A Monotonically Convergent Newton Iteration for the Quantiles of any Unimodal Distribution, with Application to the Inverse Gaussian Distribution

Göknur Giner[1,3] Gordon K. Smyth[1,2,4]

28 May 2014
Last revised 11 July 2014

(1) Bioinformatics Division, The Walter and Eliza Hall Institute of Medical Research 1G Royal Parade, Parkville, Vic 3052, Australia. (2) Department of Mathematics and Statistics and (3) Department of Medical Biology, The University of Melbourne, Parkville, Victoria 3010, Australia. (4) Corresponding author.

Abstract

One of the most basic and commonly used numerical computations in probability and statistics is to evaluate the random deviate corresponding to any given tail probability for a given probability distribution. The deviate corresponding to a given probability is called the quantile. Quantiles usually need to be computed by numerical approximation, and the need often arises to compute quantiles for probability distributions for which reliable code is not readily available. The purpose of this article is to point out a simple but elegant result that applies to all continuous unimodal distributions. Newton's method for finding the quantiles of a continuous unimodal distribution is always monotonically convergent when started from the mode of the distribution. This provides a simple, accurate and numerically reliable method of computing quantiles for any continuous unimodal distribution, given that the cumulative distribution and probability density functions can be evaluated accurately.

The monotonic Newton iteration has been implemented in the `qinvgauss` function of the R package **statmod** to compute quantiles of inverse Gaussian distributions. The resulting function proves to be faster, more accurate and more reliable than existing functions for the same purpose, even without sophisticated optimization.

Keywords

Newton's method, unimodality, quantile function, inverse Gaussian distribution

1 Introduction

One of the most basic and commonly used numerical computations in probability and statistics is to evaluate the random deviate corresponding to any given tail probability for a given probability distribution. In mathematical terms, the left tail probability as a function of the deviate is the *cumulative distribution function* (cdf). The deviate as a function of the tail probability is the inverse cdf or *quantile function*.

Only a few very simple distributions, like the uniform or exponential distributions, have quantile functions that are available as close-form expressions. For most other distributions the

quantile function must be computed by some numerical approximation. Considerable effort has been devoted over the years to the development of high-quality numerical approximations and algorithms for computing the cdf and quantile functions for commonly used distributions. The base version of the R language for example includes cdf and quantile functions for a number of popular distributions, including uniform, gamma, beta, normal, Student's t, chi-squared and F, as part of the **stats** package maintained by the R core team. The need often arises however to compute tail probabilities and quantiles for other distributions for which such well developed algorithms are not yet available.

The inverse Gaussian (IG) distribution (Tweedie, 1957) is an example of a well known distribution for which fully reliable numerical algorithms have not been available. The IG distribution is widely used in reliability and survival analysis (Whitmore, 1975; Chhikara and Folks, 1977; Bardsley, 1980; Chhikara, 1989). It is more generally used for modeling non-negative positively skewed data because of its connections to exponential families and generalized linear models (Seshadri, 1993; Blough *and others*, 1999).

In general, the cdf of a distribution is usually somewhat easier to compute than the inverse cdf. For the IGD, for example, the cdf is actually available in closed form whereas the inverse cdf is not. The cdf is usually inverted by solving the nonlinear equation defined by the cdf and the desired tail probability. Two strategies are popular. One is to solve for the quantile using a general-purpose equation solver, such as the `uniroot` function in R. This is the approach taken by the `qinvgauss` function of the **STAR** package (Pouzat, 2012). This approach is reliable but computationally slow and requires left and right bounds for the quantile to be pre-specified. The other popular approach is to use Newton's method to solve the equation after applying an initial approximation. This approach was taken by one of the current authors when developing a `qinvgauss` function for S-PLUS (Smyth and Bagshaw, 1998). It is also the approach taken by the `qinvGauss` function of the **SuppDists** package (Wheeler, 2013). This approach is potentially fast and accurate but suffers from lack of convergence.

The purpose of this article is to point out a simple but elegant result that applies to all continuous unimodal distributions. Newton's method for finding the quantiles of a continuous unimodal distribution is always monotonically convergent when started from the mode of the distribution. This provides a simple, accurate and numerically reliable method of computing quantiles for any continuous unimodal distribution, given that the cumulative distribution and probability density functions can be evaluated accurately. By combining Newton's method with some analytic information about the shape of the distribution, this method combines the reliability of the bounded solver method with the speed and precision of Newton's method. This method also avoids the need for an initial approximation. Newton's method typically converges quickly, even when the desired quantile is in the extreme tail of the distribution and far from the mode.

