1

Learning from data, and tools for the task

This text is designed as an aid, for learning and for reference, in the navigation of a world in which unprecedented new data sources, and tools for data analysis, are pervasive. It uses R for the calculations and for graphical display, but is not a book about R. Rather, it will use real examples to demonstrate an informed and critical use of the analytical abilities that the R system offers.

A large body of statistical theory underpins the analyses that R packages implement. While we will from time to time refer to theoretical results, it is not our purpose to provide a systematic account of statistical theory. Our focus is different, to teach by example a style of analysis and critique that can turn meaningful data into defensible analysis results.

We assume enough familiarity with elementary statistical ideas and terminology that readers will be comfortable with mention of such terms as *standard deviation*, *normal* distribution, *independence*, and *dependence*, in some cases used initially in advance of such formal definition as we provide. Our primary concern is with the role and meaning of this language in practical data analysis.

The opening section describes issues and interests that have motivated us, especially as they affect changes made for this new edition. Later sections discuss: data assembly, analysis, and interpretation issues; tools for, and the uses of, graphical display; data summary statistics; distributions for the random component of models; and tools that will be helpful for organizing and managing work, noting in particular the *Integrated Development Environment* (IDE) that RStudio provides.

1.1 The changing world of statistical applications

Science, as we know it, is a product of the modern world. Phenomena become part of established science once we know the circumstances under which they will re-occur. The process is straightforward for events that can be predicted with the same assurance as, for example, the occurrence of an eclipse of the sun. Not everything is thus predictable. Does exposure to words that describe common disabilities of old age cause a younger person to walk more slowly? Data have to be obtained from a suitable experiment or series of experiments.

Or it may be necessary to rely on whatever data are already available. How effective were airbags, in cars that were involved in police reported car crashes in the USA over 1997-2002, in reducing the risk of death? 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. (See the help pages `?DAAG::nassCDS` and `?gamclass::FARS`).

Disturbingly often, the careful examination of published analyses reveals serious flaws. The Reinhold and Rogoff study "Growth in time of debt" gained attention in the the news media because crucial data, with a large effect on results, had been accidentally left out. Ironically, that was the most excusable of the faults with their use of the data.

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, however, 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.

1.1.1 Models, algorithms, and machines that learn

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 serve useful purposes. 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, NZ, where the first author now lives. For that purpose, engineers require mathematical equations that can be used as models of 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.

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 processes 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 must take large amounts of often noisy data from their sensors, and then use that data as a basis for action. A difference from most of the areas of statistical application that will be discussed in this text is that major faults in the data inputs or in the data processing are likely to have an immediate and obvious result — the system will misbehave! The major risks are from faults that show up in unusual circumstances, perhaps putting human life at risk.

The absence of immediate feedback becomes a serious issue when machine learning algorithms extend their reach into social, business, and government decision making. In contrast to automated guidance systems, there may be limited direct control on the data collected, and little or no opportunity for the direct feedback that will often make data or analysis inadequacies obvious. Examples include 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 many other areas of human life.

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.

Prediction, or explanation?

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 fatalities in vehicle accidents, the interest is in the factors — seatbelt use, airbag deployment, and other factors that may influence survival. In a teacher rating system, the focus is on providing accurate and objective ratings. Use of data in ways that are not transparent, and that are not open to scrutiny, may, as with the teacher rating system, seriously compromise both fairness and effectiveness.

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. Lazer et al. (2014) warn against what 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.

Even when the primary interest is in prediction, it is often important to know what the drivers are. Comments in O'Neil (2016) are apt:

> ... 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.

Are humans 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 judgement 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 1.3.5, 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.

1.1.2 Science and statistics

Statistical methodology, and scientific processes more generally, have to be justified by their effectiveness in answering questions of interest. Their effectiveness can and should be open to empirical investigation. The critical test for laboratory science is reproducibility — are other scientists able to reproduce the results? In important areas of laboratory science, worrying evidence has emerged that suggests that a majority of published results are not reproducible. 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. Independent replication at another time and place 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.

As will be argued in the next chapter, it is wrong to treat p-values as absolute measures of the weight of evidence. Rather, they shift the weight of evidence. Comments from Fisher (Fisher, 1935, pp 13–14), who pioneered the use of p-values, are apt:

> . . . 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.

The meaning and role of p-values therefore get closer scrutiny than in earlier editions. A new and interesting twist comes from contexts where there may be many p-values — hundreds, or thousands, or more. New and fruitful ways ways then emerge to martial the evidence that they provide.

Systematic attempts to reproduce scientific results

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. The "Psychology Reproducibility" (Open Science Collaboration, 2015) project found that, depending on the measure used, around 40% of the 98 results investigated appeared reproducible. A study of this same type, for papers from pre-clinical cancer research, is at the time of writing well advanced (Errington et al., 2014).

The issues that these studies raise bear very directly on the aims that we have set for ourselves for this text. Failures in experimental design and in laboratory procedure all too frequently compromise the trustworthiness of the data used for analysis. While experimental design and more general data collection issues are not a particular focus of this text, we do want to emphasize their importance.

Data is the raw material of statistical analysis. Historical data may have important insights to offer, or even be crucial to issues of current interest. A key component of reproducibility 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 issues of the maintenance of data records through time, and to reproducible reporting. For this text, we have used the `knitr` package, which makes it possible to "knit" text and R code together in one or more documents which are then processed[1] to give the output for this text.

1.2 Statistical analysis questions, aims and strategies

Different questions, asked of the same data, will demand different analyses. Questions of interest may, given the available data, be unanswerable. Data on house prices in London, England, may not have much relevance, if the interest is in house prices in New York or Paris or London in Ontario.

Questions should be structured with a view to the intended use of results. Is the aim scientific understanding, perhaps as in the example discussed below to determine whether cuckoos do really match the eggs that they lay in the nests of other birds to the size and color of the host eggs? Or is the aim to predict, based on recent prices in the area and on house size, the price that purchasers may be willing to pay?

1.2.1 Terminology — variables, factors, and more!

The word "variable" will be used when values are on an "interval" scale (differences between points on the scale are meaningful), while "factor" will be used when values are on a categorical scale. Thus, in the data frame `DAAG::kiwishade`, `yield` is a variable, while `block` is a factor with levels `east`, `north`, and `west`. More generally, a factor may represent values on an ordinal scale. Thus the factor `tint` in the data frame `DAAG::tinting` has levels `no`, `lo`, and `hi`, which it is appropriate to treat as ordered.

Variables and factors can both appears as "terms" in a model. Other types of "terms" include interaction terms that account for, e.g., effects that are not the sum of the separate effects of two factors.

[1] The R function `knit()` starts the process

"Factor" and "term" are both used in a more general sense, as well as in the technical senses described in the previous two paragraphs. In the statement that "there are many factors that influence health", the more general usage is clearly in mind.

Where interval scales have an absolute zero point (measurement variables such as height and weight are examples), they may be referred to as "ratio" scales.

1.2.2 Graphical comparisons

Cuckoos lay eggs in the nests of other birds. The eggs are then unwittingly adopted and hatched by the host birds. Newton (1893-1896, p. 123) states that the eggs that the cuckoos lay tend to match the eggs of host bird in size, shape and color. Latter (1902) collected the data shown in Figure 1.1 in order to investigate Newton's claims. Two different forms of graphical display are used — a dotplot display, and the more summary boxplot form of display that is described in Figure 1.2 in Section 1.3.

Figure 1.2A shows the raw data values. Figure 1.2B focuses on statistics that are designed to indicate the shape of the distribution, and give a rough idea of how variation between groups compares with variation between groups. 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 con77sidered outliers. The 5 "outliers" that are indicated for meadow pipit may be in large part due to an unusually narrow box, and hence an exaggerated willingness to flag points as potential outliers.

Simplified code for Figure 1.1 is:

```
library(latticeExtra)
## Compare dotplot() with bwplot(), both from lattice package
cuckoos <- DAAG::cuckoos
gph <- dotplot(species ~ length, xlab="Length of egg (mm)",
               data=cuckoos, alpha=0.4)
av <- with(cuckoos, aggregate(length, list(species=species), FUN=mean))
gph + latticeExtra::as.layer(dotplot(species ~ x, pch=3, cex=1.25,
                                     col="black", data=av))
bwplot(species ~ length, xlab="Length of egg (mm)", data=cuckoos,
       scales=list(y=list(alternating=0)))
  # alternating=0; omit y-axis labels
```

Fuller details of the code are in the web supplement.

Figure 1.1 strongly suggests 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. 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 host. Note that apart from several outlying egg lengths in the meadow pipit nests, the length variability within each host species' nest is fairly uniform.

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 in the course of different investigations. Data on the host eggs are from various sources.
- The host egg length for the wren is an indicative length from Gordon (1894).

