```

Execute this code and then modify it to obtain the time plots of the raw residuals.

d. The raw residuals can be calculated by applying `aggregate()` to the case-by-variable form of the data obtained in part (a):

```
radon.res <- aggregate(percent ~ diameter, data = radonC, FUN = scale,
    scale = FALSE)
```

The technique of part (a) involving the `pivot_longer()` function yields the residual time plots. Modify the code above appropriately to obtain the scaled residual time plots.

28. * In the unusual case where the variance is known, the function `AIC()` defines the AIC statistic for normal theory regression models as:

$$\text{AIC} = \frac{\text{RSS}}{\sigma^2} + 2p + \text{const}, \quad \text{where} \quad \text{const} = n(1 + log(2\pi))$$

Show that, if $\sigma^2$ is replaced by the maximum likelihood estimate $\frac{\text{RSS}}{n}$, this reduces to Equation 1.5.

29. Conduct the following simulation experiment to verify the robustness property of the Mean Absolute Deviation relative to the sample standard deviation and the Interquartile Range.

a. Generate 100 independent standard normal random variates using the `rnorm()` function assigning them to an object called `x`.

b. Estimate $\sigma$ using `sd()`, `IQR()` and `mad()`, using the results in Subsection 1.3.4. Observe how close each estimate is to the true value ($\sigma = 1$).

c. Now replace the first twenty observations in `x` with independent normal variates having mean 0 and standard deviation 10. Re-estimate $\sigma$ for this contaminated dataset using each of the three estimators. Which estimate is closest to the value 1?

d. Repeat the preceding experiment using different numbers of contaminated data points and different standard deviations.

30. Compare the two plots:

```
diamonds <- ggplot2::diamonds
with(diamonds, plot(carat, price, pch=16, cex=0.25))
with(diamonds, smoothScatter(carat, price))
```

Why does the first plot give the impression that the price values are truncated at the upper end, while the second plot suggests that this may not be real? Check the help page `?adjustcolor`, and repeat the first plot with the argument `col=adjustcolor('blue', alpha=0.1)`. What does this tell you that was not obvious from the first two plots?

31. The following function can be used to obtain the data for Figure 1.24:

```
t2BF <- function(t, n=10, rscale="medium", mu=0){
if(!(length(mu)%in%(1:2)))stop("mu must be of length 1 or 2")
if(length(mu)==1){
    BayesFactor:: ttest.tstat (t=t, n1=n, rscale=rscale, simple=TRUE)} else {
     null0 <- BayesFactor::ttest.tstat(t=t, n1=n, nullInterval=mu,
                                       rscale=rscale,simple=TRUE)
alt0 <- BayesFactor::ttest.tstat(t=t, n1=n, nullInterval=mu, rscale=rscale,
                                 complement=TRUE, simple=TRUE)
alt0/null0}
}
```

The following is a cut-down version of the needed calculations:

```
bfDF <- expand.grid(p=c(0.05,0.01,0.002),n=c(10,40,160))
bfDF['t'] <- apply(bfDF,1,function(x){qt(x['p']/2, df=x['n']-1, lower.tail=FALSE)})
bfDF['bf'] <- apply(bfDF,1,function(x)t2BF(t=x['t'], n=x['n'], mu=0,
                    rscale="medium"))
bfDF['bfw'] <- apply(bfDF,1,function(x)t2BF(t=x['t'], n=x['n'], mu=0,
                    rscale="wide"))
## Now specify a null interval
bfDF['bfInterval'] <- apply(bfDF,1,function(x)t2BF(t=x['t'], n=x['n'],
                    mu=c(-0.1,0.1),rscale="medium"))
bfDF['bfIntervalw'] <- apply(bfDF,1,function(x)t2BF(t=x['t'], n=x['n'],
                    mu=c(-0.1,0.1),rscale="wide"))
```

Comment (i) on the effect of changing from `rscale="medium"` to `rscale="wide"`, and (ii) on the effect of the change from a point null to the specific null interval that was chosen.

32. *The function `bayes_inference` in the *statsr* package implements, in addition to the JZS priors that are implemented in the *BayesFactor* package, the JUI (Jeffreys Unit Information) prior that replaces the Cauchy distribution for the mean with a normal distribution. Also, it returns Bayesian credible intervals which (unlike frequentist confidence intervals) can be interpreted as a probability that the statistic of interest lies within the interval. Try the following calculations:

```
df <- data.frame(d = with(datasets::sleep, extra[group==2] - extra[group==1]))
library(statsr)
BayesFactor::ttestBF(df$d, rscale=1/sqrt(2))   # Make default setting explicit
bayes_inference(d, type='ht', data=df, statistic='mean', method='t', null=0,
                rscale=1/sqrt(2), alternative='twosided', prior_family = "JZS")
  # Set `rscale=1/sqrt(2)` (`bayes_inference()` default is 1.0)
  # `prior_family = "JZS"` is the default
# Compare with `prior_family = "JUI"`, with default settings
bayes_inference(d, type='ht', data=df, statistic='mean', method='t',
                alternative='twosided', null=0, prior_family = "JUI")
```

a. Compare the 95% Bayesian credible interval that is obtained with the corresponding frequentist confidence interval. What difference has it made, to the Bayes Factor and to the Bayesian credible interval, to replace the JZS family by the JUI family?

b. Repeat the comparisons for the comparison between `horsebean` and `linseed` feeds in the `statsr:chickwts` dataset. (See the example in the help page `?statsr:chickwts`.)