ESS 575: Bayes Theorem Lab

Team England

21 September, 2022

Team England:

- Caroline Blommel
- Carolyn Coyle
- Bryn Crosby
- George Woolsey

cblommel@mail.colostate.edu, carolynm@mail.colostate.edu, brcrosby@rams.colostate.edu, george.woolsey@colostate.edu

Problem

You are interested in estimating the posterior distribution for the mean number of individuals of an invasive plant species per m^2 in a disturbed grassland. We will call that mean θ. You have prior information telling you that the average number of these plants per m^2 is 10.2 with a standard deviation of the mean = .5. You have a set of fifty observations in hand obtained by sweaty labor in the field. Execute the following steps.

Preliminaries

1

Simulate 50 data points from a Poisson distribution with mean $\theta = 6.4$ to represent the data set. (This portrays the data that you gathered from plots, but it is lots easier to obtain.) What is the variance? Be sure to put the R function `set.seed(10)` before the call to `rpois()` to assure that we all get the same results. Call the data vector `y`.

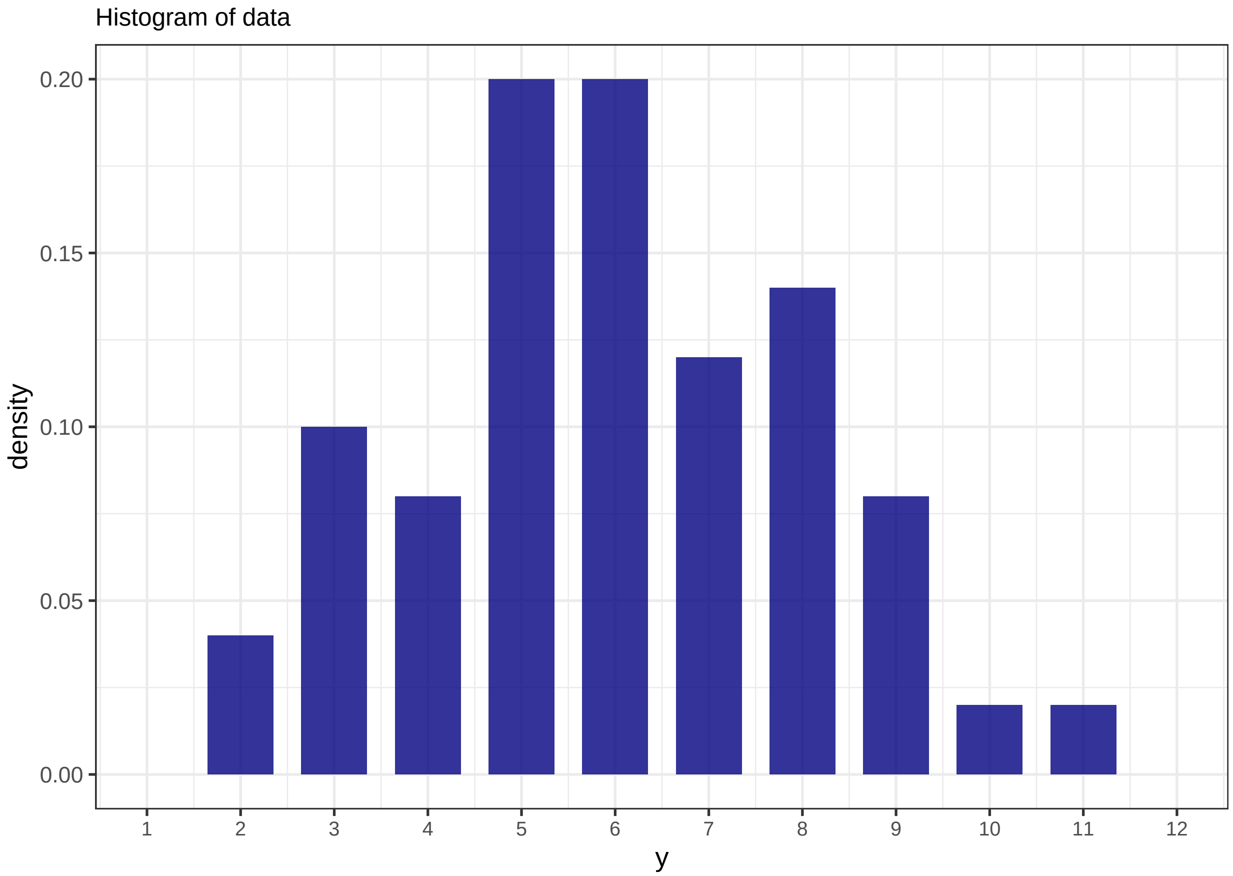
```
n <- 50
lambda <- 6.4
y <- rpois(n = n, lambda = lambda)
```