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 data frame* `cuckoos` *with information from the data frame* `cuckoohosts` *(both in the* `DAAG` *package).*

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)

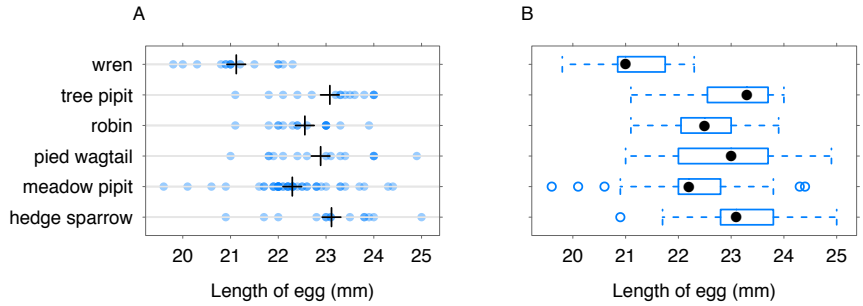


Figure 1.1: Dotplot (Panel A) and boxplot (Panel B) displays of cuckoo egg lengths. In Panel A, points that overlap have a more intense color. Means are shown as +. The boxes in Panel B take in the central 50% of the data, from 25% of the way through the data to 75% of the way through. Data are from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

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 model-based comparison between lengths for the different species is important as an aid to informed judgment.

1.2.3 Formal model-based comparison

For comparing lengths between species, 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"]],11))
```

```
hedgsparrow meadowpipit piedwagtail       robin  tree pipit        wren
      23.11       22.29       22.89       22.56       23.08       21.12
```

Commonly, a normal distribution will be used to describe the random variation, and will be fitted using least squares. The species means are then the *fitted values*, while differences from those means are the *residuals*. 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
```

A check is desirable on whether the standard deviation, used to measure variation about the species means, varies between species. The boxes in Figure 1.1, whose widths relate to a measure of variability that is an alternative to the standard deviation, hint that variation may be greater for the Pied Wagtail than for other species.

Results that can be trusted

Comments in Tukey (1997) merit close attention. It is important to separate model development steps from inference. Models aim to give real world descriptions. Exposure to diverse challenges will build (or destroy!) confidence in model-based inferences. A large part of our task in this text is to suggest effective forms of challenge. Specific types of challenge may include:

- For experiments, is the design open to criticism?
- Look for biases in processes that generated the data.
- Look for inadequacies in laboratory procedure.
- Use all relevant graphical or other summary checks to critique the model that underpins the analysis. Think carefully what test data might provide an adequate challenge, given the intended use of results.
- For experimental data, have the work replicated independently by another research group, from generation of data through to analysis,

It is important that analysts search out all available information about the processes that generated the data, and consider critically how this may affect the reliance placed on it. We should trust those results that have survived thorough and informed challenge.

Observational versus experimental data

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 1.4.2. Such data can, if the experiment has been well-designed with a view to answering the questions of interest, give highly reliable results. With data from carefully designed experiments, it is sometimes possible to infer causal relationships. Perhaps the most serious danger is that the data 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 interpretation in terms of causation, other sources of evidence must come into play.

1.2.4 How will data be used?

Studies may be designed to help scientific understanding. Consider again the data in Table 1.1. The interest of Latter's paper is primarily in establishing whether there is a relationship, and rather less in determining the nature of the relationship. Egg size and shape is one of several pieces of evidence that Latter considers. Uniquely among the birds listed, the architecture of wren nests makes it impossible for the birds to see the eggs. In wren nests, the color of the cuckoo's egg does not match the color of the wren's eggs. For the other species the color does mostly match. Latter concludes that Newton is right, that the eggs that cuckoos lay tend to match the eggs of the host bird in size and shape in ways that will make it difficult for hosts to distinguish their eggs from the cuckoo eggs.

What was measured? Is it the relevant measure?

The `science` and `socsupport` data frames (`DAAG`) are both from surveys. 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 name for their experience of science laboratory classes (smells, bangs and sparks perhaps) and field trips? Answers to other questions included in the survey shed some limited light.

In the `socsupport` dataset, an important variable is Beck Depression Inventory or `BDI`, which is based on a 21-question multiple choice self-report. The Beck Depression Inventory is the result of an extensive process of development and testing. Since its first publication in 1961, it has been extensively used, critiqued, and modified. Its results have to this extent 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.

1.2.5 The planning of data analysis

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 in any case required to identify obvious mistakes and/or quirks in the new data. Data must support the intended form of analysis. Graphs that draw attention to inadequacies may, at the same time, hint at remedies. If preliminary analysis gives a clear indication that the data should be transformed in order to give an acceptable approximation to normality, then the demand is to use the transformation. At the same time, be sensitive to the risk that use of the data to influence may bias results.

Data-based selection of comparisons

In carefully designed comparative studies where there are multiple possible comparisons, the comparisons that will be considered should be specified in advance. Prior data, if available, can assist in this choice. 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.

1.2.6 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 design of data collection, the analysis choices, and the interpretation of analysis results. These judgments, while they should be as informed as possible, cannot be avoided.

When working with a subject matter expert, it is important that lines of communication be as clear as possible. 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.

There is an inevitable risk that assumed insights and judgments, made in planning a study, will carry elements of personal bias. A well-designed study will allow some opportunity for study results to challenge the insights and understandings that underpinned the planning.

1.2.7 Notes on sampling from finite populations

Computer generated random number sequences can be used as a mechanism for ensuring random selection in sample surveys, or random assignment of experimental treatments. Examples follow.

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 15 individuals:

```
## For the sequence below, precede with set.seed(3676)
sample(1:9384, 15, replace=FALSE)
```

```
 [1] 8466  324 8363 9280 2988 3553  268 7473 3121 7034 2531 2760  851
[14] 8386 7159
```

The numbers are the numerical labels for the 15 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.

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]  3  5 10  2  7

$Treatment
[1] 4 1 9 6 8
```

```
# 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, e.g., any unwitting or witting preference for placing healthier-looking plants in the treatment group.

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.

With-replacement samples

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

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

```
 [1] 8 1 3 1 5 9 5 7 7 9
```

With replacement sampling is the basis of *bootstrap* sampling. See Subsection 2.5.3 for an example of its use.

1.3 Using graphs to make sense of data

The use of graphs to display and help understand data has a long tradition. John W. Tukey formalized and extended this tradition, giving it the name *Exploratory Data Analysis* (EDA), as explained in Hoaglin (2003). A key concern is that data should, as far as possible, have the opportunity to speak for themselves, 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 ability 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.

A note on static graphics systems in R

Three R static graphics systems get wide use, in this text as elsewhere. 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 some extra graphics language complexity.

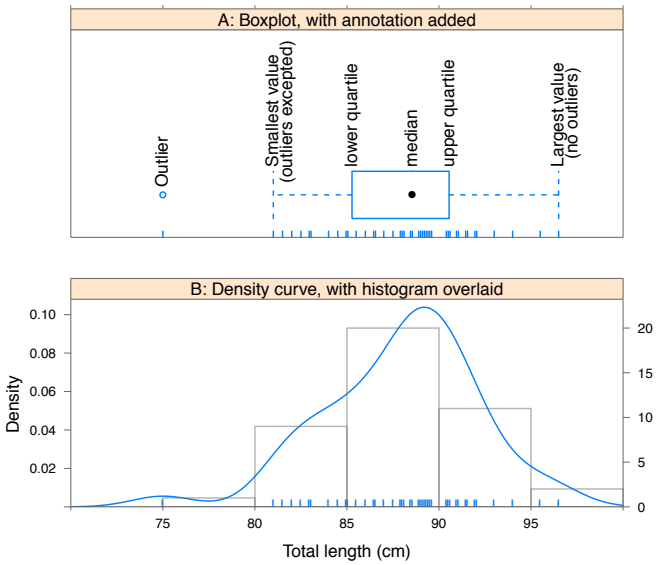


Figure 1.2: Panel A shows a boxplot, with annotation that explains boxplot features. Panel B shows a density plot, with a histogram overlaid. Histogram frequencies are shown on the right axis of Panel B. In both panels, the individual data points appear as a "rug" along the lower side of the bounding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

Later chapters will make extensive use both of *base* graphics and of `lattice` graphics, resorting to use of `ggplot2` on those few occasions when features are needed that are not readily available in `lattice` or *base* graphics. Lattice graphs will however be printed in a style (using a "theme") that is similar to the default ggplot2 style.

1.3.1 One-way layouts, perhaps broken down by groups within the data

The most basic form of display is the dotplot, which spreads the individual data points out a line. An often helpful summary form of representation, which allows a trained eye to comprehend at a glance specific important features of the data, is the boxplot.

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. In both panels, vertical bars have been added along the lower edge that show the locations of the individual points. This form of display has the name "rug".

Notice that one point lies outside the boxplot "whiskers" to the left, and is thus flagged as an outlier. An outlier is a point that, in a some defined sense, lies away from the main body of the data. For purposes of flagging points in boxplot displays as outliers, the normal distribution sets the standard. Using the default criterion. one point in 100 will on average, for data from a normal distribution, be flagged as a possible outlier. In a boxplot display of 1000 values that are drawn at random from a normal distribution, it can be expected that around 10 points will be plotted out beyond the boxplot whiskers and flagged as outliers.

