

Lab 3: Bayesian Fundamentals

Understanding Priors and Posteriors

Patrick T. Bradshaw

Updated for February 23, 2024

Working with the data from the Fracture Intervention Trial (FIT) we used last semester (see Vittinghoff et al. (2012), Ch 8), we can use Bayesian methods to estimate measures of association and arguably, interesting measures of effect, once we consider the target parameter of interest to be random variables.

First, read in the data:

```
library(foreign)
data.FIT <- read.dta("FITglm.dta")
```

Second, obtain summaries of the data (counts of events and person-time per treatment group):

```
# Obtain the total number of fractures per treatment group:
n.frac.0 <- with(data.FIT, sum(numnosp[trt01 == 0]))
n.frac.1 <- with(data.FIT, sum(numnosp[trt01 == 1]))

# Obtain the total person-time (person-years) per treatment group:
PT.0 <- with(data.FIT, sum(trialyrs[trt01 == 0]))
PT.1 <- with(data.FIT, sum(trialyrs[trt01 == 1]))
```

We assume that given the person-years of follow-up t , the counts $Y_i \mid t_i$ in treatment group i are Poisson distributed:

$$Y_i \mid t_i, \lambda_i \sim \text{Poisson}(\lambda_i t_i) \text{ for treatment: } i = 0, 1$$

with rate parameter λ_i representing number of events per person-year.

The likelihood for each treatment-specific rate $\mathcal{L}(\lambda_i \mid \mathbf{y}_i, \mathbf{t}_i)$ is given by the product of the probability mass functions:

$$\mathcal{L}(\lambda_i \mid \mathbf{y}_i, \mathbf{t}_i) = \prod_{j=1}^{N_i} \frac{(\lambda_i t_{i,j})^{y_{i,j}} \exp(-\lambda_i t_{i,j})}{(y_{i,j})!}$$

where N_i are the total number of subjects in treatment group i , and $y_{i,j}$ and $t_{i,j}$ are the number of observed fractures and years of follow-up for subject j in treatment group i .

The MLE of the treatment-specific rate parameter is given by:

$$\hat{\lambda}_i = \frac{\sum_{j=1}^{N_i} y_{i,j}}{\sum_{j=1}^{N_i} t_{i,j}}$$

and can be calculated as:

```
# Among control group:  
n.frac.0 / PT.0
```

```
[1] 0.04338471
```

```
# Among treated group:  
n.frac.1 / PT.1
```

```
[1] 0.03700692
```

Conjugate prior for λ

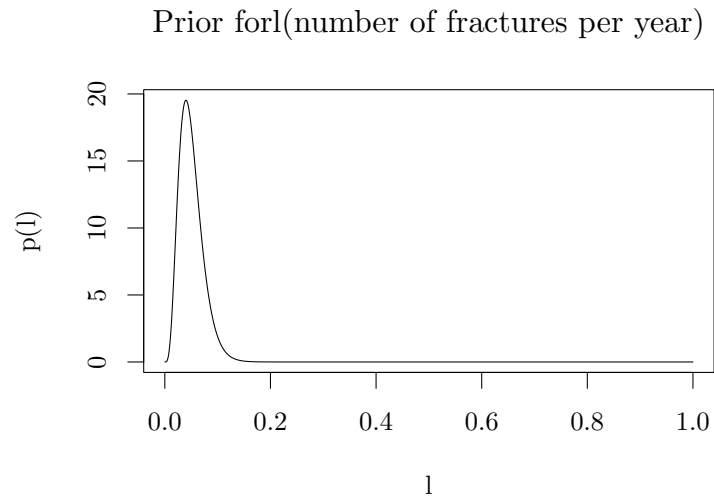
Since Y_i is Poisson distributed, the conjugate prior for the rate parameter is the Gamma distribution (using this prior, the posterior $\text{dist}(\lambda \mid \mathbf{y}, \mathbf{t})$ will also be Gamma). We first start out by visualizing the shape of this prior and characterizing its distribution assuming specific parameters that encode our prior information.

