Chapter 3

Fitting and graphing discrete distributions

{ch:discrete}

Discrete data often follow various theoretical probability models. Graphic displays are used to visualize goodness of fit, to diagnose an appropriate model, and determine the impact of individual observations on estimated parameters.

> Not everything that counts can be counted, and not everything that can be counted counts.
>
> Albert Einstein

Discrete frequency distributions often involve counts of occurrences of events, such as accident fatalities, incidents of terrorism or suicide, words in passages of text, or blood cells with some characteristic. Often interest is focused on how closely such data follow a particular probability distribution, such as the binomial, Poisson, or geometric distribution, which provide the basis for generating mechanisms that might give rise to the data. Understanding and visualizing such distributions in the simplest case of an unstructured sample provides a building block for generalized linear models (Chapter 9) where they serve as one component. The also provide the basis for a variety of recent extensions of regression models for count data (Chapter 9), allowing excess counts of zeros (zero-inflated models), left- or right- truncation often encountered in statistical practice.
TODO: DM: Fix chapter ref if ch.09 gets split

This chapter describes the well-known discrete frequency distributions: the binomial, Poisson, negative binomial, geometric, and logarithmic series distributions in the simplest case of an unstructured sample. The chapter begins with simple graphical displays (line graphs and bar charts) to view the distributions of empirical data and theoretical frequencies from a specified discrete distribution.

It then describes methods for fitting data to a distribution of a given form and simple, effective graphical methods than can be used to visualize goodness of fit, to diagnose an appropriate model (e.g., does a given data set follow the Poisson or negative binomial?) and determine the impact of individual observations on estimated parameters.

3.1 Introduction to discrete distributions

{sec:discrete-intro}

Discrete data analysis is concerned with the study of the tabulation of one or more types of events, often categorized into mutually exclusive and exhaustive categories. ***Binary events*** having two

outcome categories include the toss of a coin (head/tails), sex of a child (male/female), survival of a patient following surgery (lived/died), and so forth. ***Polytomous events*** have more outcome categories, which may be *ordered* (rating of impairment: low/medium/high, by a physician) and possibly numerically-valued (number of dots (pips), 1–6 on the toss of a die) or *unordered* (political party supported: Liberal, Conservative, Greens, Socialist).

In this chapter, we focus largely on one-way frequency tables for a single numerically-valued variable. Probability models for such data provide the opportunity to describe or explain the *structure* in such data, in that they entail some data generating mechanism and provide the basis for testing scientific hypotheses, prediction of future results. If a given probability model does not fit the data, this can often be a further opportunity to extend understanding of the data or the underlying substantive theory or both.

The remainder of this section gives a few substantive examples of situations where the well-known discrete frequency distributions (binomial, Poisson, negative binomial, geometric, and logarithmic series) might reasonably apply, at least approximately. The mathematical characteristics and properties of these theoretical distributions are postponed to Section 3.2.

In many cases, the data at hand pertain to two types of variables in a one-way frequency table. There is a basic outcome variable, k, taking integer values, $k = 0, 1, \ldots$, and called a ***count***. For each value of k, we also have a ***frequency***, n_k that the count k was observed in some sample. For example, in the study of children in families, the count variable k could be the total number of children or the number of male children; the frequency variable, n_k, would then give the number of families with that basic count k.

3.1.1 Binomial data

{sec:binom-data}

Binomial type data arise as the discrete distribution of the number of "success" events in n independent binary trials, each of which yields a success (yes/no, head/tail, lives/dies, male/female) with a constant probability p.

Sometimes, as in Example 3.1 below, the available data record only the number of successes in n trials, with separate such observations recorded over time or space. More commonly, as in Example 3.2 and Example 3.3, we have available data on the frequency n_k of $k = 0, 1, 2, \ldots n$ successes in the n trials.

{ex:arbuthnot1}

EXAMPLE 3.1: Arbuthnot data

Sex ratios— births of male to female children have long been of interest in population studies and demography. Indeed, in 1710, John Arbuthnot (Arbuthnot, 1710) used data on the ratios of male to female christenings in London from 1629–1710 to carry out the first known significance test. The data for these 82 years showed that in *every* year there were more boys than girls. He calculated that the under the assumption that male and female births were equally likely, the probability of 82 years of more males than females was vanishingly small, ($\Pr \approx 4.14 \times 10^{-25}$). He used this to argue that a nearly constant birth ratio > 1 (or $\Pr(\text{Male}) > 0.5$) could be interpreted to show the guiding hand of a divine being.

Arbuthnot's data, along with some other related variables are available in `Arbuthnot` in the HistData package. For now, we simply display a plot of the probability of a male birth over time. The plot in Figure 3.1 shows the proportion of males over years, with horizontal lines at $\Pr(\text{Male}) = 0.5$ and the mean, $\Pr(\text{Male}) = 0.517$. Also shown is a (loess) smoothed curve, which suggests that any deviation from a constant sex ratio is relatively small.

```
> data(Arbuthnot, package = "HistData")
> with(Arbuthnot, {
+   prob = Males / (Males + Females)
+   plot(x = Year, y = prob, type = "b",
+        ylim = c(0.5, 0.54), ylab = "Pr (Male)")
+   abline(h = 0.5, col = "red", lwd = 2)
```

```
+    abline(h = mean(prob), col = "blue")
+    text(x = 1640, y = 0.5, expression(H[0]: "Pr(Male)=0.5"), pos = 3, col = "red")
+    Arb.smooth <- loess.smooth(Year, prob)
+    lines(Arb.smooth, col = "blue", lwd = 2)
+    })
```

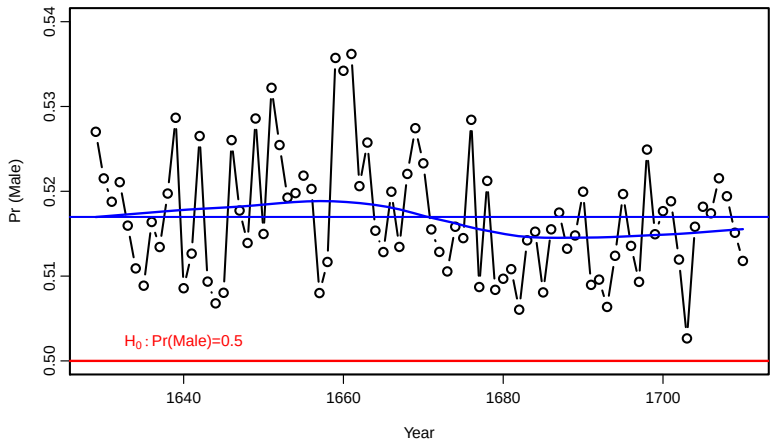

Figure 3.1: Arbuthnot's data on male/female sex ratios in London, 1629–1710, together with a (loess) smoothed curve over time and the mean Pr(Male) fig:arbuthnot1

TODO: DM: use slightly simpler alternative? :

```
> prob <- with(Arbuthnot, Males / (Males + Females))
> scatter.smooth(x = Arbuthnot$Year, y = prob, type = "b",
+                lpars = list(col = "blue", lwd = 2),
+                xlab = "Year", ylab = "Pr(Male)", ylim = c(0.5, 0.54))
> abline(h = 0.5, col = "red", lwd = 2)
> abline(h = mean(prob), col = "blue")
> text(x = 1640, y = 0.5, expression(H[0]: "Pr(Male)=0.5"), pos = 3, col = "red")
```

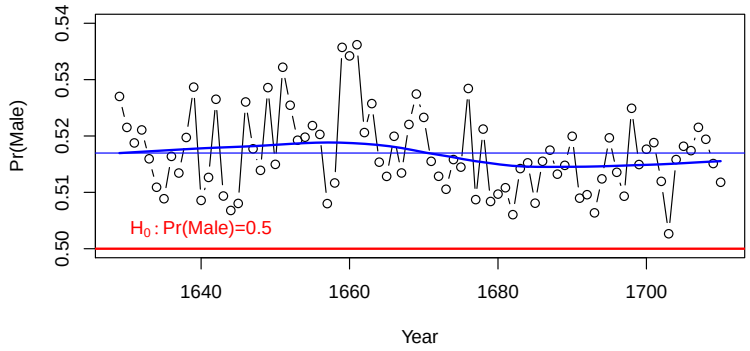

Figure 3.2: Arbuthnot's data on male/female sex ratios in London, 1629–1710, together with a (loess) smoothed curve over time and the mean Pr(Male) fig:arbuthnot12

We return to this data in a later chapter where we ask whether the variation around the mean can be explained by any other considerations, or should just be considered random variation (see Exercise 7.1) △

{ex:saxony1}

EXAMPLE 3.2: Families in Saxony

A related example of sex ratio data that ought to follow a binomial distribution comes from a classic study by A. Geissler (1889). Geissler listed the data on the distributions of boys and girls in families in Saxony for the period 1876–1885. In total, over four million births were recorded, and the sex distribution in the family was available because the parents had to state the sex of all their children on the birth certificate.

The complete data, classified by number of boys and number of girls (each 0–12) appear in Edwards (1958, Table 1).[1] Lindsey (1995, Table 6.2) selected only the 6115 families with 12 children, and listed the frequencies by number of males. The data are shown in table form in Table 3.1 in the standard form of a complete discrete distribution. The basic outcome variable, $k = 0, 1, \ldots, 12$, is the number of male children in a family and the frequency variable, n_k is the number of families with that number of boys.

{tab:saxtab}

Table 3.1: Number of male children in 6115 Saxony families of size 12

Males (k)	0	1	2	3	4	5	6	7	8	9	10	11	12	Sum
Families (n_k)	3	24	104	286	670	1,033	1,343	1,112	829	478	181	45	7	6,115

Figure 3.3 shows a bar plot of the frequencies in Table 3.1. It can be seen that the distribution is quite symmetric. The questions of interest here are: (a) how close does the data follow a binomial distribution, with a constant $\Pr(\text{Male}) = p$? (b) is there evidence to reject the hypothesis that $p = 0.5$?

```
> data(Saxony, package = "vcd")
> barplot(Saxony, xlab = "Number of males", ylab = "Number of families",
+         col = "lightblue", cex.lab = 1.5)
```

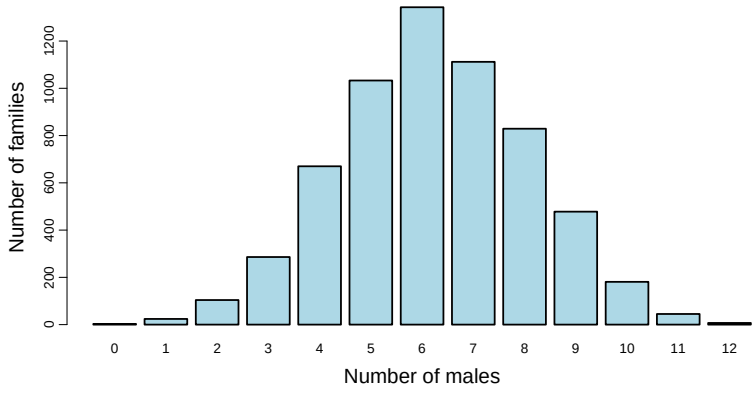

Figure 3.3: Males in Saxony families of size 12 fig:saxony-barplot

△

[1] Edwards (1958) notes that over these 10 years, many parents will have had several children, and their family composition is therefore recorded more than once. However, in families with a given number of children, each family can appear only once.

{ex:dice}

EXAMPLE 3.3: Weldon's dice

Common examples of binomial distributions involve tossing coins or dice, where some event outcome is considered a "success" and the number of successes (k) are tabulated in a long series of trials to give the frequency (n_k) of each basic count, k.

Perhaps the most industrious dice-tosser of all times, W. F. Raphael Weldon, an English evolutionary biologist and joint founding editor of *Biometrika* (with Francis Galton and Karl Pearson) tallied the results of throwing 12 dice 26,306 times. For his purposes, he considered the outcome of 5 or 6 pips showing on each die to be a success, and all other outcomes as failures.

Weldon reported his results in a letter to Francis Galton dated February 2, 1894, in order "to judge whether the differences between a series of group frequencies and a theoretical law ... were more than might be attributed to the chance fluctuations of random sampling" (Kemp and Kemp, 1991). In his seminal paper, Pearson (1900) used Weldon's data to illustrate the χ^2 goodness-of-fit test, as did Kendall and Stuart (1963, Table 5.1, p. 121).

These data are shown here as Table 3.2, in terms of the number of occurrences of a 5 or 6 in the throw of 12 dice. If the dice were all identical and perfectly fair (balanced), one would expect that $p = \Pr\{5 \text{ or } 6\} = \frac{1}{3}$ and the distribution of the number of 5 or 6 would be binomial.

A peculiar feature of these data as presented by Kendall and Stuart (not uncommon in discrete distributions) is that the frequencies of 10–12 successes are lumped together.[2] This grouping must be taken into account in fitting the distribution. This dataset is available as `WeldonDice` in the vcd package. The distribution is plotted in Figure 3.4.

Table 3.2: Frequencies of 5s or 6s in throws of 12 dice {tab:dicetab}

# 5s or 6s (k)	0	1	2	3	4	5	6	7	8	9	10+	Sum
Frequency (n_k)	185	1,149	3,265	5,475	6,114	5,194	3,067	1,331	403	105	18	26,306

```
> data(WeldonDice, package = "vcd")
> dimnames(WeldonDice)$n56[11] <- "10+"
> barplot(WeldonDice, xlab = "Number of 5s and 6s", ylab = "Frequency",
+         col = "lightblue", cex.lab = 1.5)
```

△

3.1.2 Poisson data

{sec:pois-data}

Data of Poisson type arise when we observe the counts of events k within a fixed interval of time or space (length, area, volume) and tabulate their frequencies, n_k. For example, we may observe the number of radioactive particles emitted by a source per second or number of births per hour, or the number of tiger or whale sightings within some geographical regions.

In contrast to binomial data, where the counts are bounded below and above, in Poisson data the counts k are bounded below at 0, but can take integer values with no fixed upper limit. One defining characteristic for the Poisson distribution is for rare events, which occur independently with a small and constant probability, p, in small intervals, and we count the number of such occurrences.

Several examples of data of this general type are given below. {ex:horsekick1}

[2] The unlumped entries are, for (number of 5s or 6s: frequency) — (10: 14); (11: 4), (12:0), given by Labby (2009). In this remarkable paper, Labby describes a mechanical device he constructed to repeat Weldon's experiment physically and automate the counting of outcomes. He created electronics to roll 12 dice in a physical box, and hooked that up to a webcam to capture an image of each toss and used image processing software to record the counts.

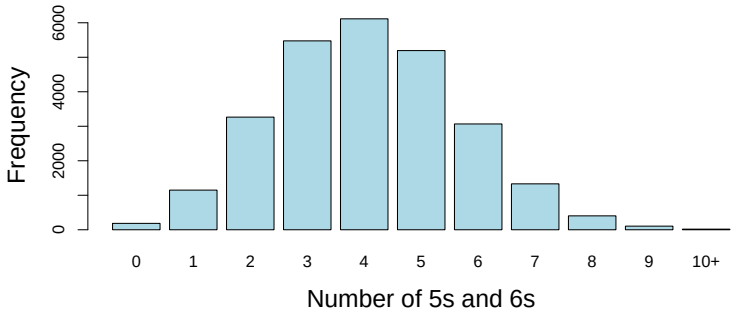

fig:dice

Figure 3.4: Weldon's dice data

EXAMPLE 3.4: Death by horse kick

One of the oldest and best known examples of a Poisson distribution is the data from von Bortkiewicz (1898) on deaths of soldiers in the Prussian army from kicks by horses and mules, shown in Table 3.3. Ladislaus von Bortkiewicz, an economist and statistician, tabulated the number of soldiers in each of 14 army corps in the 20 years from 1875-1894 who died after being kicked by a horse (Andrews and Herzberg, 1985, p. 18). Table 3.3 shows the data used by Fisher (1925) for 10 of these army corps, summed over 20 years, giving 200 'corps-year' observations. In 109 corps-years, no deaths occurred; 65 corps-years had one death, etc.

The data set is available as *HorseKicks* in the vcd package. The distribution is plotted in Figure 3.5.

{tab:horsetab}

Table 3.3: von Bortkiewicz's data on deaths by horse kicks

Number of deaths (k)	0	1	2	3	4	Sum
Frequency (n_k)	109	65	22	3	1	200

```
> data(HorseKicks, package = "vcd")
> barplot(HorseKicks, xlab = "Number of deaths", ylab = "Frequency",
+         col = "lightblue", cex.lab = 1.5)
```

△

{ex:madison1}

EXAMPLE 3.5: Federalist papers

In 1787–1788, Alexander Hamilton, John Jay, and James Madison wrote a series of newspaper essays to persuade the voters of New York State to ratify the U.S. Constitution. The essays were titled *The Federalist Papers* and all were signed with the pseudonym "Publius." Of the 77 papers published, the author(s) of 65 are known, but *both* Hamilton and Madison later claimed sole authorship of the remaining 12. Mosteller and Wallace (1963, 1984) investigated the use of statistical methods to identify authors of disputed works based on the frequency distributions of certain key function words, and concluded that Madison had indeed authored the 12 disputed papers.[3]

[3] It should be noted that this is a landmark work in the development and application of statistical methods to the analysis of texts and cases of disputed authorship. In addition to *may*, they considered many such marker words, such as *any*, *by*, *from*, *upon*, and so forth. Among these, the word *upon* was the best discriminator between the works known by Hamilton (3 per 1000 words) and Madison (1/6 per 1000 words). In this work, they pioneered the use of Bayesian discriminant analysis, and the use of cross-validation to assess the stability of estimates and their conclusions.

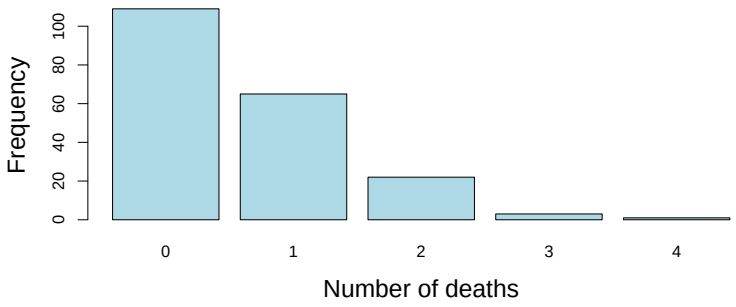

Figure 3.5: HorseKicks data fig:horsekicks

Table 3.4 shows the distribution of the occurrence of one of these "marker" words, the word *may* in 262 blocks of text (each about 200 words long) from issues of the *Federalist Papers* and other essays known to be written by James Madison. Read the table as follows: in 156 blocks, the word *may* did not occur; it occurred once in 63 blocks, etc. The distribution is plotted in Figure 3.6.