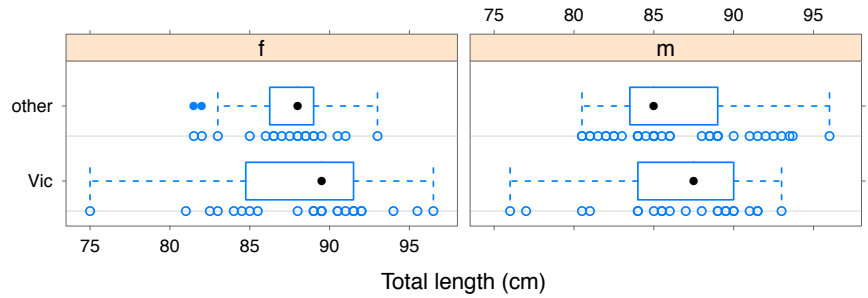


Figure 1.3: Total lengths of possums, broken down by `sex` and (within panels) by geographical location (Victorian or other).

The histogram is a basic (and over-used) exploratory tool for displaying the frequency distribution of a set of data. The area of each rectangle of a histogram is proportional to the number of observations whose values lie within the width of the rectangle. A mound-shaped histogram may make it plausible that the data follow a normal distribution (the "bell curve"). Especially in small samples, however, there needs to be caution in interpreting the shape. The shape can be highly irregular, and can depend on the choice of breakpoints.

A histogram is a crude form of a 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 can work reasonably well.

Dotplot, boxplot, and other such one-way summaries, can often be usefully broken down by one or more factors between panels, as well as within panels. Figure 1.3 overlays dotplots onto boxplot summaries of the distributions of Australian possum length data, broken down by `sex` and (within panels) by geographical region (Victorian or other). Code is:

```
possum <- DAAG::possum
gph <- bwplot(Pop~totlngth|sex, data=possum, pch=16)+
       latticeExtra::layer(panel.dotplot(x, unclass(y)-0.4, pch=1))
```

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). In each case, the user has to exercise good judgement in deciding what action to take. This will depend on the context. Subsection 1.3.6 comments in more detail.

1.3.2 Patterns in univariate time series

In Figure 1.4, "measles" includes both what is nowadays called measles and the closely related *rubella* or German measles.[2] Panel A uses a logarithmic vertical scale. Panel B uses an unlogged scale and takes advantage of the fact that deaths from measles are of the order,

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

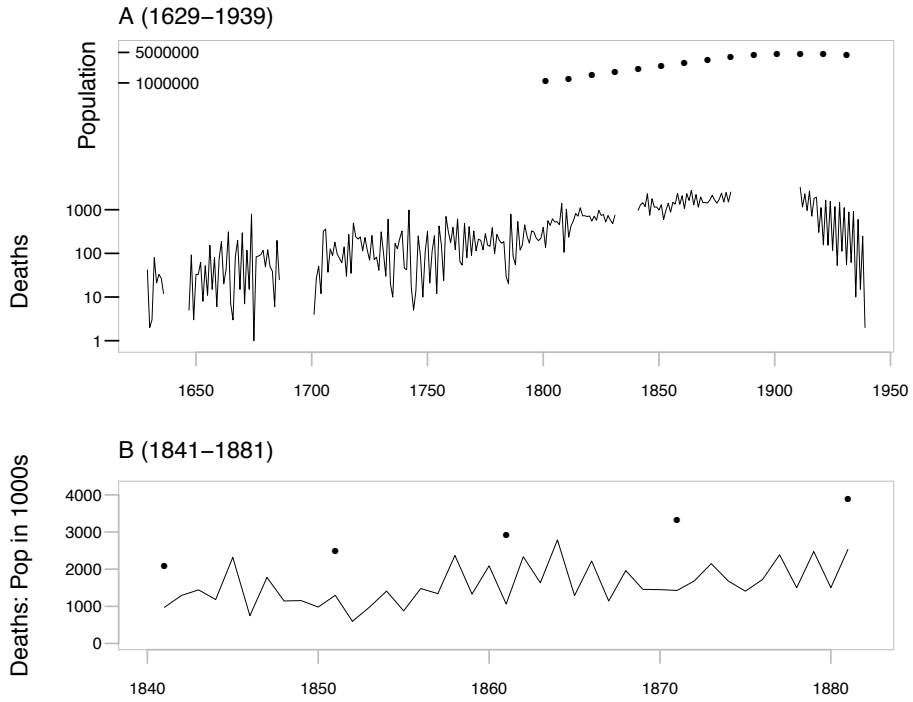


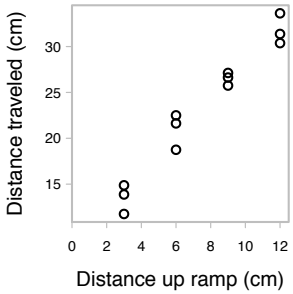
Figure 1.4: The two panels provide different insights into data on mortality from measles, in London over 1629-1939. Panel A uses a logarithmic scale to show the numbers of deaths from measles in London for the period from 1629 through 1939 (black curve). (See the comments on log scales that accompany Figure 1.7). The black dots show, for the period 1800 through to 1939, the London population. 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).

in any year, of one thousandth of the population. Thus, deaths in thousands and population in millions can be shown on the same scale.

Simplified code is:

```
measles <- DAAG::measles
## Panel A
plot(log10(measles), xlab="", ylim=log10(c(1,5000*1000)),
     ylab="Deaths; Population", yaxt="n")
ytiks <- c(1, 10, 100, 1000, 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*1000), pch=16, cex=.5)
axis(2, at=log10(ytiks), labels=paste(ytiks), las=2)
## 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)
```

Panel A shows broad trends over time, but is of no use for identifying changes on the



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

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, by which years in which there are many deaths are commonly followed by years in which there are fewer deaths, is thus evident. (To obtain this level of detail for the whole period from 1629 until 1939, multiple panels would be necessary.)

1.3.3 Bivariate data

Response lines and curves

The data shown to the right of Figure 1.5, and plotted in the figure, was 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 poor use of the data to treat the four starting points as factor levels in a one-way analysis.

For these data, the physics can be used to suggest 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.

Example — a smooth trend line

Figure 1.6 shows data from a study that measured both electrical resistance and apparent juice content for slabs of kiwifruit. The curve in Panel B, obtained using the lowess method that is discussed further in Subsection 4.4.3, estimates the relationship between electrical resistance and apparent juice content. The code for Panel B is:

```
fruitohms <- DAAG::fruitohms
plot(ohms/1000 ~ juice, xlab="Apparent juice content (%)",
     ylab="Resistance (kOhms)", data=fruitohms, fg="gray")
## Add a smooth trend curve (Panel B)
with(fruitohms, lines(lowess(juice, ohms/1000), lwd=2,
                      col="gray40"))
```

The response pattern is clearly inconsistent with a straight line. The fitted smooth curve suggests an approximate linear relationship for juice content up to somewhat over 35%.

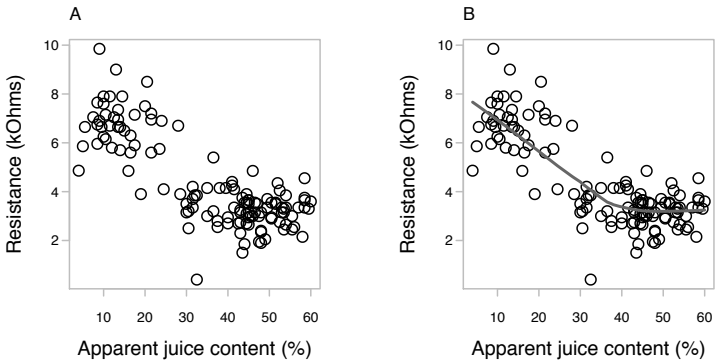


Figure 1.6: Electrical resistance versus apparent juice content. Panel B repeats panel A, but with a smooth curve fitted to the data.

Once the juice content reaches around 45%, the curve becomes a horizontal line, and there is no evident further change in resistance.

A curve fitted using `lowess()` or another such smoothing function can provide a useful benchmark against which to compare the curve given by a theoretical or other mathematical form of equation that the data are thought to follow.

What is the appropriate scale?

Figures 1.7A and 1.7B plot brain weight (g) against body weight (kg), for 28 animals.

```
Animals <- MASS::Animals
plot(I(brain/10) ~ I(body/10), data=Animals, fg="gray",
     xlab="Body (unit=10kg)", ylab="Brain (unit=10g)")  # Panel A
plot(log(brain/10) ~ log(body/10), data=MASS::Animals,  # Panel B
     xlab="Body (unit=10kg)", ylab="Brain (unit=10g)", asp=1)
```

Figure 1.7A reveals almost nothing about the relationship between brain weight and body weight. It does indicate that the distributions of data values are highly positively skew, on both axes. Panel B uses a logarithmic scale, which spreads the data out more evenly. Points along both axes that differ by a common factor (tick marks are shown for a factor of 100) are an equal distance apart. The argument `asp=1` is used to ensure that distances that reflect a change by the same factor are the same distance apart on both x- and y-axes.

A logarithmic scale is appropriate for quantities that change multiplicatively. For example, 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 so-called exponential growth. Growth in the bodily measurements of organisms may similarly be multiplicative, with large organisms increasing in some time interval by the same approximate fraction as smaller organisms. Growth rate on a natural logarithmic scale ($\log_e$) equals the relative growth rate. Derivation of this result is a straightforward use of the differential calculus.

