ST417 Introduction to Bayesian Modelling

Conjugate Modelling (Poisson-Gamma)

Slides based on the source provided courtesy of Prof. D.Draper, UCSC

Integer-Valued Outcomes

Case Study: *Hospital length of stay for birth of premature babies.* What follows is a real data set, obtained as a random sample of $n = 14$ women who came to a hospital in Santa Monica, CA, in 1988 to **give birth to premature babies**.

One (**integer-valued**) outcome of interest was

$y =$ **length of hospital stay** (LOS).

Here's a preliminary look at the data in `R`:

```
> y
[1] 1 2 1 1 1 2 2 4 3 6 2 1 3 0
> sort( y )
[1] 0 1 1 1 1 1 2 2 2 2 3 3 4 6
```

Integer-Valued Outcomes (continued)

```
> table( y )
0 1 2 3 4 6
1 5 4 2 1 1
> stem( y, scale = 2 )
The decimal point is at the |
  0 | 0
  1 | 00000
  2 | 0000
  3 | 00
  4 | 0
  5 |
  6 | 0
> mean( y )
[1] 2.071429
> sd( y )
[1] 1.54244
> var( y )
[1] 2.37912
```

Poisson Modelling

One possible model for non-negative integer-valued outcomes is the

Poisson distribution

$$P(Y_i = y_i|\lambda) = \left\{ \begin{array}{cc} \frac{\lambda^{y_i} e^{-\lambda}}{y_i!} & \text{for } y_i = 0, 1, \ldots \\ 0 & \text{otherwise} \end{array} \right\}, \tag{1}$$

for some $\lambda > 0$.

As usual `Maple` can be used to work out the **mean** and **variance** of this distribution:

```
> assume( lambda > 0 );
> p := ( y, lambda ) -> lambda^y * exp( - lambda ) / y!;
                                              y
                                         lambda  exp(-lambda)
                    p := (y, lambda) -> --------------------
                                              y!
> simplify( sum( p( y, lambda ), y = 0 .. infinity ) );
                                         1
> simplify( sum( y * p( y, lambda ), y = 0 .. infinity ) );
                                      lambda~
```

Informal Model-Checking

```
> simplify( sum( ( y - lambda )^2 * p( y, lambda ),
    y = 0 .. infinity ) );
                                        lambda~
```

Thus if $(Y|\lambda) \sim \text{Poisson}(\lambda), E(Y) = V(Y) = \lambda$, which people sometimes express by saying that the **variance-to-mean ratio** (VTMR) for the Poisson is 1.

`R` can be used to check informally whether the Poisson is a **good fit** to the LOS data:

```
> dpois( 0:7, mean( y ) )
[1] 0.126005645 0.261011693 0.270333539 0.186658872 0.096662630
[6] 0.040045947 0.013825386 0.004091186
> print( n <- length( y ) )
[1] 14
> table( y ) / n
         0          1          2          3          4          6
0.07142857 0.35714286 0.28571429 0.14285714 0.07142857 0.07142857
```

Informal Model-Checking (continued)

```
> cbind( c( dpois( 0:6, mean( y ) ),
    1 - sum( dpois( 0:6, mean( y ) ) ) ),
    apply( outer( y, 0:7, '==' ), 2, sum ) / n )

             [,1]       [,2]
[1,] 0.126005645 0.07142857
[2,] 0.261011693 0.35714286
[3,] 0.270333539 0.28571429
[4,] 0.186658872 0.14285714
[5,] 0.096662630 0.07142857
[6,] 0.040045947 0.00000000
[7,] 0.013825386 0.07142857
[8,] 0.005456286 0.00000000
```

The second column in the above table records the values of the **Poisson probabilities** for $\lambda = 2.07$, the mean of the y_i, and the third column is the **empirical relative frequencies**; informally the fit is reasonably good.

Another informal check comes from the fact that the sample mean and variance are 2.07 and $1.542^2 \doteq 2.38$, which are reasonably close.

Does Exchangeability → Sampling Distribution Here? (No.)

Exchangeability. As with the AMI mortality case study, before the data arrive I recognize that my uncertainty about the Y_i is exchangeable, and you would expect from a generalization of the binary-outcomes version of de Finetti's Theorem that the structure of a **plausible Bayesian model** for the data would then be

$$\begin{array}{rcll} \theta & \sim & p(\theta) & \text{(prior)} \\ (Y_i|\theta) & \overset{\text{IID}}{\sim} & F(\theta) & \text{(likelihood)}, \end{array} \tag{2}$$

where θ is some parameter (vector) and $F(\theta)$ is some **parametric family of distributions** on the non-negative integers indexed by θ.

Thus, in view of the preliminary examination of the data above, a **plausible Bayesian model** for these data is

$$\begin{array}{rcll} \lambda & \sim & p(\lambda) & \text{(prior)} \\ (Y_i|\lambda) & \overset{\text{IID}}{\sim} & \text{Poisson}(\lambda) & \text{(likelihood)}, \end{array} \tag{3}$$

where λ is a **positive real number**.

Model Uncertainty

NB (1) This approach to model-building involves a form of **cheating**, because we've **used the data twice**: once to choose the model, and again to draw conclusions conditional on the chosen model.

The result in general can be a failure to **assess** and **propagate model uncertainty** (e.g., Draper 1995).

(2) **Frequentist** modeling often employs this **same kind of cheating** in specifying the likelihood function.

(3) There are two Bayesian ways out of this dilemma: **cross-validation** and **Bayesian non-parametric/semi-parametric** methods.

The **latter** is beyond the scope of this course; I'll give examples of the **former** later.

To get more practice with Bayesian calculations I'm going to **ignore the model uncertainty problem for now** and pretend that somehow we knew that the Poisson was a good choice.

The likelihood function in model (3) is

Poisson Likelihood

$$
\begin{aligned}
l(\lambda|y) &= c\, p_{Y_1,\ldots,Y_n}(y_1,\ldots,y_n|\lambda) \\
&= c \prod_{i=1}^{n} p_{Y_i}(y_i|\lambda) \qquad (4) \\
&= c \prod_{i=1}^{n} \frac{\lambda^{y_i} e^{-\lambda}}{y_i!} = c\,\lambda^{s}\, e^{-n\lambda},
\end{aligned}
$$

where $y = (y_1, \ldots, y_n)$ and $s = \sum_{i=1}^{n} y_i$; here $\left(\prod_{i=1}^{n} y_i!\right)^{-1}$ can be **absorbed** into the generic positive c because it doesn't involve λ.

Thus (as was true in the Bernoulli model) $s = \sum_{i=1}^{n} y_i$ is **sufficient** for λ in the Poisson model, and we can write $l(\lambda|s)$ instead of $l(\lambda|y)$ if we want.

If a **conjugate** prior $p(\lambda)$ for λ exists it must be such that the product $p(\lambda)\, l(\lambda|s)$ has the same mathematical form as $p(\lambda)$.

Examination of (4) reveals that the same trick works here as with Bernoulli data, namely **taking the prior to be of the same form as the likelihood**:

$$
p(\lambda) = c\,\lambda^{\alpha-1} e^{-\beta\lambda} \qquad (5)
$$

The Gamma Distribution

for some $\alpha > 0, \beta > 0$ — this is the **Gamma distribution** $\lambda \sim \Gamma(\alpha, \beta)$ for $\lambda > 0$ (see GCSR, Appendix A).

As usual `Maple` can work out the **normalizing constant**:

```
> assume( lambda > 0, alpha > 0, beta > 0 );
> p1 := ( lambda, alpha, beta ) -> lambda^( alpha - 1 ) *
    exp( - beta * lambda );
                                          (alpha - 1)
     p1 := (lambda, alpha, beta) -> lambda            exp(-beta lambda)
> simplify( integrate( p1( lambda, alpha, beta ),
    lambda = 0 .. infinity ) );
                              (-alpha~)
                    beta~            GAMMA(alpha~)
```

Thus $c^{-1} = \beta^{-\alpha}\,\Gamma(\alpha)$ and the **proper definition** of the Gamma distribution is

$$\text{If } \lambda \sim \Gamma(\alpha, \beta) \text{ then } p(\lambda) = \frac{\beta^{\alpha}}{\Gamma(\alpha)}\,\lambda^{\alpha-1}\,e^{-\beta\,\lambda} \tag{6}$$

for $\alpha > 0, \beta > 0$.

