

---

## Introduction: Count Data Containing Dispersion

This chapter is an overview summarizing relevant, established, and well-studied distributions for count data that motivate the consideration of the Conway–Maxwell–Poisson (COM–Poisson) distribution. Each of the discussed models provides an improved flexibility and computational ability for analyzing count data; yet associated restrictions help readers to appreciate the need for and usefulness of the COM–Poisson distribution, thus resulting in an explosion of research relating to this model. For completeness of discussion, each of these sections includes discussion of the relevant R packages and their contained functionality to serve as a starting point for forthcoming discussions throughout subsequent chapters. Along with the R discussion, illustrative examples aid readers in understanding distribution qualities and related statistical computational output. This background provides insights into the real implications of apparent data dispersion in count data models and the need to properly address it.

This introductory chapter proceeds as follows. Section 1.1 introduces the most well-known model for count data: the Poisson distribution. Its probabilistic and statistical properties are discussed, along with R tools to perform computations. Section 1.2, however, notes a major limitation of the Poisson distribution – namely its inability to properly model dispersed count data. Focusing first on the phenomenon of data over-dispersion, this section focuses attention on the negative binomial (NB) distribution – the most popular count distribution that allows for data over-dispersion. Section 1.3 meanwhile recognizes the existence of count data that express data under-dispersion and the resulting need for model consideration that can accommodate this property. While several flexible models allowing for data over- or under-dispersion exist in the literature, this section focuses attention on the generalized Poisson (GP) distribution for modeling such data because it is arguably (one of) the most popular option(s) for modeling

such data. Section 1.4 offers an overarching perspective about these models as special cases of a larger class of weighted Poisson distributions. Finally, Section 1.5 motivates an interest in the COM–Poisson distribution and summarizes the rest of the book, including the unifying background that will be referenced in subsequent chapters.

### 1.1 Poisson Distribution

The Poisson distribution is the most studied and applied distribution referenced to describe variability in count data. A random variable  $X$  with a Poisson( $\lambda$ ) distribution has the probability mass function

$$P(X = x) = \frac{\lambda^x e^{-\lambda}}{x!}, \quad x = 0, 1, 2, \dots, \quad (1.1)$$

or, on the log-scale,

$$\begin{aligned} \ln [P(X = x)] &= x \ln \lambda - \ln (x!) - \lambda \\ &= x \ln \lambda - \sum_{j=1}^x \ln (j) - \lambda, \end{aligned}$$

where  $\lambda$  is the associated intensity parameter; illustrative examples of the distributional form assuming various values of  $\lambda$  are provided in Figure 1.1.

Derived as the limiting distribution of a binomial( $n, p$ ) distribution where  $n \rightarrow \infty$  and  $p \rightarrow 0$  such that  $np = \lambda$ , the beauty of this distribution lies in its simplicity. Both its mean and variance equal the intensity parameter  $\lambda$ ; thus, the dispersion index is

$$DI(X) = \frac{V(X)}{E(X)} = \frac{\lambda}{\lambda} = 1. \quad (1.2)$$

The probability mass function satisfies the recursion

$$\frac{P(X = x - 1)}{P(X = x)} = \frac{x}{\lambda}, \quad (1.3)$$

with its moment generating function  $M_X(t) = e^{\lambda(e^t - 1)}$ , and the Poisson distribution is a member of the exponential family of the form

$$P(X = x; \theta) = H(x) \exp [\eta(\theta)T(x) - \Psi(\theta)], \quad x \in \mathbb{N}, \quad (1.4)$$

where, for  $\theta = \lambda$ ,  $\eta(\theta) = \ln(\lambda)$ ,  $\Psi(\theta) = \lambda$ ,  $T(x) = x$ , and  $H(x) = (x!)^{-1}$ . The simplicity of the Poisson distribution, however, can also be viewed as theoretically constraining and not necessarily representative of real count

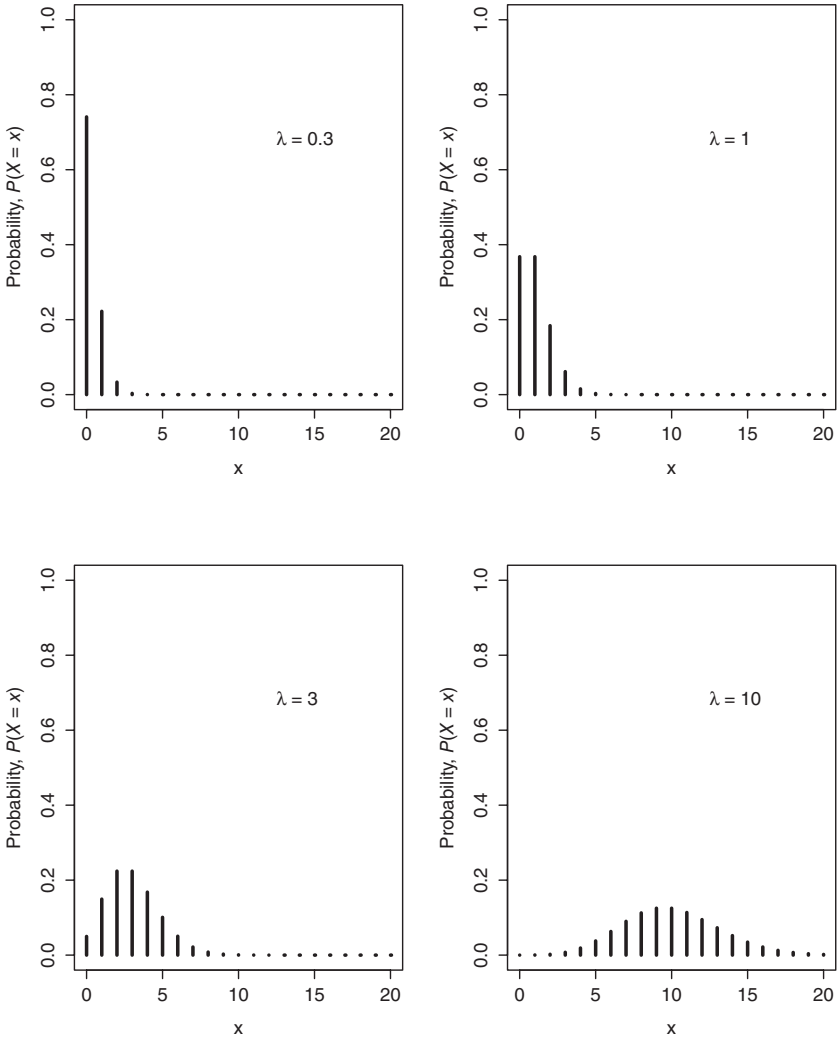


Figure 1.1 Poisson probability mass function illustrations for  $\lambda \in \{0.3, 1, 3, 10\}$ .

data distributions. Thus, applying statistical methods that are motivated and/or developed by the Poisson model assumption can cause significant repercussions with regard to statistical inference. This matter is discussed in more detail in the subsequent sections in Chapter 1 and throughout this reference.