The variance (λ) of the simulated Poisson distribution with mean $\theta = 6.4$ is: **4.4**

2

Plot a histogram of the data with density on the y-axis. It turns out that the histogram function in R is not really appropriate for discrete data (why?). Discover how to do a proper histogram for these data. Look into the `arm` package

```
# summarize data for plotting
plt_1 <-
data.frame(
    y = y
  ) %>%
  dplyr::count(y) %>%
  dplyr::mutate(density = n / sum(n)) %>%
# plot
  ggplot(., mapping = aes(x = y, y = density)) +
    geom_col(width = 0.7, fill = "navy", alpha = 0.8) +
    scale_x_continuous(
      breaks = seq(min(y)-1, max(y)+1, 1)
      , limits = c(min(y)-1, max(y)+1)
    ) +
    xlab("y") +
    ylab("density") +
    labs(
      title = "Histogram of data"
    ) +
    theme_bw() +
    theme(
      plot.title = element_text(size = 10)
      , axis.text.x = element_text(size = 8)
    )
plt_1
```



The base histogram function in R is not really appropriate for discrete data because histograms are suited to continuous data which can be binned. Plotting data for categorical or discrete data is best accomplished in R using `ggplot2::geom_col` or `ggplot2::geom_bar`

3

Set values for the prior mean (`mu.prior`) and standard deviation (`sigma.prior`). You have prior information telling you that the average number of these plants per m^2 is 10.2 with a standard deviation of the mean = .5.

```
mu.prior <- 10.2
sigma.prior <- 0.5
```

4

Set up a vector containing a sequence of values for θ, the mean number of invasive plants, You want this vector to approximate a continuous θ, so be sure it contains values that are not too far apart. Use code like this: `theta = seq(0, 15, step)` where you set `step = .01`. Setting a value for `step` with global scope is important. You will use it later when you integrate.

```
step <- .01
theta <- seq(0, 15, step)
```

The prior distribution of θ

5

Write the mathematical expression for a gamma prior on θ. Be as specific as possible given the information at hand. Write an R function for the prior on θ. The function for the prior should return a vector of gamma probability densities, one for each value of θ. It should have arguments 1) the vector for θ you created in the previous step as well as 2) the prior mean and 3) the prior standard deviation. The mean and the standard deviation, of course, will need to be moment-matched to the proper parameters of the gamma distribution in your function. Recall that when a function is composed of a single statement as it is here, the statement can simply follow the function template on the same line; curly brackets are not needed. So, in this case `mu.prior = 10.2` and `sigma.prior = 0.5`. You could hard-code these in the function template, but that is bad practice.

Mathematical expression for a gamma prior on $[\theta]$

$$[\theta] = \mathsf{gamma}\left(\frac{10.2^2}{0.5^2}, \frac{10.2}{0.5^2}\right)$$