The monotonic convergence method has been implemented to find quantiles of the inverse Gaussian distribution in the `qinvgauss` function of the R package **statmod** (Smyth, 2014). Despite the simplicity of the method, the function proves to be faster and more accurate than existing functions for the same task. The same idea has wide application and it likely to prove useful for other distributions.

Section 2 of this article reviews some properties of unimodal distributions, including the IG. Section 3 develops a Newton iteration for the quantiles, showing that it must always converge. Section 4 and Section 5 describe code implementations for the IG distribution. Section 6 compares the speed and accuracy of the new code to that of existing IG functions in other packages.

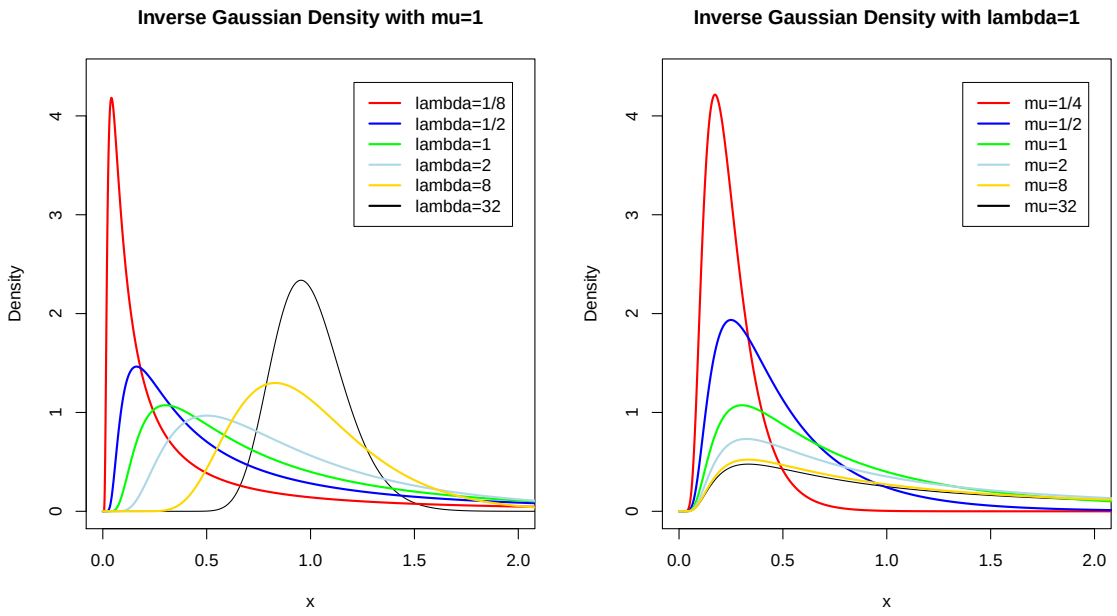

Figure 1: Probability density functions of inverse Gaussian distributions. The left panel shows densities for different λ with $\mu = 1$. The mode is always between 0 and 1. The distribution becomes more skew as λ decreases. The right panel shows densities for different μ for $\lambda = 1$. The densities are all unimodal with mode between 0 and μ.

2 Unimodal Distributions

Let $f(x)$ be a continuous probability density function (pdf) with support on (a, b), where a and b may be infinite. We say the distribution is unimodal if $f(x)$ has a unique maximum value at the mode $x = m$ and there are no other local maxima. This implies that $f(x)$ is non-decreasing for $x \leq m$ and non-increasing for $x \geq m$.

Let $F(x)$ be the cdf corresponding to $f(x)$. If the distribution is unimodal, then it follows from $F'(x) = f(x)$ that $F(x)$ has a point of inflexion at m. In that case, F is convex on (a, m) and concave on (m, b).

Many continuous distributions are unimodal. Here we concentrate on the inverse Gaussian distribution IG(μ,ϕ), with pdf

$$f(x; \mu, \phi) = \left(2\pi x^3 \phi\right)^{-1/2} \exp\left\{-\frac{(x-\mu)^2}{2\phi x \mu^2}\right\}$$

for $x > 0$, $\mu > 0$ and $\phi > 0$. The support of the distribution is the positive reals, so $a = 0$ and $b = \infty$. The mean of the distribution is $E(x) = \mu$ and the variance is $\text{var}(x) = \phi\mu^3$. Motivated by generalized linear model theory (McCullagh and Nelder, 1989), we call ϕ the dispersion parameter. Another popular parametrization of the IG distribution uses $\lambda = 1/\phi$, which we call the shape parameter.