### 1.1.1 R Computing

The `stats` package contains functions to compute the probability, distribution function, quantile function, and random number generation associated with the Poisson distribution. All of the relevant commands require the Poisson rate parameter  $\lambda$  (`lambda`) as an input value. The `dpois` function computes the probability/density  $P(X = x)$  for a random variable  $X$  at observation  $x$ . The command has the default setting as described (`log = FALSE`), while changing the indicator input to `log = TRUE` computes the probability on the natural-log scale. The `ppois` function computes the cumulative probability  $P(X \leq q)$  given a quantile value  $q$ , while `qpois` determines the quantile  $q$  (i.e. the smallest integer) for which the cumulative probability  $P(X \leq q) \geq p$  for some given probability  $p$ . This quantile determination stems from the discrete nature of the Poisson probability distribution. Both commands contain the default settings `lower.tail = TRUE` and `log.p = FALSE`. The condition `lower.tail = TRUE` infers interest regarding the cumulative probability  $P(X \leq q)$  while `lower.tail = FALSE` focuses on its complement  $P(X > q)$  (i.e. the upper tail). The indicator `log.p = FALSE` (`TRUE`) meanwhile infers whether to consider probabilities on the original or natural-log scale, respectively. Finally, the `rpois` function produces a length  $n$  (`n`) vector of count data randomly generated via the Poisson distribution.

Demonstrative examples utilizing the respective functions are provided in Code 1.1, all of which assume the Poisson rate parameter  $\lambda = 3$ . The command `dpois(x=5, lambda=3)` determines that  $P(X=x) = 0.1008188$ ; this value is illustrated in Figure 1.1 for  $\lambda = 3$ . Meanwhile, `dpois(x=5, lambda=3, log = TRUE)` shows that  $\ln(P(X=x)) = \ln(0.1008188) = -2.29443$ . The `ppois` functions demonstrate the difference between computing the lower versus upper tail, respectively; naturally, the sum of the two results equals 1. The command `qpois(p=0.9, lambda=3)` produces the expected result of 5 because we see that the earlier `ppois(q=5, lambda=3)` result showed that  $P(X \leq 5) = 0.9160821 > 0.9$ . Meanwhile, one can see that `qpois(p=0.9, lambda=3, lower.tail = FALSE)` produces the value 1 by considering the corresponding `ppois` commands:

```
ppois(q=0, lambda=3, lower.tail=FALSE) produces the result 0.9502129
ppois(q=1, lambda=3, lower.tail=FALSE) produces the result 0.8008517.
```

Recall that the discrete nature of the Poisson distribution requires a modified approach for determining the quantile value; the resulting quantile is

Code 1.1 Examples of R function use for Poisson distributional computing:

`dpois`, `ppois`, `qpois`, `rpois`.

```
> dpois(x=5, lambda=3)
[1] 0.1008188
> dpois(x=5, lambda=3, log = TRUE)
[1] -2.29443
> ppois(q=5, lambda=3)
[1] 0.9160821
> ppois(q=5, lambda=3, lower.tail = FALSE)
[1] 0.08391794
> qpois(p=0.9, lambda=3)
[1] 5
> qpois(p=0.9, lambda=3, lower.tail = FALSE)
[1] 1
> rpois(n=10, lambda=3)
[1] 3 4 3 5 2 0 5 5 4 3
```

determined such that the cumulative probability of interest is at least as much as the desired probability of interest. This definition suggests that, when considering the upper tail probability, the resulting quantile now implies that the corresponding upper tail probability is no more than the desired probability of interest. As noted above,  $P(X > 0) = 0.9502129$  and  $P(X > 1) = 0.8008517$ ; because the desired upper tail probability in the example is 0.9, we see that 0 produces an upper tail probability that is too large for consideration, while the upper tail probability associated with 1 is the first integer that satisfies  $P(X > x) \leq 0.9$ , thus producing the solution as 1. Finally, for completeness, the `rpois` function produces 10 randomly generated potential observations stemming from a Poisson(3) distribution. Given the probability mass function illustration provided in Figure 1.1 for  $\lambda = 3$ , these outcomes appear reasonable.

## 1.2 Data Over-dispersion

Over-dispersion (relative to a comparable Poisson model) describes distributions whose variance is larger than the mean, i.e.  $DI(X) > 1$  for a random variable  $X$ . This is a well-studied phenomenon that occurs in most real-world datasets. Over-dispersion can be caused by any number of situations, including data heterogeneity, the existence of positive correlation between responses, excess variation between response probabilities or counts, and violations in data distributional assumptions. Apparent over-dispersion can also exist in datasets because of outliers or, in the case of regression

models, the model may not include important explanatory variables or a sufficient number of interaction terms, or the link relating the response to the explanatory variables may be misspecified. Under such circumstances, over-dispersion causes problems because resulting standard errors associated with parameter estimation may be underestimated, thus producing biased inferences. Interested readers should see Hilbe (2007) for a comprehensive discussion regarding over-dispersion and its causes.

The most popular distribution to describe over-dispersed data is the NB distribution. A random variable  $X$  with an  $NB(r, p)$  distribution has the probability mass function

$$P(X = x) = \binom{r+x-1}{x} p^x (1-p)^r \quad (1.5)$$

$$= \frac{\Gamma(r+x)}{x! \Gamma(r)} p^x (1-p)^r, \quad x = 0, 1, 2, \dots, \quad (1.6)$$

and can be viewed as the probability of attaining a total of  $x$  successes with  $r > 0$  failures in a series of independent Bernoulli( $p$ ) trials, where  $0 < p < 1$  denotes the success probability associated with each trial. Alternatively, the NB distribution can be derived via a mixture model of a Poisson( $\lambda$ ) distribution, where  $\lambda$  is gamma distributed<sup>1</sup> with shape and scale parameters,  $r$  and  $p/(1-p)$ , respectively. The latter approach is a common technique for addressing heterogeneity. Other possible distributions for  $\lambda$  include the generalized gamma (which produces a generalized form of the NB distribution (Gupta and Ong, 2004)), the inverse Gaussian, and the generalized inverse Gaussian (which produces the Sichel distribution (Atkinson and Yeh, 1982; Ord and Whitmore, 1986)). Various other mixing distributions have also been considered; see Gupta and Ong (2005) for discussion.