For simplicity, we will assume that the prior is the same for both treatment groups (but this is not necessary). Before you look at your data, you expect that the rate of fractures (per person-year) is very small – probably on the order of 5 fractures per 100 person-years. You know the mean of a Gamma distribution parameterized by shape and rate¹ parameters $((\alpha, \beta)$, respectively) is given by $\mathbb{E}[\lambda] = \alpha/\beta$, where α roughly corresponds to the number of events, and β corresponds to the number of years of follow-up. So by choosing $\alpha = 5$ and $\beta = 100$ the average prior rate for the Poisson outcome will be 0.05 fractures per year. This is the same amount of information you would get from a study where you observed 5 events out of 100 person-years.

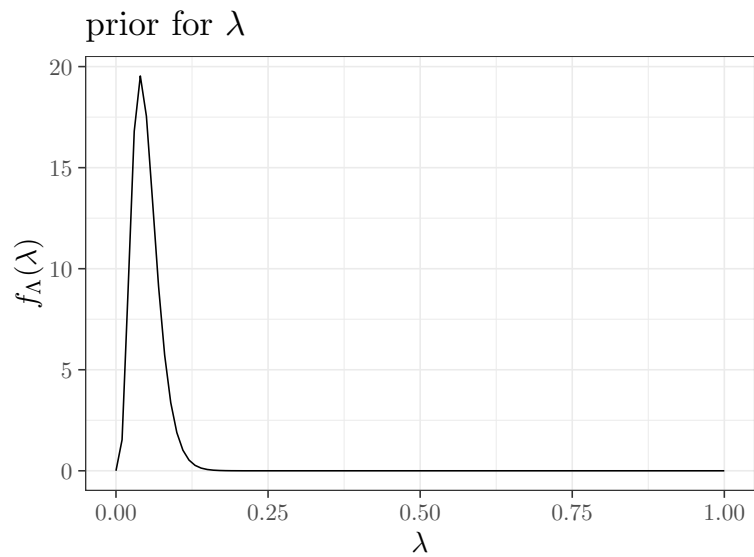
```
alpha <- 5
beta <- 100

lambda <- seq(0,1,.001) # Create a grid ranging from 0-1 in steps of 0.001
prior.lambda <- dgamma(lambda, alpha, beta)
plot(
  lambda,
  prior.lambda,
  type = "l",
  ylab = expression(paste("p(", lambda, ")")),
  xlab = expression(lambda),
  main = expression(paste(
    "Prior for ", lambda,
    " (number of fractures per year)"
  ))
)
```

¹Note that the rate for the Gamma is different than the rate for the Poisson.



```
library(ggplot2)
ggplot() +
  geom_function(fun = function(x) {
    dgamma(x, alpha, beta)}) +
  labs(y = "$f_{\\Lambda}(\\lambda)$", x = "$\\lambda$") +
  ggtitle('prior for $\\lambda$') +
  theme_bw()
```



Note that the prior allows for λ to range over $(0, \infty)$, but we truncate at 1 because the density approaches 0 very quickly.

Question

If your prior is specified as above, what does this imply about the probability that the rate is between 2/100 and 7/100?

Answer

$$\Pr[2/100 < Y < 7/100] = F(7/100) - F(2/100) = \int_{2/100}^{7/100} p(y) dy$$

where $p(y)$ is the PDF of the Gamma distribution given $\alpha = 5$ and $\beta = 100$.