Note that the mean μ can be viewed as a scaling parameter. If X is distributed as IG(μ,ϕ), then X/μ is also IG with mean 1 and dispersion $\phi\mu$. Note also that $\phi\mu$ is the squared coefficient of variation for the IG distribution, and write $\kappa = 3\phi\mu/2$. The IG distribution is unimodal

(Johnson and Kotz, 1970, p. 142) with mode at

$$\mu\left\{\left(1+\kappa^2\right)^{1/2}-\kappa\right\}.$$

Note that the second factor in the mode is strictly between 0 and 1, showing that the mode is always greater than zero but less than μ. Figure 1 shows the pdf of the IG distribution for various choices of μ and λ.

3 Newton's method for quantiles

Consider the problem of finding the value $x = x_\infty$ that solves $g(x) = 0$, where g is a continuously differential function. Newton's method starts with an initial estimate x_0, and approximates $g(x)$ by the tangent line at x_0 to obtain a new approximation x_1. Continuing in this way, the $(n+1)$th estimate of the root is obtained as

$$x_{n+1} = x_n - \frac{g(x_n)}{g'(x_n)},$$

where g' is the derivative of g.

Newton's method is quadratically convergent if started sufficiently close to the root. It is hard however to characterize how close the starting value needs to be to achieve convergence, and in general there is no guarantee that iteration will not diverge into a region where the function g is undefined.

There is one important case for which convergence can be guaranteed. If $x_0 > x_\infty$ and g is convex in the interval (x_∞, x_0), then the steps $-g(x_n)/g'(x_n)$ will all be negative and successive estimates x_n will converge to x_∞ from above. Similarly, if $x_0 < x_\infty$ and g is concave in (x_∞, x_0), then the steps $-g(x_n)/g'(x_n)$ will all be positive and successive estimates x_n will converge to x_∞ from below. We call this monotonic convergence because the successive estimates form a monotonic bounded sequence.

To compute the quantiles of a distribution it is necessary to solve $g(q) = F(q) - p$ for q. The solution $q = F^{-1}(p)$ is the quantile of the distribution corresponding to tail probability p. If we define $F^{-1}(0) = a$ and $F^{-1}(1) = b$, then the quantile q is uniquely defined for any unimodal distribution and for any $0 \le p \le 1$. If $0 < p < 1$, Newton's method for finding q yields the iteration

$$q_{n+1} = q_n + \frac{p - F(q_n)}{f(q_n)}. \tag{1}$$

The point of this article is to observe that Newton's method for the quantile is always monotonically convergent, for any unimodal distribution, if we choose the initial estimate q_0 to be the mode of the distribution. If the mode is above the desired quantile q, then F is convex on (q, m). If the mode is below the desired quantile q, then F is concave on (m, q). In either case, these conditions guarantee that Newton's method will converge monotonically to the required quantile.

Figure 2 illustrates the monotonic Newton iteration for finding quantiles of the IG distribution. Newton's iterations are shown for computing the $p = 0.01$ and $p = 0.99$ quantiles for different mean and dispersion values. As the figure shows, the iteration descends to the 0.01 quantile and ascends to the 0.99 quantile.

Sometimes it is desirable to compute tail probabilities on the log-scale, so as to avoid floating point underflow or to avoid subtractive cancelation errors when the probability is compared to