The R Implementation of the Gamma Distribution

As usual R can also be used to explore the behavior of this family of distributions **as a function of its inputs** α and β, but you need to watch out — there are **two different parameterizations** in common usage:

```
> help( dgamma )
GammaDist                package:stats                R Documentation
The Gamma Distribution
Description:
     Density, distribution function, quantile function and random
     generation for the Gamma distribution with parameters 'shape' and
     'scale'.
Usage:
     dgamma(x, shape, rate = 1, scale = 1/rate, log = FALSE)
     pgamma(q, shape, rate = 1, scale = 1/rate, lower.tail = TRUE,
            log.p = FALSE)
     qgamma(p, shape, rate = 1, scale = 1/rate, lower.tail = TRUE,
            log.p = FALSE)
     rgamma(n, shape, rate = 1, scale = 1/rate)
Arguments:
    x, q: vector of quantiles.
```

The Gamma Distribution in R (continued)

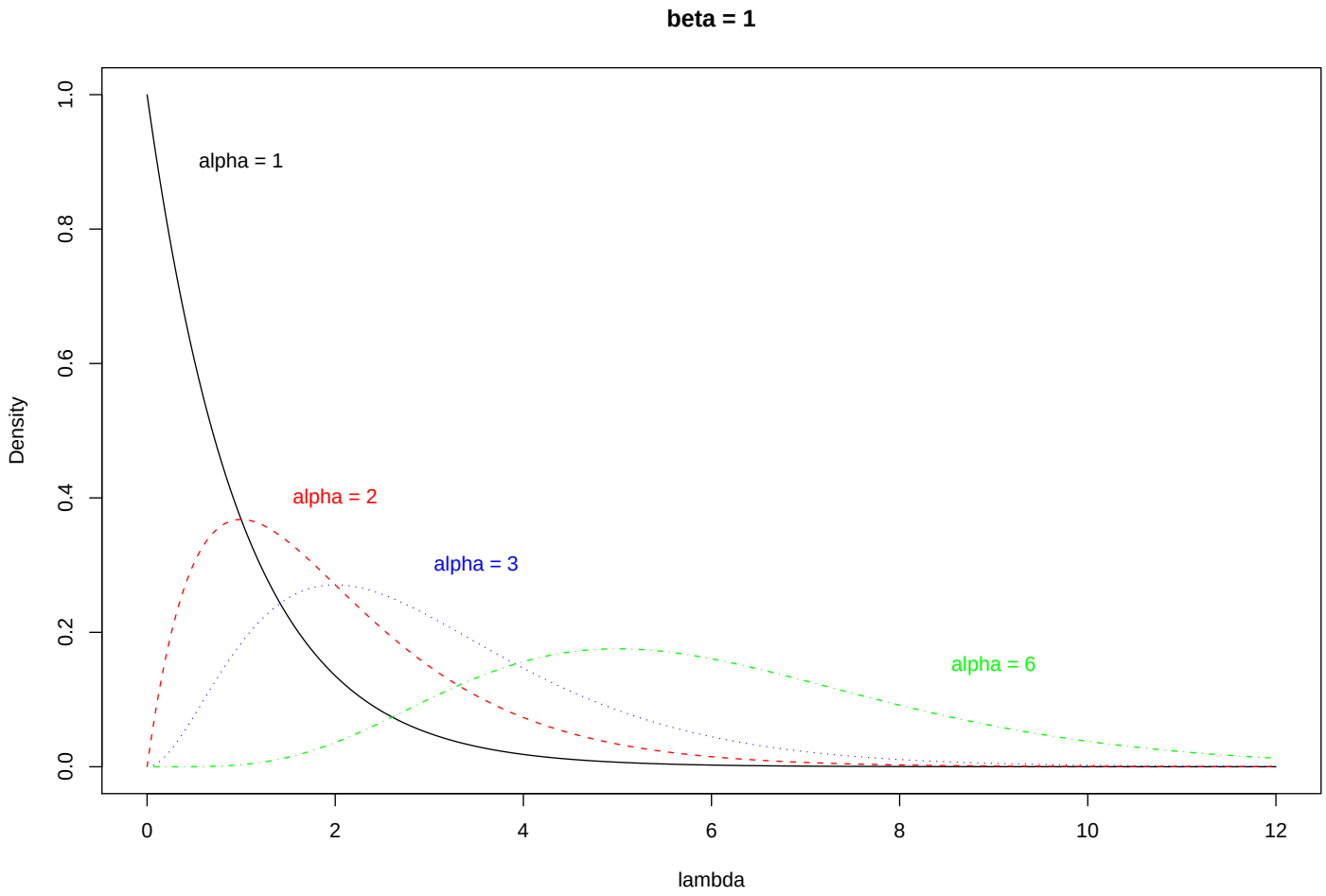
```
       p: vector of probabilities.
       n: number of observations. If 'length(n) > 1', the length is
          taken to be the number required.
    rate: an alternative way to specify the scale.
shape, scale: shape and scale parameters.  Must be positive, 'scale'
          strictly.
log, log.p: logical; if 'TRUE', probabilities/densities p are returned
          as log(p).
lower.tail: logical; if TRUE (default), probabilities are P[X <= x],
          otherwise, P[X > x].
Details:
     If 'scale' is omitted, it assumes the default value of '1'.
     The Gamma distribution with parameters 'shape' = a and 'scale' = s
     has density
                f(x)= 1/(s^a Gamma(a)) x^(a-1) e^-(x/s)
     for x >= 0, a > 0 and s > 0. (Here Gamma(a) is the function
     implemented by R's 'gamma()' and defined in its help.  Note that
     a=0 corresponds to the trivial distribution with all mass at point
     0.)
```

The Gamma Distribution in R (continued)

The quantity in R corresponding to our α is evidently **shape**, but notice that what R calls **scale** is $\frac{1}{\beta}$ for us; the name for β in R is **rate**, the reciprocal of **scale**:

```
lambda.grid.1 <- seq( 0, 12, length = 500 )
postscript( "gamma-beta-equals-1.ps" )
plot( lambda.grid.1, dgamma( lambda.grid.1, shape = 1, rate = 1 ),
  xlab = 'lambda', ylab = 'Density', type = 'l', main = 'beta = 1' )
text( 1, 0.9, 'alpha = 1' )
lines( lambda.grid.1, dgamma( lambda.grid.1, shape = 2, rate = 1 ),
  lty = 2, col = 'red' )
text( 2, 0.4, 'alpha = 2', col = 'red' )
lines( lambda.grid.1, dgamma( lambda.grid.1, shape = 3, rate = 1 ),
  lty = 3, col = 'blue' )
text( 3.5, 0.3, 'alpha = 3', col = 'blue' )
lines( lambda.grid.1, dgamma( lambda.grid.1, shape = 6, rate = 1 ),
  lty = 4, col = 'green' )
text( 9, 0.15, 'alpha = 6', col = 'green' )
dev.off( )
```

α Controls Shape in the Gamma Family



The `R` name for α is a **good choice**: α evidently controls the **shape** of the Gamma family.

What **distributional shape** does the Gamma approach as $\alpha \to \infty$?

The Exponential Distribution

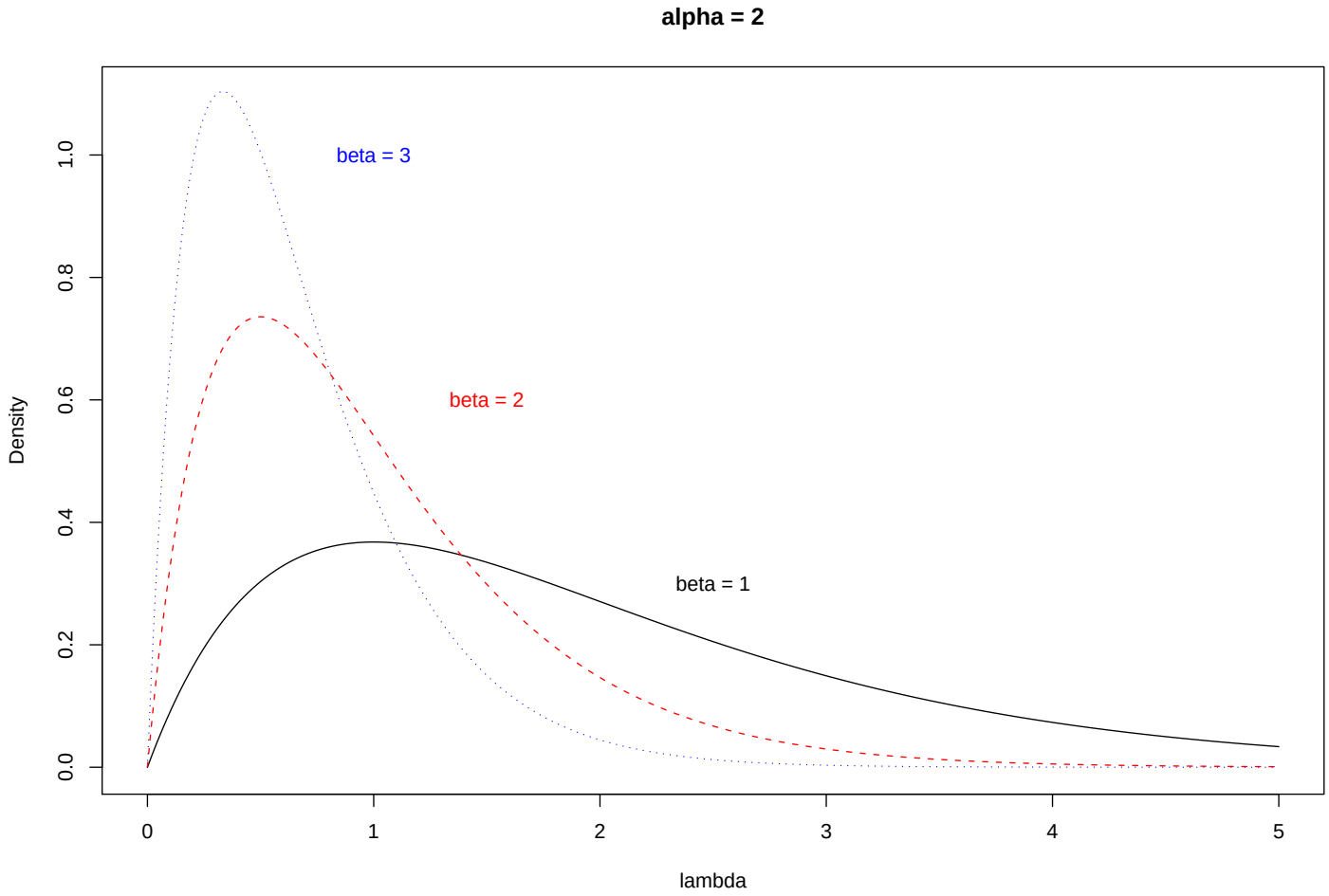
When $\alpha = 1$ the Gamma distributions have a special form that you'll probably recognize — they're the **exponential** distributions $\mathcal{E}(\beta)$: for $\beta > 0$