Pretty much anyone who works with real data — biologists, economists, physical scientists — should make themselves comfortable with the use and interpretation of logarithmic scales. There is a brief discussion of other transformations in Subsection 2.6.9.

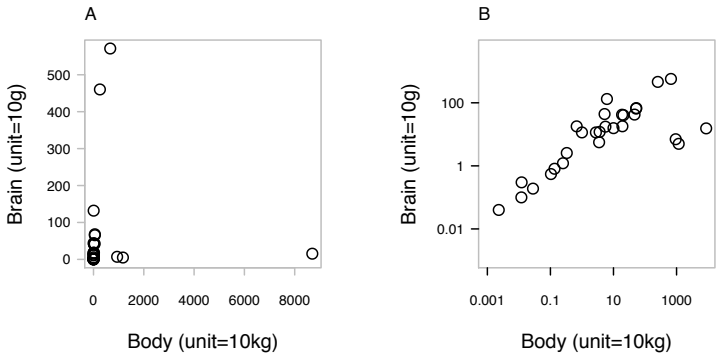


Figure 1.7: Brain weight versus body weight, for 28 animals that vary greatly in size. Panel A uses untransformed scales, while Panel B uses logarithmic scales, on both axes.

1.3.4 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.

The data frame `jobs` (`DAAG`) gives the number of workers (in thousands) in the Canadian labor force, broken down by region (BC, Alberta, Prairies, Ontario, Quebec, Atlantic), for the 24-month period from January 1995 to December 1996. Over this time, Canada was emerging from a deep economic recession. Columns 1–6 have the respective numbers for six different regions. The ranges of values in the columns are:

```
## Apply function range to columns of data frame jobs (DAAG)
jobs <- DAAG::jobs
sapply(jobs, range)
```

```
      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
```

In order to see where the economy was taking off most rapidly, it is tempting to plot all series on the same graph. In order that similar changes on the scale will correspond to similar proportional changes, a logarithmic scale is used in Figure 1.8A:

```
## Panel A: Basic plot; all series in a single panel; use log y-scale
basicGphA <-
  xyplot(Ontario+Quebec+BC+Alberta+Prairies+Atlantic ~ Date,
         outer=FALSE, data=jobs, type="l",
         ylab="Number of workers", scales=list(y=list(log="e")),
         auto.key=list(space="right", lines=TRUE, points=FALSE))
```

The graphics object `basicGphA` has been saved so that it can be updated, as demonstrated

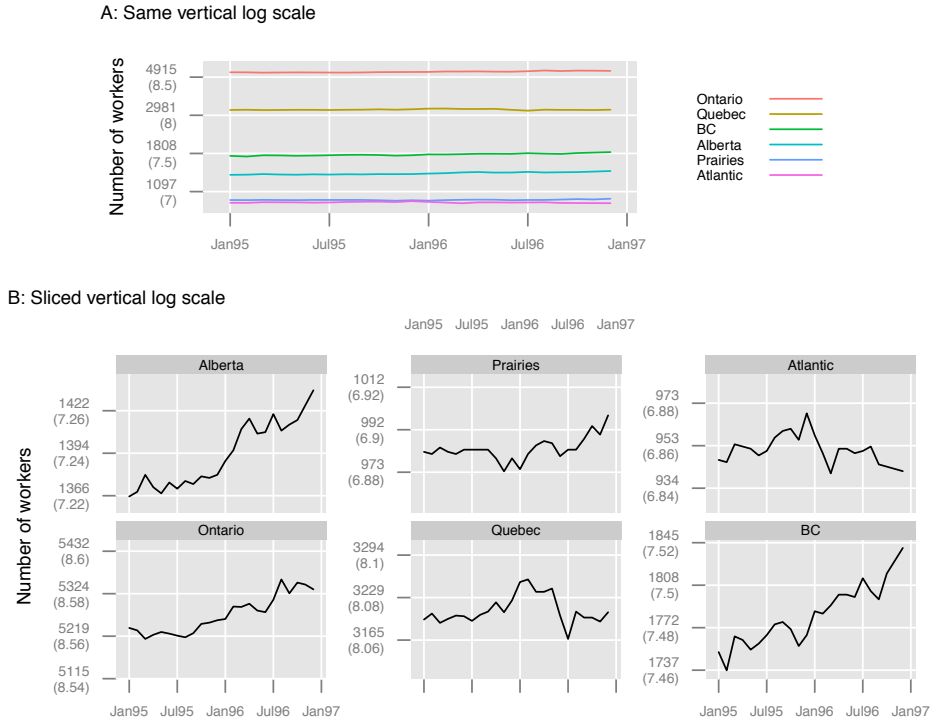


Figure 1.8: Data are numbers in the labor force (thousands) for various regions of Canada, at quarterly intervals over 1995-1996. Panel A uses the same logarithmic y-scale for all regions. Panel B shows the same data, but now with separate ("sliced") logarithmic y-scales on which the same percentage 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 almost exactly 2%.

in the footnote, to give the graph shown in Figure 1.8A.[3]

The use of column names that are joined with "+", has the result that the columns are plotted in parallel. 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 in the footnote shows how the labeling on the x- and y-axes was obtained.

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 did not remedy this problem.

Figure 1.8B shows a more informative alternative. The six different panels use different

[3]
```
## 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(jobs[,-7])), 5))
ylabelsA <- paste(round(ylabposA),"\n(", log(ylabposA), ")", sep="")
gphA <- update(basicGphA, xlab="",
               scales=list(x=list(at=datelabpos, labels=datelabs),
                           y=list(at=ylabposA, labels=ylabelsA)))
```

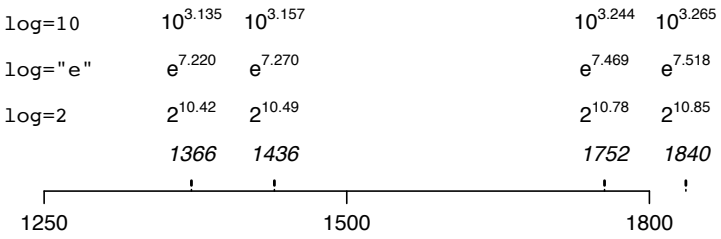


Figure 1.9: Labeling of the values for Alberta (1366, 1436) and BC (1752, 1840), with alternative logarithmic scale choices of labeling.

slices of the same logarithmic scale. Here is simplified code for Figure 1.8B. The regions are again taken in the order of numbers of workers in December 1996.

```
## Panel B
basicGphB <-
  xyplot(Ontario+Quebec+BC+Alberta+Prairies+Atlantic ~ Date,
         data=jobs, outer=TRUE, type="l", layout=c(3,2),
         xlab="", ylab="Number of workers",
         scales=list(y=list(relation="sliced", log=TRUE)))
  # Provinces are in order of number of workers in Dec96
```

Use of `outer=TRUE`, ensures that the separate columns (regions) are plotted on separate panels. Equal distances on the scale now correspond to equal relative changes. It is now clear that Alberta and BC experienced the most rapid job growth during the period, 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 1.9 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, Alberta increased by 70 from 1366 to 1436, which is a factor of $1436/1366 \simeq 1.051$. BC increased by 88 from 1752 to 1840 which is a factor of 1.050. The proper comparison is not between the absolute increases of 70 and 88, but between the relative increases by very nearly identical multipliers of 1.051 and 1.050. Panel B of Figure 1.9 gives a visual perspective on this and other such comparisons.

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`.)

*Labeling technicalities

For lattice functions, the arguments `log=2` or `log="e"` or `log=10` are available. These use the relevant logarithmic axis labeling, as in Figure 1.9, for axis labels. In base graphics, with the argument `log="x"`, the default is to label in the original units.

An alternative, both for traditional and lattice graphics, is to enter the logged values, using whatever basis is preferred (2 or `"e"` or 10), into the graphics formula. Unless other tick labels are provided, the tick marks will then be labeled with the logged values for the relevant basis.

Note again the reason for placing *y*-axis tick marks a distance 0.02 apart on the $\log_e$ physical distance scale used in Figure 1.8. On a $\log_e$ scale (natural logarithms) a change of 0.02 is, to a close approximation, a 2% change.

