Exercise 1.3. Traffic accidents

Advanced Statistical Modelling

October 10, 2023

Ximena Moure

1 Draw these three prior distributions in the same graph.

- Bru's prior: Gamma(0.01, 0.001).
  The selection of these specific parameters leads to a distribution that is greatly spread out and skewed, peaking near zero. This could indicate an initial belief that lower accident rates are slightly more likely, yet there's a substantial level of uncertainty, which also accommodates the potential for higher rates.

- Clàudia's prior: Gamma(6.25, 2.5).
  This particular Gamma distribution (Gamma(6.25,2.5)) would result in a distribution that is skewed to the right, with a peak near the mean value of 2.5, and a variance of 1.

- Carles' prior: Flat (improper prior, $\pi(\mu)=1$ for $\mu>0$). A flat prior assigns equal probability to all values of μ within the specified range. Graphically, this is represented by a horizontal line, which indicates that the density (height of the line) is the same for all μ.

The following is the R code to draw the three prior distributions in the same graph.

```
library(ggplot2)

# Define the parameters for the Gamma distributions

prior_bru <- c(alpha = 0.01 , beta = 0.001)
prior_claudia <- c(alpha = 6.25 , beta = 2.5)

# Create a sequence of mu values instead of simulating values
mu <- seq(0, 10, 0.01)

# Calculate the pdf (probability density functions) of the Gamma distributions at
# each value of mu
pdf_bru <- dgamma(mu, shape = prior_bru[1], rate = prior_bru[2])
pdf_claudia <- dgamma(mu, shape = prior_claudia[1], rate = prior_claudia[2])

# Create a dataframe for plotting
data <- data.frame(
  mu = mu,
  pdf_bru = pdf_bru,
  pdf_claudia = pdf_claudia,
  flat_prior = rep(1, length(mu))
)

# Plot the three prior distributions
ggplot(data, aes(x = mu)) +
  geom_line(aes(y = pdf_bru, color = "Bru's Prior")) +
  geom_line(aes(y = pdf_claudia, color = "Clàudia's Prior")) +
```

```
geom_line(aes(y = flat_prior, color = "Carles' Prior")) +
labs(x =  expression(lambda),y = "Density", title = "Prior Distributions") +
theme_minimal() +
scale_color_manual(values = c("red", "blue", "green"))
```

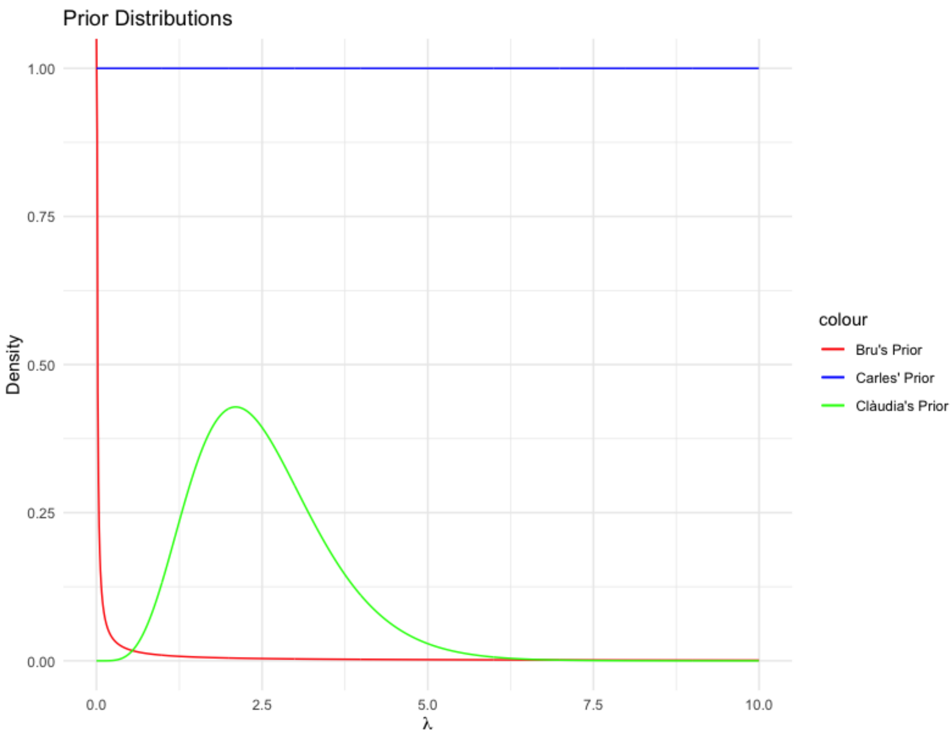

Figure 1: Plot with the three prior distributions

2 Draw the likelihood function

The likelihood function for a given value of μ is given by:

$$L(\mu) = \prod_{i=1}^{n} \frac{e^{-\mu} \mu^{x_i}}{x_i!}$$

where n is the number of observations, and x_i are the observed counts of accidents.