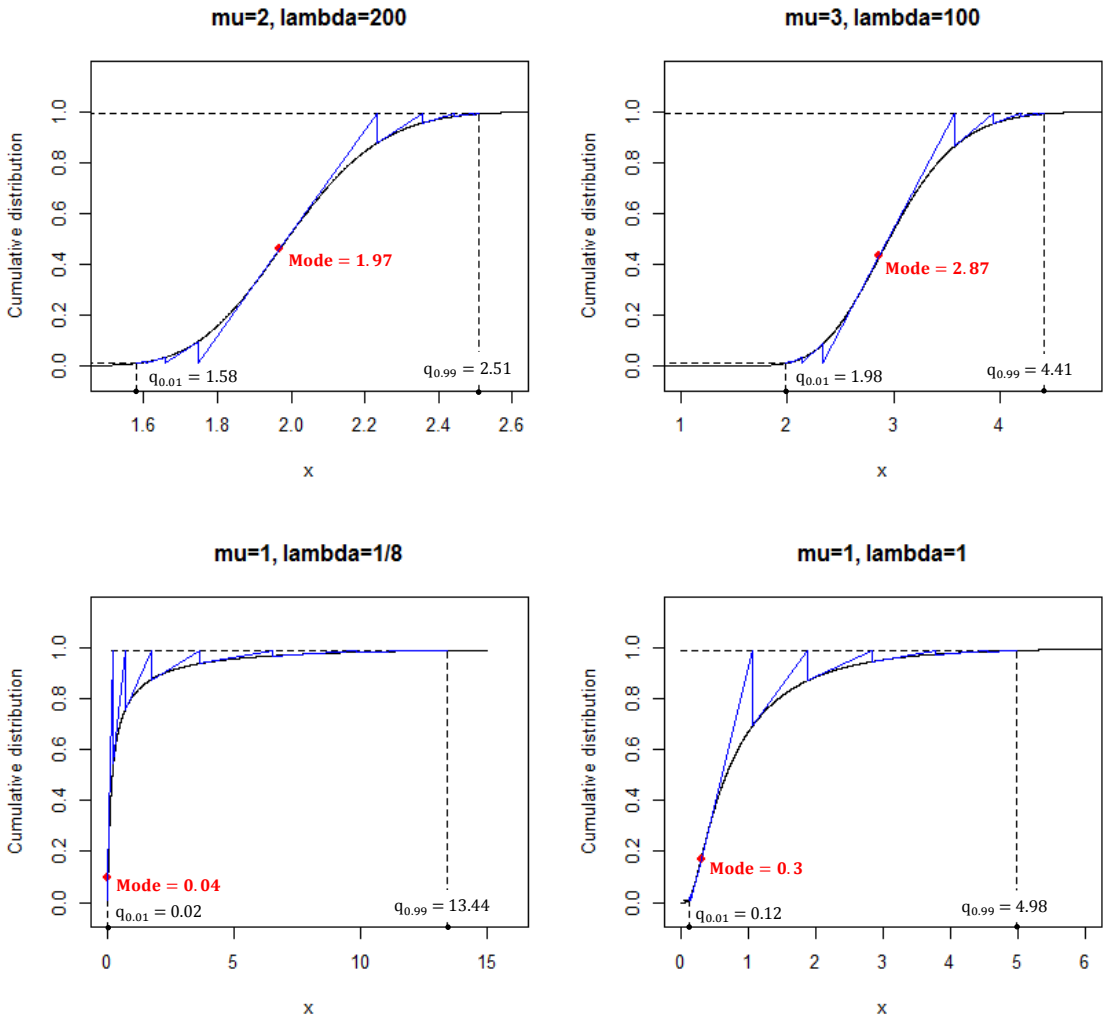

Figure 2: Monotonic Newton's method for quantiles of inverse Gaussian distributions. The cdf has a point of inflexion, marked by a red dot, at the mode of the distribution. Blue lines show the progress of the iteration for the 0.01 or 0.99 quantiles. Since the cdf is convex to the left of the mode and concave to the right, starting the iteration at the point of inflexion ensures convergence to the required quantiles.

1. If $\log p$ is supplied instead of p, and $F(q)$ can be accurately computed on the log-scale, then the $(p - F(q_n))/f(q_n)$ step of the Newton iteration can be computed on the log-scale to improve floating point accuracy. Write $d_n = \exp|\log p - \log F(q_n)|$. If log-probabilities are supplied, the Newton iteration can be redefined as

$$q_{n+1} = q_n + \exp\{\log p + \mathrm{log1p}(\mathrm{d_n}) - \log f(q_n)\}$$

when $p > F(q_n)$, or

$$q_{n+1} = q_n + \exp\{\log F(q_n) + \mathrm{log1p}(-\mathrm{d_n}) - \log f(q_n)\}$$

when $p < F(q_n)$. Here, the function `log1p(x)` evaluates $\log(1+x)$ using a Taylor series expansion to avoid subtractive cancelation in the neighborhood $x = 0$. Such a function is available in the standard distribution of R. The log-density $\log f(q_n)$ is computed by `dinvgauss` with `log=TRUE` and $\log F(q_n)$ is computed by `pinvgauss` with `log.p=TRUE`.