Table 3.4: Number of occurrences of the word *may* in texts written by James Madison tab:fedtab

Occurrences of *may* (k)	0	1	2	3	4	5	6	Sum
Blocks of text (n_k)	156	63	29	8	4	1	1	262

```
> data(Federalist, package = "vcd")
> barplot(Federalist,
+         xlab = "Occurrences of 'may'", ylab = "Number of blocks of text",
+         col = "lightgreen", cex.lab = 1.5)
```

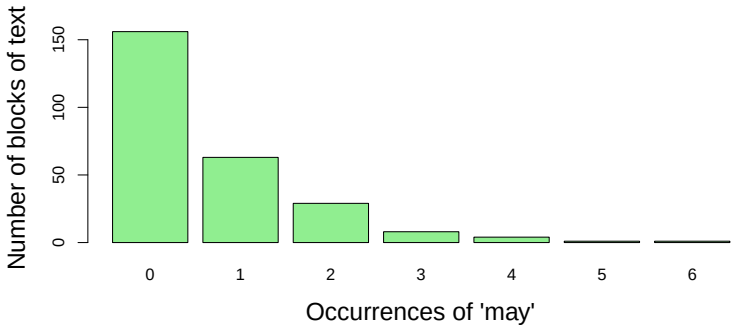

Figure 3.6: Mosteller and Wallace Federalist data fig:federalist

△

{ex:cyclists1}

EXAMPLE 3.6: London cycling deaths

Aberdein and Spiegelhalter (2013) observed that from November 5–13, 2013, six people were

killed while cycling in London. How unusual is this number of deaths in less than a two-week period? Was this a freak occurrence, or should Londoners petition for cycling lanes and greater road safety? To answer these question, they obtained data from the UK Department of Transport *Road Safety Data* from 2005–2012 and selected all accident fatalities of cyclists within the city of London.

It seems reasonable to assume that, in any short period of time, deaths of people riding bicycles are independent events. If, in addition, the probability of such events is constant over this time span, the Poisson distribution should describe the distribution of $0, 1, 2, 3, \ldots$ deaths. Then, an answer to the main question can be given in terms of the probability of six (or more) deaths in a comparable period of time.

Their data, comprising 208 counts of deaths in the fortnightly periods from January 2005 to December 2012 are contained in the data set `CyclingDeaths` in vcdExtra. To work with the distribution, we first convert this to a one-way table.

```
> data("CyclingDeaths", package = "vcdExtra")
> CyclingDeaths.tab <- table(CyclingDeaths$deaths)
> CyclingDeaths.tab


  0   1   2   3
114  75  14   5
```

The maximum number of deaths was 3, which occurred in only 5 two-week periods. The distribution is plotted in Figure 3.7.

```
> barplot(CyclingDeaths.tab,
+         xlab = "Number of deaths", ylab = "Number of fortnights",
+         col = "pink", cex.lab = 1.5)
```

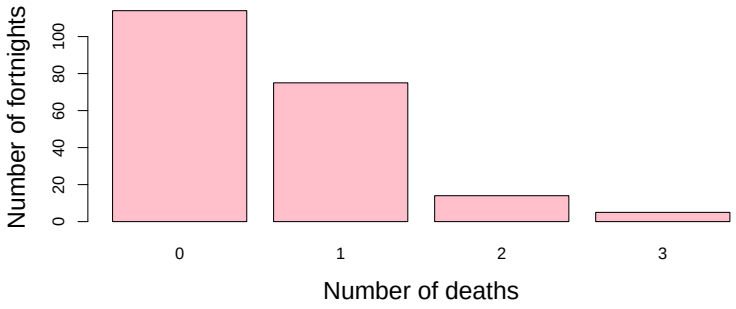

Figure 3.7: Frequencies of number of cyclist deaths in two-week periods in London, 2005–2012 fig:cyclists2

We return to this data in Example 3.10 and answer the question of how unusual six or more deaths would be in a Poisson distribution.

△

3.1.3 Type-token distributions

{sec:type-token}

There are a variety of other types of discrete data distributions. One important class is ***type-token*** distributions, where the basic count k is the number of distinct types of some observed event, $k =$

$1, 2, \ldots$ and the frequency, n_k, is the number of different instances observed. For example, distinct words in a book, words that subjects list as members of the semantic category "fruit", musical notes that appear in a score, and species of animals caught in traps can be considered as types, and the occurrences of of those type comprise tokens.

This class differs from the Poisson type considered above in that the frequency for value $k = 0$ is *unobserved*. Thus, questions like (a) How many words did Shakespeare know? (b) How many words in the English language are members of the "fruit" category? (c) How many wolves remain in Canada's Northwest territories? depend on the unobserved count for $k = 0$. They cannot easily be answered without appeal to additional information or statistical theory. {ex:butterfly}

EXAMPLE 3.7: Butterfly species in Malaya

In studies of the diversity of animal species, individuals are collected and classified by species. The distribution of the number of species (types) where $k = 1, 2, \ldots$ individuals (tokens) were collected forms a kind of type-token distribution. An early example of this kind of distribution was presented by Fisher *et al.* (1943). Table 3.5 lists the number of individuals of each of 501 species of butterfly collected in Malaya. There were thus 118 species for which just a single instance was found, 74 species for which two individuals were found, down to 3 species for which 24 individuals were collected. Fisher et al. note however that the distribution was truncated at $k = 24$. Type-token distributions are often J-shaped, with a long upper tail, as we see in Figure 3.8.

Table 3.5: Number of butterfly species n_k for which k individuals were collected {tab:buttertab}

Individuals (k)	1	2	3	4	5	6	7	8	9	10	11	12	
Species (n_k)	118	74	44	24	29	22	20	19	20	15	12	14	
Individuals (k)	13	14	15	16	17	18	19	20	21	22	23	24	Sum
Species (n_k)	6	12	6	9	9	6	10	10	11	5	3	3	501

```
> data(Butterfly, package = "vcd")
> barplot(Butterfly, xlab = "Number of individuals", ylab = "Number of species",
+         cex.lab = 1.5)
```

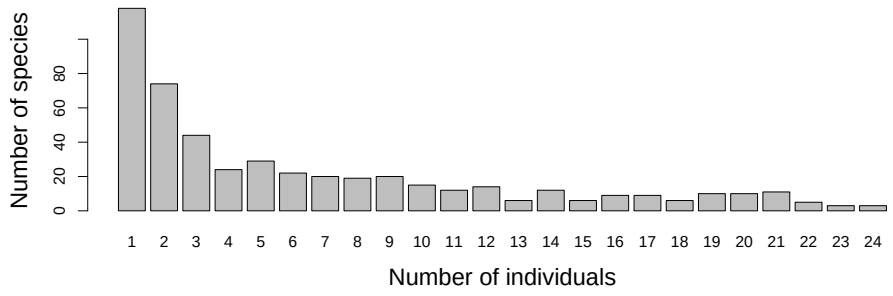

Figure 3.8: Butterfly species in Malaya fig:butterfly

△

3.2 Characteristics of discrete distributions

{sec:discrete-distrib}

This section briefly reviews the characteristics of some of the important discrete distributions encountered in practice and illustrates their use with R. An overview of these distributions is shown in Table 3.6. For more detailed information on these and other discrete distributions, Johnson *et al.* (1992) and Wimmer and Altmann (1999) present the most comprehensive treatments; Zelterman (1999, Chapter 2) gives a compact summary.

Table 3.6: Discrete probability distributions tab:distns

Discrete distribution	Probability function, $p(k)$	parameter(s)
Binomial	$\binom{n}{k} p^k (1-p)^{n-k}$	p = Pr (success); n = # trials
Poisson	$e^{-\lambda}\lambda^k/k!$	λ = mean
Negative binomial	$\binom{n+k-1}{k} p^n (1-p)^k$	p; n = # *successful* trials
Geometric	$p(1-p)^k$	p
Logarithmic series	$\theta^k/[-k\log(1-\theta)]$	θ

For each distribution, we describe properties and generating mechanisms, and show how its parameters can be estimated and how to plot the frequency distribution. R has a wealth of functions for a wide variety of distributions. For ease of reference, their names and types for the distributions covered here are shown in Table 3.7. The naming scheme is simple and easy to remember: for each distribution, there are functions, with a prefix letter, `d`, `p`, `q`, `r`, followed by the name for that class of distribution:[4]

d a density function,[5] $\Pr\{X = k\} \equiv p(k)$ for the probability that the variable X takes the value k.
p a cumulative probability function, or CDF, $F(k) = \sum_{X \le k} p(k)$.
q a quantile function, the inverse of the CDF, $k = F^{-1}(p)$. The quantile is defined as the smallest value x such that $F(k) \ge p$.
r a random number generating function for that distribution.

In the R console, `help(Distributions)` gives an overview listing of the distribution functions available in the stats package.

Table 3.7: R functions for discrete probability distributions tab:distfuns

Discrete distribution	Density (pmf) function	Cumulative (CDF)	Quantile CDF^{-1}	Random # generator
Binomial	`dbinom()`	`pbinom()`	`qbinom()`	`rbinom()`
Poisson	`dpois()`	`ppois()`	`qpois()`	`rpois()`
Negative binomial	`dnbinom()`	`pnbinom()`	`qnbinom()`	`rnbinom()`
Geometric	`dgeom()`	`pgeom()`	`qgeom()`	`rgeom()`
Logarithmic series	`dlogseries()`	`plogseries()`	`qlogseries()`	`rlogseries()`

[4] The CRAN Task View on Probability Distributions, `http://cran.r-project.org/web/views/Distributions.html`, provides a general overview and lists a wide variety of contributed packages for specialized distributions, discrete and continuous.

[5] For discrete random variables this is usually called the probability mass function (pmf).

{sec:binomial}

3.2.1 The binomial distribution

The binomial distribution, $\mathrm{Bin}(n, p)$, arises as the distribution of the number k of events of interest which occur in n independent trials when the probability of the event on any one trial is the constant value $p = \Pr(\text{event})$. For example, if 15% of the population has red hair, the number of red-heads in randomly sampled groups of $n = 10$ might follow a binomial distribution, $\mathrm{Bin}(10, 0.15)$; in Weldon's dice data (Example 3.3), the probability of a 5 or 6 should be $\frac{1}{3}$ on any one trial, and the number of 5s or 6s in tosses of 12 dice would follow $\mathrm{Bin}(12, \frac{1}{3})$.

Over n independent trials, the number of events k may range from 0 to n; if X is a random variable with a binomial distribution, the probability that $X = k$ is given by

$$\mathrm{Bin}(n, p) : \Pr\{X = k\} \equiv p(k) = \binom{n}{k} p^k (1-p)^{n-k} \qquad k = 0, 1, \ldots, n \ , \tag{3.1}$$

{eq:binom}

where $\binom{n}{k} = n!/k!(n-k)!$ is the number of ways of choosing k out of n. The first three (central) moments of the binomial distribution are as follows (letting $q = 1 - p$),

$$\begin{aligned} \mathrm{Mean}(X) &= np \\ \mathrm{Var}(X) &= npq \\ \mathrm{Skew}(X) &= npq(q-p) \ . \end{aligned}$$

It is easy to verify that the binomial distribution has its maximum variance when $p = \frac{1}{2}$. It is symmetric (Skew(x)=0) when $p = \frac{1}{2}$, and negatively (positively) skewed when $p < \frac{1}{2}$ ($p > \frac{1}{2}$).

If we are given data in the form of a discrete (binomial) distribution (and n is known), then the maximum likelihood estimator of p can be obtained as the weighted mean of the values k with weights n_k,

$$\hat{p} = \frac{\bar{x}}{n} = \frac{(\sum_k k \times n_k) / \sum_k n_k}{n} \ ,$$

and has sampling variance $\mathcal{V}(\hat{p}) = pq/n$.

TODO: DM: either add ref to some text explaining Maximum Likelihood estimation, or maybe add a section similar to old book to the Appendix), or add a note in preface that this is assumed to be known.

3.2.1.1 Calculation and visualization

As indicated in Table 3.7 (but without listing the parameters of these functions), binomial probabilities can be calculated with `dbinom(x, n, p)`, where `x` is a vector of the number of successes in `n` trials and `p` is the probability of success on any one trial. Cumulative probabilities, summed up to a vector of quantiles, `Q` can be calculated with `pbinom(Q, n, p)`, and the quantiles (the smallest value x such that $F(x) \geq P$) with `qbinom(P, n, p)`. To generate `N` random observations from a binomial distribution with `n` trials and success probability `p` use `rbinom(N, n, p)`[6].

For example, to find and plot the binomial probabilities corresponding to Weldon's tosses of 12 dice, with $k = 0, \ldots, 12$ and $p = \frac{1}{3}$, we could do the following

```
> k <- seq(0, 12)
> Pk <- dbinom(k, 12, 1/3)
> plot(x = k, y = Pk, type = "h",
+      xlab = "Number of successes", ylab = "Probability",
+      lwd = 8, lend = "square")
> lines(x = k, y = Pk)
```

TODO: DM: Why not directly using a barplot?

[6] Note that the actual R function arguments differ from the ones used here.

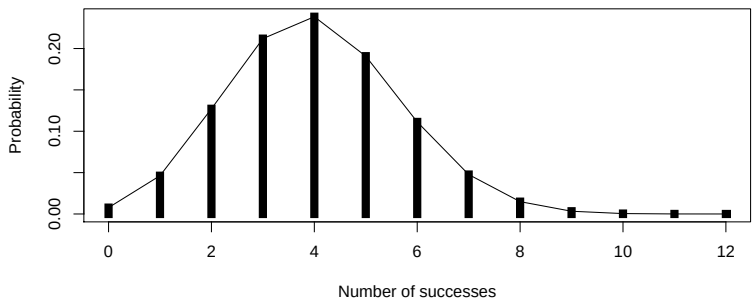

Figure 3.9: Binomial distribution for $k = 0, \ldots, 12$ successes in 12 trials and p=1/3 fig:dbinom1

```
> k <- 0 : 12
> Pk <- dbinom(k, 12, 1/3)
> b <- barplot(Pk, names.arg = k,
+              xlab = "Number of successes", ylab = "Probability")
> lines(x = b, y = Pk, col = "red")
```

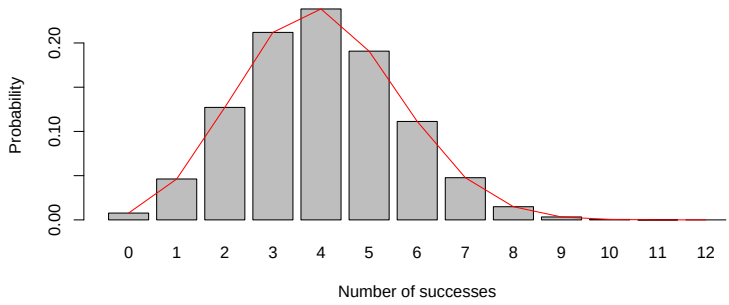

Figure 3.10: Binomial distribution for $k = 0, \ldots, 12$ successes in 12 trials and p=1/3 fig:dbinom12

Note that in the call to `plot()`, `type = "h"` draws histogram-typed lines to the bottom of the vertical axis, and `lwd = 8` makes them wide. The call to `lines()` shows another way to plot the data, as a probability polygon. We illustrate other styles for plotting in Section 3.2.2, Example 3.11 below. {ex:dice2}

EXAMPLE 3.8: Weldon's dice

Going a bit further, we can compare Weldon's data with the theoretical binomial distribution as shown below. Because the *WeldonDice* data collapsed the frequencies for 10–12 successes as 10+, we do the same with the binomial probabilities. The expected frequencies (`Exp`), if Weldon's dice tosses obeyed the binomial distribution, are calculated as $N \times p(k)$ for $N = 26,306$ tosses. In addition, we compute the differences of the observed (`Freq`) and expected (`Exp`) frequencies as column `Diff`, to be used for the χ^2 test for goodness of fit described later in Section 3.3, but a glance these are all negative for $k = 0, \ldots 4$ and positive thereafter.

```
> Weldon_df <- as.data.frame(WeldonDice) # convert to data frame
>
> k <- 0 : 12                          # same as seq(0, 12)
> Pk <- dbinom(k, 12, 1/3)             # binomial probabilities
> Pk <- c(Pk[1:10], sum(Pk[11:13]))    # sum values for 10+
> Exp <- round(26306 * Pk)             # expected frequencies
> Diff <- Weldon_df$Freq - Exp         # raw residuals
> Chisq <- Diff^2 / Exp
> data.frame(Weldon_df, Prob = round(Pk, 5), Exp, Diff, Chisq)

   n56 Freq    Prob  Exp Diff   Chisq
1    0  185 0.00771  203  -18 1.59606
2    1 1149 0.04624 1216  -67 3.69161
3    2 3265 0.12717 3345  -80 1.91330
4    3 5475 0.21195 5576 -101 1.82945
5    4 6114 0.23845 6273 -159 4.03013
6    5 5194 0.19076 5018  176 6.17298
7    6 3067 0.11127 2927  140 6.69628
8    7 1331 0.04769 1255   76 4.60239
9    8  403 0.01490  392   11 0.30867
10   9  105 0.00331   87   18 3.72414
11 10+   18 0.00054   14    4 1.14286
```

△

Finally, we can use programming features in R to calculate and plot probabilities for binomial distributions over a range of both `k` and `p` as follows, for the purposes of graphing the distributions as one or both varies. The following code uses `expand.grid()` to create a data frame `KP` containing all combinations of `k = 0:12` and `p = c(1/6, 1/3, 1/2, 2/3)`. These values are then supplied as arguments to `dbinom()`. For the purpose of plotting, the decimal value of `p` is declared as a factor.

```
> KP <- expand.grid(k = 0 : 12, p = c(1/6, 1/3, 1/2, 2/3))
> bin_df <- data.frame(KP, prob = dbinom(KP$k, 12, KP$p))
> bin_df$p <- factor(bin_df$p, labels = c("1/6", "1/3", "1/2", "2/3"))
> str(bin_df)

'data.frame': 52 obs. of  3 variables:
 $ k   : int  0 1 2 3 4 5 6 7 8 9 ...
 $ p   : Factor w/ 4 levels "1/6","1/3","1/2",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ prob: num  0.1122 0.2692 0.2961 0.1974 0.0888 ...
```