$$\text{If } \lambda \sim \mathcal{E}(\beta) \text{ then } p(\lambda) = \left\{ \begin{array}{cc} \beta\, e^{-\beta\,\lambda} & \text{for } \lambda > 0 \\ 0 & \text{otherwise} \end{array} \right\}. \tag{7}$$

What about the **effect** of β, or its reciprocal, on the distribution?

```
lambda.grid.2 <- seq( 0, 5, length = 500 )
plot( lambda.grid.2, dgamma( lambda.grid.2, shape = 2, rate = 1 ),
  xlab = 'lambda', ylab = 'Density', type = 'l', main = 'alpha = 2',
  ylim = c( 0, 1.1 ) )
text( 2.5, 0.3, 'beta = 1' )
lines( lambda.grid.2, dgamma( lambda.grid.2, shape = 2, rate = 2 ),
  lty = 2, col = 'red' )
text( 1.5, 0.6, 'beta = 2', col = 'red' )
lines( lambda.grid.2, dgamma( lambda.grid.2, shape = 2, rate = 3 ),
  lty = 3, col = 'blue' )
text( 1, 1, 'beta = 3', col = 'blue' )
```

β (and $\frac{1}{\beta}$) Control the Spread



In the Gamma family the parameter β controls the **spread** of the distribution, but $\frac{1}{\beta}$ controls the **scale**, in the sense that as $\frac{1}{\beta}$ increases the distribution becomes **more spread out**.

$\frac{1}{\beta}$ Is a Scale Parameter For the Gamma Distribution

Definition Given a random quantity y whose density $p(y|\sigma)$ depends on a parameter $\sigma > 0$, if it's possible to express $p(y|\sigma)$ in the form $\frac{1}{\sigma} f(\frac{y}{\sigma})$, where $f(\cdot)$ is a function which does not depend on y or σ, then σ is called a **scale parameter** for the parametric family p.

Letting $f(t) = e^{-t}$ and taking $\sigma = \frac{1}{\beta}$, You can see that the Gamma family can be expressed in this way, so $\frac{1}{\beta}$ is a **scale parameter** for the Gamma distribution.

As usual `Maple` can also work out the **mean** and **variance** of this family:

```
> p := ( lambda, alpha, beta ) -> beta^alpha * lambda^( alpha - 1 ) *
    exp( - beta * lambda ) / GAMMA( alpha );
                                    alpha       (alpha - 1)
                                beta      lambda             exp(-beta lambda)
p := (lambda, alpha, beta) -> ----------------------------------------
                                                GAMMA(alpha)
> simplify( integrate( p( lambda, alpha, beta ),
    lambda = 0 .. infinity ) );
                                        1
```

Conjugate Updating With the Poisson Likelihood

```
> simplify( integrate( lambda * p( lambda, alpha, beta ),
    lambda = 0 .. infinity ) );
                                   alpha~
                                   ------
                                   beta~
> simplify( integrate( ( lambda - alpha / beta )^2 *
    p( lambda, alpha, beta ), lambda = 0 .. infinity ) );
                                   alpha~
                                   ------
                                        2
                                   beta~
```

Thus if $\lambda \sim \Gamma(\alpha, \beta)$ then $E(\lambda) = \frac{\alpha}{\beta}$ and $V(\lambda) = \frac{\alpha}{\beta^2}$, and

conjugate updating is now **straightforward**: with $y = (y_1, \ldots, y_n)$ and $s = \sum_{i=1}^{n} y_i$, by Bayes's Theorem

$$\begin{aligned} p(\lambda|y) &= c\, p(\lambda)\, l(\lambda|y) \\ &= c \left(c\, \lambda^{\alpha-1}\, e^{-\beta\lambda} \right) \left(c\, \lambda^{s}\, e^{-n\lambda} \right) \\ &= c\, \lambda^{(\alpha+s)-1}\, e^{-(\beta+n)\lambda}, \end{aligned} \tag{8}$$

and the **resulting distribution** is just $\Gamma(\alpha + s, \beta + n)$.

Conjugate Poisson Analysis

This can be **summarized** as follows:

$$\left\{ \begin{array}{c} (\lambda|\alpha,\beta) \sim \Gamma(\alpha,\beta) \\ (Y_i|\lambda) \overset{\text{IID}}{\sim} \text{Poisson}(\lambda), \\ i = 1,\ldots,n \end{array} \right\} \rightarrow (\lambda|s) \sim \Gamma(\alpha^*,\beta^*), \tag{9}$$

where $(\alpha^*,\beta^*) = (\alpha+s,\beta+n)$ and $s = \sum_{i=1}^{n} y_i$ is a **sufficient statistic** for λ in this model.

The posterior mean of λ here is evidently $\frac{\alpha^*}{\beta^*} = \frac{\alpha+s}{\beta+n}$, and the prior and data means are $\frac{\alpha}{\beta}$ and $\bar{y} = \frac{s}{n}$, so (as was the case in the Bernoulli model) the posterior mean can be written as a **weighted average** of the prior and data means:

$$\frac{\alpha+s}{\beta+n} = \left(\frac{\beta}{\beta+n}\right)\left(\frac{\alpha}{\beta}\right) + \left(\frac{n}{\beta+n}\right)\left(\frac{s}{n}\right). \tag{10}$$

Thus the **prior sample size** n_0 in this model is just β (which makes sense given that $\frac{1}{\beta}$ is the scale parameter for the Gamma distribution), and the prior acts like a **dataset** consisting of β observations with mean $\frac{\alpha}{\beta}$.

The $\Gamma(\epsilon, \epsilon)$ Prior

LOS data analysis. Suppose that, before the current data set is scheduled to arrive, I know **little** about the mean length of hospital stay of women giving birth to premature babies.

Then for my prior on λ I'd like to specify a member of the $\Gamma(\alpha, \beta)$ family which is relatively **flat in the region in which the likelihood function is appreciable**.