4 Example code for the inverse Gaussian distribution

The following is minimal R code to implement the monotonic Newton iteration to compute quantiles of the IG distribution. For simplicity, this code does not do any argument checking. The code assumes that x and q values are positive and the probabilities are strictly between 0 and 1. The first argument is assumed to be a vector, whereas the parameters μ and ϕ are scalars.

```
dinvgauss <- function(x, mu=1, phi=1)
# Probability density function of inverse Gaussian distribution
{
  d <- (-log(phi)-log(2*pi)-3*log(x))/2-((x-mu)/mu)^2/(2*phi*x)
  exp(d)
}

pinvgauss <- function(q, mu=1, phi=1)
# Cumulative distribution function of inverse Gaussian distribution
{
  q <- q/mu
  phi <- phi*mu
  pq <- sqrt(phi*q)
  pnorm((q-1)/pq) + exp( 2/phi + pnorm(-(q+1)/pq,log.p=TRUE) )
}

qinvgauss  <- function(p, mu=1, phi=1, maxit=50L, tol=1e-5)
# Quantiles of the inverse Gaussian distribution
{
  n <- length(p)
  phi <- phi*mu

# Start iteration at mode (with mu=1)
  kappa <- 1.5*phi
  q <- rep_len(sqrt(1+kappa^2)-kappa,n)

# Newton iteration
  iter <- 0
```

```
  i <- rep_len(TRUE,n)
  while(any(i)) {
    iter <- iter+1
    if(iter > maxit) {
      warning("max iterations exceeded")
      break
    }
    dq <- (p[i] - pinvgauss(q[i], phi=phi)) / dinvgauss(q[i], phi=phi)
    q[i] <- q[i] + dq
    i[i] <- (abs(dq) > tol)
  }
  q*mu
}
```

5 Implementation in the statmod package

The previous section gave minimal example code. Complete full-featured functions `dinvgauss`, `pinvgauss`, `qinvgauss` and `rinvgauss` are implemented in the **statmod** package (Smyth, 2014). The **statmod** functions allow parameter vectors and include careful checking for boundary values, invalid or `NA` input arguments. They also provide the ability to work with log-probabilities. In general, they obey all the conventions obeyed by the probability functions in the **stats** package.

Variability can be specified either by way of a dispersion (ϕ) or shape (λ) parameter:

```
> args(qinvgauss)
function (p, mean = 1, shape = NULL, dispersion = 1, lower.tail = TRUE, log.p = FALSE,
    maxit = 50L, tol = 1e-05, trace = FALSE)
```

Boundary or impossible arguments are detected:

```
> qinvgauss(c(0,0.5,1,2,NA))
[1] 0.0000000 0.6758413       Inf        NA        NA
```

as are invalid parameter arguments

```
> qinvgauss(0.5, mean=c(0,1,2))
[1]        NA 0.6758413 1.0284598
```

Attributes of input arguments are also preserved on output:

```
> p
         X1        X2
A 0.6001382 0.3434530
B 0.4918840 0.4987219
> qinvgauss(p)
         X1        X2
A 0.8485613 0.4759428
B 0.6637441 0.6739203
```

6 Comparison with existing inverse Gaussian functions

Functions for quantiles of IG distributions can also be found in the **SuppDists** (Wheeler, 2013), **STAR** (Pouzat, 2012) and **predfinitepop** (Ovando *and others*, 2014) packages. Here we compare

Table 1: Speed and accuracy of `qinvgauss()` functions in different packages. Second column gives elapsed time to compute a million quantiles. Third and fourth columns are median and maximum absolute errors.

Package	Time (sec)	Median Error	Max Error
Section 4 code	6.42	1.267e-13	3.901e-11
statmod	6.77	1.267e-13	3.901e-11
SuppDists	14.73	2.774e-10	1.029e-08
predfinitepop	195.44	8.388e-08	1.179e-06
STAR	326.51	1.166e-08	6.650e-06

the speed and accuracy of the **statmod** `qinvgauss` function with the corresponding functions in the other packages.