The moment generating function of the  $NB(r, p)$  random variable  $X$  is

$$M_X(t) = \left( \frac{p}{1 - (1-p)e^t} \right)^r, \quad t < -\ln(1-p),$$

<sup>1</sup> For a gamma( $\alpha, \beta$ ) distributed random variable  $X$  with shape and scale parameters  $\alpha$  and  $\beta$ , respectively, its probability density function (pdf) is  $f(x) = \frac{1}{\Gamma(\alpha)\beta^\alpha} x^{\alpha-1} e^{-x/\beta}$  (Casella and Berger, 1990).

which produces a respective mean and variance,

$$\mu \doteq E(X) = \frac{r(1-p)}{p} \quad \text{and} \quad (1.7)$$

$$V(X) = \frac{r(1-p)}{p^2} = \mu + \frac{1}{r}\mu^2, \quad (1.8)$$

where  $r > 0$  can be viewed as a dispersion parameter. Given the dispersion parameter  $r$ , this distribution can be represented as an exponential family (Equation (1.4)), where  $\theta = p$ ,  $H(x; r) = \binom{r+x-1}{x}$ ,  $T(x) = x$ ,  $\eta(p) = \ln p$ , and  $\psi(p; r) = r \ln(1-p)$ . Equation (1.8) demonstrates that the NB distribution can accommodate data over-dispersion ( $DI(X) > 1$ ) because one can clearly see that the distribution's variance is greater than or equal to its mean since  $r > 0$ . Further, the NB distribution contains the Poisson as a limiting case; as  $r \rightarrow \infty$  and  $p \rightarrow 1$  such that  $r(1-p) \rightarrow \lambda$ ,  $0 < \lambda < \infty$ , not only do the NB mean and variance both converge to  $\lambda$ , but the NB probabilities likewise converge to their respective Poisson counterparts. Figure 1.2 illustrates the distributional convergence of the  $NB(r, p)$  to the  $Poisson(\lambda = 3)$  distribution, where  $r \rightarrow \infty$  and  $p \rightarrow 1$  such that  $r(1-p) = 3$ . The NB distribution likewise contains the geometric( $p$ ) as a special case when  $r = 1$ .

The NB distribution can alternatively be represented as  $NB(r, r/(r + \mu))$  with the probability mass function

$$P(X = x) = \binom{x+r-1}{x} \left(\frac{r}{r+\mu}\right)^x \left(\frac{\mu}{r+\mu}\right)^r, \quad x = 0, 1, 2, \dots, \quad (1.9)$$

where  $r > 0$ ,  $\mu > 0$ ; this formulation explicitly has a mean  $\mu$  and a variance  $\mu + \mu^2/r$ . The MASS package in R utilizes this parametrization and defines the dispersion parameter as `theta` such that  $V(X) = \mu + \mu^2/\theta$ , i.e.  $\theta \doteq r$ ; we will revisit this in Chapter 5. While the NB distribution has been well studied and statistical computational ability is supplied in numerous software packages (e.g. R and SAS), an underlying constraint regarding the NB distribution leads to its inability to address data under-dispersion (i.e. the dispersion index is less than 1, or the variance is smaller than the mean).

### 1.2.1 R Computing

The `stats` package provides functionality for determining the probability, distribution function, quantile function and random number generation for the NB distribution. These commands all require the inputs `size` ( $r$ ) and either the success probability  $p$  (`prob`) or mean  $\mu$  (`mu`), depending on the

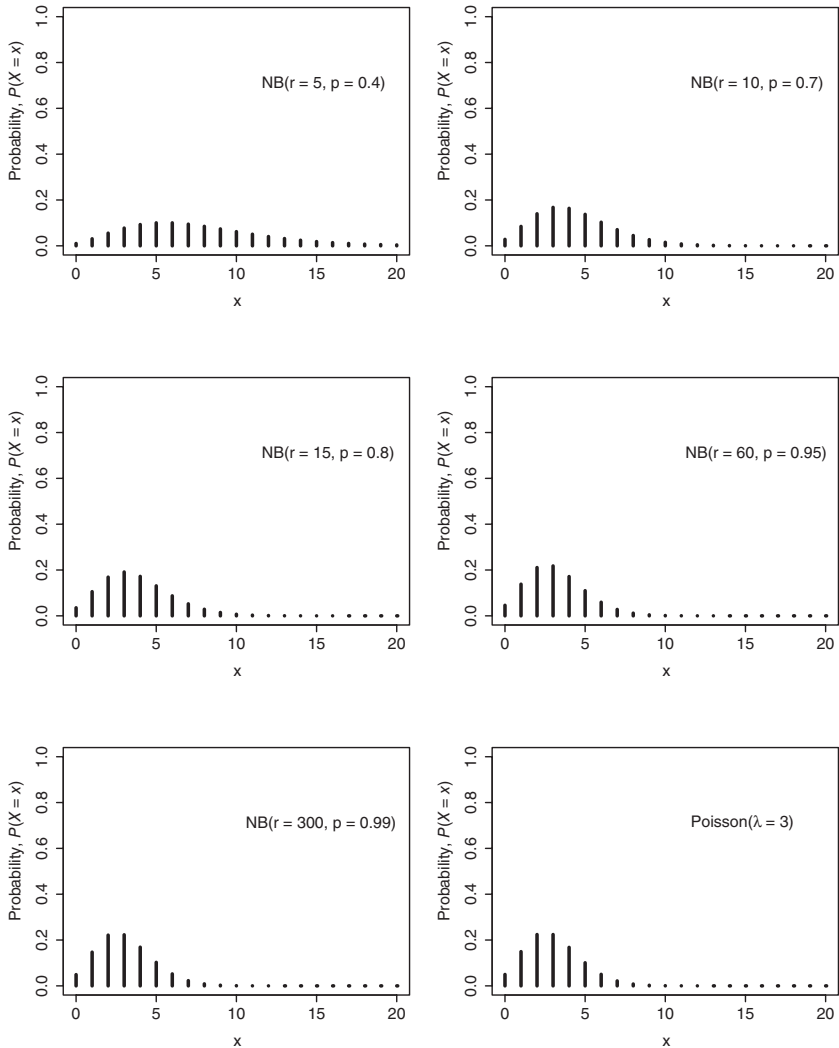


Figure 1.2 Negative binomial distribution illustrations for values of  $(r, p) \in \{(5, 0.4), (10, 0.7), (15, 0.8), (60, 0.95), (300, 0.99)\}$  and the  $\text{Poisson}(\lambda = 3)$  probability mass function. This series of density plots nicely demonstrates the distributional convergence of the negative binomial to the Poisson as  $r \rightarrow \infty$  and  $p \rightarrow 1$  such that  $r(1 - p) \rightarrow \lambda$ .

choice of parametrization. The function `dnbinom` computes the probability  $P(X = x)$  for a random variable  $X$  at observation  $x$ , either on the original scale (`log = FALSE`; this is the default setting) or on a natural-log scale