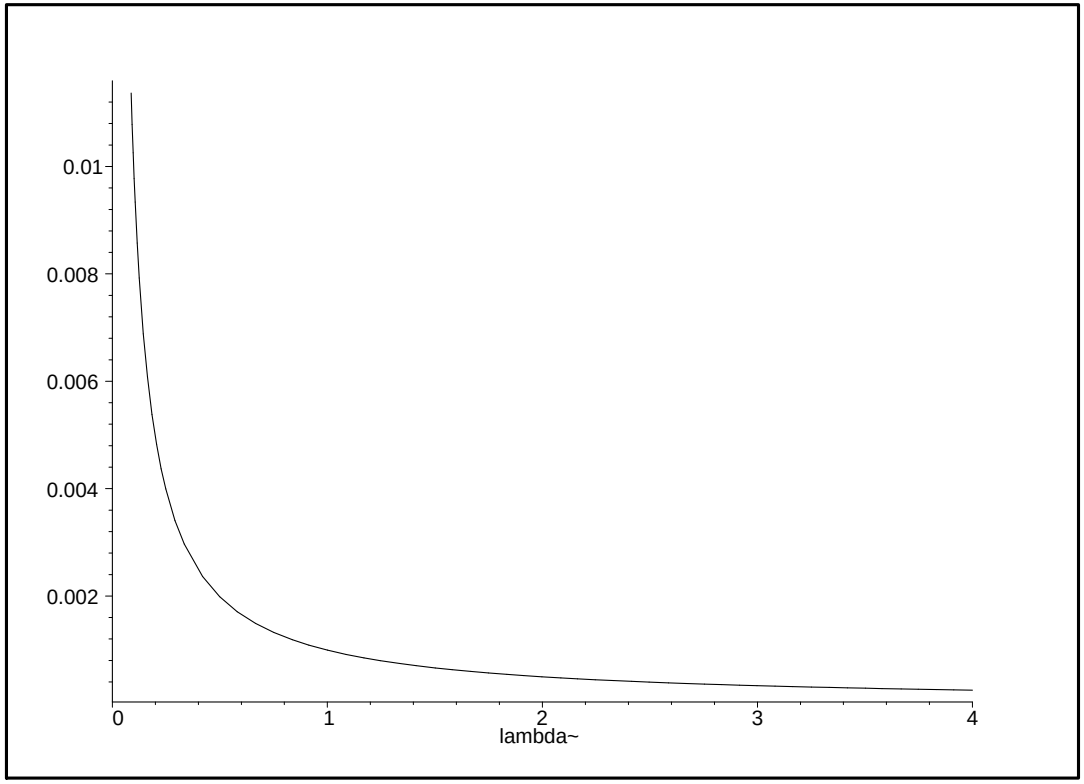
A **convenient** and **fairly all-purpose default choice** of this type is $\Gamma(\epsilon, \epsilon)$ for some small ϵ like 0.001.

When used as a prior this distribution has **prior sample size** ϵ; it also has mean 1, but that usually doesn't matter when ϵ is **tiny**.

```
> plot( p( lambda, 0.001, 0.001 ), lambda = 0 .. 4, color = black );
```

As the **graph** on the next page shows, this **distribution** is **rather flat** over the **entire region** $(1, \infty)$; it has an **unpleasant spike near 0**, but this is only a **potential problem** when the **likelihood density** is **concentrated near 0** (in that case You may be inserting **stronger prior information** than You **intended**).

The Empirical Rule



With the LOS data $s = 29$ and $n = 14$, so the **likelihood** for λ is like a $\Gamma(30, 14)$ density, which has mean $\frac{30}{14} \doteq 2.14$ and SD $\sqrt{\frac{30}{14^2}} \doteq 0.39$.

Thus by the **Empirical Rule** the likelihood is appreciable in the range $(\text{mean} \pm 3\,\text{SD}) \doteq (2.14 \pm 1.17) \doteq (1.0, 3.3)$, and you can see from the plot above that the prior is indeed **relatively flat** in this region.

LOS Data Analysis (continued)

From the **Bayesian updating** in (9), with a $\Gamma(0.001, 0.001)$ prior the **posterior** is $\Gamma(29.001, 14.001)$.

It's useful, in summarizing the **updating** from prior through likelihood to posterior, to make a table that records measures of **center** and **spread** at each point along the way.

For example, the $\Gamma(0.001, 0.001)$ **prior**, when regarded (as usual) as a **density** for λ, has mean 1.000 and SD $\sqrt{1000} \doteq 31.6$ (i.e., informally, as far as we're concerned, before the data arrive λ could be **anywhere between 0 and (say) 100**).

And the $\Gamma(29.001, 14.001)$ **posterior** has mean $\frac{29.001}{14.001} \doteq 2.071$ and SD $\sqrt{\frac{29.001}{14.001^2}} \doteq 0.385$, so after the data have arrived we know **quite a bit more than before**.

There are two main ways to summarize the **likelihood** — Fisher's approach based on **maximizing** it, and the Bayesian approach based on regarding it as a density and **integrating** over it — and it's instructive to compute them both and **compare**.

Likelihood Calculations

The **likelihood-integrating** approach (which, at least in one-parameter problems, is essentially equivalent to Fisher's (1935) attempt at **fiducial** inference) treats the $\Gamma(30, 14)$ likelihood as a density for λ, with mean $\frac{30}{14} \doteq 2.143$ and SD $\sqrt{\frac{30}{14^2}} \doteq 0.391$.

As for the **likelihood-maximizing** approach, from (4) the log likelihood function is

$$ll(\lambda|y) = ll(\lambda|s) = \log\left(c\,\lambda^s e^{-n\lambda}\right) = c + s\log\lambda - n\lambda, \tag{11}$$

and this is **maximized** as usual (check that it's the max) by setting the **derivative** equal to 0 and solving:

$$\frac{\partial}{\partial\lambda} ll(\lambda|s) = \frac{s}{\lambda} - n = 0 \quad \text{iff} \quad \lambda = \hat{\lambda}_{\text{MLE}} = \frac{s}{n} = \bar{y}. \tag{12}$$

Since the MLE $\hat{\lambda}_{\text{MLE}}$ turns out to be our old friend the **sample mean** $\bar{y}$, you might be tempted to conclude immediately that $\widehat{SE}\left(\hat{\lambda}_{\text{MLE}}\right) = \frac{\hat{\sigma}}{\sqrt{n}}$, where $\hat{\sigma} = 1.54$ is the sample SD, and indeed it's true in repeated sampling that $V\left(\bar{Y}\right) = \frac{V(Y_1)}{n}$; but the **Poisson distribution** has variance $V(Y_1) = \lambda$, so that

Calibrating the MLE

$\sqrt{V\left(\bar{Y}\right)} = \frac{\sqrt{\lambda}}{\sqrt{n}}$, and there's no guarantee in the Poisson model that the best way to estimate $\sqrt{\lambda}$ in this standard error calculation is with the sample SD $\hat{\sigma}$ (in fact we have a **strong hint** from the above MLE calculation that the sample variance is **irrelevant** to the estimation of λ in the Poisson model, since the sample variance does not arise in the Poisson likelihood).

The right (large-sample) likelihood-based **standard error** for $\hat{\lambda}_{\text{MLE}}$, using the **Fisher information** logic we examined earlier, is obtained from the following calculation:

$$\begin{aligned}
\frac{\partial^2}{\partial \lambda^2} \log l(\lambda|y) &= -\frac{s}{\lambda^2}, \quad \text{so} \qquad (13)\\
\hat{I}\left(\hat{\lambda}_{\text{MLE}}\right) &= \left[-\frac{\partial^2}{\partial \lambda^2} \log l(\lambda|y)\right]_{\lambda=\hat{\lambda}_{\text{MLE}}} \\
&= \left(\frac{s}{\lambda^2}\right)_{\lambda=\bar{y}} = \frac{s}{\bar{y}^2} = \frac{n}{\bar{y}}, \quad \text{and} \\
\hat{V}\left(\hat{\lambda}_{\text{MLE}}\right) &= \hat{I}^{-1}\left(\hat{\lambda}_{\text{MLE}}\right) = \frac{\bar{y}}{n} = \frac{\hat{\lambda}_{\text{MLE}}}{n}.
\end{aligned}$$

Prior-Likelihood-Posterior Summaries

So in this case study Fisher's **likelihood-maximizing** approach would **estimate** λ by $\hat{\lambda}_{\text{MLE}} = \bar{y} = \frac{29}{14} \doteq 2.071$, with a **give-or-take** of $\widehat{SE}\left(\hat{\lambda}_{\text{MLE}}\right) = \frac{\sqrt{\hat{\lambda}_{\text{MLE}}}}{\sqrt{n}} = \frac{1.44}{\sqrt{14}} \doteq 0.385$.