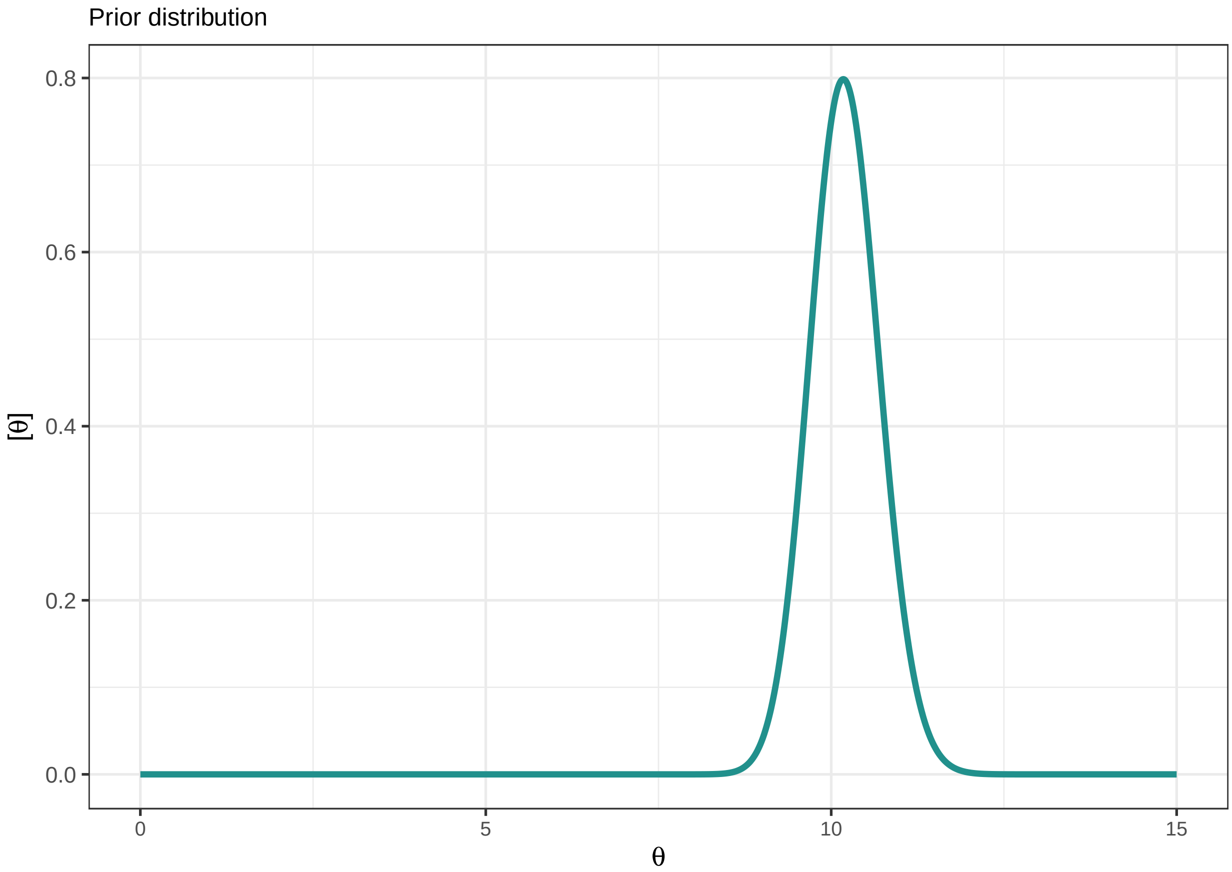
Function for the prior

```
# define function
prior <- function(theta, mu, sigma){
  ########################
  # assume gamma dist
  ########################
  # calculate the shape alpha for gamma dist
  alpha <- (mu^2)/(sigma^2)
  # calculate the rate beta for gamma dist
  beta <- (mu)/(sigma^2)
  ## probability density f'
  d_gamma <- dgamma(x = theta, shape = alpha, rate = beta)
  return(d_gamma)
}
# return prior on theta
prior_theta <- prior(theta = theta, mu = mu.prior, sigma = sigma.prior)
```