Code 1.2 Examples of R commands for NB distributional computing: `dnbinom`, `pnbinom`, `qnbinom`, `rnbinom`.

```
> dnbinom(x=5, size=10, prob=0.7)
[1] 0.1374203
> dnbinom(x=5, size=10, prob=0.7, log = TRUE)
[1] -1.984712
> pnbinom(q=5, size=10, prob=0.7)
[1] 0.7216214
> pnbinom(q=5, size=10, prob=0.7, lower.tail = FALSE)
[1] 0.2783786
> qnbinom(p=0.9, size=10, prob=0.7)
[1] 8
> qnbinom(p=0.9, size=10, prob=0.7, lower.tail = FALSE)
[1] 1
> rnbinom(n=10, size=10, prob=0.7)
[1] 1 8 7 3 5 8 4 2 5 3
```

(`log = TRUE`). For a given quantile value  $q$ , the `pnbinom` function determines the cumulative probability  $P(X \leq q)$ , where the default settings, `lower.tail = TRUE` and `log.p = FALSE`, imply that the resulting cumulative probability is attained by accumulating the probability from the lower tail and on the original probability scale. The command `qnbinom` meanwhile determines the smallest discrete quantile value  $q$  that satisfies the cumulative probability  $P(X \leq q) \geq p$  for a given probability  $p$ . This function likewise assumes the default settings, `lower.tail = TRUE` and `log.p = FALSE`, such that the quantile  $q$  is determined from the lower tail and on the original probability scale. For both of these commands, changing the default settings to `lower.tail = FALSE` and `log.p = TRUE`, respectively allows analysts to instead consider quantile determination on the basis of the upper tail probability  $P(X > q)$ , and via a probability computation on the basis of the natural-log scale. Finally, the `rnbinom` function randomly generates  $n$  (`n`) observations from an NB distribution with the specified size (`size`) and success probability (`prob`).

The NB( $r = 10$ ,  $p = 0.7$ ) distribution is provided in Figure 1.2 and serves as a graphical reference for the illustrative commands featured in Code 1.2. All of the demonstrated functions assume  $r = 10$  and  $p = 0.7$  as the associated NB size and success probability parameters. The first command (`dnbinom(x=5, size=10, prob=0.7)`) shows that  $P(X = x) = 0.1374203$ ; this probability is shown in the associated plot in Figure 1.2. Meanwhile, `dnbinom(x=5, size=10, prob=0.7, log = TRUE)` shows that  $\ln(P(X = x)) = \ln(0.1374203) = -1.984712$ .

The `pnbinom` functions show the results when computing the lower versus upper tail, respectively; naturally, the sum of the two computations equals 1. Calling `qnbinom(p=0.9, size=10, prob=0.7)` produces the result 8, while `qnbinom(p=0.9, size=10, prob=0.7, lower.tail = FALSE)` yields the value 1. Finally, the `rnbinom` command produces 10 randomly generated potential observations stemming from an  $NB(r = 10, p = 0.7)$  distribution.

### 1.3 Data Under-dispersion

Where data over-dispersion describes excess variation in count data, under-dispersion describes deficient variation in count data. Data under-dispersion (relative to the Poisson model) refers to count data that are distributed such that the variance is smaller than the mean, i.e. its dispersion index  $DI(X) < 1$  for a random variable  $X$ .

There remains some measures of debate regarding the legitimacy of data under-dispersion as a real concept. Some researchers attribute under-dispersion to the data generation (e.g. small sample values) or to the modeling process (e.g. model over-fitting), noting that the arrival process, birth–death process, or binomial thinning mechanisms can also lead to under-dispersion (Kokonendji, 2014; Lord and Guikema, 2012; Puig et al., 2016). As an example, for renewal processes where the distribution of the interarrival times has an increasing hazard rate, the distribution of the number of events is under-dispersed (Barlow and Proschan, 1965). Efron (1986), however, argues that “there are often good physical reasons for not believing in under-dispersion, however, especially in binomial and Poisson situations.”

Whether real or apparent, examples across disciplines are surfacing with more frequency where data under-dispersion is present; thus there exists the need to represent such data. The most popular model that can accommodate data dispersion (whether over- or under-dispersion) is the GP distribution – a flexible two-parameter distribution that contains the Poisson distribution as a special case (Consul, 1988). A random variable  $X$  that is  $GP(\lambda_1, \lambda_2)$  distributed has the probability mass function

$$P(X = x) = \begin{cases} \frac{\lambda_1(\lambda_1 + \lambda_2 x)^{x-1}}{x!} \exp(-\lambda_1 - \lambda_2 x), & x = 0, 1, 2, \dots \\ 0, & x \geq m \text{ where } \lambda_1 + \lambda_2 m \leq 0 \end{cases} \quad (1.10)$$

for  $\lambda_1 > 0$  and  $-1 < \lambda_2 < 1$  (Consul and Jain, 1973). This distribution has the respective mean and variance,

$$E(X) = \frac{\lambda_1}{1 - \lambda_2} \quad (1.11)$$

$$V(X) = \frac{\lambda_1}{(1 - \lambda_2)^3}, \quad (1.12)$$

and can accommodate any form of data dispersion via  $\lambda_2$ . The  $GP(\lambda_1, \lambda_2)$  distribution contains the special-case  $Poisson(\lambda_1)$  distribution, where  $\lambda_2 = 0$ ; this is the case of equi-dispersion relative to the Poisson model. Meanwhile, for  $\lambda_2 > (<)0$ , the GP distribution accommodates data over-dispersion (under-dispersion). Figure 1.3 illustrates various probability mass functions for different values of  $\lambda_1$  and  $\lambda_2 \in \{-0.5, 0, 0.5\}$ . These choices for  $\lambda_1$  and  $\lambda_2$  demonstrate the change in shape and skewness for this unimodal distribution and also illustrate the data over- or under-dispersion as a function of  $\lambda_2$ . The middle column of Figure 1.3 contains the respective  $Poisson(\lambda_1 = 2, 3, 6)$  probability distributions.

The GP distribution allows for over- or under-dispersion; however, extreme under-dispersion can result in probability models that do not satisfy the basic probability axioms (Famoye, 1993). Alternative count distributions exist that allow for data under-dispersion, such as the condensed Poisson, the Gamma count, and the double Poisson distributions; see Sellers and Morris (2017) for discussion regarding these distributions. Nonetheless, the GP distribution maintains its status as a very popular and well-studied count distribution that allows for data dispersion.