This data can be plotted using `xyplot()` in lattice, using the `groups` argument to make separate curves for each value of `p`. The following code generates Figure 3.11.

```
> library(lattice)
> mycol <- palette()[2:5]
> xyplot(prob ~ k, data = bin_df, groups = p,
+   xlab = list("Number of successes", cex = 1.25),
+   ylab = list("Probability",  cex = 1.25),
+   type = "b", pch = 15:17, lwd = 2, cex = 1.25, col = mycol,
+   key = list(
+     title = "Pr(success)",
+     points = list(pch = 15 : 17, col = mycol, cex = 1.25),
+     lines = list(lwd = 2, col = mycol),
+     text = list(levels(bin_df$p)),
+     x = 0.9, y = 0.98, corner = c(x = 1, y = 1)
+     )
+   )
```

TODO: DM: Avoid lattice to reduce complexity. Either only use ggplot2 throughout the chpater, or use base graphics here, and ggplot2 for the remaining plots:

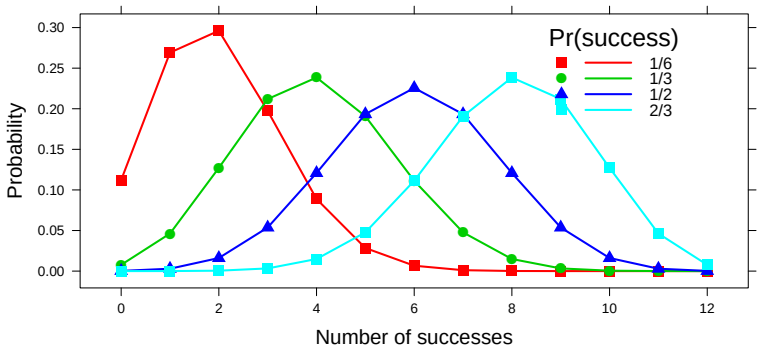

Figure 3.11: Binomial distributions for $k = 0, \ldots, 12$ successes in $n = 12$ trials, and four values of p

fig:dbinom2-plot

```
> p <- c(1/6, 1/3, 1/2, 2/3)
> k <- 0 : 12
> Prob <- outer(k, p, function(k, p) dbinom(k, 12, p))
> str(Prob)

 num [1:13, 1:4] 0.1122 0.2692 0.2961 0.1974 0.0888 ...
```

```
> col <- palette()[2:5]
> matplot(k, Prob,
+         type = "o", pch = 15 : 17, col = col, lty = 1,
+         xlab = "Number of Successes", ylab = "Probability")
> legend("topright", legend = c("1/6","1/3","1/2","2/3"),
+        pch = 15 : 17, lty = 1, col = col, title = "Pr(Success)")
```

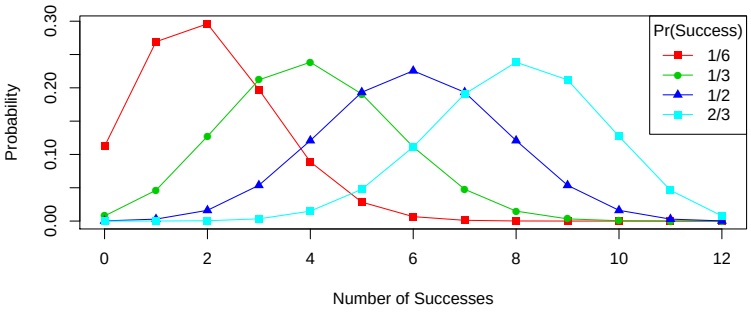

Figure 3.12: Binomial distributions for $k = 0, \ldots, 12$ successes in $n = 12$ trials, and four values of p

fig:dbinom2-plot2

3.2.2 The Poisson distribution

{sec:poisson}

The Poisson distribution gives the probability of an event occurring $k = 0, 1, 2, \ldots$ times over a large number of independent "trials", when the probability, p, that the event occurs on any one trial

(in time or space) is small and constant. Hence, the Poisson distribution is usually applied to the study of rare events such as highway accidents at a particular location, deaths from horse kicks, or defects in a well-controlled manufacturing process. Other applications include: the number of customers contacting a call center per unit time; the number of insurance claims per unit region or unit time; number of particles emitted from a small radioactive sample.

For the Poisson distribution, the probability function is

$$\text{Pois}(\lambda) : \Pr\{X = k\} \equiv p(k) = \frac{e^{-\lambda}\,\lambda^k}{k!} \qquad k = 0, 1, \ldots \tag{3.2}$$ {eq:poisf}

where the rate parameter, λ (> 0), turns out to be the mean of the distribution. The first three (central) moments of the Poisson distribution are:

$$\begin{aligned} \text{Mean}(X) &= \lambda \\ \text{Var}(X) &= \lambda \\ \text{Skew}(X) &= \lambda^{-1/2} \end{aligned}$$

So, the mean and variance of the Poisson distribution are always the same, which is sometimes used to identify a distribution as Poisson. For the binomial distribution, the mean (Np) is always greater than the variance (Npq); for other distributions (negative binomial and geometric) the mean is less than the variance. The Poisson distribution is always positively skewed, but skewness decreases as λ increases.

The maximum likelihood estimator of the parameter λ in Eqn. (3.2) is just the mean of the distribution,

$$\hat{\lambda} = \bar{x} = \frac{\sum_k k\, n_k}{\sum_k n_k} \ . \tag{3.3}$$ {eq:pois-lambda}

Hence, the expected frequencies can be estimated by substituting the sample mean into Eqn. (3.2) and multiplying by the total sample size N.

There are many useful properties of the Poisson distribution.**TODO**: DM: Better add some book ref? Wikipedia is not persistent ... [7] Among these:

- Poisson variables have a nice reproductive property: if $X_1, X_2, \ldots X_m$ are independent Poisson variables with the same parameter λ, then their sum, $\sum X_i$ is a Poisson variate with parameter $m\lambda$; if the Poisson parameters differ, the sum is still Poisson with parameter $\sum \lambda_i$.
- For two or more independent Poisson variables, $X_1 \sim \text{Pois}(\lambda_1), X_2 \sim \text{Pois}(\lambda_2), \ldots$, with rate parameters $\lambda_1, \lambda_2 \ldots$, the distribution of any X_i, *conditional on their sum*, $\sum_j X_j = n$, is binomial, $\text{Bin}(n, p)$, where $p = \lambda_i / \sum_j \lambda_j$.
- As λ increases, the Poisson distribution becomes increasingly symmetric, and approaches the normal distribution $N(\lambda, \lambda)$ with mean and variance λ as $\lambda \to \infty$. The approximation is quite good with $\lambda > 20$.
- If $X \sim \text{Pois}(\lambda)$, then $\sqrt{X}$ converges much faster to a normal distribution $N(\lambda, \frac{1}{4})$, with mean $\sqrt{\lambda}$ and constant variance $\frac{1}{4}$. Hence, the square root transformation is often recommended as a *variance stabilizing* transformation for count data when classical methods (ANOVA, regression) assuming normality are employed.

{ex:soccer}

EXAMPLE 3.9: UK Soccer scores

Table 3.8 gives the distributions of goals scored by the 20 teams in the 1995/96 season of the Premier League of the UK Football Association as presented originally by Lee (1997), and now available as the two-way table `UKSoccer` in the vcd package. Over a season each team plays each

[7] See: `http://en.wikipedia.org/wiki/Poisson_distribution`

Table 3.8: Goals scored by home and away teams in 380 games in the Premier Football League, 1995/96 season {tab:soccer1}

Home Team Goals	Away Team Goals 0	1	2	3	4+	Total
0	27	29	10	8	2	76
1	59	53	14	12	4	142
2	28	32	14	12	4	90
3	19	14	7	4	1	45
4+	7	8	10	2	0	27
Total	140	136	55	38	11	380

other team exactly once, so there are a total of $20 \times 19 = 380$ games. Because there may be an advantage for the home team, the goals scored have been classified as "home team" goals and "away team" goals in the table. Of interest for this example is whether the number of goals scores by home teams and away teams follow Poisson distributions, and how this relates to the distribution of the total number of goals scored.

If we assume that in any small interval of time there is a small, constant probability that the home team or the away team may score a goal, the distributions of the goals scored by home teams (the row totals in Table 3.8) may be modeled as Pois(λ_H) and the distribution of the goals scored by away teams (the column totals) may be modeled as Pois(λ_A).

If the number of goals scored by the home and away teams are independent[8], we would expect that the total number of goals scored in any game would be distributed as Pois($\lambda_H + \lambda_A$). These totals are shown in Table 3.9.

{tab:soccer2} **Table 3.9:** Total goals scored in 380 games in the Premier Football League, 1995/95 season

Total goals	0	1	2	3	4	5	6	7
Number of games	27	88	91	73	49	31	18	3

As a preliminary check of the distributions for the home and away goals, we can determine if the means and variances are reasonably close to each other. If so, then the total goals variable should also have a mean and variance equal to the sum of those statistics for the home and away goals.

In the R code below, we first convert the two-way frequency table `UKSoccer` to a data frame in frequency form. We use `within()` to convert `Home` and `Away` to numeric variables, and calculate `Total` as their sum.

```
> data(UKSoccer, package = "vcd")
>
> soccer.df <- as.data.frame(UKSoccer, stringsAsFactors = FALSE)
> soccer.df <- within(soccer.df, {
+   Home <- as.numeric(Home)        # make numeric
+   Away <- as.numeric(Away)        # make numeric
+   Total <- Home + Away            # total goals
```

[8] This question is examined visually in Chapter 5 (Example 5.5) and Chapter 6 (Example 6.11), where we find that the answer is "basically, yes".

```
+ })
> str(soccer.df)

'data.frame': 25 obs. of  4 variables:
 $ Home : num  0 1 2 3 4 0 1 2 3 4 ...
 $ Away : num  0 0 0 0 0 1 1 1 1 1 ...
 $ Freq : num  27 59 28 19 7 29 53 32 14 8 ...
 $ Total: num  0 1 2 3 4 1 2 3 4 5 ...
```

To calculate the mean and variance of these variables, first expand the data frame to 380 individual observations using `expand.dft()`. Then use `apply()` over the rows to calculate the mean and variance in each column.

```
> soccer.df <- expand.dft(soccer.df)    # expand to ungrouped form
> apply(soccer.df, 2, FUN = function(x) c(mean = mean(x), var = var(x)))

       Home   Away  Total
mean 1.4868 1.0632 2.5500
var  1.3164 1.1728 2.6175
```

The means are all approximately equal to the corresponding variances. More to the point, the variance of the `Total` score is approximately equal to the sum of the individual variances. Note also there does appear to be an advantage for the home team, of nearly half a goal.

△

{ex:cyclists2}

EXAMPLE 3.10: London cycling deaths

A quick check of whether the numbers of deaths among London cyclists follows the Poisson distribution can be carried out by calculating the mean and variance. The ***index of dispersion***, the ratio of the variance to the mean, is commonly used to quantify whether a set of observed frequencies is more or less dispersed than a reference (Poisson) distribution.

```
> with(CyclingDeaths, c(mean = mean(deaths),
+                       var = var(deaths),
+                       ratio = mean(deaths) / var(deaths)))

   mean     var   ratio
0.56731 0.52685 1.07679
```

Thus, there was an average of about 0.57 deaths per fortnight, or a bit more than 1 per month, and no evidence for over- or underdispersion.

We can now answer the question of whether it was an extraordinary event to observe six deaths in a two-week period, by calculating the probability of more than 5 deaths using `ppois()`.

```
> mean.deaths <- mean(CyclingDeaths$deaths)
> ppois(5, mean.deaths, lower.tail = FALSE)

[1] 2.8543e-05
```

This probability is extremely small, so we conclude that the occurrence of six deaths was a singular event. The interpretation of this result might indicate an increased risk to cycling in London, and might prompt further study of road safety. △

3.2.2.1 Calculation and visualization

For the Poisson distribution, you can generate probabilities using `dpois(x, lambda)` for the numbers of events in `x` with rate parameter `lambda`. As we did earlier for the binomial distribution, we can calculate these for a collection of values of `lambda` by using `expand.grid()` to create all combinations of with the values of `x` we wish to plot.

{ex:dpois-plot}

EXAMPLE 3.11: Plotting styles for discrete distributions

In this example, we illustrate some additional styles for plotting discrete distributions, using both lattice `xyplot()` and the ggplot2 package. The goal here is to visualize a collection of Poisson distributions for varying values of λ.

We first create the 63 combinations of `x = 0 : 20` for three values of λ, `lambda = c(1, 4, 10)`, and use these columns as arguments to `dpois()`. Again, `lambda` is a numeric variable, but the plotting methods are easier if this variable is converted to a factor.

```
> KL <- expand.grid(k = 0 : 20, lambda = c(1, 4, 10))
> pois_df <- data.frame(KL, prob = dpois(KL$k, KL$lambda))
> pois_df$lambda = factor(pois_df$lambda)
> str(pois_df)

'data.frame': 63 obs. of  3 variables:
 $ k     : int  0 1 2 3 4 5 6 7 8 9 ...
 $ lambda: Factor w/ 3 levels "1","4","10": 1 1 1 1 1 1 1 1 1 1 ...
 $ prob  : num  0.3679 0.3679 0.1839 0.0613 0.0153 ...
```

Discrete distributions are often plotted as bar charts or in histogram-like form, as we did for the examples in Section 3.1, rather than the line-graph form used for the binomial distribution in Figure 3.11. With `xyplot()`, the plot style is controlled by the `type` argument, and the code below uses `type = c("h", "p")` to get *both* histogram-like lines to the origin and points. As well, the plot formula, `prob ~ k | lambda` instructs `xyplot()` to produce a multi-panel plot, conditioned on values of `lambda`. These lines produce Figure 3.13.

```
> xyplot(prob ~ k | lambda, data = pois_df,
+   type = c("h", "p"), pch = 16, lwd = 4, cex = 1.25, layout = c(3, 1),
+   xlab = list("Number of events (k)", cex = 1.25),
+   ylab = list("Probability", cex = 1.25))
```

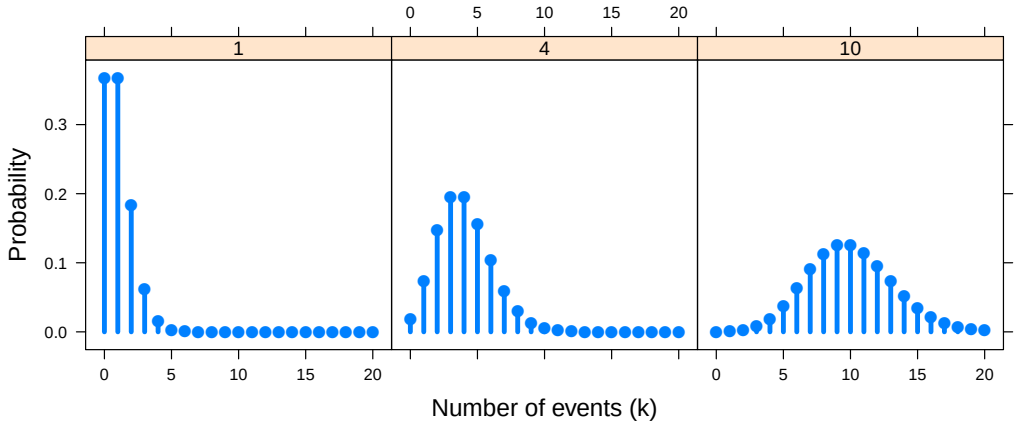

Figure 3.13: Poisson distributions for $\lambda = 1, 4, 10$, in a multi-panel display fig:dpois-xyplot1

The line-graph plot style of Figure 3.11 has the advantage that it is easier to compare the separate distributions in a single plot (using the `groups` argument) than across multiple panels (using a conditioning formula). It has the disadvantages that (a) a proper legend is difficult to construct with lattice, and (b) is difficult to read, because you have to visually coordinate the curves in the plot with the values shown in the legend. Figure 3.14 solves both problems using the directlabels package.

```
> mycol <- palette()[2:4]
> plt <- xyplot(prob ~ k, data = pois_df, groups = lambda,
+   type = "b", pch = 15 : 17, lwd = 2, cex = 1.25, col = mycol,
+   xlab = list("Number of events (k)", cex = 1.25),
+   ylab = list("Probability",  cex = 1.25))
>
> library(directlabels)
> direct.label(plt, list("top.points", cex = 1.5, dl.trans(y = y + 0.1)))
```

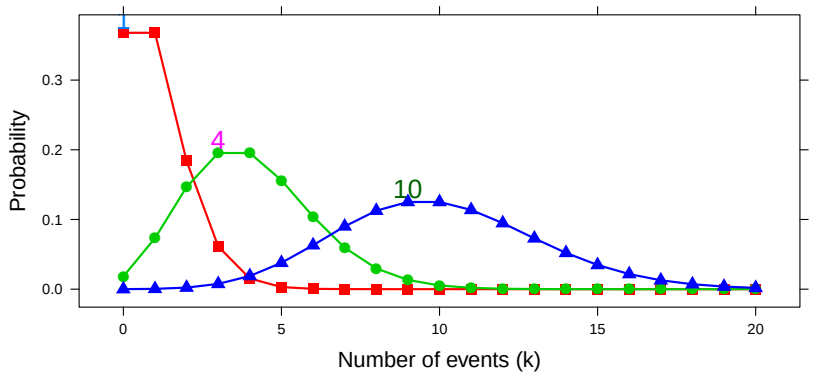

Figure 3.14: Poisson distributions for $\lambda = 1, 4, 10$, using direct labels fig:dpois-xyplot2

Note that the plot constructed by `xyplot()` is saved as a ("trellis") object, `plt`. The function `direct.label()` massages this to add the labels directly to each curve. In the second argument above, `"top.points"` says to locate these at the maximum value on each curve.

Finally, we illustrate the use of ggplot2 to produce a single-panel, multi-line plot of these distributions. The basic plot uses `aes(x = k, y = prob, ...)` to produce a plot of `prob` vs. `k`, assigning color and shape attributes to the values of `lambda`.

```
> library(ggplot2)
> gplt <- ggplot(pois_df, aes(x = k, y = prob, colour = lambda, shape = lambda)) +
+   geom_line(size = 1) + geom_point(size = 3) +
+         xlab("Number of events (k)") +
+         ylab("Probability")
```

ggplot2 allows most details of the plot to be modified using `theme()`. Here we use this to move the legend inside the plot, and enlarge the axis labels and titles.

```
> gplt + theme(legend.position = c(0.8, 0.8)) +  # manually move legend
+        theme(axis.text = element_text(size = 12),
+              axis.title = element_text(size = 14, face = "bold"))
```

△

3.2.3 The negative binomial distribution

{sec:negbin}

The negative binomial distribution is a type of waiting-time distribution, but also arises in statistical applications as a generalization of the Poisson distribution, allowing for ***overdispersion*** (variance > mean). See Hilbe (2011) for a comprehensive treatment of negative binomial statistical models with many applications in R.