We standardize the likelihood, which means adjusting it so that its integral (or area under the curve) equals 1. This allows the likelihood to be plotted and compared on the same scale as probability distributions such as priors and posteriors. Also, it is easier to interpret, as it gives a clear picture of how likely different parameter values are given the data.

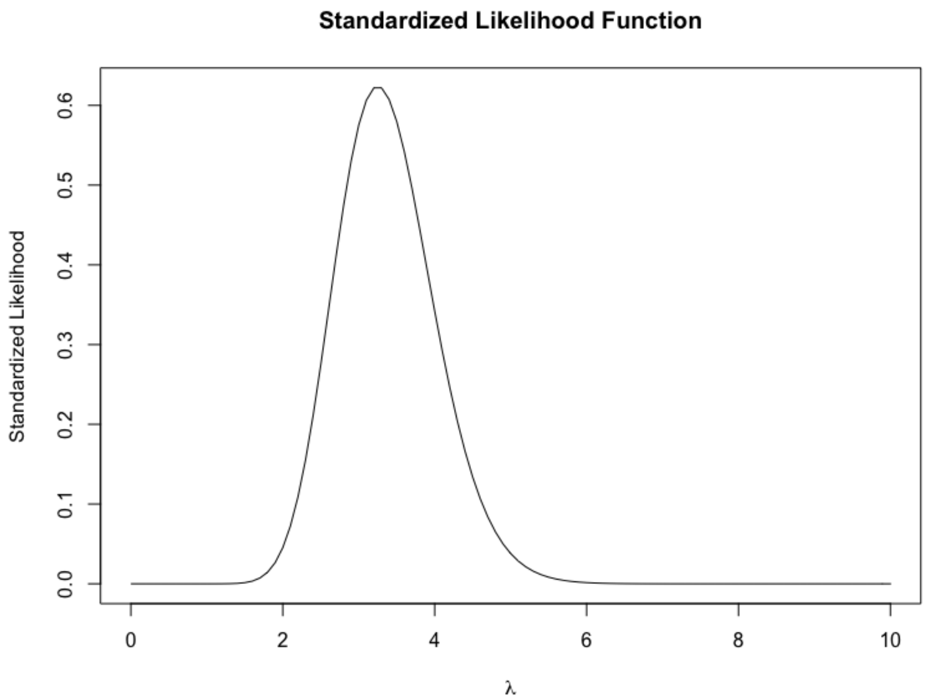

Figure 2: Likelihood function

Code:

```
# data
y <- c(3, 2, 0, 8, 2, 4, 6, 1)
n <- length(y)

# Define the standardized likelihood function
sd.like <- function(th) {
  (th^sum(y)*exp(-n*th))/integrate(function(th)(th^sum(y)*exp(-n*th)), lower = 0, upper
      = 10)$value
```

```
}

# Generate x-values (lambda values) for plotting
x_vals <- seq(0, 10, 0.1)

# Evaluate the standardized likelihood at the x-values
sd_like_vals <- sapply(x_vals, sd.like)

# Plot the standardized likelihood
plot(x_vals, sd_like_vals, type = "l", xlab = expression(lambda), ylab = "Standardized Likelihood", main = "Standardized Likelihood Function")
```

3 Draw the three posterior distributions in the same graph

Posterior distributions are derived by multiplying the prior distribution by the likelihood function and then normalizing the result. In the cases of Bru and Clàudia, the multiplication involving the Gamma prior will yield another Gamma distribution, thanks to the conjugate prior property. On the other hand, for Carles, the flat prior will merely retain the form of the likelihood function, although normalization will be required to convert it into a proper distribution.