1.3.5 Graphical displays for categorical data

Code used to enter the counts into the array `stones` that is shown to the right of Figure 1.10, then creating the table shown underneath that results from adding over `Size`, 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)
```

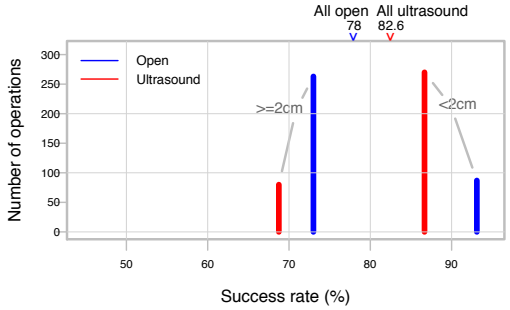
The table `stones` has three `margins` — `Success`, `Method`, and `Size`. The table `margin12` retains the first two of these margins only.

Figure 1.10 illustrates the possible hazards of adding 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 as either at least 2 cm or less than 2 cm. For each subject, the outcome was assessed as successful ("yes") or unsuccessful ("no").

Both for small stones and for large stones separately, surgery appears more successful than ultrasound. The blue vertical bar Figure 1.10 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 2.1.3.)

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.

Mosaic plots, implemented using `mosaicplot()` from base graphics or `vcd::mosaicplot()`, offer an alternative type of display. Figure 1.10 makes the point of interest for the kidney stone surgery data more simply and directly.



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

1.3.6 What to look for in plots

The notes that follow draw attention to a few of the more important features that may be apparent from visual inspection.

Outliers

Outliers are points that appear to be isolated from the main body of the data. Such points (whether errors or genuine values) are liable to distort any model that we fit. What appears as an outlier depends, inevitably, on the view that is presented. On a fairly simple level, the view is affected by whether or not, and on how, the data are transformed.

Boxplots, and the normal probability plot that will be discussed in Subsection 1.5.4, 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. The presence of outliers can indicate departure from model assumptions.

Asymmetry of the distribution

Positive skewness is a common form of departure from normality. The largest values are widely dispersed (there is a tail to the right), 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 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 uses of 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 1.4.3.)

When variability increases as data values increase, the logarithmic transformation will often help. If the variability is constant on a logarithmic scale, then the relative variation on the original scale is constant.

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.

Non-linearity

We should not fit a linear model to data where relationships are demonstrably non-linear. Often it is possible to transform variables so that it makes sense to model their effects as linear. Where none of the common standard transformations (the logarithmic is the commonest) meets requirements, methodology is available that will fit quite general forms of smoothing curve. See especially Sections 4.4 and 4.5.

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.

1.4 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. If no information is lost, the gain in simplicity of analysis can make the use of summary data highly worthwhile.

It is important, when data are summarized, not to introduce distortions that are artifacts of the way that the data have been summarized – examples will be given. The potential for loss or misrepresentation of information is an especial issue when counts are summarized across the margins of multi-way tables, or when correlation coefficients are calculated.

1.4.1 Counts

Data in the data frame `DAAG::nswpsid1` are derived 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. Are the two groups genuinely comparable? This can be checked by comparing them with respect to various measures other than their exposure (or not) to the labor training program.

Thus, what are the relative numbers in each of the two groups who had completed high school (`nodeg = 0`), as opposed to those who had not (`nodeg = 1`)?

```
## Table of counts example: data frame nswpsid1 (DAAG)
nswpsid1 <- DAAG::nswpsid1
tab <- with(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
```

Notice the use of the argument `useNA="ifany"` in the call to `table()`. This ensures that any `NA`s in either of the margins of the table will be tabulated.

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. These data will be investigated further in Section 10.1.

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

Each year the National Highway Traffic Safety Administration in the USA uses a random sampling method, with sampling fractions that differ according to class of accident, to collect data from all police-reported crashes in which there is a harmful event (people or property), and from which at least one vehicle is towed. The data in `nassCDS` (`DAAG`) are restricted to front-seat occupants.[4]

Factors whose effect warrant investigation include, as a minimum: `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 range of variation in the sampling weights is huge, as indicated by the boxplot representation of the weights in Figure 1.11: 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 crashes:

[4] They hold 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`.

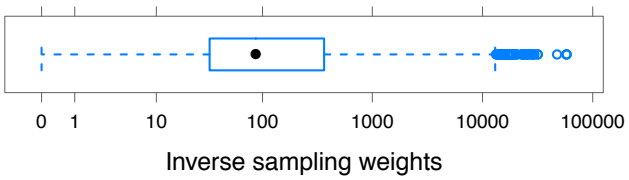


Figure 1.11: Boxplot representation of values in the column `weight`, in the dataset `DAAG::nassCDS`. (A `log(weight+1)` scale): has been used.)

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

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

```
nassCDS <- DAAG::nassCDS
Atab <- xtabs(weight ~ airbag + dead, data=nassCDS)
## 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          5,445,245.9    39,676.0         7.2
airbag        6,622,691.0    25,919.1         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))
SAtabf <- ftable(addmargins(SAtab, margin=3,
                            FUN=DeadPer1000), col.vars=3)
print(SAtabf, digits=2, method="compact", big.mark=",")
```

```
seatbelt airbag | dead       alive        dead       Total DeadPer1000
none     none          1,342,021.9    24,066.7 1,366,088.6         8.8
         airbag          871,875.4    13,759.9   885,635.3         7.8
belted   none          4,103,224.0    15,609.4 4,118,833.4         1.9
         airbag        5,750,815.6    12,159.2 5,762,974.8         1.1
```

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 the spuriously large advantage that the table that had added over the `seatbelt` margin gave 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 is data from the Fatality Analysis Recording System (FARS). 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. The gamclass::FARS dataset has data for the years 1998 to 2010.

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 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.

1.4.2 Summaries of information from data frames

The data frame kiwishade (from DAAG) has yield measurements from 48 kiwifruit vines. Plots, made up of 4 vines each, were the experimental units.

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.

Figure 1.12 plots both the aggregated means and the individual vine results. The code is given as a footnote.[5] As treatments were applied to whole plots, a focus on the individual

[5]
```
## 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")}
gph <- dotplot(shade~yield | block, data=kiwishade, col="gray40", aspect=0.65,
               panel=panelfun, key=keyset, layout=c(3,1))
```

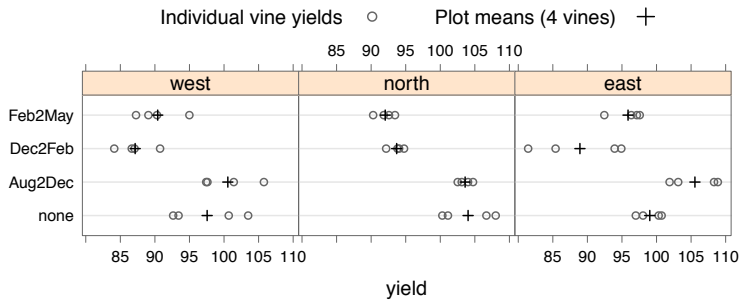


Figure 1.12: Individual yields and plot-level mean yields of kiwifruit (in kg) for each of four treatments (season) and blocks (exposure).

vine results exaggerates the extent of information that is available, in each block, for comparing treatments. For gaining an accurate impression of the strength of the evidence, focus the eye on the means, shown as +.

The code used for the plot used 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)
```

```
  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 used with 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. The question then becomes: "Is the mean an effective form of summary?" If there were occasional highly aberrant values, use of medians might be preferable. Use of summary data gives the freedom to choose the most appropriate form of summary.

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, is available for 2000 administrative regions. Climate information is available on a much finer scale, on a grid of about 80 000 pixels at 0.5° latitude and longitude resolution. Should the analysis work with a dataset that consists of 2000 administrative regions, or with the much

larger dataset that has one row for each of the 80 000 pixels? The following are reasons that might have argued for working with the summarized data:

- Dengue status is a summary figure that is given by administrative region. Use of the individual pixel values to calculate summary climate statistics prior to the analysis, by administrative region, gives the user control over the form of statistical summary that will be used. If, for example, values for some pixels are extreme relative to other pixels in the administrative region, medians may be more appropriate than means. In some regions, the range of climatic variation may be extreme. The mean will give the same weight to sparsely populated cold mountainous locations as to highly populated hot and humid locations on nearby plains.
- Correlation between observations that are close together geographically, 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 both for the interpretation of plots and, often, for the analysis. Regions that are geographically close will often have similar climates and the same dengue status.
- Analysis is more straightforward with the reduced size of dataset. It is easier to do standard forms of data checking. Standard forms of scatterplot less readily degenerate into a dense mass of black ink.

There are many possible ways to calculate a central value, of which the mean and the median are the most common. (In fact, however, the paper used the disaggregated data.)

1.4.3 Measures of spread — standard deviation and inter-quartile range

An important measure of variation in a population is the population standard deviation (often written σ), which is almost always unknown. The variance σ^2, which is the square of the standard deviation, is widely used in formal inference.

Given a random sample $x_1, x_2, \ldots, x_n$, the sample standard deviation is:

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

In R, use the function `sd()` to calculated the standard deviation, or `var()` to calculate the variance. The standard deviation is in the same units as the original measurements. For s to be an accurate estimate of σ, the sample must be large.

Cuckoo eggs example

Consider again the data on cuckoo eggs that we discussed in Subsection 1.2.2. 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 then subject to large statistical uncertainty.

The footnote has code for the calculations used to create the table.[6]

Degrees of freedom

The denominator $n-1$ 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 the variance is possible. The degrees of freedom are the number of data values, additional to the first data value.

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.

Inter-quartile range (IQR)

The standard deviation is similar in concept to the inter-quartile range H, noted in connection with Figure 1.2 in Section 1.3. The inter-quartile range is the difference between the first and third quartiles. (The region between the lower and upper quartiles takes in 50% of the data.)

For data that are approximately normally distributed, note the relationship

$$s \approx 0.75H.$$

For data are normally distributed, one standard deviation either side of the mean takes in slightly more than 68% of the data.

Note also the median absolute deviation, calculated using the function `mad()`. This calculates the median of the absolute deviations from the median, multiplied by the value given to the argument `constant`. The default is `constant` = 1.4286, set to ensure that in a large sample of normally distributed values the value returned should approximately equal the standard deviation. This is a more reliable estimate than that based on the IQR.

The pooled standard deviation

Consider two independent samples of sizes n_1 and n_2, respectively, randomly selected from populations that have the same standard deviation but for which the means may differ. After estimating the two means, n_1+n_2-2 degrees of freedom remain for estimating the (common) standard deviation. The ("pooled") standard deviation estimate is calculated by summing squares of differences of each data value from their respective sample mean, dividing by the degrees of freedom n_1+n_2-2, and taking the square root:

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

Use of this pooled estimate of the standard deviation is appropriate if variation in the two populations is plausibly similar. The pooled standard deviation is estimated with more degrees of freedom, and therefore, more accurately, than either of the separate standard deviations.

[6]
```
## 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,],')'),
```

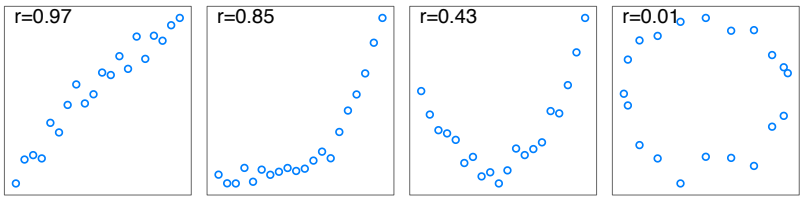


Figure 1.13: Different relationships between y and x. In the second panel, the Pearson correlation is 0.85, while the Spearman rank correlation is 0.92.

Elastic bands example

Consider data from an experiment in which 21 elastic bands were randomly divided into two groups, one of 10 and one of 11. Bands in the first group were immediately tested for the amount that they stretched under a weight of 1.35 kg. The other group were dunked in hot water at 65°C for four minutes, then left at air temperature for ten minutes, and then tested for the amount that they stretched under the same 1.35 kg weight as before. The results were:

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 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.

1.4.4 Correlation

The usual 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. Check also that the separate distributions of the two variables are roughly normal, or at least not highly skew. If the relationship is monotonic, but is not linear and/or the separate distributions are asymmetric, a Spearman rank correlation may be more appropriate.

Figure 1.13 gives four graphs to consider. For which does it make sense to calculate

1. a Pearson correlation coefficient?
2. a Spearman rank correlation?

The figure that appears in the upper left in each panel is the Pearson correlation. For the second panel, the Pearson correlation is 0.85, while the Spearman correlation, which better captures the strength of the relationship, is 0.92. Here a linear fit clearly is inadequate. The magnitude of the correlation r, or of the squared correlation r^2, does not of itself indicate whether the fit is adequate.

Note also the Kendall correlation, obtained by specifying `method="kendall"` when `cor.test()` is called. This is applicable in contexts where the same individuals are assessed by two different judges, and estimates the probability that the two judges will

assign the same ranking to an individual.

Other 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 it will be unclear whether any correlation at the level of individuals reflects a correlation between village averages or a correlation between individuals within villages, or some of each. The two correlations may not be the same, and may even go in differentmdirections. 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. Thus, consider 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.

Note that while a correlation coefficient may sometimes be a useful single number summary of the relationship between x and y, regression methods offer a much richer framework for the examination of such relationships.

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 2.3.4 explains the assumptions that use of this function requires.

1.5 Distributions: models for the random component

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

$$y = \mu + \varepsilon$$

where μ is a constant, and ε is the random component. In the models that will be discussed in this section, μ will be the population mean, or expected value.

Alternative names that may be used for the random component are *noise*, or *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 an extensive range of functions that have direct application in cases where the deterministic model component is constant. 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).

1.5.1 Discrete distributions

Probabilities for a discrete random variable with values 0, 1, 2, ..., are for naming purposes treated as densities that are constant on unit length intervals, chosen thus:

```
0          1          2          ...
```

```
(-0.5,0.5)  (0.5,1.5)  (1.5,2.5)        ...
```

Hence the notation `dbinom()` and `dpois()`, for what are really discrete probabilities, not densities!

The usual starting points for discussing discrete distributions are the binomial and 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 π. The name Bernoulli is used for the special case when `size` $n = 1$. A binomial random variable with `size` $n > 1$ is the sum of n independent Bernoulli variables. Standard simple applications are as a model for the number of heads in a sequence of fair coin tosses, or for the number of daughters in a family.

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):

```
round(setNames(dbinom(0:10, size=10, prob=0.15), 0:10), 3)
```

```
    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 Poisson distribution: `dpois()`, `ppois()`, `qpois()`, `rpois()`
Values are 0, 1, 2, ..., where the argument `lambda` specifies the poisson 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 λ (the Greek letter "lambda") which happens to coincide 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 that raisins appear independently in different buns, probabilities for numbers of raisins in a bun are:

```
## Probabilities of 0, 1, 2, 3, 4 raisins
round(setNames(dpois(x = 0:9, lambda = 3), 0:9),3)
```

```
    0     1     2     3     4     5     6     7     8     9
0.050 0.149 0.224 0.224 0.168 0.101 0.050 0.022 0.008 0.003
```

```
## Probability of > 9 raisins
setNames(1-ppois(q = 9, lambda = 3), ">9")
```

```
      >9
0.001102
```

The functions `ppois()`, `qpois()`, and `rpois()` can be used in exactly the same way as binomial family functions.

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.

Means and standard deviations

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` = π is $n\pi$.

The standard deviation is a summary measure of the spread of a probability distribution. For predicting the value of a random variable, a high standard deviation reflects more uncertainty than a low value. The standard deviation for a binomial random variable is $\sqrt{n\pi(1-\pi)}$. The variance, which is the square of the standard deviation, is $n\pi(1-\pi)$.

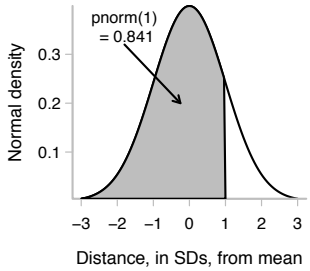
For the number of defectives in our sample of 10 items with $\pi = 0.15$, the variance is $10\times0.15\times0.85 \simeq 1.275$, with standard deviation = $\sqrt{1.275} \simeq 1.129$.

Poisson: The variance of a Poisson random variable is equal to its mean, i.e., λ. Thus if the number of raisins in a bun is Poisson with mean $\lambda = 3$, the variance is also 3. The standard deviation is $\sqrt{\lambda} \simeq 1.732$.

1.5.2 Continuous distributions

Models for measurement data will usually require use of a *continuous* distribution. It is no longer useful to consider the probability that a measurement takes a particular value. Instead, a continuous random variable is summarized 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 1.14, is widely used to model continuous measurement data (a transformation may be required, as in Figure 1.7 in Subsection 1.3.3, for the normal model to be useful). The height of the curve is a function of the distance from the mean.



Calculations using pnorm()

	Probability
pnorm(0)	0.5000
pnorm(1)	0.8413
pnorm(-1.96)	0.0250
pnorm(1.96)	0.9750
pnorm(1.96, mean = 2)	0.4840
pnorm(1.96, sd = 2)	0.8365

Figure 1.14: 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.

The density curve shown in Figure 1.14 is for a *standard* normal distribution, i.e., with 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 μ and the variance to σ.

Code that plots 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 deviate", ylab="Density", yaxs="i")
 # yaxs="i" locates the axes at the limits of the data
```

Functions for calculations relating to the normal distributions are dnorm() (used for plotting the density curve in Figure 1.14), 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 deviate with mean 0 and standard deviation 1 is less than 1.0 is:

```
pnorm(1.0)          # by default, mean=0 and SD=1
```

```
[1] 0.8413
```

This corresponds to the area of the shaded region in Figure 1.14.

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.[7]

Other continuous distributions: An especially simple model is the *uniform*, for which an observation is equally likely to take any value in a given interval. The probability density is constant on a fixed interval.

[7]
```
## 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)
```

The *exponential* 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. Details on computing probabilities for these distributions are in the exercises.

Different ways to represent distributions

In Figure 1.2A in Section 1.3 it was noted that, with the default boxplot settings, 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 may lie outside the whiskers, mostly at the lower end if the distribution is left-skewed (i.e., with a longish tail to the left), and mostly at the upper end if the distribution is right-skewed. If the distribution is symmetric, but "heavy-tailed", then the expected 1% of values that are out beyond the boxplot whiskers will have no preference for side.

1.5.3 Use of simulation to estimate sampling distributions

In a simulation, repeated random samples are taken from a specified distribution. Statistics, perhaps estimates that are derived from one or other model, can then be calculated for each successive sample. The information on the sampling distibutions of the statistics of interest offers an *empirical*, i.e., sampling based, alterative to the direct use of statistical theory. It can be used when theoretical results are not available or are of uncertain relevance.

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. In situations, such as when checking calculations, that require the repeating of 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 as part of the workspace. When the workspace is loaded again, the seed will be restored to its value when the workspace was last saved. Any new simulations will then be independent of those prior to saving the workspace.

Sampling from discrete distributions

To generate the numbers of daughters in a simulated sample of 25 four-child families, assuming independence between successive births and a probabiity of 0.5 for a daughter, use the rbinom() function thus:

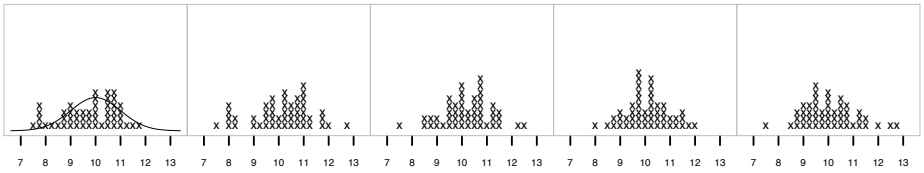


Figure 1.15: Each panel shows a simulated distribution of 50 values from a normal distribution with mean = 10 and sd = 1. The underlying theoretical normal curve is overlaid on the far left panel.