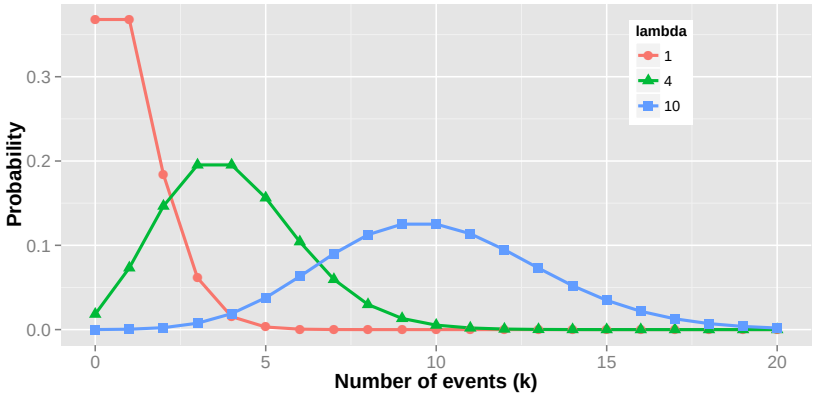

Figure 3.15: Poisson distributions for $\lambda = 1, 4, 10$, using ggplot fig:dpois-ggplot2

One form of the negative binomial distribution (also called the ***Pascal distribution***) arises when a series of independent Bernoulli trials is observed with constant probability p of some event, and we ask how many non-events (failures), k, it takes to observe n successful events. For example, in tossing one die repeatedly, we may consider the outcome "1" as a "success" (with $p = \frac{1}{6}$) and ask about the probability of observing $k = 0, 1, 2, \ldots$ failures before getting $n = 3$ 1s.

The probability function with parameters n (a positive integer, $0 < n < \infty$) and $p\,(0 < p < 1)$ gives the probability that k non-events (failures) are observed before the n-th event (success), and can be written[9]

{eq:negbinf}

$$\text{NBin}(n, p) : \Pr\{X = k\} \equiv p(k) = \binom{n+k-1}{k} p^n (1-p)^k \qquad k = 0, 1, \ldots, \infty \tag{3.4}$$

This formulation makes clear that a given sequence of events involves a total of $n + k$ trials of which there are n successes, with probability p^n, and k are failures, with probability $(1 - p)^k$. The binomial coefficient, $\binom{n+k-1}{k}$ gives the number of ways to choose the k successes from the remaining $n + k - 1$ trials preceding the last success.

The first three central moments of the negative binomial distribution are:

$$\begin{aligned} \text{Mean}(X) &= nq/p = \mu \\ \text{Var}(X) &= nq/p^2 \\ \text{Skew}(X) &= \frac{2-p}{\sqrt{nq}} \ , \end{aligned}$$

where $q = 1 - p$. The variance of X is therefore greater than the mean, and the distribution is always positively skewed.

A more general form of the negative binomial distribution (the ***Polya distribution***) allows n to take non-integer values and to be an unknown parameter. In this case, the combinatorial coefficient, $\binom{n+k-1}{k}$ in Eqn. (3.4) is calculated using the gamma function, $\Gamma(\bullet)$, a generalization of the factorial for non-integer values, defined so that $\Gamma(x + 1) = x!$ when x is an integer.

[9] There are a variety of other parameterizations of the negative binomial distribution, but all of these can be converted to the form shown here, which is relatively standard, and consistent with R. They differ in whether the parameter n relates to the number of successes or the total number of trials, and whether the stopping criterion is defined in terms of failures or successes. See: http://en.wikipedia.org/wiki/Negative_binomial_distribution for details on these variations.

Then the probability function Eqn. (3.4) becomes

{eq:negbinf2}

$$\Pr\{X = k\} \equiv p(k) = \frac{\Gamma(n+k)}{\Gamma(n)\Gamma(k+1)} p^n (1-p)^k \qquad k = 0, 1, \ldots, \infty \ . \tag{3.5}$$

Greenwood and Yule (1920) developed the negative binomial distribution as a model for accident proneness or susceptibility of individuals to repeated attacks of disease. They assumed that for any individual, i, the number of accidents or disease occurrences has a Poisson distribution with parameter λ_i. If individuals vary in proneness, so that the λ_i have a gamma distribution, the resulting distribution is the negative binomial.

In this form, the negative binomial distribution is frequently used as an alternative to the Poisson distribution when the assumptions of the Poisson (constant probability and independence) are not satisfied, or when the variance of the distribution is greater than the mean (overdispersion). This gives rise to an alternative parameterization in terms of the mean (μ) of the distribution and its relation to the variance. From the relation of the mean and variance to the parameters n, p given above,

$$\text{Mean}(X) = \mu = \frac{n(1-p)}{p} \quad \Longrightarrow \quad p = \frac{n}{n+\mu} \tag{3.6}$$

$$\text{Var}(X) = \frac{n(1-p)}{p^2} \quad \Longrightarrow \quad \text{Var}(X) = \mu + \frac{\mu^2}{n} \tag{3.7}$$

This formulation allows the variance of the distribution to exceed the mean, and in these terms, the "size" parameter n is called the ***dispersion parameter***.[10] Increasing this parameter corresponds to less heterogeneity, variance closer to the mean, and therefore greater applicability of the Poisson distribution.

3.2.3.1 Calculation and visualization

In R, the density (pmf), distribution (CDF), quantile and random number functions for the negative binomial distribution are a bit special, in that the parameterization can be specified using either (n, p) or (n, μ) forms, where $\mu = n(1-p)/p$. In our notation, probabilities can be calculated using `dnbinom()` using the call `dbinom(k, n, p)` or the call `dbinom(k, n, mu=)`, as illustrated below:

```
> k <- 2
> n <- 2 : 4
> p <- 0.2
> dnbinom(k, n, p)

[1] 0.07680 0.03072 0.01024

> (mu <- n * (1 - p) / p)

[1]  8 12 16

> dnbinom(k, n, mu = mu)

[1] 0.07680 0.03072 0.01024
```

Thus, for the distribution with `k = 2` failures and `n = 2 : 4` successes with probability `p = 0.2`, the values `n = 2:4` correspond to means $\mu = 8, 12, 16$ as shown above.

[10] Other terms are "shape parameter," with reference to the mixing distribution of Poissons with varying λ, "heterogeneity parameter," or "aggregation parameter."

As before, we can calculate these probabilities for a range of the combinations of arguments using `expand.grid()`. In the example below, we allow three values for each of `n` and `p` and calculate all probabilities for all values of `k` from 0 to 20. The result, `nbin_df` is like a 3-way, $21 \times 3 \times 3$ array of `prob` values, but in data frame format.

```
> XN <- expand.grid(k = 0 : 20, n = c(2, 4, 6), p = c(0.2, 0.3, 0.4))
> nbin_df <- data.frame(XN, prob = dnbinom(XN$k, XN$n, XN$p))
> nbin_df$n <- factor(nbin_df$n)
> nbin_df$p <- factor(nbin_df$p)
> str(nbin_df)

'data.frame': 189 obs. of  4 variables:
 $ k   : int  0 1 2 3 4 5 6 7 8 9 ...
 $ n   : Factor w/ 3 levels "2","4","6": 1 1 1 1 1 1 1 1 1 1 ...
 $ p   : Factor w/ 3 levels "0.2","0.3","0.4": 1 1 1 1 1 1 1 1 1 1 ...
 $ prob: num  0.04 0.064 0.0768 0.0819 0.0819 ...
```

With 9 combinations of the parameters, it is most convenient to plot these in separate panels, in a 3×3 display. The formula `prob ~ k | n + p` in the call to `xyplot()` constructs plots of `prob` vs. `k` conditioned on the combinations of `n` and `p`.

```
> xyplot(prob ~ k | n + p, data = nbin_df,
+   xlab = list("Number of failures (k)", cex = 1.25),
+   ylab = list("Probability",  cex = 1.25),
+   type = c("h", "p"), pch = 16, lwd = 2,
+   strip = strip.custom(strip.names = TRUE)
+         )
```

It can be readily seen that the mean increases from left to right with `n`, and increases from top to bottom with decreasing `p`. For these distributions, we can also calculate the theory-implied means, μ, across the entire distributions, $k = 0, 1, \ldots \infty$, as shown below.

```
> NP <- expand.grid(n=c(2, 4, 6), p=c(0.2, 0.3, 0.4))
> NP <- within(NP, { mu = n*(1-p)/p })
> # show as matrix
> matrix(NP$mu, 3, 3, dimnames=list(n=c(2,4,6), p=(2:4)/10))

   p
n   0.2     0.3 0.4
  2   8  4.6667   3
  4  16  9.3333   6
  6  24 14.0000   9
```

TODO: DM: maybe simpler?

```
> n <- c(2, 4, 6)
> p <- c(0.2, 0.3, 0.4)
> NP <- outer(n, p, function(n, p) n * (1 - p) / p)
> dimnames(NP) <- list(n = n, p = p)
> NP

   p
n   0.2     0.3 0.4
  2   8  4.6667   3
  4  16  9.3333   6
  6  24 14.0000   9
```

3.2.4 The geometric distribution

{sec:geometric}

The special case of the negative binomial distribution when $n = 1$ is a geometric distribution. We observe a series of independent trials and count the number of non-events (failures) preceding the

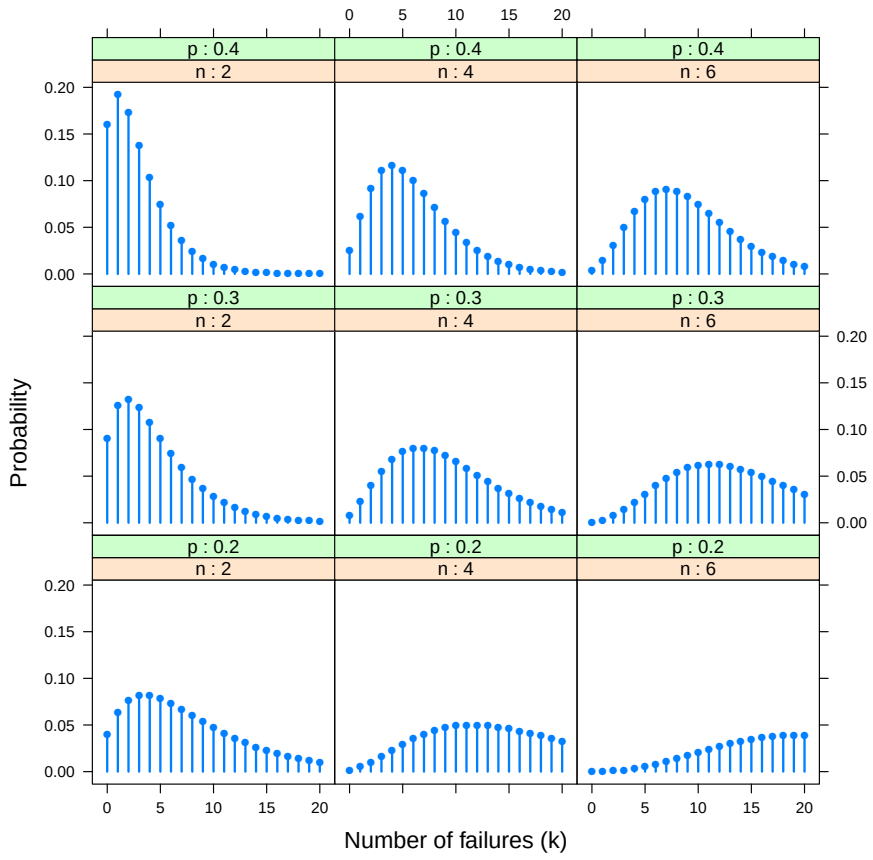

Figure 3.16: Negative binomial distributions for $n = 2, 4, 6$ and $p = 0.2, 0.3, 0.4$, using xyplot fig:dnbin3

first successful event. The probability that there will be k failures before the first success is given by

{eq:geomf}
$$\text{Geom}(p) : \Pr\{X = k\} \equiv p(k) = p(1-p)^k \qquad k = 0, 1, \ldots \ . \tag{3.8}$$

For this distribution the central moments are:

$$\begin{aligned} \text{Mean}(X) &= 1/p \\ \text{Var}(X) &= (1-p)/p^2 \\ \text{Skew}(X) &= (2-p)/\sqrt{1-p} \end{aligned}$$

Note that estimation of the parameter p for the geometric distribution can be handled as the special case of the negative binomial by fixing $n = 1$, so no special software is needed. Going the other way, if $X_1, X_2, \ldots X_n$ are independent geometrically distributed as $\text{Geom}(p)$, then their sum, $Y = \sum_j^n X_j$ is distributed as $\text{NBin}(p, n)$.

In R, the standard set of functions for the geometric distribution are available as `dgeom(x, prob)`, `pgeom(q, prob)`, `qgeom(p, prob)` and `rgeom(n, prob)` where `prob` represents p here. Visualization of the geometric distribution follows the pattern used earlier for other discrete distributions.

3.2.5 The logarithmic series distribution

The logarithmic series distribution is a long-tailed distribution introduced by Fisher *et al.* (1943) in connection with data on the abundance of individuals classified by species of the type shown for the distribution of butterfly species in Table 3.5.

The probability distribution function with parameter p is given by

{eq:logseriesf}
$$\text{LogSer}(p) : \Pr\{X = k\} \equiv p(k) = \frac{p^k}{-(k \log(1-p))} = \alpha\, p^k/k \qquad k = 1, 2, \ldots, \infty \ , \tag{3.9}$$

where $\alpha = -1/\log(1-p)$ and $0 < p < 1$. For this distribution, the first two central moments are:

$$\begin{aligned} \text{Mean}(X) &= \alpha \left(\frac{p}{1-p} \right) \\ \text{Var}(X) &= -p \frac{p + \log(1-p)}{(1-p)^2 \log^2(1-p)} \end{aligned}$$

Fisher derived the logarithmic series distribution by assuming that for a given species the number of individuals trapped has a Poisson distribution with parameter $\lambda = \gamma t$, where γ is a parameter of the species (susceptibility to entrapment) and t is a parameter of the trap. If different species vary so that the parameter γ has a gamma distribution, then the number of representatives of each species trapped will have a negative binomial distribution. However, the observed distribution is necessarily truncated on the left, because one cannot observe the number of species never caught (where $k = 0$). The logarithmic series distribution thus arises as a limiting form of the zero-truncated negative binomial.

Maximum likelihood estimation of the parameter p in the log-series distribution is described by Böhning (1983), extending a simpler Newton's method approximation by Birch (1963). The vcdExtra package contains the set of R functions, `dlogseries(x, prob)`, `plogseries(q, prob)`, `qlogseries(p, prob)` and `rlogseries(n, prob)` where `prob` represents p here.

TODO: implement the log-series in `goodfit()` and `distplot()` so this distribution can be used in later sections.

3.2.6 Power series family

{sec:pwrseries}

We mentioned earlier that the Poisson distribution was unique among all discrete (one parameter) distributions, in that it is the only one whose mean and variance are equal (Kosambi, 1949). The relation between mean and variance of discrete distributions also provides the basis for integrating them into a general family. All of the discrete distributions described in this section are in fact special cases of a family of discrete distributions called the power series distributions by Noack (1950) and defined by

$$p(k) = a(k)\theta^k / f(\theta) \qquad k = 0, 1, \ldots \ ,$$

with parameter $\theta > 0$, where $a(k)$ is a coefficient function depending only on k and $f(\theta) = \sum_k a(k)\theta^k$ is called the series function. The definitions of these functions are shown in Table 3.10.

Table 3.10: The Power Series family of discrete distributions

tab:pwrseries

Discrete Distributiion	Probability function, $p(k)$	Series parameter, θ	Series function, $f(\theta)$	Series coefficient, $a(k)$
Poisson	$e^{-\lambda}\lambda^k/k!$	$\theta = \lambda$	e^{θ}	$1/k!$
Binomial	$\binom{n}{k} p^k (1-p)^{n-k}$	$\theta = p/(1-p)$	$(1+\theta)^n$	$\binom{n}{k}$
Negative binomial	$\binom{n+k-1}{k} p^n (1-p)^k$	$\theta = (1-p)$	$(1-\theta)^{-k}$	$\binom{n+k-1}{k}$
Geometric	$p(1-p)^k$	$\theta = (1-p)$	$(1-\theta)^{-k}$	1
Logarithmic series	$\theta^k/[-k\log(1-\theta)]$	$\theta = \theta$	$-\log(1-\theta)$	$1/k$

These relations among the discrete distribution provide the basis for graphical techniques for diagnosing the form of discrete data described later in this chapter (Section 3.5.4).

3.3 Fitting discrete distributions

{sec:discrete-fit}

In applications to discrete data such as the examples in Section 3.1, interest is often focused on how closely such data follow a particular distribution, such as the Poisson, binomial, or geometric distribution. A close fit provides for interpretation in terms of the underlying mechanism for the distribution; conversely, a bad fit can suggest the possibility for improvement by relaxing one or more of the assumptions. We examine more detailed and nuanced methods for diagnosing and testing discrete distributions in Section 3.4 and Section 3.5 below.

Fitting a discrete distribution involves three basic steps:

1. Estimating the parameter(s) of the distribution from the data, for example, p for the binomial, λ for the Poisson, n and p for the negative binomial. Typically, this is carried out by maximum likelihood methods, or a simpler method of moments, which equates sample moments (mean, variance, skewness) to those of the theoretical distribution, and solves for the parameter estimates. These methods are illustrated in Section 3.3.1.
2. From this, we can calculate the fitted probabilities, $\hat{p}_k$ that apply for the given distribution, or equivalently, the model expected frequencies, $N\hat{p}_k$, where N is the total sample size.
3. Finally, we can calculate goodness-of-fit tests measuring the departure between the observed and fitted frequencies.

Often goodness-of-fit is examined with a classical (Pearson) ***goodness-of-fit*** (GOF) chi-squared test,

$$X^2 = \sum_{k=1}^{K} \frac{(n_k - N\hat{p}_k)^2}{N\hat{p}_k} \sim \chi^2_{(K-s-1)} \ , \tag{3.10}$$ {eq:chi2}

where there are K frequency classes, s parameters have been estimated from the data and $\hat{p}_k$ is the estimated probability of each basic count, under the null hypothesis that the data follows the chosen distribution.