### 1.3.1 R Computing

The GP distribution is a popular model for describing count data that express either over- or under-dispersion, and this is reflected through the multiple R packages available for statistical computing. Basic functionality exists in the packages `HMMpa` (Witowski and Foraita, 2018), `LaplacesDemon` (Statisticat and LLC., 2021), and `RNGforGPD` (Li et al., 2020), while commands to conduct GP regression are available in the `VGAM` (Yee, 2008) package.

The `HMMpa` and `LaplacesDemon` packages each contain commands that can compute the probability mass function of a GP distribution. `HMMpa` provides the `dgenpois(x, lambda1, lambda2)` function, where `lambda1` and `lambda2` are  $\lambda_1$  and  $\lambda_2$  as defined in Equation (1.10). `LaplacesDemon` meanwhile contains the function `dgpois(x, lambda,`

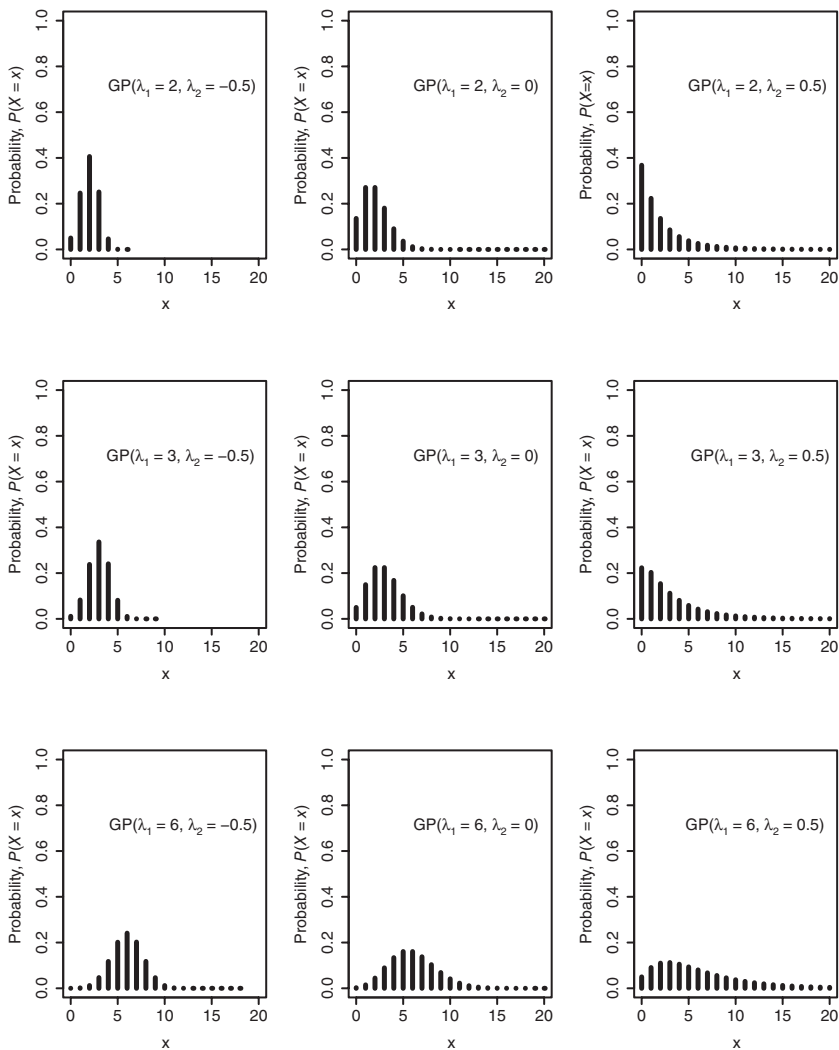


Figure 1.3 Generalized Poisson probability mass function illustrations for values of  $\lambda_1 > 0$ , and dispersion parameter  $\lambda_2 \in \{-0.5, 0, 0.5\}$ . For  $\lambda_1 > 0$  and  $-1 < \lambda_2 < 1$  such that  $\lambda_2 > (<)0$  denotes data over-dispersion (under-dispersion), the generalized Poisson distribution has the mean  $E(X) = \frac{\lambda_1}{1-\lambda_2}$  and variance  $V(X) = \frac{\lambda_1}{(1-\lambda_2)^3}$ .

omega, log=FALSE) that computes the probability mass function via an alternate parametrization, namely

$$P(X = x) = \frac{\lambda(1 - \omega)[\lambda(1 - \omega) + \omega x]^{x-1}}{x!} \exp(-\lambda(1 - \omega) - \omega x),$$

$$x = 0, 1, 2, \dots, \quad (1.13)$$

for parameters  $\lambda > 0$  and  $0 \leq \omega < 1$  (as reported in Statisticat and LLC. (2021)). Under this parametrization,  $\omega = 0$  reduces the GP distribution to the Poisson( $\lambda$ ) distribution. Equations (1.10) and (1.13) are equivalent with  $\lambda_1 = \lambda(1 - \omega)$  and  $\lambda_2 = \omega$ . The `dgpois` logical input `log` determines whether the probability mass function is provided on the original (`log=FALSE`; this is the default) or natural-log (`log=TRUE`) scale. The two functions `dgenpois` and `dgpois` produce identical outcomes for `lambda1 = lambda(1-omega)` and `lambda2 = omega` for appropriate values of  $x$ .

While `dgenpois` and `dgpois` both have the capability to compute  $P(X=x)$  for a GP Poisson random variable  $X$ , these functions should be used with caution. The GP parametrization that motivates `dgpois` stems from an applied focus involving claim count data with the argument that such data are not commonly under-dispersed so that distributional focus assumes nonnegative  $\omega$  (Ntzoufras et al., 2005). Equation (1.13) thus has a mean and variance

$$E(X) = \lambda \quad (1.14)$$

$$V(X) = \frac{\lambda}{(1 - \omega)^2} \quad (1.15)$$

that results in the dispersion index,  $DI(X) = 1/(1 - \omega)^2 \geq 1$ . The `dgpois` function, however, appears to accurately compute probabilities associated with data under-dispersion (i.e. satisfying  $-1 < \omega < 0$ ); hence analysts can safely maintain  $|\omega| < 1$ . The `dgenpois` function meanwhile computes the first component of Equation (1.10) (i.e.  $\frac{\lambda_1(\lambda_1 + \lambda_2 x)^{x-1}}{x!} \exp(-\lambda_1 - \lambda_2 x)$  for  $x = 0, 1, 2, \dots$ ); however, it does not set  $P(X = x) = 0$  for those  $x \geq m$ , where  $\lambda_1 + \lambda_2 m \geq 0$ . As a result, the function can compute extraneous output; Figure 1.4 provides an illustrative example. As demonstrated in Figure 1.4(a), because the `dgenpois` function does not properly account for values  $x \geq m$ , where  $\lambda_1 + \lambda_2 m \geq 0$  for some  $m$ , the resulting outcomes defy the probability axioms. In this illustration, we see that  $m = 6$ ; thus  $P(X = x)$  should equal 0 for  $x \geq 6$ . Reported computations for  $x > 6$ , however,