```
set.seed(31)  # Use to reproduce the sample below
rbinom(25, size=4, prob=0.5)
```

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

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

```
rpois(20, 3)
```

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

Sampling from the normal and other continuous distributions

The function `rnorm()` generates random deviates 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]  0.0087 -2.1096 -0.4879 -0.2962  0.3986  0.4652  0.7066 -0.2313
 [9]  1.9908  0.1407
```

Figure 1.15 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.[8]

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 such as are shown in Figure 1.15 are not a good basis for deciding whether sample values are consistent with a normal distribution. For that purpose, an effective tool is the normal probability plot, used as will now be demonstrated.

[8]
```
## 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))
```

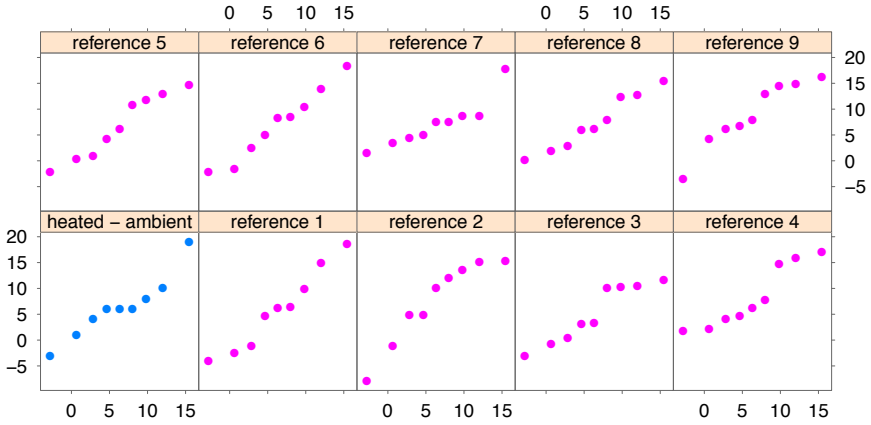


Figure 1.16: The lower left panel is the normal probability plot for heated–ambient differences. Remaining panels show normal probability plots for samples of nine numbers from a normal distribution.

1.5.4 Graphical checks for normality

To create a normal probability (or normal quantile-quantile), the sorted data values are plotted against the expected ordered values for a normal distribution. The "expected" value of a statistic is its mean, as estimated from the fitted model. 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 test for "stretchiness", bands were then 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
```

In a later section, the heated–ambient differences will be the basis for a formal comparison. The normal probability plot for these differences is in the lower left panel of Figure 1.16. 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 probability 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))
```

Displays in the style of Figure 1.16 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`.

The limitations of checks for normality

In practice, exact normality is unlikely. Nor, depending somewhat on the use that will be made of the data, is exact normality necessary. Concern arises when there are gross departures from normality. Check especially for data that are skew, and for outliers. Check also for data that take relatively few discrete values, perhaps as a result of excessive rounding.

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 commonly 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.

The sampling distribution of the mean — the Central Limit Theorem

The sampling distribution of the mean is the distribution of the means of repeated random samples of a given size n. The standard deviation of this sampling distribution has the name *standard error of the mean* (SEM). If the population mean is μ and the standard deviation is σ, then

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

A consequence of the "Central Limit Theorem" is that for a statistic such as a mean or a regression slope, the effects of averaging may give a close approximation to normality, even when the underlying population is clearly not normally distributed. This theorem implies that, for large enough n, the sampling distribution of the mean will closely approximate the normal.[9] The sample size n needed so that the normal is a good approximation will depend on the distribution of the population from which samples are taken.

Modest departures from normality will commonly be of minor consequence where samples are large. Graphical checks, and formal statistical tests for normality, will both detect non-normality in this large sample context where it commonly gives the least reason for concern. Graphical checks have the advantage that they give indications of the extent and nature of non-normality.

Figure 1.17 shows, as solid curves, simulated sampling distributions of the mean (sample sizes 3, 9, and 30), for samples from a distribution that is mildly skew. Panel A shows density curves, while Panel B shows normal probability plots. The dashed curve, which is the same in each panel, shows the population distribution. As the sample size increases from

[9]More precisely, the distribution of the sample mean approximates the normal distribution with arbitrary accuracy, for a large enough sample, assuming values are independent, and a finite population standard deviation. There are similar results for a number of other sample statistics.

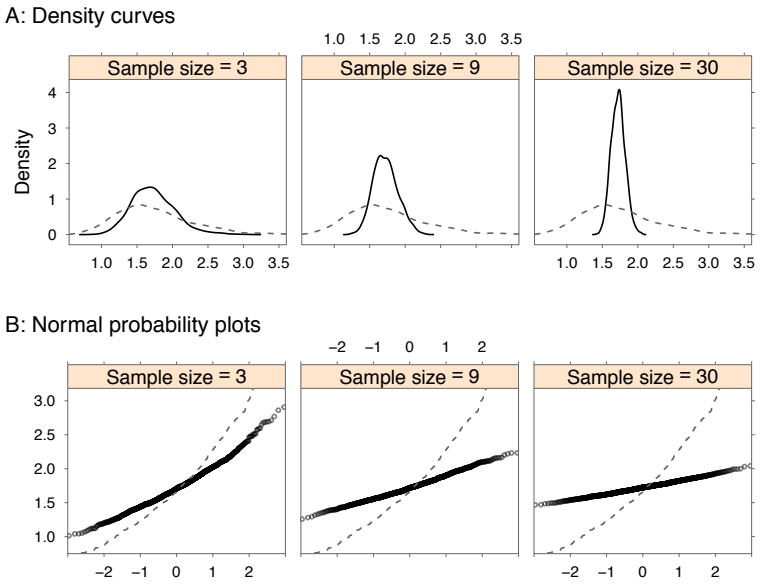


Figure 1.17: Data values are from simulations of the sampling distribution of the mean, for a distribution that is mildly skew. Panel A shows density curves, while Panel B shows normal probability plots. The plot for the population, repeated in each panel, is shown as a dashed curve. Simulated sampling 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.