To calculate in R we can take the difference in the cumulative distribution functions (the CDF for the Gamma distribution, $F(x) = \Pr(X < x)$ in R is calculated with the `pgamma` function):

```
pgamma(7 / 100, alpha, beta) - pgamma(2 / 100, alpha, beta)
```

```
[1] 0.7743554
```

Or, even more fun, we can ask R to evaluate the integral between 2/100 and 7/100 (using numerical approximation) on the density function directly:

```
integrate(function(x) dgamma(x, alpha, beta), 2/100, 7/100)
```

```
0.7743554 with absolute error < 8.6e-15
```

Exercise

Try evaluating the integral from 0 to `Inf`(∞) to verify that the PDF integrates to 1!

```
pgamma(Inf, alpha, beta)
```

```
[1] 1
```

```
integrate(function(x) dgamma(x, alpha, beta), 0, Inf)
```

```
1 with absolute error < 1.2e-06
```

The mean and variance (α/β^2) of this prior depends on the specific values for α and β . Note that different values can yield the same mean $= \alpha/\beta$ but the shape can change widely. You can use this fact to regulate how likely you think certain values might be *a priori*, but maintaining the location of your prior expectation.

Characterizing the posterior

The posterior distribution $\text{dist}(\lambda_i | \mathbf{y}, \mathbf{t})$ can be shown to be Gamma (because of conjugacy between the Gamma prior and Poisson likelihood):

$$\text{Posterior} = p(\lambda_i | \mathbf{y}_i, \mathbf{t}_i) \propto \overbrace{p(\mathbf{y}_i, \mathbf{t}_i | \lambda_i)}^{\text{Likelihood}} \times \underbrace{p(\lambda_i)}_{\text{Prior}}$$

In this case, it has parameters:

$$\alpha_i^* = \alpha + \sum_j y_{i,j}$$

$$\beta_i^* = \beta + \sum_j t_{i,j}.$$

Intuitively, we have taken the information from the prior and incorporated (added) the information from the observed data (counts: $\sum y$ and follow-up time: $\sum t$), yielding an update to our belief about the distribution of λ_i . We can characterize this posterior in a similar manner to how we did the prior:

```
# Specify the parameters from the posterior:
alpha.star.0 <- alpha + n.frac.0
beta.star.0 <- beta + PT.0

alpha.star.1 <- alpha + n.frac.1
beta.star.1 <- beta + PT.1

# Generate posterior densities over the range of lambda:
posterior.lambda.0 <- dgamma(lambda, alpha.star.0, beta.star.0)
posterior.lambda.1 <- dgamma(lambda, alpha.star.1, beta.star.1)

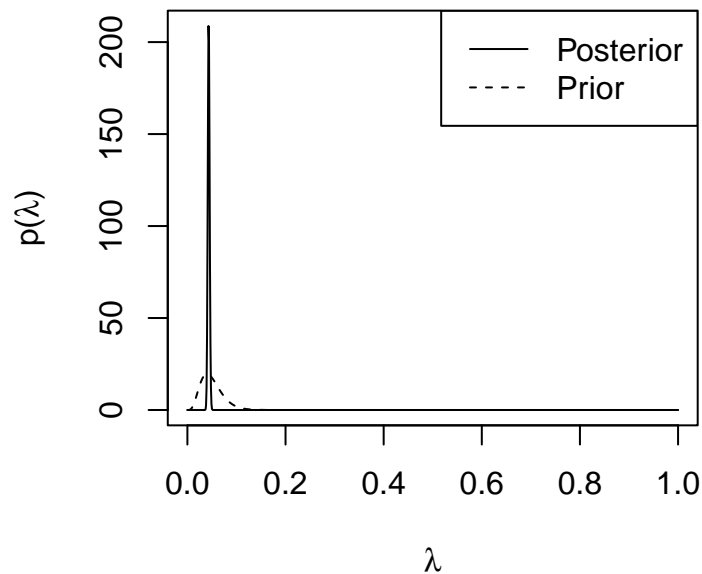
# Plot:
plot(
  lambda,
```

```

posterior.lambda.0,
type = "l",
ylab = expression(paste("p(", lambda, ")")),
xlab = expression(lambda),
main = expression(
  atop(
    "Distributions for "*
    lambda*
    " (number of fractures","per year) in Control Group"
  ))
)
lines(lambda, prior.lambda, lty = "dashed")
legend("topright",
      legend = c("Posterior", "Prior"),
      lty = c(1, 2))