In Bayesian analysis, the posterior distribution is proportional to the product of the prior distribution and the likelihood function:

$$\text{Posterior} \propto \text{Prior} \times \text{Likelihood}$$

When a flat prior is used, the prior does not contribute any additional information, and the posterior is shaped by the likelihood:

$$\text{Posterior} \propto \text{Likelihood}$$

In the case of Carles, a flat prior is assumed, which means the prior is a constant value for all . Because the prior is constant, it might seem like the posterior is simply proportional to the likelihood. However, for the posterior to be a proper probability distribution, it must integrate to 1 over the parameter space.

The posterior parameters for Bru and Claudia are calculated based on the formula for updating a Gamma prior with Poisson likelihood.

When you observe data $y = \{y_1, y_2, \ldots, y_n\}$ from a Poisson distribution with a Gamma prior on λ, the posterior distribution for λ is updated as follows:

$$\text{Posterior: } Gamma(\alpha_{\text{post}}, \beta_{\text{post}})$$

where

$$\alpha_{\text{post}} = \alpha_{\text{prior}} + \sum y_i$$

$$\beta_{\text{post}} = \beta_{\text{prior}} + n$$

n= number of data points

Code:

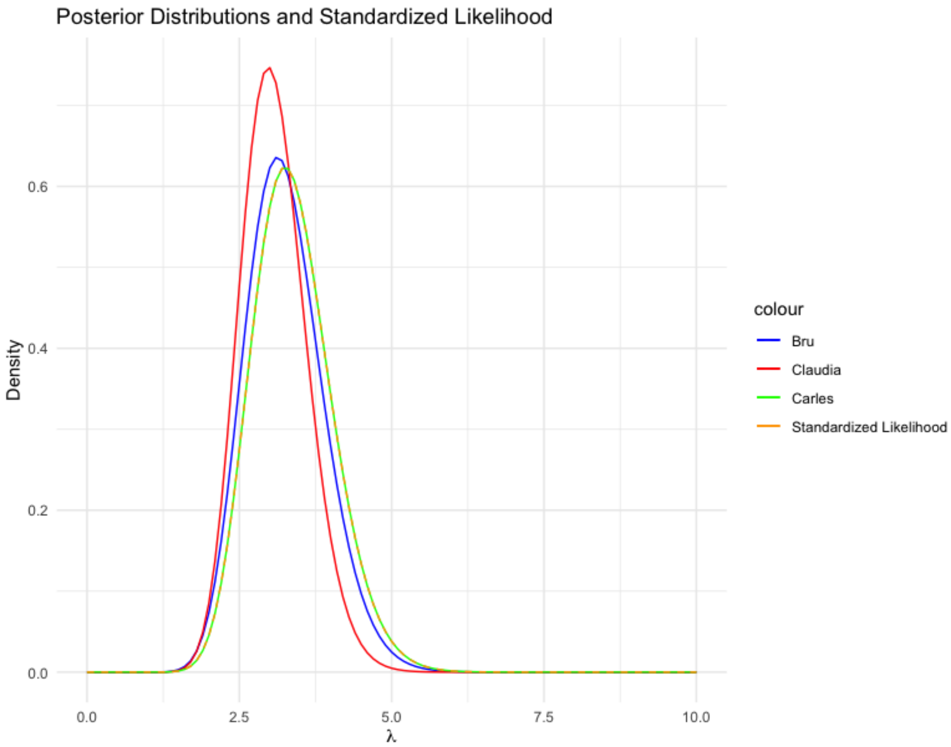

Figure 3: Posterior distributions

```r

library(ggplot2)

# Define the data and the number of data points
y <- c(3, 2, 0, 8, 2, 4, 6, 1)
n <- length(y)

# Define the prior parameters
prior1 <- c(alpha =  0.01, beta = 0.001)
prior2 <- c(alpha =  6.25, beta = 2.5)
prior3 <- c(alpha =  1, beta = 0)  # Note: This flat prior is improper and won't work with dgamma

# Calculate the posterior parameters for Bru and Claudia
posterior1 <- c(a = prior1[1] + sum(y), b = prior1[2] + n)
posterior2 <- c(a = prior2[1] + sum(y), b = prior2[2] + n)

# Define the standardized likelihood function
sd.like <- function(th) {
  (th^sum(y)*exp(-n*th))/integrate(function(th)(th^sum(y)*exp(-n*th)), lower = 0, upper = 10)$value
}

# For Carles, calculate the unnormalized posterior
# This function defines the unnormalized posterior distribution for
# Carles based on the likelihood of a Poisson distribution.
```

```r
carles_posterior_unnormalized <- function(th) {
  th^sum(y) * exp(-n*th)
}

# Normalize Carles' posterior
normalization_constant_carles <- integrate(carles_posterior_unnormalized, lower = 0,
    upper = 10)$value
carles_posterior <- function(th) {
  carles_posterior_unnormalized(th) / normalization_constant_carles
}

# Define a sequence of x values for plotting
x_vals <- seq(0, 10, 0.1)

# Evaluate the posterior distributions and the standardized likelihood at the x values
bru_posterior_vals <- dgamma(x_vals, shape = posterior1[1], rate = posterior1[2])
claudia_posterior_vals <- dgamma(x_vals, shape = posterior2[1], rate = posterior2[2])
carles_posterior_vals <- sapply(x_vals, carles_posterior)
sd_like_vals <- sapply(x_vals, sd.like)

# Create a data frame for plotting
plot_data <- data.frame(
  x_vals,
  bru_posterior_vals,
  claudia_posterior_vals,
  carles_posterior_vals,
  sd_like_vals
)

# Plot the posterior distributions and the standardized likelihood
ggplot(plot_data, aes(x = x_vals)) +