An alternative test statistic is the likelihood-ratio G^2 statistic,

{eq:g2}
$$G^2 = \sum_{k=1}^{K} n_k \log(n_k / N\hat{p}_k) \ , \tag{3.11}$$

when the $\hat{p}_k$ are estimated by maximum likelihood, which also has an asymptotic $\chi^2_{(K-s-1)}$ distribution. "Asymptotic" means that these are *large sample tests*, meaning that the test statistic follows the χ^2 distribution increasingly well as $N \to \infty$. A common rule of thumb is that all *expected* frequencies should exceed one and that fewer than 20% should be less than 5.

{ex:horsekick2}

EXAMPLE 3.12: Death by horse kick

We illustrate the basic ideas of goodness-of fit tests with the `HorseKick` data, where we expect a Poisson distribution with parameter λ = mean number of deaths. As shown in Eqn. (3.3), this is calculated as the frequency (n_k) weighted mean of the k values, here, number of deaths.

In R, such one-way frequency distributions should be converted to data frames with numeric variables. The calculation below uses `weighted.mean()` with the frequencies as weights, and finds $\lambda = 0.61$ as the mean number of deaths per corps-year.

```
> # goodness-of-fit test
> tab <- as.data.frame(HorseKicks, stringsAsFactors = FALSE)
> colnames(tab) <- c("nDeaths", "Freq")
> str(tab)

'data.frame': 5 obs. of  2 variables:
 $ nDeaths: chr  "0" "1" "2" "3" ...
 $ Freq   : int  109 65 22 3 1

> (lambda <- weighted.mean(as.numeric(tab$nDeaths), w = tab$Freq))

[1] 0.61
```

From this, we can calculate the probabilities (`phat`) of `k = 0 : 4` deaths, and hence the expected (`exp`) frequencies in a Poisson distribution.

```
> phat <- dpois(0 : 4, lambda = lambda)
> exp <- sum(tab$Freq) * phat
> chisq <- (tab$Freq - exp)^2 / exp
>
> GOF <- data.frame(tab, phat, exp, chisq)
> GOF

  nDeaths Freq      phat       exp     chisq
1       0  109 0.5433509 108.67017 0.0010011
2       1   65 0.3314440  66.28881 0.0250573
3       2   22 0.1010904  20.21809 0.1570484
4       3    3 0.0205551   4.11101 0.3002534
5       4    1 0.0031346   0.62693 0.2220057
```

Finally, the Pearson χ^2 is just the sum of the `chisq` values and `pchisq()` is used to calculate the p-value of this test statistic—the probability of obtaining this χ^2 or a more extreme value if our assumption on the underlying distribution is true.

```
> sum(chisq)  # chi-square value

[1] 0.70537

> pchisq(sum(chisq), df = nrow(tab) - 2, lower.tail = FALSE)

[1] 0.87194
```

The result, $\chi^2_3 = 0.70537$ shows an extremely good fit of these data to the Poisson distribution, perhaps exceptionally so.[11]

△

3.3.1 R tools for discrete distributions

{sec:fitdistr}

In R, the function `fitdistr()` in the MASS is a basic work horse for fitting a variety of distributions by maximum likelihood and other methods, giving parameter estimates and standard errors. Among discrete distributions, the binomial, Poisson and geometric distributions have closed-form maximum likelihood estimates; the negative binomial distribution, parameterized by (n, μ), is estimated iteratively by direct optimization.

These basic calculations are extended and enhanced for one-way discrete distributions in the vcd function `goodfit()`, which computes the fitted values of a discrete distribution (either Poisson, binomial or negative binomial) to the count data. If the parameters are not specified they are estimated either by ML or Minimum Chi-squared. `print()` and `summary()` methods for the "goodfit" objects give, respectively, a table of observed and fitted frequencies, and the Pearson and/or likelihood ratio goodness-of-fit statistics. Plotting methods for visualizing the discrepancies between observed and fitted frequencies are described and illustrated in Section 3.3.2.

{ex:saxfit}

EXAMPLE 3.13: Families in Saxony

This example uses `goodfit()` to fit the binomial to the distribution of the number of male children in families of size 12 in Saxony. Note that for the binomial, both n and p are considered as parameters, and by default n is taken as the maximum count.

```
> data(Saxony, package = "vcd")
> Sax_fit <- goodfit(Saxony, type = "binomial")
> unlist(Sax_fit$par) # estimated parameters

    prob     size
 0.51922 12.00000
```

So, we estimate the probability of a male in these families to be $p = 0.519$, a value that is quite close to the value found in Arbuthnot's data ($p = 0.517$).

It is useful to know that `goodfit()` returns a list structure of named components which are used by method functions for class "goodfit" objects. The `print.goodfit()` method prints the table of observed and fitted frequencies. `summary.goodfit()` calculates and prints the likelihood ratio χ^2 GOF test when the ML estimation method is used.

```
> names(Sax_fit)      # components of "goodfit" objects

[1] "observed" "count"    "fitted"   "type"     "method"
[6] "df"       "par"
```

[11] An exceptionally good fit occurs when the p-value for the test χ^2 statistic is so high, as to suggest that something unreasonable under random sampling might have occurred. The classic example of this is the controversy over Gregor Mendel's experiments of cross-breeding garden peas with various observed (phenotype) characteristics, where R. A. Fisher (1936) suggested that observed frequencies of combinations like (smooth/wrinkled), (green/yellow) in a 2^{nd} generation were uncomfortably too close to the $3:1$ ratio predicted by genetic theory.

```
> Sax_fit            # print method


Observed and fitted values for binomial distribution
with parameters estimated by `ML'

 count observed     fitted
     0        3    0.93284
     1       24   12.08884
     2      104   71.80317
     3      286  258.47513
     4      670  628.05501
     5     1033 1085.21070
     6     1343 1367.27936
     7     1112 1265.63031
     8      829  854.24665
     9      478  410.01256
    10      181  132.83570
    11       45   26.08246
    12        7    2.34727

> summary(Sax_fit)    # summary method


 Goodness-of-fit test for binomial distribution

                     X^2 df   P(> X^2)
Likelihood Ratio 97.007 11 6.9782e-16
```

Note that the GOF test gives a highly significant p-value (pratically zero), indicating significant lack of fit to the binomial distribution.[12] Some further analysis of this result is explored in examples below. △

{ex:dicefit}

EXAMPLE 3.14: Weldon's dice

Weldon's dice data, explored in Example 3.3, are also expected to follow 9a binomial distribution, here with $p = \frac{1}{3}$. However, as given in the data set `WeldonDice`, the frequencies for counts 10–12 were grouped as "10+". In this case, it necessary to supply the correct value of $n = 12$ as the value of the `size` parameter in the call to `goodfit()`.

```
> data(WeldonDice, package = "vcd")
> dice_fit <- goodfit(WeldonDice, type = "binomial", par = list(size = 12))
> unlist(dice_fit$par)

    prob     size
 0.33769 12.00000
```

The probability of a success (a 5 or 6) is estimated as $\hat{p} = 0.3377$, not far from the theoretical value, $p = 1/3$.

```
> print(dice_fit, digits = 0)


Observed and fitted values for binomial distribution
with parameters estimated by `ML'
```

[12] A handy rule-of-thumb is to think of the ratio of χ^2/df, because, under the null hypothesis of acceptable fit, $\mathcal{E}(\chi^2/df) = 1$, so ratios exceeding ≈ 2.5 are troubling. Here, the ratio is $97/11 = 8.8$, so the lack of fit is substantial.

```
 count observed     fitted
     0      185 1.8742e+02
     1     1149 1.1467e+03
     2     3265 3.2156e+03
     3     5475 5.4650e+03
     4     6114 6.2694e+03
     5     5194 5.1144e+03
     6     3067 3.0422e+03
     7     1331 1.3295e+03
     8      403 4.2366e+02
     9      105 9.6003e+01
    10       18 1.4684e+01
    11        0 1.3613e+00
    12        0 5.7838e-02

> summary(dice_fit)


 Goodness-of-fit test for binomial distribution

                     X^2 df P(> X^2)
Likelihood Ratio 11.506  9   0.2426
```

Here, we find an acceptable fit for the binomial distribution. △

{ex:HKfit}

EXAMPLE 3.15: Death by horse kick

This example reproduces the calculations done "manually" in Example 3.12 above. We fit the Poisson distribution to the *HorseKicks* data by specifying `type = "poisson"` (actually, that is the default for `goodfit()`).

```
> data("HorseKicks", package = "vcd")
> HK_fit <- goodfit(HorseKicks, type = "poisson")
> HK_fit$par

$lambda
[1] 0.61

> HK_fit


Observed and fitted values for poisson distribution
with parameters estimated by `ML'

 count observed    fitted
     0      109 108.67017
     1       65  66.28881
     2       22  20.21809
     3        3   4.11101
     4        1   0.62693
```

The `summary` method uses the LR test by default, so the `X^2` value reported below differs slightly from the Pearson χ^2 value shown earlier.

```
> summary(HK_fit)


 Goodness-of-fit test for poisson distribution

                      X^2 df P(> X^2)
Likelihood Ratio 0.86822  3  0.83309
```

△

{ex:Fedfit}

EXAMPLE 3.16: Federalist papers

In Example 3.5 we examined the distribution of the marker word "may" in blocks of text in the *Federalist Papers* written by James Madison. A naive hypothesis is that these occurrences might follow a Poisson distribution, that is, as independent occurrences with constant probability across the 262 blocks of text. Using the same methods as above, we fit these data to the Poisson distribution:

```
> data("Federalist", package = "vcd")
> Fed_fit0 <- goodfit(Federalist, type = "poisson")
> unlist(Fed_fit0$par)

 lambda
0.65649

> Fed_fit0


Observed and fitted values for poisson distribution
with parameters estimated by `ML'

 count observed     fitted
     0      156 135.891389
     1       63  89.211141
     2       29  29.283046
     3        8   6.407995
     4        4   1.051694
     5        1   0.138085
     6        1   0.015109
```

The GOF test below shows a substantial lack of fit, rejecting the assumptions of the Poisson model.

```
> summary(Fed_fit0)


 Goodness-of-fit test for poisson distribution

                     X^2 df   P(> X^2)
Likelihood Ratio 25.243  5 0.00012505
```

Mosteller and Wallace (1963) determined that the negative binomial distribution provided a better fit to these data than the Poisson. We can verify this as follows:

```
> Fed_fit1 <- goodfit(Federalist, type = "nbinomial")
> unlist(Fed_fit1$par)

   size    prob
1.18633 0.64376

> summary(Fed_fit1)


 Goodness-of-fit test for nbinomial distribution

                   X^2 df P(> X^2)
Likelihood Ratio 1.964  4  0.74238
```

Recall that the Poisson distributionassumes that the probability of a word like *may* appearing in a block of text is small and constant and that for the Poisson, $\mathcal{E}(x) = \mathcal{V}(x) = \lambda$. One interpretation of

the better fit of the negative binomial is that the use of a given word occurs with Poisson frequencies, but Madison varied its rate λ_i from one block of text to another according to a gamma distribution, allowing the variance to be greater than the mean.

△

3.3.2 Plots of observed and fitted frequencies

{sec:fitplot}

In the examples of the last section, we saw cases where the GOF tests showed close agreement between the observed and model-fitted frequencies, and cases where they diverged significantly, to cause rejection of a hypothesis that the data followed the specified distribution.

What is missing from such numerical summaries is any appreciation of the *details* of this statistical comparison. Plots of the observed and fitted frequencies can help to show both the shape of the theoretical distribution we have fitted and the pattern of any deviations between our data and theory.

In this section we illustrate some simple plotting tools for these purposes, using the `plot.goodfit()` method for "goodfit" objects.[13] The left panel of Figure 3.17 shows the fit of the Poisson distribution to the Federalist papers data, using one common form of plot that is sometimes used for this purpose. In this plot, observed frequencies are shown by bars and fitted frequencies are shown by points, connected by a smooth (spline) curve.

Such a plot, however, is dominated by the largest frequencies, making it hard to assess the deviations among the smaller frequencies. To make the smaller frequencies more visible, Tukey (1977) suggest plotting the frequencies on a square-root scale, which he calls a ***rootogram***. This plot is shown in the right panel of Figure 3.17.

```
> plot(Fed_fit0, scale = "raw", type = "standing")
> plot(Fed_fit0, type = "standing")
```

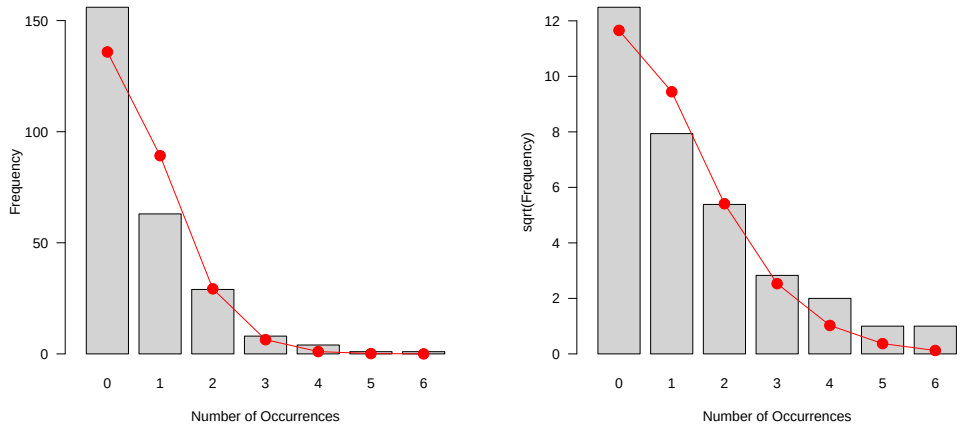

Figure 3.17: Plots for the Federalist Papers data, fitting the Poisson model. Each panel shows the observed frequencies as bars and the fitted frequencies as a smooth curve. Left: raw frequencies; right: plotted on a square root scale to emphasize smaller frequencies.

fig:Fedu-plots1

Additional improvements over the standard plot on the scale of raw frequencies are shown in

[13] Quantile-quantile (QQ) plots are a common alternative for for the goal of comparing observed and expected values under some distribution. These plots are useful for unstructured samples, but less so when we want to also see the shape of a distribution, as is the case here.

Figure 3.18, both of which use the square root scale. The left panel moves the rootogram bars so their tops are at the expected frequencies (giving a ***hanging rootogram***). This has the advantage that we can more easily judge the pattern of departures against the horizontal reference line at 0, than against the curve.

```
> plot(Fed_fit0, type = "hanging")
> plot(Fed_fit0, type = "deviation")
```

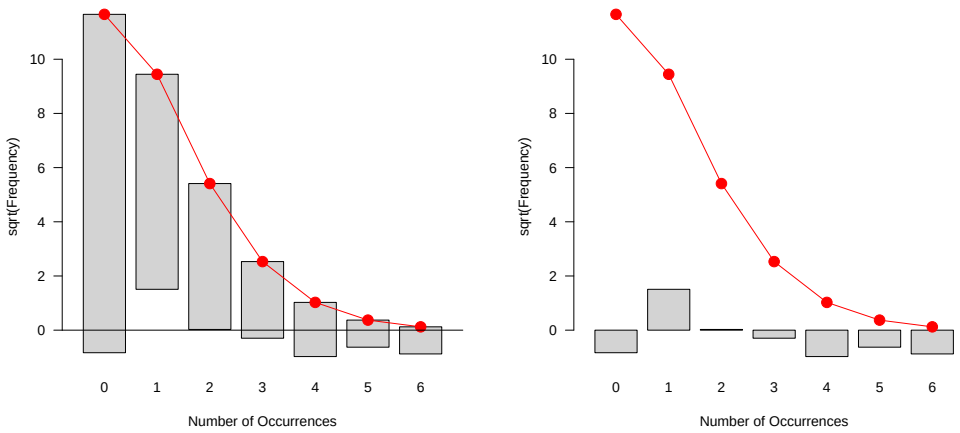

Figure 3.18: Plots for the Federalist Papers data, fitting the Poisson model. Left: hanging rootogram; right: deviation rootogram. `fig:Fed0-plots2`

A final variation is to emphasize the differences between the observed and fitted frequencies by drawing the bars to show the gaps between the 0 line and the (observed−expected) difference (Figure 3.18, right).

All of these plots are actually produced by the `rootogram()` function in vcd. The default is `type = "hanging"`, and there are many options to control the plot details.

The plots in Figure 3.17 and Figure 3.18 used the ill-fitting Poisson model on purpose to highlight how these plots show the departure between the observed and fitted frequencies. Figure 3.19 compares this with the negative binomial model, `Fed_fit1`, which we saw has a much better, and acceptable fit.

```
> plot(Fed_fit0, main = "Poisson")
> plot(Fed_fit1, main = "Negative binomial")
```

Comparing the two plots in Figure 3.19, we can see that the Poisson model underestimates the frequencies of 0 counts and the larger counts for 4–6 occurrences. The deviations for the negative binomial are small and unsystematic.

Finally, Figure 3.20 shows hanging rootograms for two atrociously bad models for the data on butterfly species in Malaya considered in Example 3.7. As we will see in Section 3.4, this long-tailed distribution is better approximated by the logarithmic series distribution, but this distribution is presently not handled by `goodfit()`.

```
> data(Butterfly, package = "vcd")
> But_fit1 <- goodfit(Butterfly, type = "poisson")
> But_fit2 <- goodfit(Butterfly, type = "nbinomial")
> plot(But_fit1, main = "Poisson")
> plot(But_fit2, main = "Negative binomial")
```

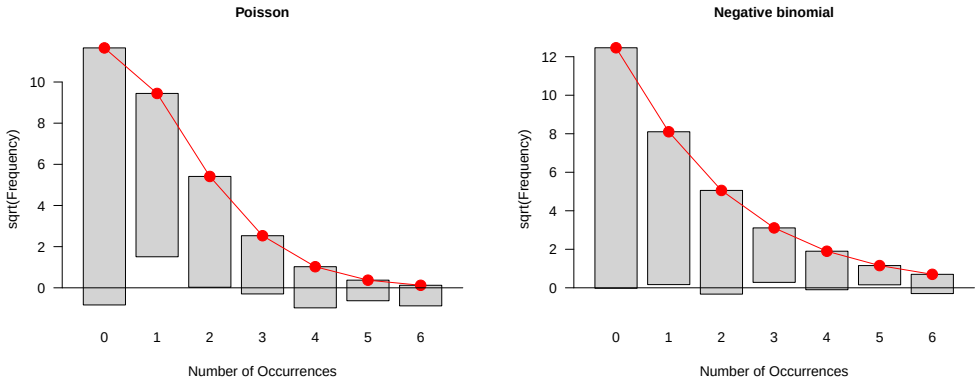

Figure 3.19: Hanging rootograms for the Federalist Papers data, comparing the Poisson and negative binomial models. fig:Fed0-Fed1

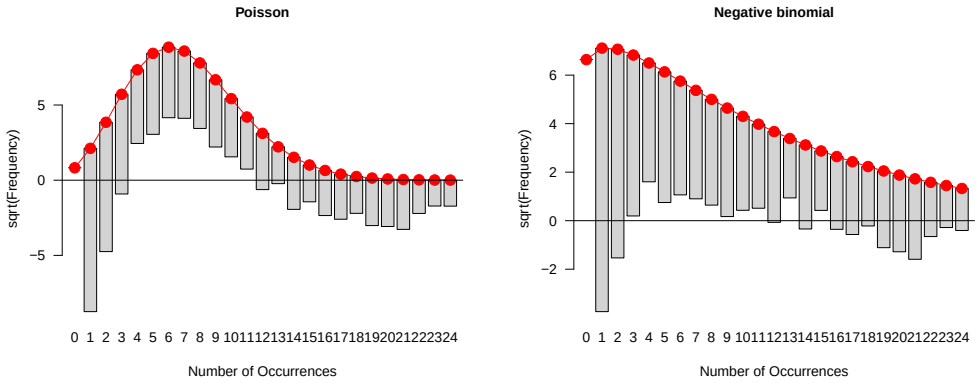

Figure 3.20: Hanging rootograms for the Butterfly data, comparing the Poisson and negative binomial models. The lack of fit for both is readily apparent. fig:But-Fit

3.4 Diagnosing discrete distributions: Ord plots

{sec:discrete-ord}

Ideally, the general form chosen for a discrete distribution should be dictated by substantive knowledge of a plausible mechanism for generating the data. When such knowledge is lacking, however, we may not know which distribution is most appropriate for some particular set of data. In these cases, the question is often turned around, so that we seek a distribution that fits well, and then try to understand the mechanism in terms of aspects of the underlying probability theory (independent trials, rare events, waiting-time to an occurrence, and so forth).