All of this may be **summarized** in the following table:

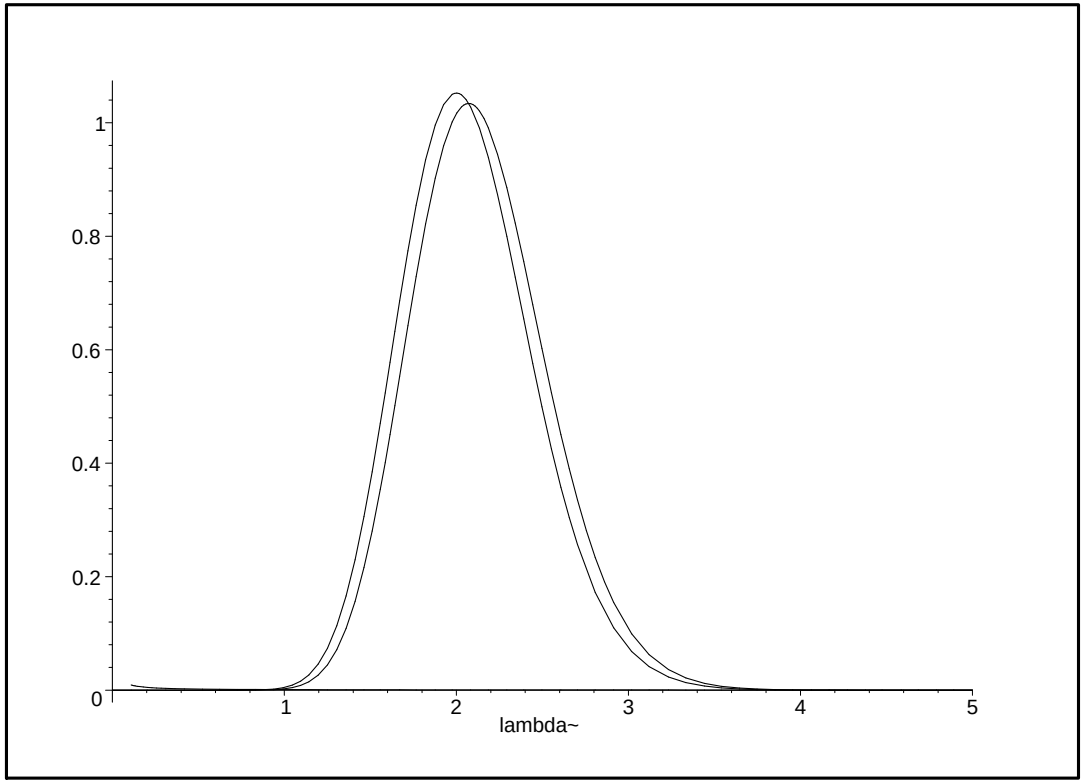
		Likelihood		
	Prior	Maximizing	Integrating	Posterior
Mean/Estimate	1.00	2.071	2.143	2.071
SD/SE	31.6	0.385	0.391	0.385

The **discrepancies** between the likelihood-maximizing and likelihood-integrating columns in this table would be smaller with a larger sample size and would **tend to 0** as $n \to \infty$.

The **prior-likelihood-posterior plot** comes out like this:

```
> plot( { p( lambda, 0.001, 0.001 ), p( lambda, 30, 14 ),
    p( lambda, 29.001, 14.001 ) }, lambda = 0 .. 5, color = black );
```

Prior-Likelihood-Posterior Summaries (continued)



For **interval estimation** in the maximum-likelihood approach the best we could do, using the technology I've described to you so far, would be to appeal to the **CLT** (even though n is only 14) and use $\hat{\lambda}_{\text{MLE}} \pm 1.96\,\widehat{SE}(\hat{\lambda}_{\text{MLE}}) \doteq 2.071 \pm (1.96)(0.385) \doteq (1.316, 2.826)$ as an **approximate 95% confidence interval** for λ.

The Interval Should Be Asymmetric in This Problem

You can see from the previous plot that the likelihood function is **asymmetric**, so a more careful method (e.g., the **bootstrap**; Efron 1979) would be needed to create a better interval estimate from the likelihood point of view.

Some trial and error with `Maple` can be used to find the lower and upper limits of the **central 95% posterior interval** for λ:

```
> evalf( Int( p( lambda, 29.001, 14.001 ), lambda = 0 .. 1.316 ) );
                                .01365067305
> evalf( Int( p( lambda, 29.001, 14.001 ), lambda = 0 .. 1.4 ) );
                                .02764660367
> evalf( Int( p( lambda, 29.001, 14.001 ), lambda = 0 .. 1.387 ) );
                                .02495470339
> evalf( Int( p( lambda, 29.001, 14.001 ), lambda = 2.826 .. infinity ) );
                                .03403487851
> evalf( Int( p( lambda, 29.001, 14.001 ), lambda = 2.890 .. infinity ) );
                                .02505307631
```

Thus a **95% (central) posterior interval** for λ, given a diffuse prior, runs from **1.387** to **2.890**, and is (correctly) **asymmetric** around the posterior mean of 2.071.

The R Solution

R can be used to work out the **limits of this interval** even more readily:

```
> help( qgamma )
GammaDist                    package:base                    R Documentation
The Gamma Distribution
Description:
     Density, distribution function, quantile function and random
     generation for the Gamma distribution with parameters 'shape' and
     'scale'.
Usage:
     dgamma(x, shape, scale=1, log = FALSE)
     pgamma(q, shape, scale=1, lower.tail = TRUE, log.p = FALSE)
     qgamma(p, shape, scale=1, lower.tail = TRUE, log.p = FALSE)
     rgamma(n, shape, scale=1)
Arguments:
    x, q: vector of quantiles.
       p: vector of probabilities.
       n: number of observations.
shape, scale: shape and scale parameters.
log, log.p: logical; if TRUE, probabilities p are given as log(p).
lower.tail: logical; if TRUE (default), probabilities are P[X <= x],
```

The R Solution (continued)

```
          otherwise, P[X > x].
Value:
     'dgamma' gives the density, 'pgamma' gives the distribution
     function 'qgamma' gives the quantile function, and 'rgamma'
     generates random deviates.
See Also:
     'gamma' for the Gamma function, 'dbeta' for the Beta distribution
     and 'dchisq' for the chi-squared distribution which is a special
     case of the Gamma distribution.
> qgamma( 0.025, 29.001, 1 / 14.001 )
[1] 1.387228
> qgamma( 0.975, 29.001, 1 / 14.001 )
[1] 2.890435
```

Maple or R can also be used to obtain the **probability content**, according to the posterior distribution, of the approximate 95% (large-sample) likelihood-based interval:

```
> evalf( Int( p( lambda, 29.001, 14.001 ), lambda = 1.316 .. 2.826 ) );
                              .9523144484
```

Predictive Distributions

So the **maximization** approach has led to **decent approximations** here (later I'll give examples where maximum likelihood doesn't do well in small samples).

Predictive distributions in this model can be computed by `Maple` **in the usual way**: e.g., to compute $p(y_{n+1}|y)$ for $y = (y_1, \ldots, y_n)$ we want to evaluate

$$\begin{aligned}
p(y_{n+1}|y) &= \int_0^\infty p(y_{n+1}, \lambda|y)\, d\lambda \\
&= \int_0^\infty p(y_{n+1}|\lambda, y)\, p(\lambda|y)\, d\lambda \qquad (14) \\
&= \int_0^\infty p(y_{n+1}|\lambda)\, p(\lambda|y)\, d\lambda \\
&= \int_0^\infty \frac{\lambda^{y_{n+1}} e^{-\lambda}}{y_{n+1}!} \frac{(\beta^*)^{\alpha^*}}{\Gamma(\alpha^*)} \lambda^{\alpha^*-1} e^{-\beta^*\lambda}\, d\lambda, \\
&= \frac{(\beta^*)^{\alpha^*}}{\Gamma(\alpha^*)\, y_{n+1}!} \int_0^\infty \lambda^{(\alpha^*+y_{n+1})-1} e^{-(\beta^*+1)\lambda}\, d\lambda,
\end{aligned}$$

where $\alpha^* = \alpha + s$ and $\beta^* = \beta + n$; in these expressions y_{n+1} is a **non-negative integer**.

Predictive Distributions (continued)

```
> assume( astar > 0, bstar > 0, yf > 0 );
> simplify( bstar^astar * int( lambda^( astar + yf - 1 ) *
    exp( - ( bstar + 1 ) * lambda ), lambda = 0 .. infinity ) /
    ( GAMMA( astar ) * yf! ) );
                astar~                  (-astar~ - yf~)
          bstar~       (bstar~ + 1)                      GAMMA(astar~ + yf~)
          ------------------------------------------------------------
                          GAMMA(astar~) GAMMA(yf~ + 1)
```

A bit of **rearranging** then gives that for $y_{n+1} = 0, 1, \ldots,$

$$p(y_{n+1}|y) = \frac{\Gamma(\alpha^* + y_{n+1})}{\Gamma(\alpha^*)\,\Gamma(y_{n+1}+1)} \left(\frac{\beta^*}{\beta^*+1}\right)^{\alpha^*} \left(\frac{1}{\beta^*+1}\right)^{y_{n+1}}. \qquad (15)$$

This is called the **Poisson-Gamma** distribution, because (14) is asking us to take a **mixture** (weighted average) of Poisson distributions, using probabilities from a Gamma distribution as the mixing weights.

(15) is a generalization of the **negative binomial** distribution (e.g., Johnson and Kotz 1994), which you may have encountered in your earlier probability study.