6

Plot the prior distribution of θ, the probability density of θ as a function of the values of θ.

```
plt_2 <-
# plot
data.frame(
  theta
  , prior_theta
) %>%
ggplot(., mapping = aes(x = theta, y = prior_theta)) +
  geom_line(lwd = 1.2, color = viridisLite::viridis(n = 3)[2]) +
  xlab(latex2exp::TeX("$\\theta$")) +
  ylab(latex2exp::TeX("$\\[\\theta\\]$")) +
  labs(
    title = "Prior distribution"
  ) +
  theme_bw() +
  theme(
    plot.title = element_text(size = 10)
    , axis.text.x = element_text(size = 8)
  )
plt_2
```



7

Check your moment matching by generating 100,000 random variates from a gamma distribution with parameters matched to the prior mean and standard deviation. Now compute the mean and standard deviation of the random variates. They should be very close to 10.2 and .5.

```
n <- 100000
## calculate the shape alpha for gamma dist
alpha <- (mu.prior^2)/(sigma.prior^2)
## calculate the rate beta for gamma dist
beta <- (mu.prior)/(sigma.prior^2)
```

```
## probability density f'
r_gamma <- rgamma(n = n, shape = alpha, rate = beta)
```

The mean of the simulated gamma distribution with mean mu.prior = 10.2 and variance sigma.prior = 0.5 is: **10.20**

The standard deviation of the simulated gamma distribution with mean mu.prior = 10.2 and variance sigma.prior = 0.5 is: **0.50**

The likelihood

8

What is the mathematical expression for the likelihood $[\mathbf{y} \mid \theta]$, assuming that the data are conditionally independent? Be as specific as possible using the information at hand.

Write an R function for the likelihood. The function must use all `50` observations to compute the total likelihood across all of the data points (not the log likelihood) for each value of the vector θ. It should have arguments for the vector θ and the data. The function should create and return a vector with elements $[\mathbf{y} \mid \theta_i]$. Note that this is the total probability density of all of the data for *each* value of θ_i, not the probability density of a single data point. In reality, θ is a continuous random variable, the mean of the Poisson distribution. We are discretizing it here into small intervals. The function template will be something like:

Mathematical expression for the likelihood $[\mathbf{y} \mid \theta]$

$$[\mathbf{y} \mid \theta] = \prod_{i=1}^{50} \mathsf{Poisson}(\mathrm{y_i} \mid \theta)$$