Although it is possible to fit each of several possibilities, the summary goodness-of-fit statistics can easily be influenced by one or two disparate cells, or additional (ignored or unknown) factors. One simple alternative is a plot suggested by Ord (1967) which may be used to diagnose the form of the discrete distribution.

Ord showed that a linear relationship of the form:

{eq:ord}

$$\frac{k\,p(k)}{p(k-1)} \equiv \frac{k\,n_k}{n_{k-1}} = a + b\,k \tag{3.12}$$

holds for each of the Poisson, binomial, negative binomial, and logarithmic series distributions, and these distributions are distinguished by the signs of the intercept, a, and slope, b, as shown in Table 3.11.

Table 3.11: Diagnostic slope and intercept for four discrete distributions. The ratios kn_k/n_{k-1} plotted against k should appear as a straight line, whose slope and intercept determine the particular distribution.

{tab:ordparm}

Slope (b)	Intercept (a)	Distribution (parameter)	Parameter estimate
0	+	Poisson (λ)	$\lambda = a$
−	+	Binomial (n, p)	$p = b/(b-1)$
+	+	Negative binomial (n, p)	$p = 1 - b$
+	−	Log. series (θ)	$\theta = b$
			$\theta = -a$

The slope, b, in Eqn. (3.12) is zero for the Poisson, negative for the binomial, and positive for the negative binomial and logarithmic series distributions; the latter two are distinguished by their intercepts. In practical applications of this idea, the details are important: how to fit the line, and how to determine if the pattern of signs are sufficient to reasonably provide a diagnosis of the distribution type.

One difficulty in applying this technique is that the number of points (distinct values of k) in the Ord plot is often small, and the sampling variances of $k\,n_k/n_{k-1}$ can vary enormously. A little reflection indicates that points where n_k is small should be given less weight in determining the slope of the line (and hence determining the form of the distribution). In applications it has been found that using a weighted least squares fit of $k\,n_k/n_{k-1}$ on k, using weights of $w_k = \sqrt{n_k - 1}$ produces reasonably good automatic diagnosis of the form of a probability distribution. Moreover, to judge whether a coefficient is positive or negative, a small tolerance is used; if none of the distributions can be classified, no parameters are estimated. Caution is advised in accepting the conclusion, because it is based on these simple heuristics.

In the vcd package this method is implemented in the `Ord_plot()` function. The essential ideas are illustrated using the *`Butterfly`* data below, which produces Figure 3.21. Note that the function returns (invisibly) the values of the intercept and slope in the weighted LS regression.

```
> ord <- Ord_plot(Butterfly,
+                 main = "Butterfly species collected in Malaya",
+                 gp = gpar(cex = 1), pch = 16)
> ord

Intercept     Slope
 -0.70896   1.06082
```

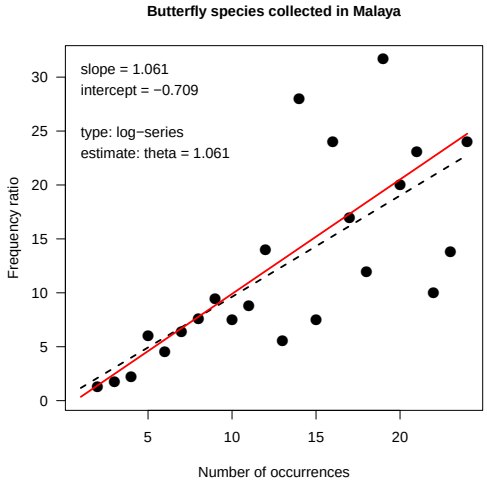

Figure 3.21: Ord plot for the Butterfly data. The slope and intercept in the plot correctly diagnoses the log-series distribution. fig:ordplot1

In this plot, the black line shows the usual OLS regression fit of frequency, n_k on number of occurrences, k; the red line shows the weighted least squares fit, using weights of $\sqrt{n_k - 1}$. In this case, the two lines are fairly close together, as regards their intercepts and slopes. The positive slope and negative intercept diagnoses this as a log-series distribution.

In other cases, the number of distinct points (values of k) is small, and the sampling variances of the ratios $k\, n_k / n_{k-1}$ can vary enormously. The following examples illustrate some other distributions and some of the details of the heuristics.

3.4.0.1 Ord plot examples

{ex:horsekick3}

EXAMPLE 3.17: Death by horse kick

The results below show the calculations for the horse kicks data, with the frequency ratio $k\, n_k / n_{k-1}$ labeled `y`.

```
> data(HorseKicks, package = "vcd")
> nk <- as.vector(HorseKicks)
> k <- as.numeric(names(HorseKicks))
> nk1 <- c(NA, nk[-length(nk)])
> y <- k * nk / nk1
> weight <- sqrt(pmax(nk, 1) - 1)
> (ord_df <- data.frame(k, nk, nk1, y, weight))

  k  nk nk1       y  weight
1 0 109  NA      NA 10.3923
2 1  65 109 0.59633  8.0000
```

```
3 2  22  65 0.67692  4.5826
4 3   3  22 0.40909  1.4142
5 4   1   3 1.33333  0.0000

> coef(lm(y ~ k, weights = weight, data = ord_df))

(Intercept)           k
   0.656016   -0.034141
```

The weighted least squares line, with weights w_k, has a slope (-0.03) close to zero, indicating the Poisson distribution.[14] The estimate $\lambda = a = .656$ compares favorably with the MLE, $\lambda = 0.610$ and the value from the Poissonness plot, shown in the following section. The call to `Ord_plot()` below produces Figure 3.22.

```
> Ord_plot(HorseKicks,
+          main = "Death by horse kicks", gp = gpar(cex = 1), pch = 16)
```

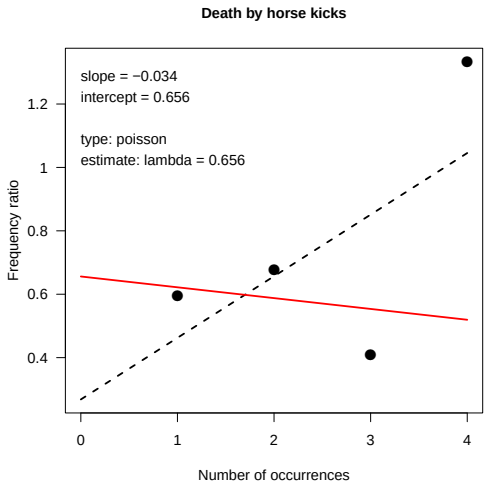

Figure 3.22: Ord plot for the HorseKicks data. The plot correctly diagnoses the Poisson distribution. `fig:ordplot2`

△

{ex:madison3}

EXAMPLE 3.18: Federalist papers

Figure **??** (left) shows the Ord plot for the *Federalist* data. The slope is positive, so either the negative binomial or log series are possible, according to Table 3.11. The intercept is essentially zero, which is ambiguous. However, the logarithmic series requires $b \approx -a$, so the negative binomial is a better choice. Mosteller and Wallace (1963, 1984) did in fact find a reasonably good fit to this distribution. Note that there is one apparent outlier, at $k = 6$, whose effect on the OLS line is to increase the slope and decrease the intercept. △

[14] The heuristic adopted in `Ord_plot()` uses a tolerance of 0.1 to decide if a coefficient is negative, zero, or positive.

```
> Ord_plot(Federalist, main = "Instances of 'may' in Federalist papers",
+          gp = gpar(cex = 1), pch = 16)
```

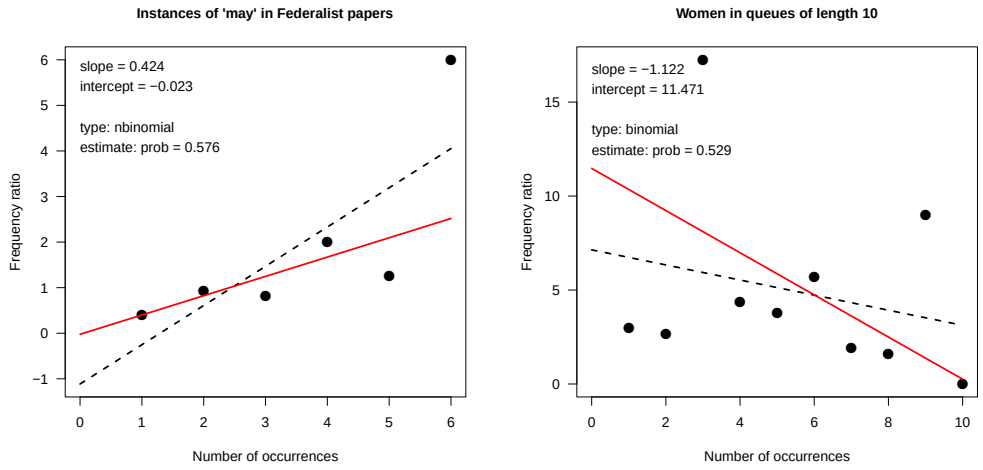

Figure 3.23: Ord plots for the Federalist (left) and WomenQueue (right) data sets. fig:ordplot3plot

{ex:queues}

EXAMPLE 3.19: Women in queues

Jinkinson and Slater (1981), Hoaglin and Tukey (1985) give the frequency distribution of the number of females observed in 100 queues of length 10 in a London Underground station, recorded in the data set *WomenQueue* in vcd.

```
> data(WomenQueue, package = "vcd")
> WomenQueue

nWomen
 0  1  2  3  4  5  6  7  8  9 10
 1  3  4 23 25 19 18  5  1  1  0
```

If it is assumed that people line up independently, and that men and women are equally likely to be found in a queue (not necessarily reasonable assumptions), then the number of women out of 10 would have a (symmetric) binomial distribution with parameters $n = 10$ and $p = \frac{1}{2}$. However, there is no real reason to expect that males and females are equally likely to be found in queues in the London underground, so we may be interested in estimating p from the data and determining if a binomial distribution fits.

```
> Ord_plot(WomenQueue, main = "Women in queues of length 10",
+          gp = gpar(cex = 1), pch = 16)
```

Figure **??** (right) shows the Ord plot for these data. The negative slope and positive intercept clearly diagnose this distribution as binomial. The rough estimate of $\hat{p} = b/(1-b) = 0.53$ indicates that women are slightly more prevalent than men in these data for the London underground. △

3.4.0.2 Limitations of Ord plots

Using a single simple diagnostic plot to determine one of four common discrete distributions is advantageous, but your enthusiasm should be dampened by several weaknesses:

- The Ord plot lacks resistance, since a single discrepant frequency affects the points n_k/n_{k-1} for both k and $k+1$.
- The sampling variance of $k\, n_k/n_{k-1}$ fluctuates widely (Hoaglin and Tukey, 1985, Jinkinson and Slater, 1981). The use of weights w_k helps, but is purely a heuristic device. The `Ord_plot()` function explicitly shows both the OLS line and the WLS line, which provides some indication of the effect of the points on the estimation of slope and intercept.

3.5 Poissonness plots and generalized distribution plots

{crete-Poissonness}

The ***Poissonness plot*** (Hoaglin, 1980) is a robust plot to sensitively determine how well a one-way table of frequencies follows a Poisson distribution. It plots a quantity called a count metameter against k, designed so that the result will be points along a straight line when the data follow a Poisson distribution. When the data deviate from a Poisson, the points will be curved. Hoaglin and Tukey (1985) develop similar plots for other discrete distributions, including the binomial, negative binomial, and logarithmic series distributions. We first describe the features and construction of these plots for the Poisson distribution and then (Section 3.5.4) the extension to other distributions.

3.5.1 Features of the Poissonness plot

The Poissonness plot has the following desirable features:

- ***Resistance***: a single discrepant value of n_k affects only the point at value k. (In the Ord plot it affects each of its neighbors.)
- ***Comparison standard***: An approximate confidence interval can be found for each point, indicating its inherent variability and helping to judge whether each point is discrepant.
- ***Influence***: Extensions of the method result in plots which show the effect of each point on the estimate of the main parameter of the distribution (λ in the Poisson).

3.5.2 Plot construction

Assume, for some fixed λ, each observed frequency, n_k equals the expected frequency, $m_k = Np_k$. Then, setting $n_k = Np_k = Ne^{-\lambda}\, \lambda^k/k!$, and taking logs of both sides gives

$$\log(n_k) = \log\, N - \lambda + k\, \log\, \lambda - \log\, k!\ .$$

This can be rearranged to a linear equation in k,

{eq:poispl}

$$\phi\,(n_k) \equiv \log\left(\frac{k!\, n_k}{N}\right) = -\lambda + (\log\, \lambda)\, k\ . \tag{3.13}$$

The left side of Eqn. (3.13) is called the ***count metameter***, and denoted $\phi\,(n_k)$. Hence, plotting $\phi(n_k)$ against k should give a straight line of the form $\phi(n_k) = a + bk$ with

- slope = $\log\, \lambda$
- intercept = $-\lambda$

when the observed frequencies follow a Poisson distribution. If the points in this plot are close enough to a straight line, then an estimate of λ may be obtained from the slope b of the line, $\hat{\lambda} = e^b$ should be reasonably close in value to the MLE of λ, $\hat{\lambda} = \bar{x}$. In this case, we might as well use the MLE as our estimate.

3.5.2.1 Leveled plot

If we have a preliminary estimate λ_0 of λ, we can use this to give a new plot where the reference line is horizontal, making comparison of the points with the line easier. In this leveled plot the vertical coordinate $\phi(n_k)$ is modified to

$$\phi'(n_k) = \phi(n_k) + \lambda_0 - k \log \lambda_0 \ . \tag{3.14}$$ {eq:pois-leveled}

When the data follow a Poisson distribution with parameter λ, the modified plot will have

- slope = $\log \lambda - \log \lambda_0 = \log(\lambda / \lambda_0)$
- intercept = $\lambda_0 - \lambda$

In the ideal case, where our estimate of λ_0 is close to the true λ, the line will be approximately horizontal at $\phi(n_k)' = 0$. The modified plot is particularly useful in conjunction with the confidence intervals for individual points described below.

3.5.2.2 Confidence intervals

The goal of the Poissonness plot is to determine whether the points are "sufficiently linear" to conclude that the Poisson distribution is adequate for the data. Confidence intervals for the points can help you decide, and also show the relative precision of the points in these plots.

For example, when one or two points deviate from an otherwise nearly linear relation, it is helpful to determine whether the discrepancy is consistent with chance variation. As well, we must recognize that classes with small frequencies n_k are less precise than classes with large frequencies.

Hoaglin and Tukey (1985) develop approximate confidence intervals for $\log(m_k)$ for each point in the Poissonness plot. These are calculated as

$$\phi\left(n_k^*\right) \pm h_k \tag{3.15}$$ {eq:poisCI}

where the count metameter function is calculated using a modified frequency n_k^*, defined as

$$n_k^* = \begin{cases} n_k - .8 n_k - .67 & n \geq 2 \\ 1/e & n = 1 \\ \text{undefined} & n = 0 \end{cases}$$

and h_k is the half-width of the 95% confidence interval,

$$h_k = 1.96 \frac{\sqrt{1 - \widehat{p}_k}}{[n_k - (.25 \widehat{p}_k + .47) \sqrt{n_k}]^{1/2}}$$

and $\hat{p}_k = n_k / N$.

3.5.3 The `distplot()` function

Poissonness plots (and versions for other distributions) are produced by the function `distplot()` in vcd. As with `Ord_plot()`, the first argument is either a vector of counts, a one-way table of frequencies of counts or a data frame or matrix with frequencies in the first column and the corresponding counts in the second column. Nearly all of the examples in this chapter use one-way tables of counts.

The `type` argument specifies the type of distribution. For `type = "poisson"`, specifying a value for `lambda` $= \lambda_0$ gives the leveled version of the plot. {ex:horsekick4}

EXAMPLE 3.20: Death by horse kick

The calculations for the Poissonness plot, including confidence intervals, are shown below for the *HorseKicks* data. The call to `distplot()` produces the plot in the left panel of Figure 3.24.

```
> data("HorseKicks", package="vcd")
> dp <- distplot(HorseKicks, type = "poisson",
+   xlab="Number of deaths", main="Poissonness plot: HorseKicks data")
> print(dp, digits=4)

  Counts Freq Metameter CI.center CI.width CI.lower CI.upper
1      0  109    -0.607   -0.6131   0.1305  -0.7436  -0.4827
2      1   65    -1.124   -1.1343   0.2069  -1.3412  -0.9274
3      2   22    -1.514   -1.5451   0.4169  -1.9620  -1.1281
4      3    3    -2.408   -2.6607   1.3176  -3.9783  -1.3431
5      4    1    -2.120   -3.1203   2.6887  -5.8089  -0.4316
```

In this plot, the open circles show the calculated observed values of the count `Metameter` = $\phi(n_k)$. The smaller filled points show the centers of the confidence intervals, `CI.center` = $\phi(n_k^*)$ (Eqn. (3.15)), and the dashed lines show the extent of the confidence intervals.