```

Distributions for λ (number of fractures
per year) in Control Group



```

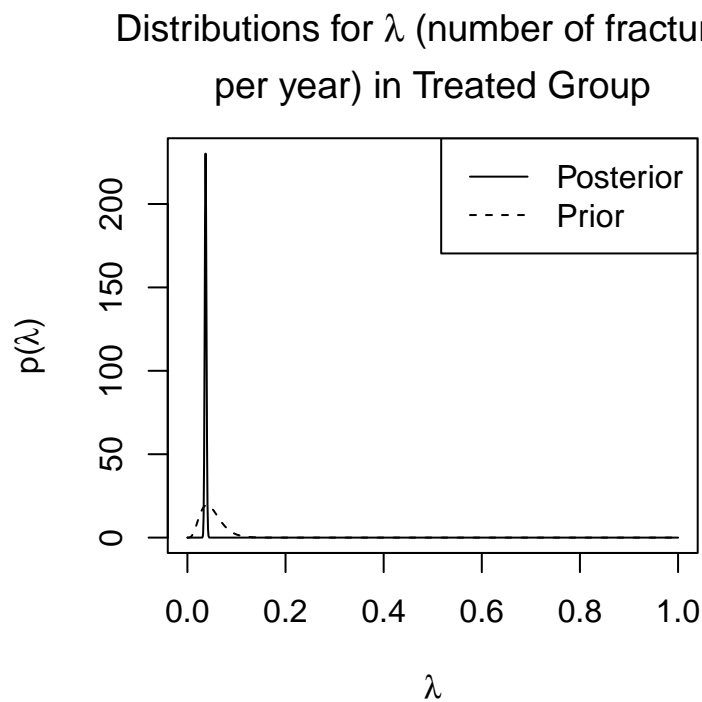
plot(
  lambda,
  posterior.lambda.1,
type = "l",
ylab = expression(paste("p(", lambda, ")")),

```

```

xlab = expression(lambda),
main = expression(
  atop(
    "Distributions for "*"
    lambda*"
    " (number of fractures","per year) in Treated Group"
  ))
)
lines(lambda, prior.lambda, lty = "dashed")
legend("topright",
      legend = c("Posterior", "Prior"),
      lty = c(1, 2))

```



Because the mean of a Gamma distribution is the ratio of the shape to rate parameter, we can calculate the **posterior** mean:

```

# In control:
alpha.star.0 / beta.star.0

```

```
[1] 0.04343842
```



```
# In treated:  
alpha.star.1 / beta.star.1
```

```
[1] 0.03711221
```

The expected posterior rates are essentially the same as the rates that would be calculated from the original data (without any prior).

Question

Why didn't the prior have much influence?

Hint. Look at the expressions for the parameters of the posterior α_i^* and β_i^* . Consider how much information the *data* contributed to the posterior parameters, and how much information the *prior* contributed.

Now calculate the **posterior probability** of the rates being between 2/100 and 7/100:

```
# Control:  
pgamma(7/100, alpha.star.0, beta.star.0) -  
pgamma(2/100, alpha.star.0, beta.star.0)
```

```
[1] 1
```

```
# Treated:  
pgamma(7/100, alpha.star.1, beta.star.1) -  
pgamma(2/100, alpha.star.1, beta.star.1)
```

```
[1] 1
```

After seeing the data, we are now virtually 100% certain that the true rate in both groups lies between 2/100 and 7/100.

Question

Why did our certainty for values within this range increase so much after incorporating the data?