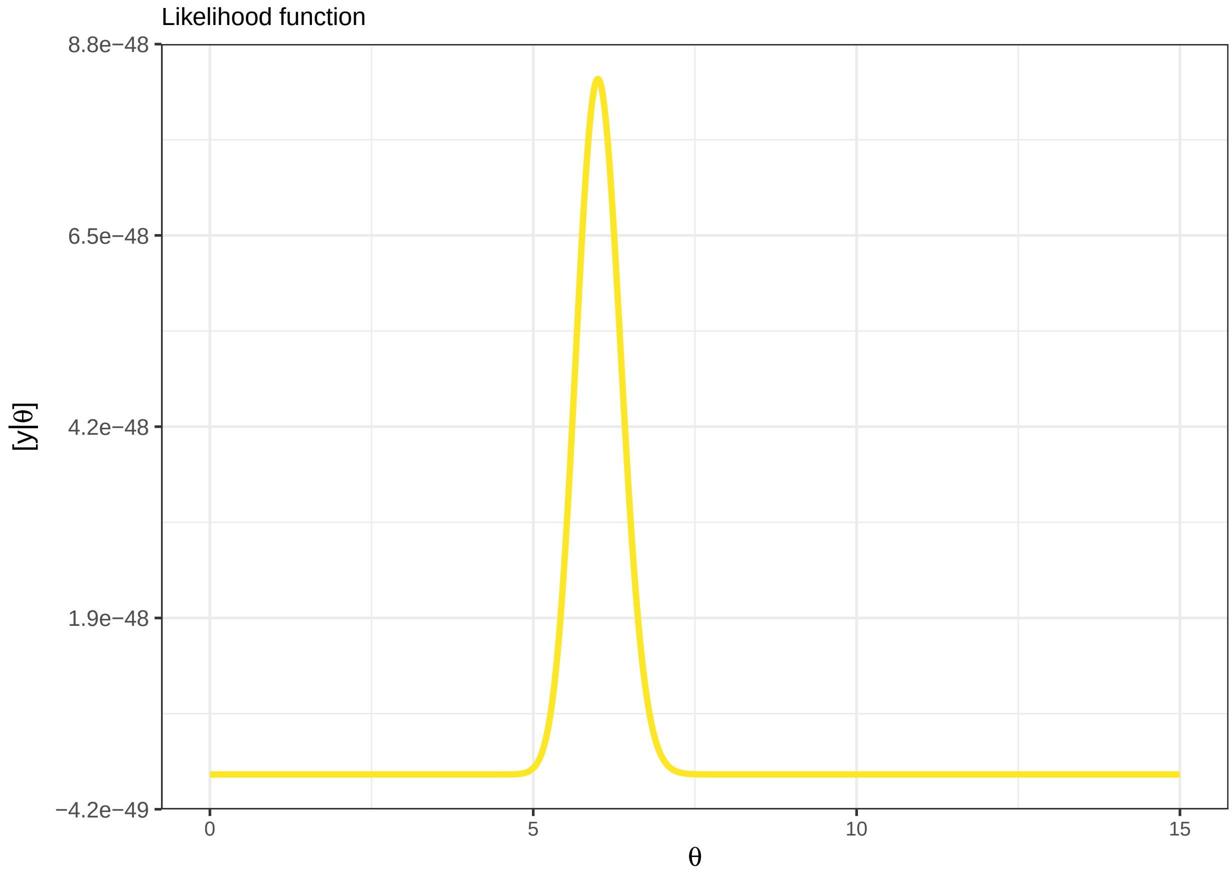
Function for the likelihood

```
like <- function(theta, y){
  #your code to calculate total likelihood of the data conditional on each value of theta
  temp_v <- numeric(length(theta))
  for(i in 1:length(theta)){
    temp_v[i] <- prod(
      dpois(x = y, lambda = theta[i], log = FALSE)
    )
  }
  return(temp_v)
}
likelihood_y_theta <- like(theta = theta, y = y)
```

9

Plot the likelihood $[\mathbf{y} \mid \theta_i]$ holding the data constant and varying θ. What is this plot called? Can you say anything about the area under the curve? What happens to the inference we can make based on likelihood alone if we multiply the curve by a constant?

```
plt_3 <-
# plot
data.frame(
  theta
  , likelihood_y_theta
) %>%
ggplot(., mapping = aes(x = theta, y = likelihood_y_theta)) +
  geom_line(lwd = 1.2, color = viridisLite::viridis(n = 3)[3]) +
  scale_y_continuous(labels = scales::label_scientific(digits = 2)) +
  xlab(latex2exp::TeX("$\\theta$")) +
  ylab(latex2exp::TeX("$\\[y | \\theta\\]$")) +
  labs(
    title = "Likelihood function"
  ) +
  theme_bw() +
  theme(
    plot.title = element_text(size = 10)
    , axis.text.x = element_text(size = 8)
  )
plt_3
```



We use $[y \mid \theta]$ to assess the likelihood of different values of θ in light of the data. In this case, the function does not sum or integrate to one over all possible values of the parameter. The inference we can make based on likelihood alone if we multiply the curve by a constant is the same but would be proportional to the constant.

The joint distribution

10

What is the mathematical expression for the joint distribution $[\theta, \mathbf{y}]$? Your answer should be as specific as possible. I am not looking for the non-specific equation $[\theta, \mathbf{y}] = [\mathbf{y} \mid \theta][\theta]$. Create an R function for the joint distribution of the parameters and the data as the product of the prior and the likelihood functions. Call this function `joint`. The function should simply call the previous two functions and multiply them. Plot `joint(theta)` as a function of `theta`. Does this seem reasonable? Why are the values on the y axis so small? Think about what is going on here.