  geom_line(aes(y = bru_posterior_vals, color = 'Bru')) +
  geom_line(aes(y = claudia_posterior_vals, color = 'Claudia')) +
  geom_line(aes(y = carles_posterior_vals, color = 'Carles')) +
  geom_line(aes(y = sd_like_vals, color = 'Standardized Likelihood'), linetype = "
      dashed") +
  scale_color_manual(
    values = c('Bru' = 'blue', 'Claudia' = 'red', 'Carles' = 'green', 'Standardized
        Likelihood' = 'orange'),
    breaks = c('Bru', 'Claudia', 'Carles', 'Standardized Likelihood')
  ) +
  labs(title = "Posterior Distributions and Standardized Likelihood", x = expression(
      lambda), y = "Density") +
  theme_minimal()
```

4 Calculate a 90% credible interval of the number of accidents for next weekend for each of the three Bayesian models

The task is to calculate a 90% credible interval for the number of accidents for the next weekend under the three different Bayesian models. The predictive posterior distribution is the appropriate tool for this task because it estimates the distribution of future observations based on the observed data and the posterior distribution of the parameter.

In the Bayesian framework, making predictions about future observations naturally involves using the posterior distribution of the parameter, which has been updated with the observed data. The predictive posterior distribution is the Bayesian way to make such predictions.

In order to calculate the credible interval, I used a simulation based approach.

- For Bru: The 90% credible interval is [1,7]. This means that, according to Bru's model and given the data and prior information, there's a 90% probability that the number of accidents next weekend will fall between 1 and 7.

- For Clàudia: The 90% credible interval is [0,6].This means that, according to Claudia's model and given the data and prior information, there's a 90% probability that the number of accidents next weekend will fall between 0 and 6.

- For Carles:The 90% credible interval is [1,7]. This means that, according to Bru's model and given the data and prior information, there's a 90% probability that the number of accidents next weekend will fall between 1 and 7.

Code:

```
# Function to calculate the 90% credible interval
 M <- 100000



 y <- c(3, 2, 0, 8, 2, 4, 6, 1)
 n <- length(y)

 posterior1 <- c(a = 0.01 + sum(y), b = 0.001 + n)
 posterior2 <- c(a = 6.25 + sum(y), b = 2.5 + n)
 posterior3 <- c(a = 1 + sum(y), b = 0 + n)
 #1
 posterior1.sim <- rgamma(M, posterior1[1], posterior1[2])
 pre.posterior1.sim <- rpois(M,posterior1.sim)
 table(pre.posterior1.sim)

```

```r
interval1 <- quantile(pre.posterior1.sim, probs = c(0.05, 0.95))
print(interval1)

#2
posterior2.sim <- rgamma(M, posterior2[1], posterior2[2])
pre.posterior2.sim <- rpois(M,posterior2.sim)
table(pre.posterior2.sim)

interval2 <- quantile(pre.posterior2.sim, probs = c(0.05, 0.95))
print(interval2)


#3
posterior3.sim <- rgamma(M, posterior3[1], posterior3[2])
pre.posterior3.sim <- rpois(M,posterior3.sim)
table(pre.posterior3.sim)

interval3 <- quantile(pre.posterior3.sim, probs = c(0.05, 0.95))
print(interval3)


# Output the 90% credible intervals
list(Bru = interval1, Claudia = interval2, Carles = interval3)
```