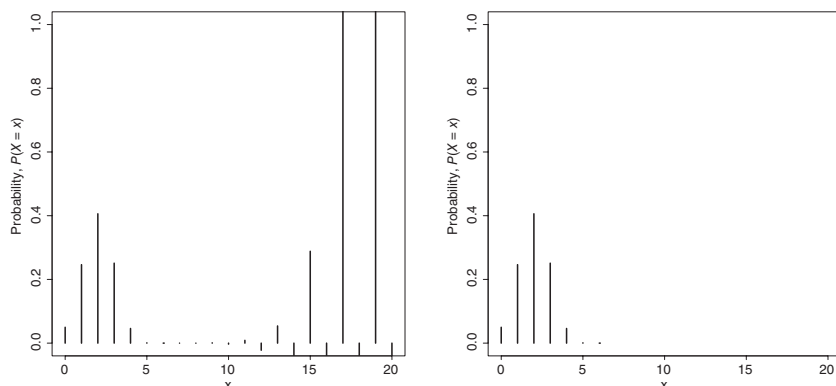


Figure 1.4 The probability mass function  $P(X = x)$  created for  $x \in \{0, \dots, 20\}$  for a generalized Poisson distribution (a) via `dgenpois` (HMMpa) with  $\lambda_1 = 3$ ,  $\lambda_2 = -0.5$ ; and (b) via `dgpois` (LaplacesDemon) with  $\lambda = 2$ ,  $\omega = -0.5$ . The resulting plots should be identical because  $\lambda_1 = \lambda(1 - \omega)$  and  $\lambda_2 = \omega$ .

instead bifurcate between outcomes that increase in absolute value, whether negative or positive (thus further producing outcomes that are greater than 1); both of these scenarios contradict probability axioms. Thus, in order to get the `dgenpois` function to provide appropriate output, it is important to insert the condition `(lambda1+lambda2*x) >= 0`; see below for illustrative R code and output that can produce probabilities as shown in Figure 1.4(b).

```
> x<- 0:20
> lambda1=3
> lambda2=-0.5
> ifelse((lambda1+lambda2*x) >= 0, dgenpois(x, lambda1, lambda2), 0)
 [1] 0.0497870684 0.2462549959 0.4060058497 0.2510214302 0.0459849301
 [6] 0.0009477042 0.0000000000 0.0000000000 0.0000000000 0.0000000000
[11] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
[16] 0.0000000000 0.0000000000 0.0000000000 0.0000000000 0.0000000000
[21] 0.0000000000
```

The command `dgpois(x=0:20, lambda=2, omega=-0.5)` likewise produces Figure 1.4(b); this is because `dgpois` properly detects the need to set  $P(X = x) = 0$  for  $x \geq 7$ . This function, however, does so by producing a warning and NaNs as outcomes for those probabilities  $P(X = x)$ ,  $x \geq m_2$  for some  $m_2$  such that  $\lambda_1 + \lambda_2 m_2 = \lambda(1 - \omega) + \omega m_2 < 0$ ; see the following illustration for details. The term  $\lambda(1 - \omega) + \omega x$  is defined as `lambda.star` in the `dgpois` function and is referenced in the following warning message. In this example,  $x \geq 7$  produces NaN (i.e. in the eighth position).

```

> dgpois(x=0:20, lambda=2, omega=-0.5)
[1] 0.0497870684 0.2462549959 0.4060058497 0.2510214302 0.0459849301
[6] 0.0009477042 0.0000000000          NaN          NaN          NaN
[11]          NaN          NaN          NaN          NaN          NaN
[16]          NaN          NaN          NaN          NaN          NaN
[21]          NaN
Warning message:
In log(lambda.star) : NaNs produced

```

The `dgpois` and `dgenpois`  $P(X = x)$  outputs are equivalent for  $x = 0, \dots, 6$  (see Figure 1.4); thus both of these functions are capable of computing the first condition of the GP probability mass function as shown in Equations (1.10) and (1.13). Analysts are thus encouraged to first confirm that the constraint  $\lambda_1 + \lambda_2 x = \lambda(1 - \omega) + \omega x \geq 0$  is satisfied in order to ensure proper GP probability computation.

HMMpa also contains the functions `pgenpois` and `rngenpois` to conduct cumulative probability computation and random number generation, respectively, based on the GP distribution. Both functions require the parameter inputs `lambda1` and `lambda2`; `pgenpois` needs the added input `q` to determine the cumulative probability  $P(X \leq q)$  for a quantile value  $q$ , while `rngenpois` further requires the value `n` to obtain  $n$  randomly generated observations from a  $GP(\lambda_1, \lambda_2)$  distribution. Recognizing the aforementioned issue, however, that the `genpois` functions contained in HMMpa do not first constrain the support space for  $x$  such that  $\lambda_1 + \lambda_2 x \geq 0$ , one should ensure that this caveat holds for any subsequent use of `dgenpois` or `pgenpois` in order to have confidence in the resulting output. The HMMpa function `rngenpois` appears to operate properly as a random number generator based on the GP distribution; the function selects proper values associated with the true support space. The `RNGforGPD` package offers alternative commands with the ability to randomly generate univariate or multivariate generalized Poisson data. The `GenUniGpois` function generates univariate GP data via one of five methods (inversion, build-up, chop-down, normal-approximation, and branching) selected by the analyst. For the given rate and dispersion parameters, `theta` and `lambda` respectively, and `method`, `GenUniGpois` can generate `n` univariate data from a  $GP(\theta = \lambda_1, \lambda = \lambda_2)$  distribution, where we note the aforementioned variable substitutions to adhere to Equation (1.10) for  $\lambda_1 > 0$  and  $-\lambda_1/4 \leq \lambda_2 < 1$ .

As with other GP representations, the `RNGforGPD` package recognizes the Poisson model as a special case of the GP distribution when `lambda = lambda_2 = 0`; under this circumstance, any data-generation method can be

specified. Analysts should otherwise be mindful of which method is selected for random number generation as constraints exist in order to ensure performance and/or reliability. The branching method does not work for generating under-dispersed data (thus  $\lambda_2 \geq 0$ ), and the normal approximation approach is not necessarily reliable for  $\lambda_1 < 10$  (Demirtas, 2017; Li et al., 2020). The `GenMVGp` function meanwhile generates data of size `sample.size` from a multivariate GP distribution with the marginal rate and dispersion vectors `theta.vec` and `lambda.vec`, respectively, and the correlation matrix `cmat.star`; see Li et al. (2020) for details.