$n = 3$ to $n = 9$ to $n = 30$, the density curves become more nearly symmetric with a decreasing standard deviation, while the normal probability plots become increasingly linear, with a reduced slope. The reduction in slopes 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 1.17A is:

```
## 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 1.17B, 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`.

Simulation in teaching and research

In statistical theory and in practice, simulation is widely used, when analytical results are not available, to determine the sampling distribution of statistics that are of interest. In teaching, it can provide helpful insight. The R package *animation* (Xie and Cheng, 2008) has a variety of simulations that are intended for use in teaching or self-instruction.

1.6 Organizing and managing work; reproducible reporting

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 to and fro between them. Analyses will themselves need to be documented, with the results feeding into a description that may become part of a report or paper.

Tools that will help take complexity from the analyst's mind or notebook and place it in the external world — should then be used to full advantage. Tools that will be noted can be classified, broadly, as either *Integrated Development Environment*s (IDEs), or *Graphical User Interfaces* (GUIs).

For managing work with R, we strongly recommend the highly rated RStudio IDE (go to `www.rstudio.com`). RStudio provides, from a graphical user interface, a range of abilities for organizing and managing work, for accessing help information, and for data input. Computations will usually be initiated from the command line, or from a script window.

As an alternative to the R GUI that is supplied with R binaries, there are several GUIs that provide, also, graphical and analysis abilities. Here, note the R Commander (`Rcmdr`), Deducer, and Rattle. Details can be obtained from online resources.

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. All the GUIs noted here make available the commands used by R, 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.

Our discussion will usually assume use of the command line, either directly, or from an editor window in a GUI or IDE. Further comments on RStudio now follow.

The RStudio Integrated Development Environment

The RStudio IDE has extensive features that assist with:

- The organization of work into projects;
- 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 an interface to the abilities of the `knitr` package. Subsection 1.6.1 has further details.
- The creation of packages.

RStudio has very extensive web-based documentation. Go to `https://support.rstudio.com/hc/en-us` and look under Documentation. 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.

1.6.1 Reproducible reporting — the `knitr` *package*

As noted above, the R package `knitr` has extensive abilities that allow the mixing of text and R code for automatic report generation. The R code chunks are embedded within markup that includes options that control what will be done with the code and with any computer output. When suitably processed, a document is generated that contains the text, and any 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 they have been processed for printing. Perhaps the simplest of these languages is Markdown. With the ability added to include R code and output in the final document, RStudio gives it the name R Markdown. Other possibilities are R HTML, and Sweave. Sweave is LaTeX with R markup incorporated. R Markdown is easily the simplest of these three to learn and use.

To get a quick introduction to R Markdown, start up RStudio. Then, within the R command panel, click on File | New File | R Markdown This will display a simple skeleton R Markdown document. This 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.

1.7 Further Reading and Study

Extensive R-related tutorial material is available online. See for example Galili (2015). Introductions to R include Dalgaard (2008). Braun and Murdoch (2016) is an elementary introduction to the R programming language. More technical and detailed accounts of the R language include: Chambers (2008); Matloff (2011); Wickham (2015, 2016a).

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. Thaler (2015) is an extended commentary on the mismatch between the decision making processes of the idealised agents of the models of classical economics (Thaler calls these agents "Econs"), and human agents, with serious implications for social policy.

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); 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.

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`.

References

Allison et al. 2016. Reproducibility: A tragedy of errors.
Andersen 1990. *Methodological Errors in Medical Research: an Incomplete Catalogue.*
Braun and Murdoch 2016. *A First Course in Statistical Programming with R.*
Chambers 2008. *Software for Data Analysis: Programming with R.*
Chang 2013. *R Graphics Cookbook.*
Dalgaard 2008. *Introductory Statistics with R.*
Galili 2015. Tutorials for learning R. `https://www.r-bloggers.com/how-to-learn-r-2/`
Greenland et al. 2016. Statistical tests, P values, confidence intervals, and power: a guide to misinterpretations.
Kahneman 2013. *Thinking, Fast and Slow*.
Maindonald 1992. Statistical design, analysis and presentation issues.
Matloff 2011. *The Art of R Programming*.
Murrell 2011. *R Graphics*.
`www.stat.auckland.ac.nz/~paul/RGraphics/rgraphics.html`
O'Neil 2016. *Weapons of Math Destruction*.
Robbins 2012. *Creating more effective graphs.*
Thaler 2015. *Misbehaving*.
Wickham 2015. *Advanced R.* `http://adv-r.had.co.nz/`
Wickham 2016a. *R for Data Science.* `http://r4ds.had.co.nz/`
Wilkinson and Task Force on Statistical Inference 1999. Statistical methods in psychology journals: guidelines and explanation.

See the references at the end of the book for fuller bibliographic details.

1.8 Exercises

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

   Create a new data frame by extracting these rows from `orings`, and plot `total` incidents against `temperature` for this new data frame. Obtain a similar plot for the full data set. Why is it important to show the plot for the full set of data?

2. For the data frame `possum` (`DAAG`)

   (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?

3. 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))
```

4. 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?

5. Use the function `abbreviate()` to obtain six-character abbreviations for the row names in the data frame `cottonworkers` (`DAAG` package). Plot `survey1886` against `census1886`, and plot `avwage*survey1886` against `avwage*census1886`, in each case using the six-letter abbreviations to label the points. How should each of these graphs be interpreted? [Hint: Be sure to specify `I(avwage*survey1886)` and `I(avwage*census1886)` when plotting the second of these graphs.]

6. The data frame `socsupport` (`DAAG`) has data from a survey on social and other kinds of support, for a group of university students. It includes Beck Depression Inventory (`BDI`) scores. The following are two alternative plots of `BDI` against `age`.

```
plot(BDI ~ age, data=socsupport)
plot(BDI ~ unclass(age), data=socsupport)
```

   For examination of cases where the score seems very high, which plot is more useful? Explain. Why is it necessary to be cautious in making anything of the plots for students in the three oldest age categories (`25-30`, `31-40`, `40+`)?

7. Plot a histogram of the `earconch` measurements for the `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 four columns in the dataset `Devore7::ex10.22` give tomato yields at the four levels of salinity 1.6, 3.8, 6.0, and 10.2, as measured by electrical conductivity (`EC`, in nmho/cm).

    (a) Obtain a scatterplot of `yield` against `EC`.
    (b) Obtain side by side boxplots of `yield` for each level of `EC`.
    (c) Comment upon whether the yield data are more effectively analyzed using `EC` as a quantitative or qualitative factor.

    Note: You will likely want to work with the "long" of the data, which can be obtained thus:

```
tomatoes <- data.frame(yield=unlist(Devore7::ex10.22),
                       EC=rep(c(1.6,3.8, 6.0,10.2),rep(5,4)))
```

11. Figure 1.8 showed changes in labor force numbers, in six regions of Canada, in successive quarters of 1995-1996. The population (in thousands) aged 15 years and over in each of these regions was, according to the 1996 census: `BC`: 3955; `Alberta`: 2055; `Prairies`: 1604; `Ontario`: 8249; `Quebec`: 5673; `Atlantic`: 1846. Plot a version of Figure 1.8B in which the labor force numbers are standardized by division by the number in the relevant population. Compare a plot that shows all regions in the same panel, with a plot that gives each region its own panel and its own slice of a common scale, commenting on the advantages and disadvantages of each. Is there now any reason to use a logarithmic scale?

12. The `MASS::galaxies` data frame gives speeds of 82 galaxies (see the help file and the references listed there for more information). Show a density plot for these data. Is the distribution strongly skewed? Is there evidence of clustering?

13. The `MASS::cpus` data frame gives information on 8 aspects for each of 209 different types of computers. Read the help page for more information.

    (a) Construct a scatterplot matrix for these data, as in Figure 3.3 in Subsection 3.2.1. Should any of the variables be transformed before further analysis is conducted?
    (b) How well does estimated performance (`estperf`) predict performance (`perf`)? Study this question by constructing a scatterplot of these two variables, after taking logarithms. Do the plotted points scatter about a straight line or is there an indication of nonlinearity? Is variability in performance the same at each level of performance?

14. Figure 1.7B plots brain weight (units of 10gm) versus body weight (units of 10kg), for 28 animals. Make a copy of 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?

15. 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 of the numbers 1 to 7 that will be used, one set in each block, to make the assignments of treatments to plots.

16. (a) Use `y <- rnorm(10)` to generate a random sample of size 100 from a normal distribution.
    (b) Use a loop to repeat the drawing of a normal random sample, as in (a), 25 times. For each sample store the mean in the next available element of a vector `av` that has been created for

this purpose. Calculate the standard deviation of the values in `av`, and compare the result with the theoretical value of the standard deviation for a sample that has been created in this way.

(c) Create a function that performs the calculations described in (b). Run the function several times, showing each of the distributions of 25 means in a density plot.

17. Use `mfrow` to set up the layout for a 3 by 4 array of plots. For the top panel with a samples of size 100, the middle 4 panels with samples of size 100, and the bottom 4 panels with samples of size 1000, show normal probability plots (four samples and four plots in each case.) Comment on how the appearance of the plots changes as the sample size changes.

18. Repeat exercise 17, but using the function `runif()` to take samples from a uniform distribution (by default on the interval 0 to 1.) Comment on the patterns that the points show.

19. (a) The function `pexp(x, rate=r)` computes the probability that an exponential variable is less than `x`. Suppose the time between accidents at an intersection can be modeled by an exponential distribution with a rate of .05 per day. Find the probability that the next accident will occur during the following three weeks.

(b) Use the function `rexp()` to simulate 100 exponential random numbers with rate 0.2. Obtain a density plot for the observations. Find the sample mean of the observations. Compare with the population mean (the mean for an exponential population is 1/rate).

20. (a) The statement `x <- rt(10, 1)` generates 10 random values from a t distribution with one degree of freedom. Make normal probability plots for samples of various sizes from this distribution. How large a sample is necessary, to obtain a visually consistent shape?

(b) Repeat, using statements of the form `x <- rchisq(10, 1)` to generates random values from a chi-squared distribution with one degree of freedom.

(c) Repeat, generating random values from a chi-squared distribution with 5, and then with 50, degrees of freedom.

21. The following are the total number 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 the 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)
```

22. *A Markov chain is a data sequence in which the probability of change from one state to the next depends only on the current state. The probabilities can be set out as a transition matrix, the rows give the current state, and columns the state next occupied. Thus, for the weather in a particular season of the year, a simple form of Markov chain model, for changes in the weather from one day to the next, might have the transition matrix:

$$P = \begin{bmatrix} & Sun & Cloud & Rain \\ Sun & 0.6 & 0.2 & 0.2 \\ Cloud & 0.2 & 0.4 & 0.4 \\ Rain & 0.4 & 0.3 & 0.3 \end{bmatrix}$$

It can be shown, using linear algebra, that in the long run ("in the limit") this Markov chain will visit the states according to the *stationary* distribution:

Sun	Cloud	Rain
0.429	0.286	0.286

A result called the *ergodic* theorem allows us to estimate this distribution by simulating the Markov chain for a long enough time.

```
Markov <- function (P, N=10000, initial.state=0)
{
    state <- numeric(N)
    state[1] <- initial.state + 1  # States 0:2; subscripts 1:3
    n <- nrow(P)
    for (i in 2:N){
        state[i] <- sample(1:n, size=1, prob=P[state[i-1], ])}
    state - 1
}
```

Simulate 10,000 values, and calculate the proportion of times the chain visits each of the states. Compare the proportions given by the simulation with the above theoretical proportions. Repeat several times, for each of the choices 0, 1, and 2 for `initial.state`.