Speed was determined by generating `p` as a vector of a million random uniform deviates, and running the `qinvgauss` or `qinvGauss` functions on `p` with mean and dispersion both equal to one. Precision was determined by comparing the probability vector `p` with the values obtained by passing the probabilities through `qinvgauss` and `pinvgauss`. `qinvgauss` and `pinvgauss` are inverse functions, so the final probabilities should be equal in principle to the original values. The error is measured by the absolute deviations between the original and processed probability vectors. Table 1 shows running times in seconds, and the median and maximum errors. The **statmod** `qinvgauss` function runs 2–3 times faster and produces three extra decimal places of accuracy than the nearest competitor, which is **SuppDists**. The speed improvement is achieved despite the fact that the **SuppDists** functions are coded in the C programming language whereas **statmod** is pure R. Coding the **statmod** function in C would presumably increase the speed advantage further. Precision is given for the default settings of `statmod::qinvgauss`; even greater accuracy could be achieved if desired by decreasing the `tol` argument. The simple code shown in Section 4 is very slightly faster than the **statmod** function because it doesn't do any argument checking. Timings here are for a Windows laptop with a 2.2GHz Intel i7 processor running 64-bit R 3.1.0.

Another critical consideration is reliability. The **SuppDists** function `qinvGauss` fails for some parameter values for reasons that are probably to do with lack of convergence. For example:

```
> qinvGauss(0.00013,1,3)
Error in qinvGauss(0.00013, 1, 3) :
Iteration limit exceeded in NewtonRoot()
```

By contrast, the **statmod** function `qinvgauss` runs successfully for all parameter values because divergence of the algorithm is impossible.

```
> qinvgauss(0.00013,1,3)
[1] 0.1503976
```

7 Conclusions

Newton's method is a fast and accurate method of computing quantiles if convergence of the iteration can be guaranteed. Such a guarantee is available for unimodal distributions if the starting value is chosen to be the mode of the distribution. In such a case, Newton's method

converges to the required quantile without any divergence or overstepping.

This method is applicable to any continuous distribution that can be transformed to be unimodal. For example, the beta and F distributions are not themselves unimodal for all parameter values, but become unimodal after logit and log transformations respectively.

It might be surprising that we recommend starting the iteration from the same value regardless of the quantile required. Intuitively, a starting value that is closer to the required quantile might have been expected to be better. However using an initial approximation runs the risk of divergence, and convergence of Newton's method from the mode is so rapid that the potential advantage of a closer initial approximation is minimized.

The monotonic Newton iteration has been implemented in the **statmod** package to compute quantiles of IG distributions. The resulting function is faster, more accurate and more reliable than existing functions, even without sophisticated optimization.

References

Bardsley, W. (1980). Note on the use of the inverse gaussian distribution for wind energy applications. *Journal of Applied Meteorology* **19**(9), 1126–1130.

Blough, D. K, Madden, C. W and Hornbrook, M. C. (1999). Modeling risk using generalized linear models. *Journal of Health Economics* **18**(2), 153–171.

Chhikara, R and Folks, J. (1977). The inverse gaussian distribution as a lifetime model. *Technometrics* **19**(4), 461–468.

Chhikara, R. S. (1989). *The Inverse Gaussian Distribution*. New York: Marcel Dekker.

Johnson, N. L and Kotz, S. (1970). *Continuous Univariate Distributions, Vol. 1*. New York: Wiley-Interscience.

McCullagh, P and Nelder, J. A. (1989). *Generalized Linear Models*, 2nd edition. Boca Raton, Florida: Chapman & Hall/CRC.

Ovando, J. C. M, Guzman, S. I. O and Rodriguez, A. R. (2014). **predfinitepop***: Predictive Inference on Totals and Averages of Finite Populations Segmented in Planned and Unplanned Domains*. R package version 1.0.

Pouzat, C. (2012). **STAR***: Spike Train Analysis with R*. R package version 0.3-7.

Seshadri, V. (1993). *The Inverse Gaussian Distribution: A Case Study in Exponential Families*. Oxford: Clarendon Press.

Smyth, G. K. (2014). **statmod***: Statistical Modeling*. R package version 1.4.21.

Smyth, G. K and Bagshaw, P. (1998). *invgauss: Inverse Gaussian Distribution*. Statistical Modelling in S-PLUS: A Library of Functions for S-PLUS. Last Modified 23 December 1998.

Tweedie, M. C. (1957). Statistical properties of inverse Gaussian distributions i. *The Annals of Mathematical Statistics* **28**(2), 362–377.

Wheeler, B. (2013). **SuppDists***: Supplementary Distributions*. R package version 1.1-9.1.

Whitmore, G. (1975). The inverse gaussian distribution as a model of hospital stay. *Health Services Research* **10**(3), 297.