Mathematical expression for the joint distribution $[\theta, \mathbf{y}]$

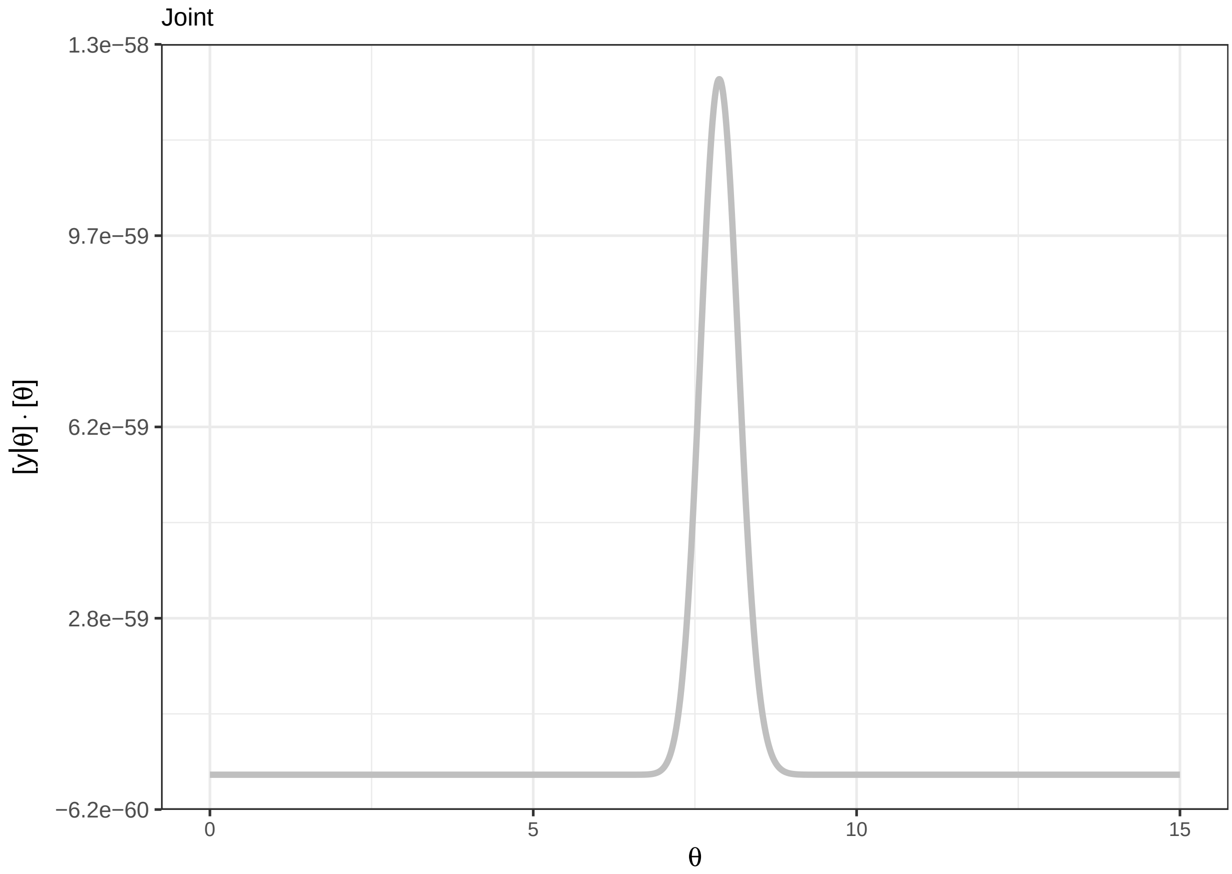
$$[\mathbf{y} \mid \theta] = \prod_{i=1}^{50} \mathsf{Poisson}(\mathrm{y_i} \mid \theta) \;\cdot\; \mathsf{gamma}\bigg(\theta \mid \frac{10.2^2}{0.5^2}, \frac{10.2}{0.5^2}\bigg)$$