The `RNGforGPD` package likewise contains the function `Quantile-Gp` that can determine the quantile  $q$  that satisfies the cumulative probability  $P(X \leq q) \geq p$  for some percentile  $p$  associated with a  $GP(\theta=\lambda_1, \lambda=\lambda_2)$  distributed random variable. This function includes the logical input `details`, where `details=FALSE` (the default setting) reports the quantile value, and `details=TRUE` provides the probability  $P(X = x)$  and cumulative probability  $P(X \leq x)$  for every  $x \leq q$ . When providing a negative dispersion parameter, it may be helpful to set `details=TRUE` as `RNGforGPD` adjusts the initially provided cumulative probabilities to account for the truncation error, and then lists the adjusted cumulative probabilities.

## 1.4 Weighted Poisson Distributions

The weighted Poisson distribution is a flexible model class for count data that can account for either over- or under-dispersion. Let  $X^w$  denote the weighted version of a Poisson random variable  $X$  with the probability mass function  $P(X = x; \lambda)$  as defined in Equation (1.1);  $X^w$  has the probability

$$P(X^w = x; \lambda) = \frac{w(x)P(X = x; \lambda)}{E_\lambda(w(X))}, \quad x = 0, 1, 2, \dots, \quad (1.16)$$

where  $w(\cdot)$  is a nonnegative weight function, and  $E_\lambda(w(X)) = \sum_{j=0}^{\infty} w(j)P(X = j; \lambda) > 0$  is the finite expectation. The weighted Poisson is actually a class of distributions that depends on their associated weight functions and does not offer its own general statistical computing packages (e.g. in R). Examples of weighted Poisson distributions include the NB and GP distributions; Table 1.1 provides the weight functions that define several examples of weighted Poisson models.

The weighted Poisson distribution has several interesting properties. For a Poisson weight function having an exponential form,

$$w(y) = \exp [rt(y)], \quad y \in \mathbb{N},$$



Table 1.1 *Weight functions associated with various examples of weighted Poisson distributions.*

Distribution	Weight function, $w(x)$
Poisson	1
negative binomial	$\Gamma(r + x)$ , where $r > 0$
generalized Poisson	$\left(\frac{\lambda_1 + \lambda_2 x}{\lambda_1}\right)^{x-1} \exp(-\lambda_2 x)$ , where $\lambda_1 > 0$ and $-1 < \lambda_2 < 1$

where  $r \in \mathbb{R}$  and  $y \rightarrow t(y)$  is a convex function (that may or may not depend on the original Poisson parameter),  $r > 0$  corresponds to a weighted Poisson distribution that is over-dispersed. Similarly,  $r = (<) 0$  implies that it is equi-dispersed (under-dispersed) (del Castillo and Pérez-Casany, 2005). The random variable  $X^w$  is over-dispersed (under-dispersed) if and only if the mean weight function  $E_\lambda(w(X; \phi))$  for a weight function  $w(x; \phi)$  that does not depend on the Poisson mean  $\lambda > 0$  is log-convex (log-concave). Further,  $E_\lambda(w(X; \phi))$  has the same direction of log-concavity as  $w(x; \phi)$ ; if  $w(x; \phi)$  is log-convex (log-concave), then  $E_\lambda(w(X; \phi))$  is likewise log-convex (log-concave). Thus, one can simply assess the shape of  $w(x; \phi)$  to determine the direction of dispersion for  $X^w$ . Accordingly, a positive weight function's log-concavity implies the log-concavity of the weighted Poisson distribution; if the weight function  $w(x; \phi)$  is log-concave, then the associated weighted Poisson distribution is likewise log-concave. These concavity results are compelling because they imply other relationships regarding distributional forms. Discrete log-concave distributions have an increasing failure rate and are unimodal, while log-convex distributions have a decreasing failure rate (DFR) and are infinitely divisible, thus implying over-dispersion (Kokonendji et al., 2008). Two weighted Poisson distributions are defined as a pointwise dual pair if their respective positive Poisson weight functions  $w_1$  and  $w_2$  satisfy  $w_1(x)w_2(x) = 1$  for all  $x \in \mathbb{N}$ . The dual of weighted Poisson distributions is closed if the two distributions have differing dispersion types, i.e. one is over-dispersed (under-dispersed) and the other is under-dispersed (over-dispersed). Further, all natural exponential families of the form

$$P(X = x; \theta, \phi) = \Gamma(x; \phi) \exp[\eta(\theta)T(x) - \Psi(\theta; \phi)], \quad x \in \mathbb{N}, \quad (1.17)$$

with a fixed  $\phi > 0$  are weighted Poisson distributions where the weight function is  $w(x; \phi) = x! \Gamma(x; \phi)$ ,  $x \in \mathbb{N}$ ; however, not all weighted Poisson distributions have the exponential family form. The weighted Poisson

distribution is likewise a member of an exponential dispersion family if it satisfies the form

$$P(X = x; \theta, \phi) = H(x; \phi) \exp \left( \frac{(\eta(\theta)T(x) - \Psi(\theta; \phi))w}{\phi} \right). \quad (1.18)$$

Weighted Poisson distributions give rise to a destructive cure rate model framework in survival analysis. Let  $M^w$  denote the number of competing causes associated with an event occurrence and have a weighted Poisson distribution as defined in Equation (1.16). Given  $M^w$ , let

$$D^w = \begin{cases} \sum_{i=1}^{M^w} B_i & M^w > 0 \\ 0 & M^w = 0, \end{cases} \quad (1.19)$$

where  $B_i$  are independent and identically Bernoulli( $p$ ) distributed random variables (independent from  $M^w$ ) noting the presence (1) or absence (0) of Cause  $i = 1, \dots, M^w$ .  $D^w$  denotes the total number of competing risks or causes that remain viable after eradication or treatment. Accordingly, the destructive weighted Poisson cure rate survival function is

$$S_p(y) = P(Y \geq y) = \sum_{d=0}^{\infty} P(D^w = d)[S(y)]^d,$$

where  $Y = \min(W_0, W_1, W_2, \dots, W_{D^w})$  measures the survival time based on  $D^w$  competing risks and their independent and identically distributed survival times  $S(y)$  (Rodrigues et al., 2011, 2012). We will revisit these ideas in Chapter 8.