The Poisson-Gamma Distribution

Maple can try to get simple expressions for the **mean** and **variance** of this distribution:

```
> assume( alpha > 0, beta > 0 );
> pg := ( y, alpha, beta ) -> GAMMA( alpha + y ) *
    ( beta / ( beta + 1 ) )^alpha * ( 1 / ( beta + 1 ) )^y /
    ( GAMMA( alpha ) * GAMMA( y + 1 ) );
                                          /  beta  \alpha /   1    \y
                                GAMMA(alpha + y) |-------|      |-------|
                                                 \beta + 1/      \beta + 1/
  pg := (y, alpha, beta) -> ----------------------------------------
                                     GAMMA(alpha) GAMMA(y + 1)
> simplify( sum( pg( y, alpha, beta ), y = 0 .. infinity ) );
                                     1
> simplify( sum( y * pg( y, alpha, beta ), y = 0 .. infinity ) );
                                   alpha~
                                   ------
                                   beta~
```

So the **mean** of the Poisson-Gamma(α^*, β^*) distribution is $E(y_{n+1}|y) = \frac{\alpha^*}{\beta^*}$.

Contrasting Inference and Prediction

```
> simplify( sum( ( y - alpha / beta )^2 * pg( y, alpha, beta ),
     y = 0 .. infinity ) );
                                alpha~ (beta~ + 1)
                                ------------------
                                          2
                                     beta~
```

And the variance of the Poisson-Gamma(α^*, β^*) distribution is

$$V(y_{n+1}|y) = \frac{\alpha^*}{\beta^*}\left(1 + \frac{1}{\beta^*}\right). \tag{16}$$

This provides an interesting **contrast between inference and prediction**: we've already seen in this model that the posterior mean and variance of λ are $\frac{\alpha^*}{\beta^*} = \frac{\alpha+s}{\beta+n}$ and $\frac{\alpha^*}{(\beta^*)^2} = \frac{\alpha+s}{(\beta+n)^2}$, respectively.

Thus λ (the **inferential** objective) and y_{n+1} (the **predictive** objective) have the same posterior mean, but the posterior variance of y_{n+1} is **much larger**, as can be seen by the following argument.

Contrasting Inference and Prediction (continued)

	Posterior	
Quantity	Mean	Variance
λ	$\frac{\alpha+s}{\beta+n}$	$\frac{\alpha+s}{(\beta+n)^2} = \frac{\alpha+s}{\beta+n}\left(0+\frac{1}{\beta+n}\right)$
y_{n+1}	$\frac{\alpha+s}{\beta+n}$	$\frac{\alpha+s}{\beta+n}\left(1+\frac{1}{\beta+n}\right)$

(1) Denoting by μ the mean of the **population** from which the Y_i are thought of as (like) a random sample, when n is large α and β will be **small** in relation to s and n, respectively, and the ratio $\bar{y} = \frac{s}{n}$ should **more and more closely approach** μ — thus for large n,

$$E(\lambda|y) = E(y_{n+1}|y) \doteq \mu. \tag{17}$$

(2) For the Poisson distribution the (population) mean μ and variance σ^2 are **equal**, meaning that for large n the ratio $\frac{\alpha+s}{\beta+n}$ will be close both to μ and to σ^2.

Thus for large n,

$$V(\lambda|y) \doteq \frac{\sigma^2}{n} \quad \text{but} \quad V(y_{n+1}|y) \doteq \sigma^2. \tag{18}$$

Predictive Model-Checking

An informal way to restate (18) is to say that accurate **prediction** of new data is an **order of magnitude harder** (in powers of n) than accurate **inference** about population parameters.

Bayesian model-checking with predictive distributions. One way to **check** a model like (9) is as follows — as i goes from 1 to n, do the following two things:

(1) Temporarily **set aside** observation y_i, obtaining a new dataset $y_{-i} = (y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_n)$ with $(n-1)$ observations.

(2) Use the current Bayesian model applied to y_{-i} to **predict** y_i, and summarize the extent to which the actual value of y_i is **surprising** in view of this predictive distribution.

A simple measure of surprise is **predictive z–scores** (later, if there's time, I'll talk about a better measure):

$$z_i = \frac{y_i - E[y_i|y_{-i}]}{\sqrt{V[y_i|y_{-i}]}}. \tag{19}$$

Predictive Model-Checking (continued)

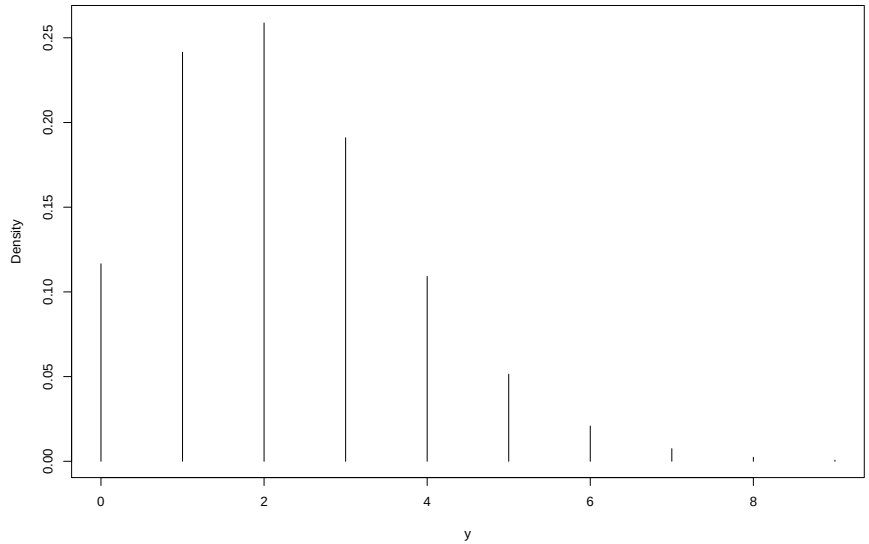
The idea is to compare the surprise measure with its **expected behavior** if the model had been **"correct"** (e.g., $z = (z_1, \ldots, z_n)$ should have mean 0 and SD 1, and a normal qqplot of the z_i values should be approximately linear).

Here's some `R` code to carry out this program on the **LOS data**.

```
> poisson.gamma <- function( y, alpha, beta ) {
    log.density <- lgamma( alpha + y ) + alpha *
      log( beta / ( beta + 1 ) ) - y * log( beta + 1 ) -
      lgamma( alpha ) - lgamma( y + 1 )
    return( exp( log.density ) )
  }
> print( y <- sort( y ) )
 [1] 0 1 1 1 1 1 2 2 2 2 3 3 4 6
> print( y.current <- y[ -1 ] )
 [1] 1 1 1 1 1 2 2 2 2 3 3 4 6
> print( n.current <- length( y.current ) )
[1] 13
> alpha <- beta <- 0.001
> print( s.current <- sum( y.current ) )
[1] 29
```

Predictive Model-Checking (continued)

```
> print( alpha.star <- alpha + s.current )
[1] 29.001
> print( beta.star <- beta + n.current )
[1] 13.001
> print( pg.current <- poisson.gamma( 0:9, alpha.star, beta.star ) )
 [1] 0.1165953406 0.2415099974 0.2587508547 0.1909752933 0.1091243547
 [6] 0.0514422231 0.0208209774 0.0074357447 0.0023899565 0.0007017815
> plot( 0:9, pg.current, type = 'n', xlab = 'y', ylab = 'Density' )
> for ( i in 0:9 ) {
    segments( i, 0, i, pg.current[ i + 1 ] )
  }
```



Predictive Model-Checking (continued)

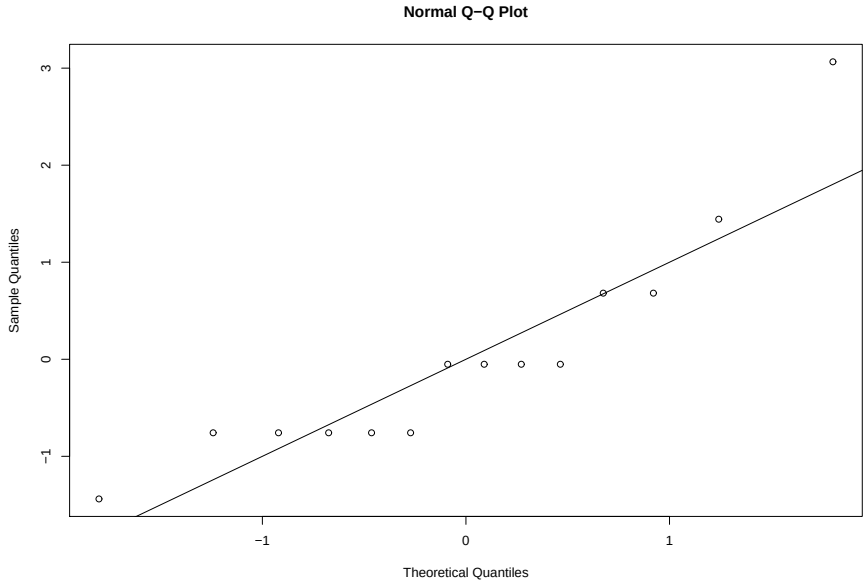
The omitted observed value of **0** is **not too unusual** in this predictive distribution.