And, we can calculate 95% credible intervals for each of these rates:

```
# Control:  
qgamma(c(0.025, 0.975), alpha.star.0, beta.star.0)
```

```
[1] 0.03983509 0.04719554
```

```
# Treated:  
qgamma(c(0.025, 0.975), alpha.star.1, beta.star.1)
```

```
[1] 0.03379074 0.04058714
```

Sensitivity analysis

The prior we initially specified was equivalent to data from a study that observed 5 fractures over 100 person-years. This does not correspond to much information, but the prior was centered near the maximum likelihood value, and didn't really place much prior probability on values outside the range of .02 to .07. To relax this, re-run the above code, placing a weaker Gamma prior on λ . Note that the shape parameter from a Gamma distribution (α) does not have to be an integer, and fractional values can represent "average" number of counts per unit time. For example, if we specify $\alpha = 0.5$ and $\beta = 10$ this still yields a distribution with the same expected value $E[\lambda] = \alpha/\beta = 0.05$ but based on less "information" (e.g. only 1/2 of an event observed over 10 person-years).

Exercise

Try with other values of α and β , keeping the mean = 0.05 to build intuition. Also try selecting values that yield larger prior means (e.g. $\alpha/\beta = .5$ with small and large values of β).

```
alpha <- 0.5
beta  <- 10
# mean of gamma = alpha/beta
# variance of gamma alpha/beta^2

# Visualize prior under new hyperparameters
ggplot() +
  geom_function(fun = function(x) {
    dgamma(x, alpha, beta)}) +
  ggtitle("Prior density on  $\lambda$ ") +
  labs(x = " $\lambda$ ", y = " $\Lambda$ ") +
  theme_bw() -> prior.ggplot
prior.ggplot

# Calculate posterior hyperparameters
alpha.star.0 <- alpha + n.frac.0
alpha.star.1 <- alpha + n.frac.1
beta.star.0 <- beta + PT.0
beta.star.1 <- beta + PT.1

prior.ggplot +
  geom_function(fun = function(x) {
    dgamma(x, alpha.star.0, beta.star.0)
```

```
  }, aes(col = "control")) +  
  geom_function(fun = function(x) {  
    dgamma(x, alpha.star.1, beta.star.1)  
  }, aes(col = "treatment")) +  
  scale_x_continuous(limits = c(0, 0.1))  
  
# Posterior means  
alpha.star.0/beta.star.0  
alpha.star.1/beta.star.1
```

Simulation

Another way we can characterize the posterior distribution is via Monte-Carlo simulation—we have used this earlier in the semester to examine other distributions for target parameters with uncertain distributions. In this example it does not buy us much since we know the specific form of the posterior, but this helps build the foundation for more sophisticated modeling. We know here that the posterior for each λ_i are distributed Gamma, and we know the parameters of this posterior, so we can randomly sample from these Gamma distributions:

```
N.sim <- 10000 # 10,000 samples

set.seed(123)

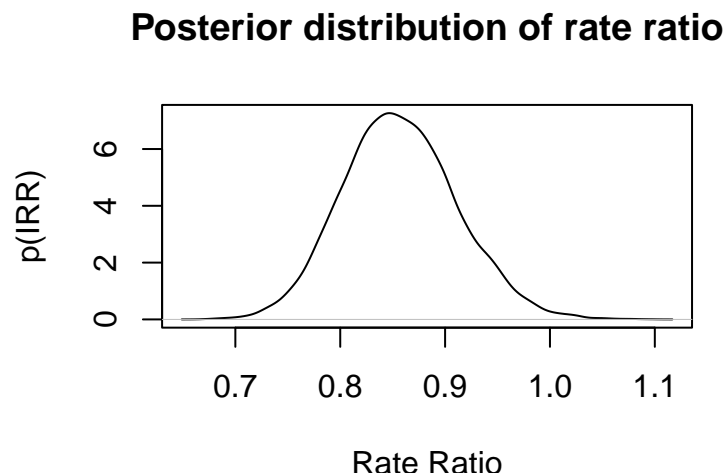
# Randomly sample from posterior rate in controls:
lambda.0 <- rgamma(N.sim, alpha.star.0, beta.star.0)

# Randomly sample from posterior rate in treated:
lambda.1 <- rgamma(N.sim, alpha.star.1, beta.star.1)
```