## 1.5 Motivation, and Summary of the Book

The Poisson distribution is a classical statistical model for modeling count data and, because its probability mass function is the simplest distribution for counts, is a “fan favorite” in the statistics community. Its underlying equi-dispersion property, however, is idealistic and constraining such that real data do not typically satisfy this attribute. Over-dispersed data are often modeled via the NB distribution; however, it cannot address data under-dispersion. A distribution that can effectively model data over- or under-dispersion would be convenient for analysts because such a construct could address any exploratory analyses regarding dispersion in a direct sense without a priori knowledge of the dispersion type in the data. More broadly, any statistical methods motivated and/or derived by such a distribution would likewise allow for more flexibility and thus more proper inference. The GP distribution is a popular two-parameter distribution that

allows for over- or under-dispersion; however, its distributional complexity and inability to properly model extreme under-dispersion are troubling. Thus, there remains the need to consider an alternate count distribution that can likewise accommodate data over- or under-dispersion.

This book introduces the reader to the COM–Poisson distribution and motivates its use in developing flexible statistical methods based on its form. This two-parameter model not only serves as a flexible distribution containing the Poisson distribution as a special case but, in its ability to capture either data over- or under-dispersion, it contains (in particular) two other classical distributions as special cases (namely, the geometric and Bernoulli distributions). The COM–Poisson distribution thereby can effectively model a range of count data distributions that contain data over- or under-dispersion, from the geometric to the Poisson to the Bernoulli distributions, simply through the addition of one parameter. The statistical methods described in this reference cover a myriad of topics, including distributional theory, generalized linear modeling, control chart theory, and count processes. Chapter 2 describes the COM–Poisson distribution in further detail and discusses its associated statistical properties. It further introduces various proposed parametrizations of the model and offers added discussion regarding the normalizing constant and its approximations. Chapter 3 introduces readers to several distributional extensions of the COM–Poisson distribution and/or other distributions that otherwise associate with the COM–Poisson model. Chapter 4 highlights bivariate and multivariate count distributions that are motivated by the COM–Poisson and discusses their respective statistical properties. Chapter 5 highlights various approaches for COM–Poisson regression under the various parametrizations, including discussions regarding model formulation and estimation approach. It further discusses subsequent advancements, including considerations of observation-level dispersion, additive models, and accounting for excess zeroes and/or data clustering. Chapter 6 introduces the reader to flexible control chart developments for discrete data, including COM–Poisson-motivated generalized control charts, cumulative sum charts, and generalized exponentially weighted moving average control charts. Chapter 7 presents methods for analyzing serially dependent count data via COM–Poisson-motivated stochastic processes, as well as time series and spatio-temporal models. Finally, Chapter 8 presents COM–Poisson-motivated cure rate models that can be used to describe time-to-event data, thus demonstrating the use of this flexible model as a tool in survival analysis. All of the chapters incorporate (where possible) discussions regarding statistical computations via R, thus introducing

readers to the opportunities for data analysis via the featured R packages and their functionality.

As demonstrated in the subsequent chapters, a great deal of work has emerged where statistical methodologies are motivated by the COM–Poisson distribution. The utility of the COM–Poisson distribution, however, is not limited to these areas. Additional COM–Poisson-related works have emerged in fields, including capture–recapture and other abundance estimation methods (Anan et al., 2017; Wu et al., 2015), and disclosure limitation (Kadane et al., 2006a). Further, the COM–Poisson distribution has been employed in a variety of applications, including biology (Ridout and Besbeas, 2004), linguistics (Shmueli et al., 2005), risk analysis (Guikema and Coffelt, 2008), transportation (Lord and Guikema, 2012; Lord et al., 2008, 2010), and marketing and eCommerce (Boatwright et al., 2003; Borle et al., 2006, 2005, 2007).

Throughout this reference, much of the discussion focuses on parameter-estimation techniques associated with the various statistical method developments. These approaches are relatively thematic, falling in line with one of three approaches: maximum likelihood estimation, generalized quasi-likelihood estimation, and Bayesian estimation (Markov Chain Monte Carlo, Metropolis–Hastings, etc.). This reference will provide a high-level discussion of the respective approaches as they relate to the featured concepts; however, it assumes that the reader has a prerequisite, rudimentary knowledge of these concepts.

A common theme regarding parameter estimation in this reference centers on its dependence on statistical computation to obtain results because the COM–Poisson distribution does not have a closed form. Various optimization tools exist, however, to aid analysts with such issues. This reference focuses on R tools where existing package functions or analyst-generated codes can utilize optimization tools such as `optim`, `nlm`, or `nlmnmb` to determine parameter estimates. Details are supplied throughout the manuscript in relation to the respective statistical methodologies under discussion. Meanwhile, hypothesis testing discussions generally center on the likelihood ratio test, while other test statistics (e.g. Rao’s score test) can likewise be considered. The likelihood ratio test statistic is

$$\Lambda = \frac{\sup_{\theta \in \Theta_0} L(\theta)}{\sup_{\theta \in \Theta} L(\theta)}, \quad (1.20)$$

where  $\theta$  denotes the collection of parameters under consideration, and  $\Theta_0$  and  $\Theta$  represent the parameter space under the null hypothesis and in general, respectively; as  $n \rightarrow \infty$ ,  $-2 \ln \Lambda$  converges to a chi-squared

Table 1.2 *Levels of model support based on AIC difference values,  $\Delta_i = AIC_i - AIC_{\min}$ , for Model  $i$  (Burnham and Anderson, 2002).*

$\Delta_i$	Empirical support level for Model $i$
[0, 2]	Substantial
[4, 7]	Considerably less
(10, $\infty$ )	Essentially none

distribution. Tests about a boundary condition under the null hypothesis meanwhile produce a likelihood ratio test statistic whose asymptotic distribution is based on the equally weighted sum of a point mass and the cumulative probability of a chi-squared distribution (i.e.  $0.5 + 0.5\chi^2$ ) (Self and Liang, 1987). For example, a common interest is to test for statistically significant dispersion where the dispersion parameter may be bounded by 0; this test is introduced in Section 2.4.5 and noted throughout subsequent chapters in this reference as the implications of this test relate to the corresponding chapter content.

Discussions will also include model comparisons to demonstrate and substantiate the COM–Poisson model’s importance and flexibility. The Akaike information criterion (AIC) and the Bayesian information criterion (BIC) are two popular measures used for model comparisons, where

$$AIC = -2 \ln(L) + 2k \quad \text{and} \quad BIC = -2 \ln(L) + k \ln(n)$$

for a model’s maximized likelihood value  $L$ , number of parameters  $k$ , and sample size  $n$ . For a collection of considered models, the selected model is desired to have the minimum AIC or BIC, respectively. In particular, this reference adopts the Burnham and Anderson (2002) approach for model comparison, where models are compared via the AIC and relative performance is measured via AIC difference values  $\Delta_i = AIC_i - AIC_{\min}$ , where  $AIC_i$  denotes the AIC associated with Model  $i$ , and  $AIC_{\min}$  is the minimum AIC among the considered models. Table 1.2 supplies the levels of model support based on recommended  $\Delta_i$  ranges.