The fitted least squares line has a slope of -0.431, which would indicate $\lambda = e^{-0.431} = 0.65$. This compares well with the MLE, $\lambda = \bar{x} = 0.61$.

Using `lambda = 0.61` as below gives the leveled version shown in the right panel of Figure 3.24.

```
> # leveled version, specifying lambda
> distplot(HorseKicks, type = "poisson", lambda = 0.61,
+   xlab="Number of deaths", main="Leveled Poissonness plot")
```

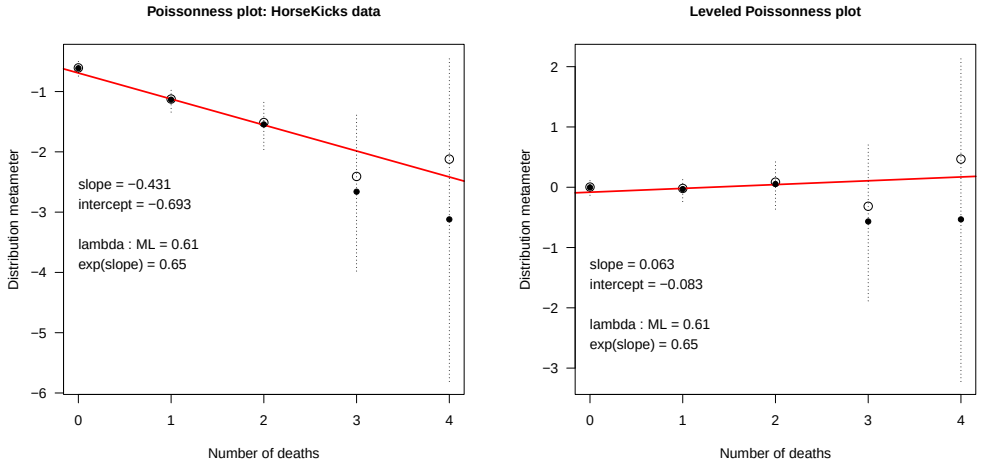

Figure 3.24: Poissonness plots for the HorseKick data. Left: standard plot; right: leveled plot. `fig:distplot1`

TODO: DM: In the leveled plot, the label for the slope is actually wrong, should be exp(slope + log λ) = 0.65

In both plots the fitted line is within the confidence intervals, indicating the adequacy of the Poisson model for these data. The widths of the intervals for $k > 2$ are graphic reminders that these observations have decreasingly low precision where the counts n_k are small.

△

sec:discrete-other}

3.5.4 Plots for other distributions

As described in Section 3.2.6, the binomial, Poisson, negative binomial, geometric, and logseries distributions are all members of the general power series family of discrete distributions. For this family, Hoaglin and Tukey (1985) develop similar plots of a count metameter against k which appear as a straight line when a data distribution follows a given family member.

The distributions which can be analyzed in this way are shown in Table 3.12, with the interpretation given to the slope and intercept in each case. For example, for the Binomial distribution, a "binomialness" plot is constructed by plotting $\log n_k^* / N\binom{n}{k}$ against k. If the points in this plot approximate a straight line, the slope is interpreted as $\log(p/(1-p))$, so the binomial parameter p may be estimated as $p = e^b/(1+e^b)$.

Table 3.12: Plot parameters for five discrete distributions. In each case the count metameter, $\phi(n_k^*)$ is plotted against k, yielding a straight line when the data follow the given distribution. {tab:distparms}

Distributiion	Probability function, $p(k)$	Count metameter, $\phi(n_k^*)$	Theoretical Slope (b)	Theoretical Intercept (a)
Poisson	$e^{-\lambda}\lambda^k/k!$	$\log(k! n_k^*/N)$	$\log(\lambda)$	-λ
Binomial	$\binom{n}{k} p^k (1-p)^{n-k}$	$\log\left(n_k^*/N\binom{n}{k}\right)$	$\log\left(\frac{p}{1-p}\right)$	$n\log(1-p)$
Negative binomial	$\binom{n+k-1}{k} p^n (1-p)^k$	$\log\left(n_k^*/N\binom{n+k-1}{k}\right)$	$\log(1-p)$	$n\log(p)$
Geometric	$p(1-p)^k$	$\log\left(n_k^*/N\right)$	$\log(1-p)$	$\log(p)$
Logarithmic series	$\theta^k/[-k\log(1-\theta)]$	$\log\left(k n_k^*/N\right)$	$\log(\theta)$	$-\log\left(-\log(1-\theta)\right)$

Source: adapted from Hoaglin and Tukey (1985), Table 9-15.

Unlike the Ord plot, a different plot is required for each distribution, because the count metameter, $\phi(n_k)$, differs from distribution to distribution. Moreover, systematic deviation from a linear relationship does not indicate which distribution provides a better fit. However, the attention to robustness, and the availability of confidence intervals and influence diagnostics make this a highly useful tool for visualizing discrete distributions. {ex:saxony-distplot}

EXAMPLE 3.21: Families in Saxony

Our analysis in Example 3.2 and Example 3.13 of the `Saxony` data showed that the distribution of male children had slightly heavier tails than the binomial, meaning the observed distribution is overdispersed. We can see this in the `goodfit()` plot shown in Figure 3.25 (left), and even more clearly in the distribution diagnostic plot produced by `distplot()` in the right panel of Figure 3.25. For a binomial distribution, we call this distribution plot a "binomialness plot".

```
> plot(goodfit(Saxony, type="binomial", par=list(size=12)))
> distplot(Saxony, type = "binomial", size = 12,
+   xlab="Number of males")
```

The weight of evidence is thus that, as simple as the binomial might be, it is inadequate to fully explain the distribution of sex ratios in this large sample of families of 12 children. To understand this data better, it is necessary to question the assumptions of the binomial (births of males are

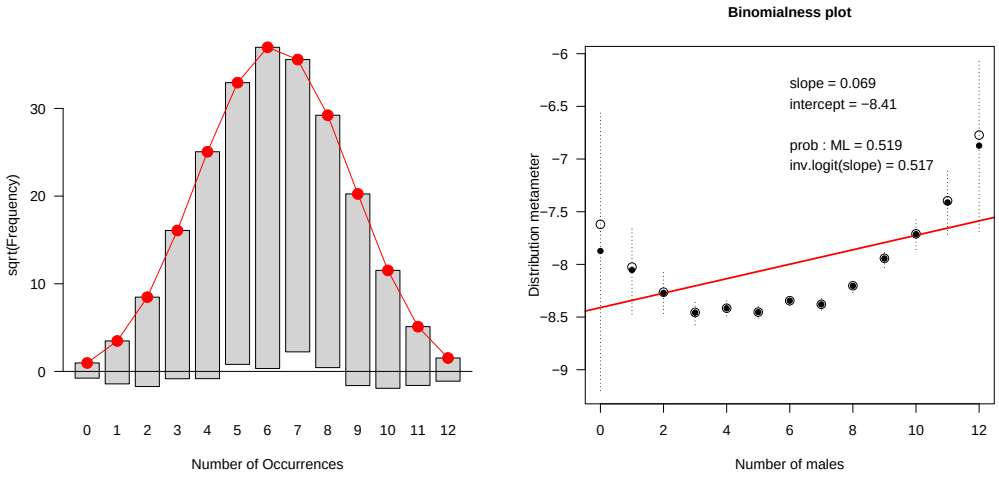

Figure 3.25: Diagnostic plots for males in Saxony families. Left: `goodfit()` plot; right: `distplot()` plot. Both plots show heavier tails than in a binomial distribution. `fig:distplots`

independent Bernoulli trials with constant probability p) as a model for this birth distribution and/or find a more adequate model.[15] △

ederalist-distplot}

EXAMPLE 3.22: Federalist papers

In Example 3.16 we carried out GOF tests for the Poisson and negative binomial models with the Federalist papers data; Figure 3.19 showed the corresponding rootogram plots. Figure 3.26 compares these two using the diagnostic plots of this section. Again the Poisson shows systematic departure from the linear relation required in the Poissonness plot, while the negative binomial model provides an acceptable fit to these data.

```
> distplot(Federalist, type = "poisson", xlab="Occurrences of 'may'")
> distplot(Federalist, type = "nbinomial", xlab="Occurrences of 'may'")
```

△

TODO: DM: The following section assumes knowledge of GLMs, introduced in a later chapter. Either remove chapter, or move to the end after the GLM chapter.

3.6 Fitting discrete distributions as generalized linear models

{sec:fitglm}

In Section 3.2.6, we described how the common discrete distributions are all members of the general power series family. This provides the basis for the generalized distribution plots described in Section 3.5.4. Another general family of distributions—the ***exponential family***—includes most of the common continuous distributions: the normal, gamma, exponential, and others, and is the basis of the class of generalized linear models (GLMs) fit by `glm()`.

[15] On these questions, Edwards (1958) reviews numerous other studies of these Geissler's data, and fits a so-called β-***binomial*** model proposed by Skellam (1948), where p varies among families according to a β distribution. He concludes that there is evidence that p varies between families of the same size. One suggested explanation is that family decisions to have a further child is influenced by the balance of boys and girls among their earlier children.

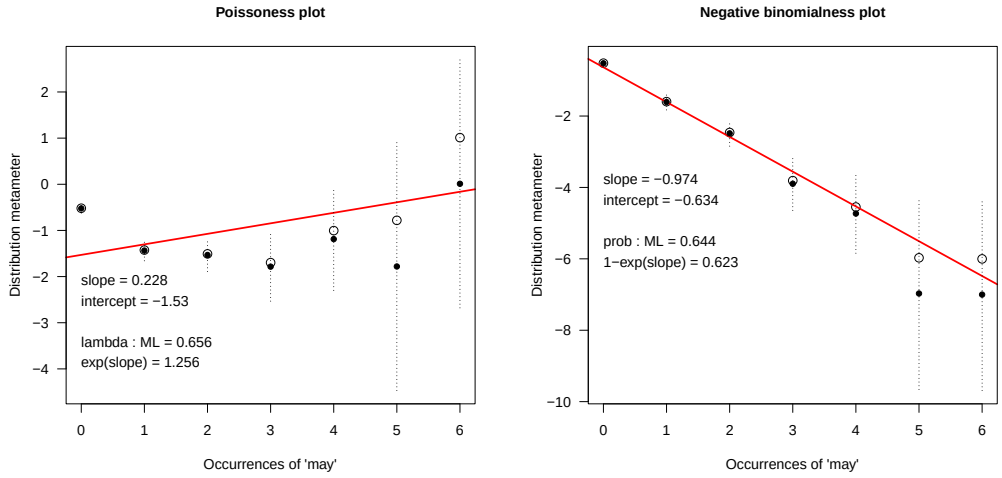

Figure 3.26: Diagnostic plots for the Federalist papers data. Left: Poissonness plot; right: negative binomialness plot. `fig:distplot5`

A clever approach by Lindsey and Mersch (1992), Lindsey (1995, §6.1) shows how various discrete (and continuous) distributions can be fit to frequency data using generalized linear models for log frequency (which are equivalent to Poisson loglinear models). The uniform, geometric, binomial, and the Poisson distributions may all be fit easily in this way, but the idea extends to some other distributions, such as the ***double binomial*** distribution, that allows a separate parameter for overdispersion relative to the binomial. A clear advantage is that this method gives estimated standard errors for the distribution parameters as well as estimated confidence intervals for fitted probabilities.

The essential idea is that, for frequency data, any distribution in the exponential family may be represented by a linear model for the logarithm of the cell frequency, with a Poisson distribution for errors, otherwise known as a "Poisson loglinear regression model". These have the form

$$\log(N\pi_k) = \text{ offset } + \beta_0 + \boldsymbol{\beta}^{\mathsf{T}} \boldsymbol{S}(k) \ ,$$

where N is the total frequency, π_k is the modeled probability of count k, $\boldsymbol{S}(k)$ is a vector of zero or more sufficient statistics for the canonical parameters of the exponential family distribution, and the offset term is a value which does not depend on the parameters.

Table 3.13 shows the sufficient statistics and offsets for several discrete distributions. See Lindsey and Mersch (1992) for further details, and definitions for the double-binomial distribution,[16] and Lindsey (1995, pp. 130–133) for his analysis of the `Saxony` data using this distribution. Lindsey and Altham (1998) provide an analysis of the complete Geissler data (provided in the data set `Geissler` in vcdExtra) using several different models to handle overdispersion. `{ex:saxony2}`

EXAMPLE 3.23: Families in Saxony

The binomial distribution and the double binomial can both be fit to frequency data as a Poisson regression via `glm()` using $\log \binom{n}{k}$ as an offset. First, we convert `Saxony` into a numeric data frame for use with `glm()`.

[16] In R, the double binomial distribution is implemented in the rmutil package, providing the standard complement of density function (`ddoublebinom()`), CDF (`pdoublebinom()`), quantiles (`qdoublebinom()`) and random generation (`rdoublebinom()`).

Table 3.13: Poisson loglinear representations for some discrete distributions {tab:expfamily}

Distribution	Sufficient statistics	Offset
Geometric	k	
Poisson	k	$-\log(k!)$
Binomial	k	$\log \binom{n}{k}$
Double binomial	$k, k\log(k) + (n-k)\log(n-k)$	$\log \binom{n}{k}$

```
> data(Saxony, package="vcd")
> Males <- as.numeric(names(Saxony))
> Families <- as.vector(Saxony)
> Sax.df <- data.frame(Males, Families)
```

To calculate the offset for `glm()` in R, note that `choose(12,0:12)` returns the binomial coefficients, and `lchoose(12,0:12)` returns their logs.

```
> # fit binomial (12, p) as a glm
> Sax.bin <- glm(Families ~ Males, offset=lchoose(12,0:12),
+                family=poisson, data=Sax.df)
>
> # brief model summaries
> LRstats(Sax.bin)

Likelihood summary table:
        AIC BIC LR Chisq Df Pr(>Chisq)
Sax.bin 191 192       97 11      7e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> coef(Sax.bin)

(Intercept)       Males
  -0.069522    0.076898
```

As we have seen, this model fits badly. The parameter estimate for `Males`, $\beta_1 = 0.0769$ is actually estimating the logit of p, $\log p/(1-p)$, so the inverse transformation gives $\hat{p} = \frac{\exp(\beta_1)}{1+\exp(\beta_1)} = 0.5192$, as we had before.

The double binomial model can be fitted as follows. The term `YlogitY` calculates $k\log(k) + (n-k)\log(n-k)$, the second sufficient statistic for the double binomial (see Table 3.13) fitted via `glm()`.

```
> # double binomial, (12, p, psi)
> Sax.df$YlogitY <-
+   Males      * log(ifelse(Males==0, 1, Males)) +
+         (12-Males) * log(ifelse(12-Males==0, 1, 12-Males))
>
> Sax.dbin <- glm(Families ~ Males + YlogitY, offset=lchoose(12,0:12),
+         family=poisson, data=Sax.df)
> coef(Sax.dbin)

(Intercept)       Males     YlogitY
  -3.096918    0.065977    0.140205

> LRstats(Sax.bin, Sax.dbin)
```

```
Likelihood summary table:
         AIC BIC LR Chisq Df Pr(>Chisq)
Sax.bin  191 192     97.0 11      7e-16 ***
Sax.dbin 109 111     13.1 10       0.22
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the above, we can see that the double binomial model `Sax.dbin` with one more parameter is significantly better than the simple binomial and represents an adequate fit to the data. The table below displays the fitted values and standardized residuals for both models.

```
> results <- data.frame(Sax.df,
+           fit.bin=fitted(Sax.bin), res.bin=rstandard(Sax.bin),
+           fit.dbin=fitted(Sax.dbin), res.dbin=rstandard(Sax.dbin))
> print(results, digits=2)

   Males Families YlogitY fit.bin res.bin fit.dbin res.dbin
1      0        3      30    0.93    1.70      3.0    0.026
2      1       24      26   12.09    3.05     23.4    0.136
3      2      104      24   71.80    3.71    104.3   -0.036
4      3      286      23  258.48    1.87    307.8   -1.492
5      4      670      22  628.06    1.94    652.9    0.778
6      5     1033      22 1085.21   -1.87   1038.5   -0.202
7      6     1343      22 1367.28   -0.75   1264.2    2.635
8      7     1112      22 1265.63   -5.09   1185.0   -2.550
9      8      829      22  854.25   -1.03    850.1   -0.846
10     9      478      23  410.01    3.75    457.2    1.144
11    10      181      24  132.84    4.23    176.8    0.371
12    11       45      26   26.08    3.42     45.2   -0.039
13    12        7      30    2.35    2.45      6.5    0.192
```

Finally, Figure 3.27 shows the rootogram for the double binomial, which can be compared with that for the binomial model shown in Figure 3.25. We can see that the fit is now quite good, particularly in the tails. The positive coefficient for the term `YlogitY` gives additional weight in the tails.

```
> with(results, vcd::rootogram(Families, fit.dbin,
+                          xlab="Number of males"))
```

△

3.6.1 Covariates, overdispersion and excess zeros

All of the examples in this chapter are somewhat special, in that in each case the data consist only of a one-way frequency distribution of a basic count variable. In more general and realistic settings, there may also be one or more explanatory variables or ***covariate***s that influence the frequency distributions of the counts. For example, in the *Saxony* data, the number of boys in families of size 12 was aggregated over the years 1876–1885, and it is possible that any deviation from a binomial distribution could be due to variation over time or unmeasured predictors (e.g., rural vs. urban, age of parents).

This is where the generalized linear model approach introduced here (treated in detail in Chapter 9), begins to shine— because it allows such covariates to be taken into account, and then questions regarding the *form* of the distribution pertain only to the variation of the frequencies not fitted by the model. The next example illustrates what can go wrong when important predictors are omitted from the analysis.

{ex:phdpubs0}

EXAMPLE 3.24: Publications of PhD candidates

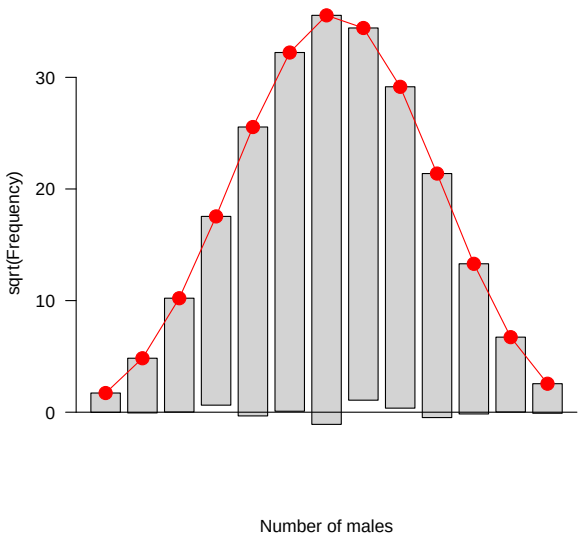

Figure 3.27: Rootogram for the double binomial model for the Saxony data. This now fits well in the tails of the distribution. fig:sax-glm5

Long (1990, 1997) gave data on the number of publications by 915 doctoral candidates in biochemistry in the last three years of their PhD studies, contained in the data set *PhdPubs* in vcdExtra. The data set also includes information on gender, marital status, number of young children, prestige of the doctoral department and number of publications by the student's mentor. The frequency distribution of number of publications by these students is shown below.