Now, we can use these samples to characterize the posterior distribution of functions of these parameters. For example, we can use posterior samples from λ_0 and λ_1 to calculate the posterior distribution of the **rate ratio**, and obtain the corresponding posterior mean, median, and the 95% credible interval:

```
rr <- lambda.1/lambda.0

plot(density(rr), xlab="Rate Ratio", ylab="p(IRR)",
     main="Posterior distribution of rate ratio", lty=1)
```



```
round(mean(rr),2)
```

```
[1] 0.86
```

```
round(quantile(rr, c(0.5, 0.025, 0.975)),2)
```

```
50% 2.5% 97.5%  
0.85 0.76 0.97
```

Compare to the IRR from a Poisson GLM:

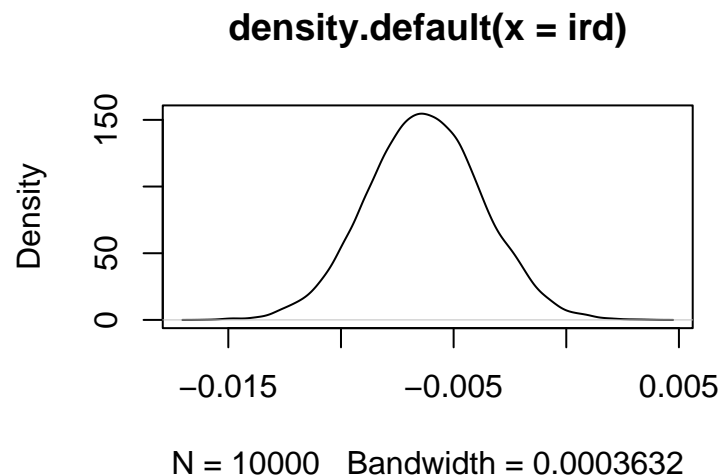
```
fit.glm <- glm(numnosp ~ trt01 + offset(log(trialyrs)), family=poisson,  
               data=data.FIT)  
IRR.pois <- exp(coef(fit.glm))  
CI.IRR.pois <- exp(confint.default(fit.glm))  
round(cbind(IRR.pois, CI.IRR.pois),2)[2,]
```

```
IRR.pois    2.5 %    97.5 %  
    0.85      0.75      0.97
```

Question

Consider how you might modify the above code to calculate an Incidence Rate Difference, and associated credible intervals.

```
ird <- lambda.1 - lambda.0  
plot(density(ird))
```



```
quantile(ird, c(0.5, 0.025, 0.975))
```

50%	2.5%	97.5%
-0.006287067	-0.011283040	-0.001281495

```
mean(ird)
```

```
[1] -0.006276404
```

One last thing...

One particularly attractive feature of the Bayesian framework, where target parameters are treated as random, is that you can estimate quantities that most people would find useful. For example, say you wanted to know the **probability** that the treatment in the above trial reduced the rate of fractures.

Using the simulated values of the individual rates, the proportion of the simulations where $\lambda_1 < \lambda_0$ (or equivalently $\lambda_1/\lambda_0 < 1$) is an estimate of the probability that the treatment reduced the rate of events. We can calculate this as:

```
mean(lambda.1/lambda.0 < 1)
```

```
[1] 0.9925
```

which suggests that the probability that the treatment “worked” is around 0.99. Note that this does not take into account magnitude, but we could always refine the question, such as: What is the probability that the treatment reduced the rate by at least 20%? We would change the code to:

```
mean(lambda.1/lambda.0 < .80)
```

```
[1] 0.1463
```

In general, Monte-Carlo random sampling is limited in its usefulness for Bayesian computation, but we will discuss an extension called *Markov-Chain Monte-Carlo* sampling (and other relatives) that can be used more generally.

Vittinghoff, Eric, David V. Glidden, Stephen C. Shiboski, and Charles E. McCulloch. 2012. “Regression Methods in Biostatistics.” Springer New York, NY. <https://doi.org/10.1007/978-1-4614-1353-0>.