Function for the joint

```r
joint <- function(theta, y, mu, sigma){
  prior <- prior(theta = theta, mu = mu, sigma = sigma)
  likelihood <- like(theta = theta, y = y)
  return(likelihood * prior)
}
joint_theta <- joint(theta = theta, y = y, mu = mu.prior, sigma = sigma.prior)
```

Plot `joint(theta)` as a function of `theta`

```r
plt_4 <-
#plot
data.frame(
  theta
  , joint_theta
) %>%
ggplot(., mapping = aes(x = theta, y = joint_theta)) +
  geom_line(lwd = 1.2, color = "gray75") +
  scale_y_continuous(labels = scales::label_scientific(digits = 2)) +
  xlab(latex2exp::TeX("$\\theta$")) +
  ylab(latex2exp::TeX("$\\[y | \\theta\\] \\cdot \\[\\theta\\]$")) +
  labs(
    title = "Joint"
  ) +
  theme_bw() +
  theme(
    plot.title = element_text(size = 10)
    , axis.text.x = element_text(size = 8)
  )
plt_4
```



Yes, the values on the plot seem reasonable because the result is the product of the likelihood function and the prior distribution. The values on the y-axis are very small because the values on the y-axis of the likelihood function were very small and the values on the y-axis of the prior distribution were less than 1.

The marginal probability of the data

11

What is the mathematical expression for the marginal probability of the data [**y**]? Again, be as specific as possible with the information you have been given. Approximate the marginal probability of the data, the integral of the likelihood multiplied by the prior, to obtain a normalization constant [**y**]. How would you accomplish this integration? (Hint–Recall the first principles definition of the definite integral.) Explain the output of this integration, a scalar. Why do we call [**y**] a "distribution" if it evaluates to a scalar?

Mathematical expression for the marginal probability of the data [y]

$$[\mathbf{y}] = \int_\theta \prod_{i=1}^{50} \mathsf{Poisson}(\mathrm{y_i} \mid \theta) \ \cdot \ \mathsf{gamma}\left(\theta \mid \frac{10.2^2}{0.5^2}, \frac{10.2}{0.5^2}\right) \mathrm{d}\theta$$

Approximate the marginal probability of the data

```
marginal_p_y <- sum(
  joint(theta = theta, y = y, mu = mu.prior, sigma = sigma.prior)
  * step
)
```

The approximate marginal probability of the data, the normalizing constant, is: **9.2e-59**

The marginal distribution of the data, the normalizing constant, (the denominator) is the area under the joint distribution. We call [**y**] a distribution even though it evaluates to a scalar because the data are fixed after they are collected.

The posterior distribution

12

What is the mathematical expression for the posterior distribution $[\theta \mid \mathbf{y}]$? Be as specific as possible using the information you have been given. Compute the posterior distribution by dividing each element of the vector of output produced by the joint function by the integral of the joint function. Plot the posterior as a function of θ.

Mathematical expression for the posterior distribution $[\theta \mid \mathbf{y}]$