```
> data("PhdPubs", package="vcdExtra")
> table(PhdPubs$articles)


  0   1   2   3   4   5   6   7   8   9  10  11  12  16  19
275 246 178  84  67  27  17  12   1   2   1   1   2   1   1
```

The naive approach, ignoring the potential predictors is just to try fitting various probability models to this one-way distribution. Rootograms for the simpler Poisson distribution and the negative binomial that allows for overdispersion are shown in Figure 3.28.

```
> library(vcd)
> plot(goodfit(PhdPubs$articles), xlab="Number of Articles",
+      main="Poisson")
> plot(goodfit(PhdPubs$articles, type="nbinomial"), xlab="Number of Articles",
+      main="Negative binomial")
```

From these plots it is clear that the Poisson distribution doesn't fit well at all, because there is a large excess of zero counts— candidates with no publications, and most of the counts of four or more publications are larger than the Poisson model predicts. The fit of the negative binomial model in the right panel of Figure 3.28 looks much better, except that for eight or more publications, there is a systematic tendency of overfitting for 8–10 and underfittting for the observed counts of 12 or more. This lack of fit is confirmed by the formal test.

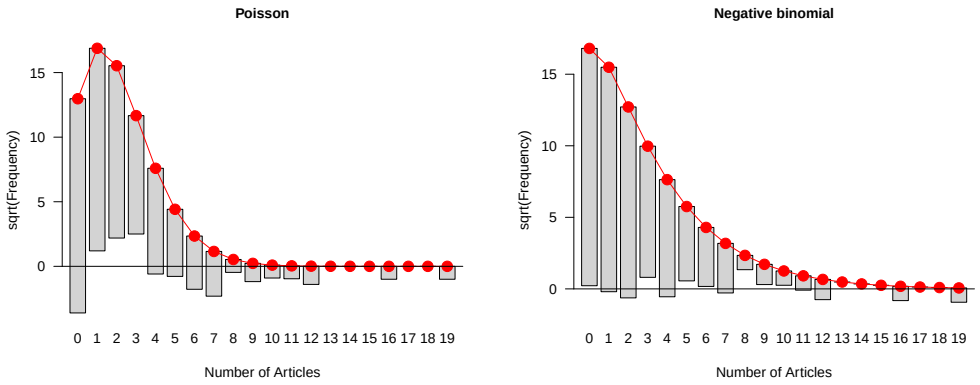

Figure 3.28: Hanging rootograms for publications by PhD candidates, comparing the Poisson and negative binomial models. The Poisson model clearly does not fit. The the negative binomial is better, but still has significant lack of fit. fig:phdpubs-rootogram

```
> summary(goodfit(PhdPubs$articles, type="nbinomial"))


 Goodness-of-fit test for nbinomial distribution

                     X^2 df  P(> X^2)
Likelihood Ratio 31.098 12 0.0019033
```

The difficulty with this simple analysis is not only that it ignores the possible predictors of publishing by these PhD candidates, but also, by doing so, it prevents a better, more nuanced explanation of the phenomenon under study. This example is re-visited in Chapter 9, Example 9.1, where we consider generalized linear models taking potential predictors into account, as well as extended *zero-inflated* models allowing special consideration of zero counts. △

3.7 Chapter summary

{sec:ch03-summary}

- Discrete distributions typically involve basic *counts* of occurrences of some event occurring with varying *frequency*. The ideas and methods for one-way tables described in this chapter are building blocks for the analysis of more complex data.

- The most commonly used discrete distributions include the binomial, Poisson, negative binomial, geometric, and logarithmic series distributions. Happily, these are all members of a family called the power series distributions. Methods of fitting an observed data set to any of these distributions are described, and implemented in the `goodfit()` function.

- After fitting an observed distribution it is useful to plot the observed and fitted frequencies. Several ways of making these plots are described, and implemented in the `rootogram()` function.

- A heuristic graphical method for identifying which discrete distribution is most appropriate for a given set of data involves plotting ratios kn_k/n_{k-1} against k. These plots are constructed by the function `Ord_plot()`.

- A more robust plot for a Poisson distribution involves plotting a count metameter, $\phi(n_k)$ against

k, which gives a straight line (whose slope estimates the Poisson parameter) if the data follow a Poisson distribution. This plot provides robust confidence intervals for individual points and provides a means to assess the influence of individual points on the Poisson parameter. These plots are provided by the function `distplot()`.

- The ideas behind the Poissonness plot can be applied to the other discrete distributions.

3.8 Lab exercises

{lab:3.1}
{sec:ch03-labs}

Exercise 3.1 The *Arbuthnot* data in HistData (Example 3.1) also contains the variable `Ratio`, giving the ratio of male to female births.

(a) Make a plot of `Ratio` over `Year`, similar to Figure 3.1. What features stand out? Which plot do you prefer to display the tendency for more male births?
(b) Plot the total number of christenings, `Males + Females` or `Total` (in 000s) over time. What unusual features do you see?

{lab:3.2}

Exercise 3.2 Use the graphical methods illustrated in Section 3.2 to plot a collection of geometric distributions for $p = 0.2, 0.4, 0.6, 0.8$, over a range of values of $k = 0, 1, \ldots 10$.

(a) With `xyplot()`, try the different plot formats using points connected with lines, as in Figure 3.11, or using points and lines down to the origin, as in the panels of Figure 3.13.
(b) Also with `xyplot()`, produce one version of a multi-line plot in a single panel that you think shows well how these distributions change with the probability p of success.
(c) Do the same in a multi-panel version, conditional on p.

{lab:3.3}

Exercise 3.3 Use the data set *WomenQueue* to:

(a) produce plots analogous to those shown in Section 3.1 (some sort of bar graph of frequencies)
(b) check for goodness-of-fit to the binomial distribution using the `goodfit()` methods described in Section 3.3.2.

{lab:3.4}

Exercise 3.4 Continue Example 3.13 on the distribution of male children in families in Saxony by fitting a binomial distribution, $\text{Bin}(n = 12, p = \frac{1}{2})$, specifying equal probability for boys and girls. [Hint: you need to specify both `size` and `prob` values for `goodfit()`.]

(a) Carry out the GOF test for this fixed binomial distribution. What is the ratio of χ^2/df? What do you conclude?
(b) Test the additional lack of fit for the model $\text{Bin}(n = 12, p = \frac{1}{2})$ compared to the model $\text{Bin}(n = 12, p = \hat{p})$ where $\hat{p}$ is estimated from the data.
(c) Use the `plot.gootfit()` method to visualize these two models.

{lab:3.5}

Exercise 3.5 For the *Federalist* data, the examples in Section 3.3.1 and Section 3.3.2 showed the negative binomial to provide an acceptable fit. Compare this with the simpler special case of geometric distribution, corresponding to $n = 1$.

(a) Use `goodfit()` to fit the geometric distribution. [Hint: use `type="nbinomial"`, but specify `size=1` as a parameter.]
(b) Compare the negative binomial and the geometric models statistically, by a likelihood-ratio test of the difference between these two models.
(c) Compare the negative binomial and the geometric models visually by hanging rootograms or other methods.

{lab:3.6}

Exercise 3.6 Mosteller and Wallace (1963, Table 2.4) give the frequencies, n_k of counts $k = 0, 1, \ldots$ of other selected marker words in 247 blocks of text known to have been written by Alexander Hamilton. The data below show the occurrences of the word *upon*, that Hamilton used much more than did James Madison.

```
> count <- 0 : 5
> Freq <- c(129, 83, 20, 9, 5, 1)
```

(a) Read these data into R and construct a one-way table of frequencies of counts or a matrix or data frame with frequencies in the first column and the corresponding counts in the second column, suitable for use with `goodfit()`.
(b) Fit and plot the Poisson model for these frequencies.
(c) Fit and plot the negative binomial model for these frequencies.
(d) What do you conclude?

{lab:3.7}

Exercise 3.7 The data frame *Geissler* in the vcdExtra package contains the complete data from Geissler's (1889) tabulation of family sex composition in Saxony. The table below gives the number of boys in families of size 11.

boys	0	1	2	3	4	5	6	7	8	9	10	11
Freq	8	72	275	837	1,540	2,161	2,310	1,801	1,077	492	93	24

(a) Read these data into R
(b) Following Example 3.13, use `goodfit()` to fit the binomial model and plot the results. Is there an indication that the binomial does not fit these data?
(c) Diagnose the form of the distribution using the methods described in Section 3.4.
(d) Try fitting the negative binomial distribution, and use `distplot()` to diagnose whether the negative binomial is a reasonable fit.

{lab:3.8}

Exercise 3.8 The data frame *Bundesliga* gives a similar data set to that for UK soccer scores (*UKSoccer*) examined in Example 3.9, but over a wide range of years. The following lines calculate a two-way table, `BL1995`, of home-team and away-team goals for the 306 games in the year 1995.

```
> data("Bundesliga", package = "vcd")
> BL1995 <- xtabs(~ HomeGoals + AwayGoals, data = Bundesliga,
+                 subset = (Year == 1995))
> BL1995

          AwayGoals
HomeGoals  0  1  2  3  4  5  6
        0 26 16 13  5  0  1  0
        1 19 58 20  5  4  0  1
        2 27 23 20  5  1  1  1
        3 14 11 10  4  2  0  0
        4  3  5  3  0  0  0  0
        5  4  1  0  1  0  0  0
        6  1  0  0  1  0  0  0
```

(a) As in Example 3.9, find the one-way distributions of `HomeGoals`, `AwayGoals` and `TotalGoals = HomeGoals + AwayGoals`.
(b) Use `goodfit()` to fit and plot the Poisson distribution to each of these. Does the Poisson seem to provide a reasonable fit?

(c) Use `distplot()` to assess fit of the Poisson distribution.
(d) What circumstances of scoring goals in soccer might cause these distributions to deviate from Poisson distributions?

Exercise 3.9 ⋆ Repeat the exercise above, this time using the data for all years in which there was the standard number (306) of games, that is for `Year>1965`, tabulated as shown below.

```
> BL <- xtabs(~ HomeGoals + AwayGoals, data = Bundesliga,
+             subset = (Year > 1965))
```

{lab:3.10}

Exercise 3.10 Using the data *CyclingDeaths* introduced in Example 3.6 and the one-way frequency table `CyclingDeaths.tab = table(CyclingDeaths$deaths)`,

- Make a sensible plot of the number of deaths over time. For extra credit, add a smoothed curve (e.g., using `lines(lowess(...))`).
- Test the goodness of fit of the table `CyclingDeaths.tab` to a Poisson distribution statistically using `goodfit()`.
- Continue this analysis using a `rootogram()` and `distplot()`.
- Write a one-paragraph summary of the results of these analyses and your conclusions.

{lab:3.11}

Exercise 3.11 ⋆ The one-way table, *Depends*, in vcdExtra and shown below gives the frequency distribution of the number of dependencies declared in $4,983$ R packages maintained on the CRAN distribution network on January 17, 2014. That is, there were 986 packages that had no dependencies, $1,347$ packages that depended on one other package, up to 2 packages that depended on 14 other packages.

TODO: Perhaps promote this table to an introductory example, leaving analysis to this exercise.

Depends	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
# Pkgs	986	1,347	993	685	375	298	155	65	32	19	9	4	9	4	2

(a) Make a histogram of this distribution.
(b) Use `Ord_plot()` to see if this method can diagnose the form of the distribution.
(c) Try to fit a reasonable distribution to describe dependencies among R packages.

{lab:3.12}

Exercise 3.12 ⋆ How many years does it take to get into the baseball Hall of Fame? The Lahman package provides a complete record of historical baseball statistics from 1871 to the present. One table, *HallOfFame*, records the history of players nominated to the Baseball Hall of Fame, and those eventually inducted. The table below, calculated in `help(HallOfFame, package="Lahman")`, records the distribution of the number of years taken (from first nomination) for the 109 players in the Hall of Fame to be inducted (1936–present). Note that `years==0` does not, and cannot, occur in this table, so the distribution is restricted to positive counts. Such distributions are called ***zero-truncated distribution***s. Such distributions are like the ordinary ones, but with the probability of zero being zero. Thus the other probabilities are scaled up (i.e., divided by $1 - \Pr(Y = 0)$) so they sum to 1.

years	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
inducted	46	10	8	7	8	4	2	4	6	3	3	1	4	1	2

(a) For the Poisson distribution, show that the zero-truncated probability function can be expressed in the form

$$\Pr\{X = k \,|\, k > 0\}) = \frac{1}{1 - e^{-\lambda}} \times \frac{e^{-\lambda}\, \lambda^k}{k!} \qquad k = 1, 2, \ldots$$

(b) Show that the mean is $\lambda/(1 - \exp(-\lambda))$.
(c) Enter these data into R as a one-way table, and use `goodfit()` to fit the standard Poisson distribution, as if you hadn't encountered the problem of zero truncation.

References

Aberdein, J. and Spiegelhalter, D. (2013). Have London's roads become more dangerous for cyclists? *Significance*, 10(6), 46–48.

Andrews, D. F. and Herzberg, A. M. (1985). *Data: A Collection of Problems from Many Fields for the Student and Research Worker*. New York, NY: Springer-Verlag.

Arbuthnot, J. (1710). An argument for devine providence, taken from the constant regularity observ'd in the births of both sexes. *Philosophical Transactions*, 27, 186–190. Published in 1711.

Birch, M. W. (1963). An algorithm for the logarithmic series distributions. *Biometrics*, 19, 651–652.

Böhning, D. (1983). Maximum likelihood estimation of the logarithmic series distribution. *Statistische Hefte (Statistical Papers)*, 24(1), 121–140.

Edwards, A. W. F. (1958). An analysis of geissler's data on the human sex ratio. *Annals of Human Genetics*, 23(1), 6–15.

Fisher, R. A. (1925). *Statistical Methods for Research Workers*. London: Oliver & Boyd.

Fisher, R. A. (1936). Has Mendel's work been rediscovered? *Annals of Science*, 1, 115–137.

Fisher, R. A., Corbet, A. S., and Williams, C. B. (1943). The relation between the number of species and the number of individuals. *Journal of Animal Ecology*, 12, 42.

Geissler, A. (1889). Beitrage zur frage des geschlechts verhaltnisses der geborenen. *Z. K. Sachsischen Statistischen Bureaus*, 35(1), n.p.

Greenwood, M. and Yule, G. U. (1920). An inquiry into the nature of frequency distributions of multiple happenings, with particular reference to the occurrence of multiple attacks of disease or repeated accidents. *Journal of the Royal Statistical Society, Series A*, 83, 255–279.

Hilbe, J. (2011). *Negative Binomial Regression*. Cambridge University Press, 2nd edn.

Hoaglin, D. C. (1980). A poissonness plot. *The American Statistician*, 34, 146–149.

Hoaglin, D. C. and Tukey, J. W. (1985). Checking the shape of discrete distributions. In D. C. Hoaglin, F. Mosteller, and J. W. Tukey, eds., *Exploring Data Tables, Trends and Shapes*, chap. 9. New York: John Wiley and Sons.

Jinkinson, R. A. and Slater, M. (1981). Critical discussion of a graphical method for identifying discrete distributions. *The Statistician*, 30, 239–248.

Johnson, N. L., Kotz, S., and Kemp, A. W. (1992). *Univariate Discrete Distributions*. New York, NY: John Wiley and Sons, 2nd edn.

Kemp, A. W. and Kemp, C. D. (1991). Weldon's dice data revisited. *The American Statistician*, 45, 216–222.

Kendall, M. G. and Stuart, A. (1963). *The Advanced Theory of Statistics*, vol. 1. London: Griffin.

Kosambi, D. D. (1949). Characteristic properties of series distributions. *Proceedings of the National Institute of Science of India*, 15, 109–113.

Labby, Z. (2009). Weldon's dice, automated. *Chance*, 22(4), 6–13.

Lee, A. J. (1997). Modelling scores in the Premier League: Is Manchester United really the best? *Chance*, 10(1), 15–19.

Lindsey, J. K. (1995). *Modelling Frequency and Count Data*. Oxford, UK: Oxford University Press.

Lindsey, J. K. and Altham, P. M. E. (1998). Analysis of the human sex ratio by using overdispersion models. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 47(1), 149–157.

Lindsey, J. K. and Mersch, G. (1992). Fitting and comparing probability distributions with log linear models. *Computational Statistics and Data Analysis*, 13, 373–384.

Long, J. S. (1990). The origins of sex differences in science. *Social Forces*, 68(4), 1297–1316.

Long, J. S. (1997). *Regression Models for Categorical and Limited Dependent Variables*. Thousand Oaks, CA: Sage Publications.

Mosteller, F. and Wallace, D. L. (1963). Inference in an authorship problem. *Journal of the American Statistical Association*, 58(302), 275–309.

Mosteller, F. and Wallace, D. L. (1984). *Applied Bayesian and Classical Inference: The Case of the Federalist Papers*. New York, NY: Springer-Verlag.

Noack, A. (1950). A class of random variables with discrete distributions. *Annals of Mathematical Statistics*, 21, 127–132.

Ord, J. K. (1967). Graphical methods for a class of discrete distributions. *Journal of the Royal Statistical Society, Series A*, 130, 232–238.

Pearson, K. (1900). On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen by random sampling. *Philosophical Magazine*, 50(5th Series), 157–175.

Skellam, J. G. (1948). A probability distribution derived from the binomial distribution by regarding the probability of success as variable between the sets of trials. *Journal of the Royal Statistical Society. Series B (Methodological)*, 10(2), 257–261.

Tukey, J. W. (1977). *Exploratory Data Analysis*. Reading, MA: Addison Wesley.

von Bortkiewicz, L. (1898). *Das Gesetz der Kleinen Zahlen*. Leipzig: Teubner.

Wimmer, G. and Altmann, G. (1999). *Thesaurus of univariate discrete probability distributions*. Stamm.

Zelterman, D. (1999). *Models for Discrete Data*. New York: Oxford University Press.