The following `R` code **loops** through the whole dataset to get the **predictive** z**–scores**.

```
alpha <- beta <- 0.001
z <- rep( 0, n )
for ( i in 1:n ) {
  y.current <- y[ -i ]
  n.current <- length( y.current )
  s.current <- sum( y.current )
  alpha.star <- alpha + s.current
  beta.star <- beta + n.current
  predictive.mean.current <- alpha.star / beta.star
  predictive.SD.current <- sqrt( ( alpha.star / beta.star ) *
    ( 1 + 1 / beta.star ) )
  z[ i ] <- ( y[ i ] - predictive.mean.current ) /
    predictive.SD.current
}
```

Predictive Model-Checking (continued)

```
z
 [1] -1.43921925 -0.75757382 -0.75757382 -0.75757382 -0.75757382
 [6] -0.75757382 -0.05138023 -0.05138023 -0.05138023 -0.05138023
[11]  0.68145253  0.68145253  1.44329065  3.06513271
mean( z )
[1] 0.03133708
sqrt( var( z ) )
[1] 1.155077
qqnorm( z )
abline( 0, 1 )
```



Predictive Model-Checking (continued)

The 14 predictive z–scores have mean **0.03** (about right) and SD **1.16** (close enough to 1 when sampling variability is considered?), and the **normal qqplot** above shows that the only really surprising observation in the data, as far as the Poisson model was concerned, is the value of **6**, which has a z–score of **3.07**.

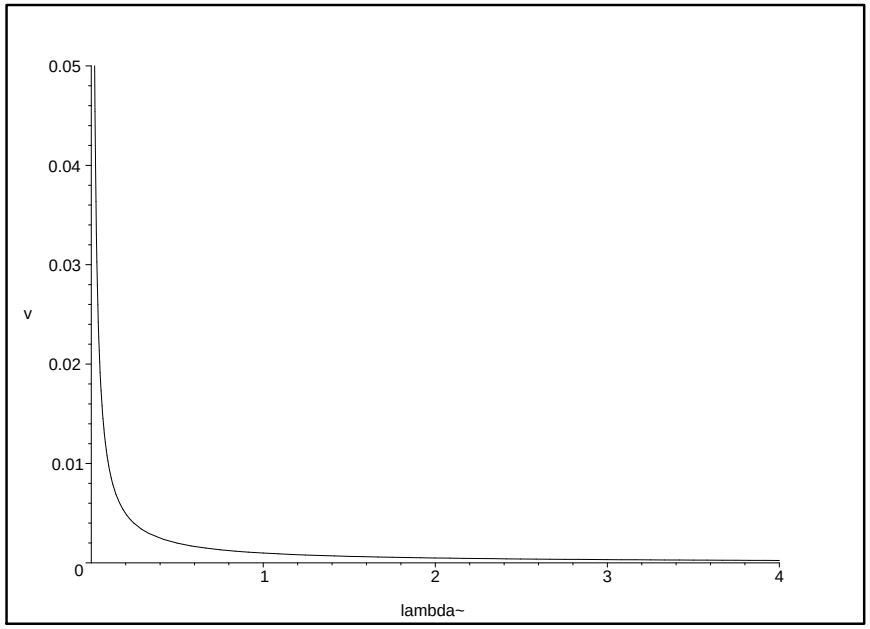
NB The figure above is only a **crude approximation** to the right qqplot, which would have to be created by **simulation**; even so it's enough to **suggest** how the model might be **improved**.

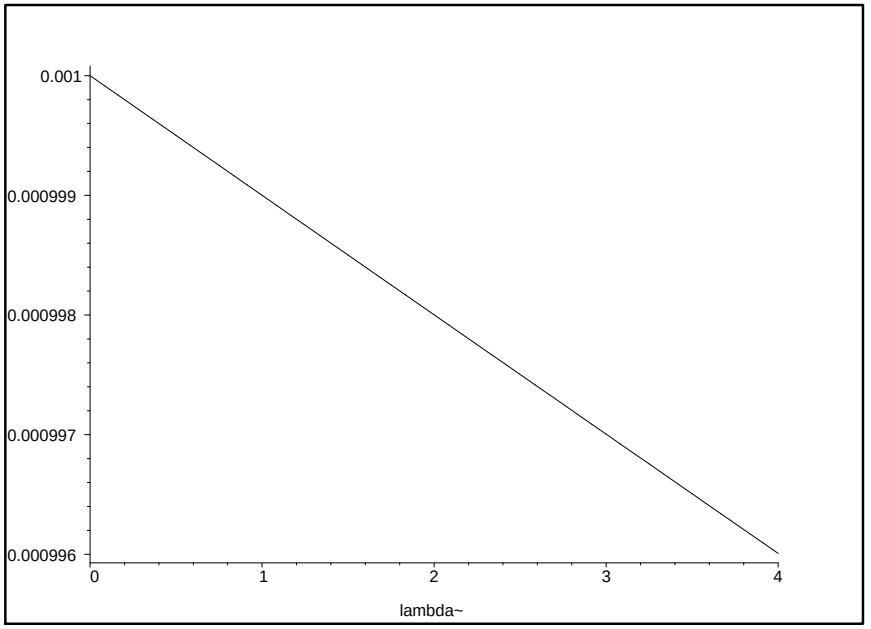
I would conclude **informally** (a) that the Poisson is a **decent** model for these data, but (b) if you wanted to expand the model in a direction suggested by this diagnostic you should look for a model with **extra-Poisson variation**: the sample VTMR in this dataset was about **1.15**.

Diffuse priors in the LOS case study. In specifying a **diffuse** prior for λ in the LOS case study, several **alternatives** to $\Gamma(\epsilon, \epsilon)$ might occur to you, including $\Gamma(1, \epsilon), \Gamma(\alpha, \beta)$ for some large α (like 20, to get a roughly **normal** prior) and small β (like 1, to have a **small prior sample size**), and $U(0, C)$ for some cutoff C (like 4) chosen to avoid **truncation** of the likelihood function, where $U(a, b)$ denotes the **uniform** distribution on (a, b).

Prior Specification: Sensitivity Analysis

```
> plot( p( lambda, 0.001, 0.001 ), lambda = 0 .. 4, v = 0 .. 0.05,
    color = black );
> plot( p( lambda, 1.0, 0.001 ), lambda = 0 .. 4, color = black );
```





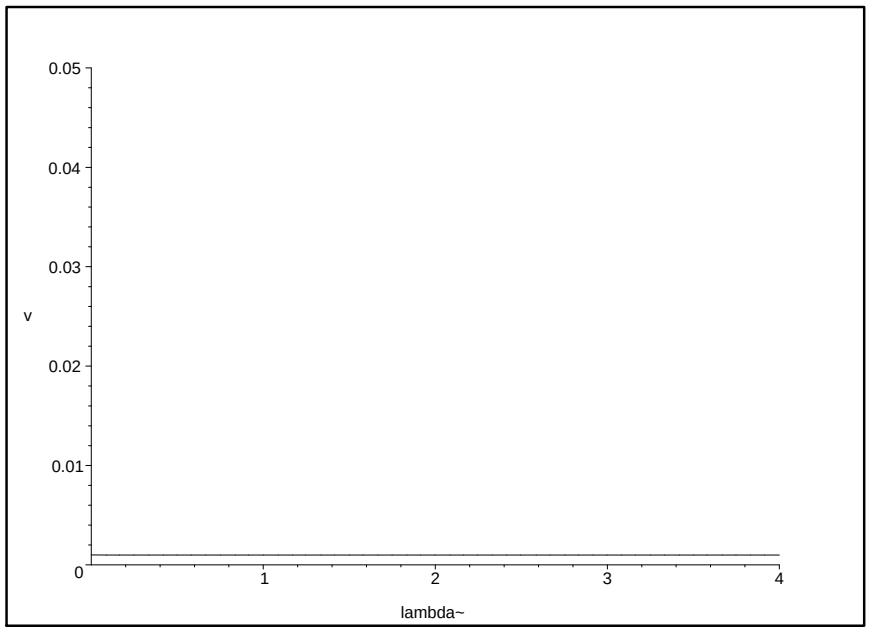
$\Gamma(1, \epsilon)$ doesn't look promising initially as a **flat** prior, but that's a consequence of `Maple`'s default choice of **vertical axis**:

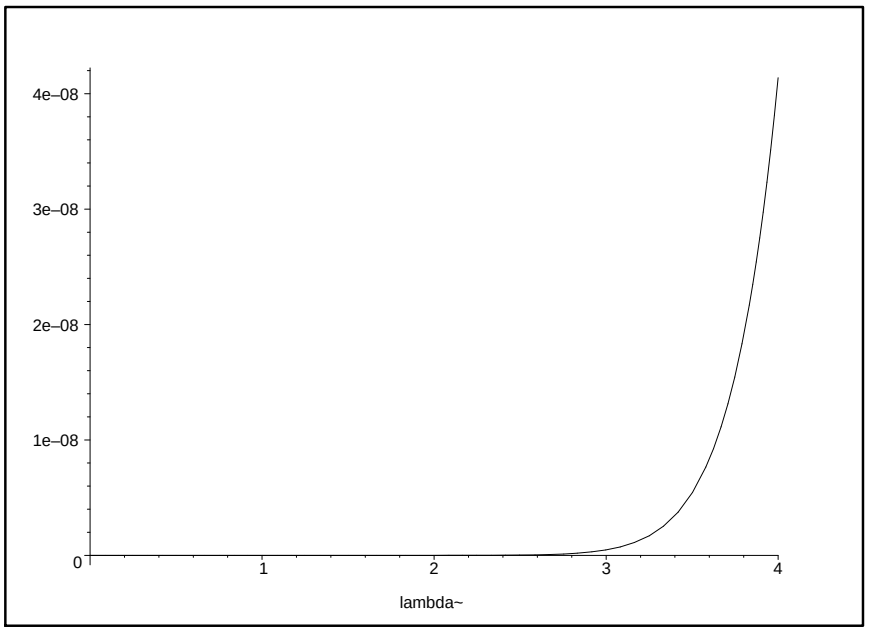
```
> plot( p( lambda, 1.0, 0.001 ), lambda = 0 .. 4, v = 0 .. 0.05,
    color = black );
```

Prior Specification: Sensitivity Analysis (continued)

```
> plot( p( lambda, 20, 1 ), lambda = 0 .. 4, color = black );
```

(Left: right-hand plot on previous page with **more sensible vertical scale**; right: $\Gamma(20, 1)$ with **not-so-sensible horizontal scale**)






```
> plot( p( lambda, 20, 1 ), lambda = 0 .. 40, color = black );
```

As is evident on the next page, $\Gamma(20, 1)$ does indeed look **not far from Gaussian**, and at first it may appear that it is indeed **relatively flat**

Prior Specification: Sensitivity Analysis (continued)

in the region where the likelihood is appreciable ($\lambda \in (1.0, 3.3)$), but we'll see below that it's actually **rather more informative** than we intend.

Recalling that the **mean** and **SD** of a $\Gamma(\alpha, \beta)$ random quantity are $\frac{\alpha}{\beta}$ and $\sqrt{\frac{\alpha}{\beta^2}}$, respectively, and that when used as a prior with the Poisson likelihood the $\Gamma(\alpha, \beta)$ distribution acts like a dataset with **prior sample size** β, you can construct the following table:

Prior				Posterior			
α	$\beta =$ Sample Size	Mean	SD	α^*	β^*	Mean	SD
0.001	0.001	1	31.6	29.001	14.001	2.071	0.385
1	0.001	1000	1000	30	14.001	2.143	0.391
20	1	20	4.47	49	15	3.267	0.467
20	0.001	20000	4472	49	14.001	3.500	0.500
$U(0, C)$ for $C > 4$		$\frac{C}{2}$	$\frac{C}{\sqrt{12}}$	30	14	2.143	0.391

Prior Specification: Sensitivity Analysis (continued)

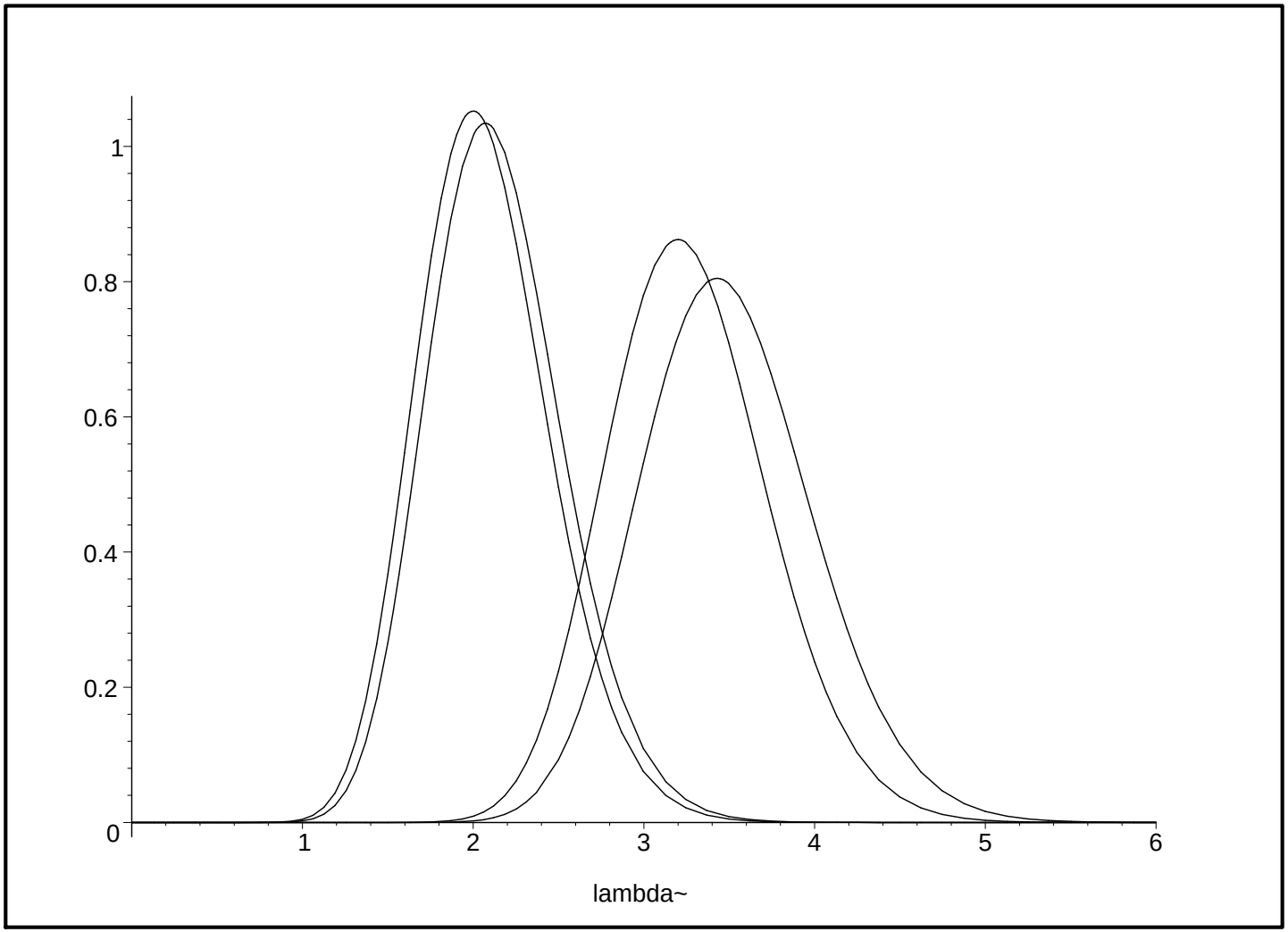
The $\Gamma(1, \epsilon)$ prior leads to an analysis that's **essentially equivalent** to the **integrated likelihood (fiducial)** approach back on page 100, and the $U(0, C)$ prior for $C > 4$ (say) produces similar results: $U(0, C)$ yields the $\Gamma(s + 1, n)$ posterior **truncated** to the right of C (and this truncation has **no effect** if you choose C big enough).

You might say that the $U(0, C)$ distribution has a **prior sample size of 0** in this analysis, and its prior mean $\frac{C}{2}$ and SD $\frac{C}{\sqrt{12}}$ (both of which can be made arbitrarily large by letting C grow without bound) are **irrelevant** (an example of how intuition can change when you depart from the class of **conjugate** priors).

```
> plot( { p( lambda, 29.001, 14.001 ), p( lambda, 30, 14.001 ),
    p( lambda, 49, 15 ), p( lambda, 49, 14.001 ) }, lambda = 0 .. 6,
    color = black );
```

The **moral** from the table above and the graph on the next page is that with only $n = 14$ observations, **some care is needed** (e.g., through **pre-posterior** analysis) to achieve a prior that **doesn't affect the posterior very much**, if that's the scientifically appropriate **information content** of the prior.

Prior Specification: Sensitivity Analysis (continued)



(Reading from left to right, **posteriors** with the following **priors**: $\Gamma(0.001, 0.001)$, $\Gamma(1, 0.001)$, $\Gamma(20, 1)$, $\Gamma(20, 0.001)$)