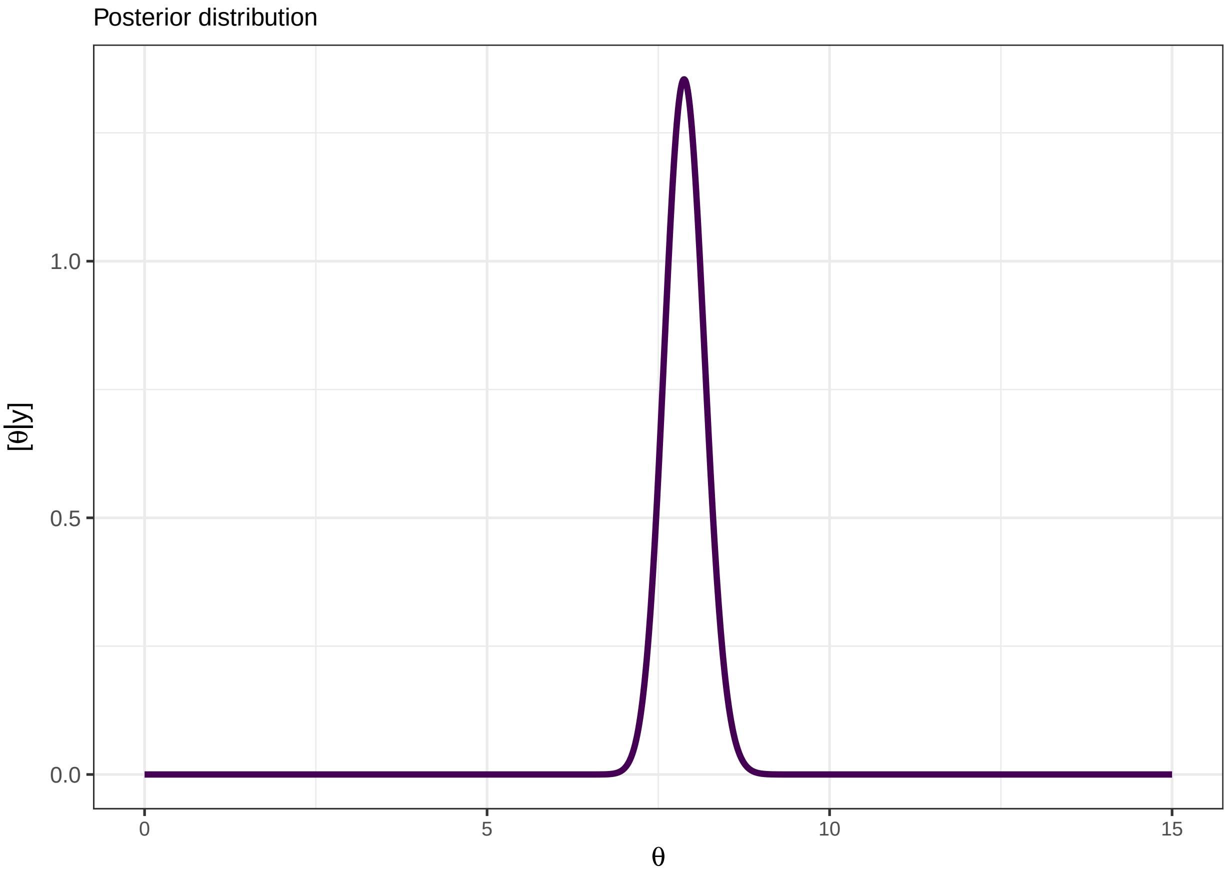
$$[\theta \mid \mathbf{y}] = \frac{\prod_{i=1}^{50} \textsf{Poisson}(\mathrm{y_i} \mid \theta) \ \cdot \ \textsf{gamma}\left(\theta \mid \frac{10.2^2}{0.5^2}, \frac{10.2}{0.5^2}\right)}{\int_\theta \prod_{i=1}^{50} \textsf{Poisson}(\mathrm{y_i} \mid \theta) \ \cdot \ \textsf{gamma}\left(\theta \mid \frac{10.2^2}{0.5^2}, \frac{10.2}{0.5^2}\right) \mathrm{d}\theta}$$

Compute the posterior distribution

```
posterior_dist <- joint(theta = theta, y = y, mu = mu.prior, sigma = sigma.prior) /
  sum(
    joint(theta = theta, y = y, mu = mu.prior, sigma = sigma.prior)
    * step
  )
```

Plot the posterior as a function of θ

```
plt_5 <-
#plot
data.frame(
  theta
  , posterior_dist
) %>%
ggplot(., mapping = aes(x = theta, y = posterior_dist)) +
  geom_line(lwd = 1.2, color = viridisLite::viridis(n = 3)[1]) +
  xlab(latex2exp::TeX("$\\theta$")) +
  ylab(latex2exp::TeX("$\\[\\theta | y\\]$")) +
  labs(
    title = "Posterior distribution"
  ) +
  theme_bw() +
  theme(
    plot.title = element_text(size = 10)
    , axis.text.x = element_text(size = 8)
  )
plt_5
```

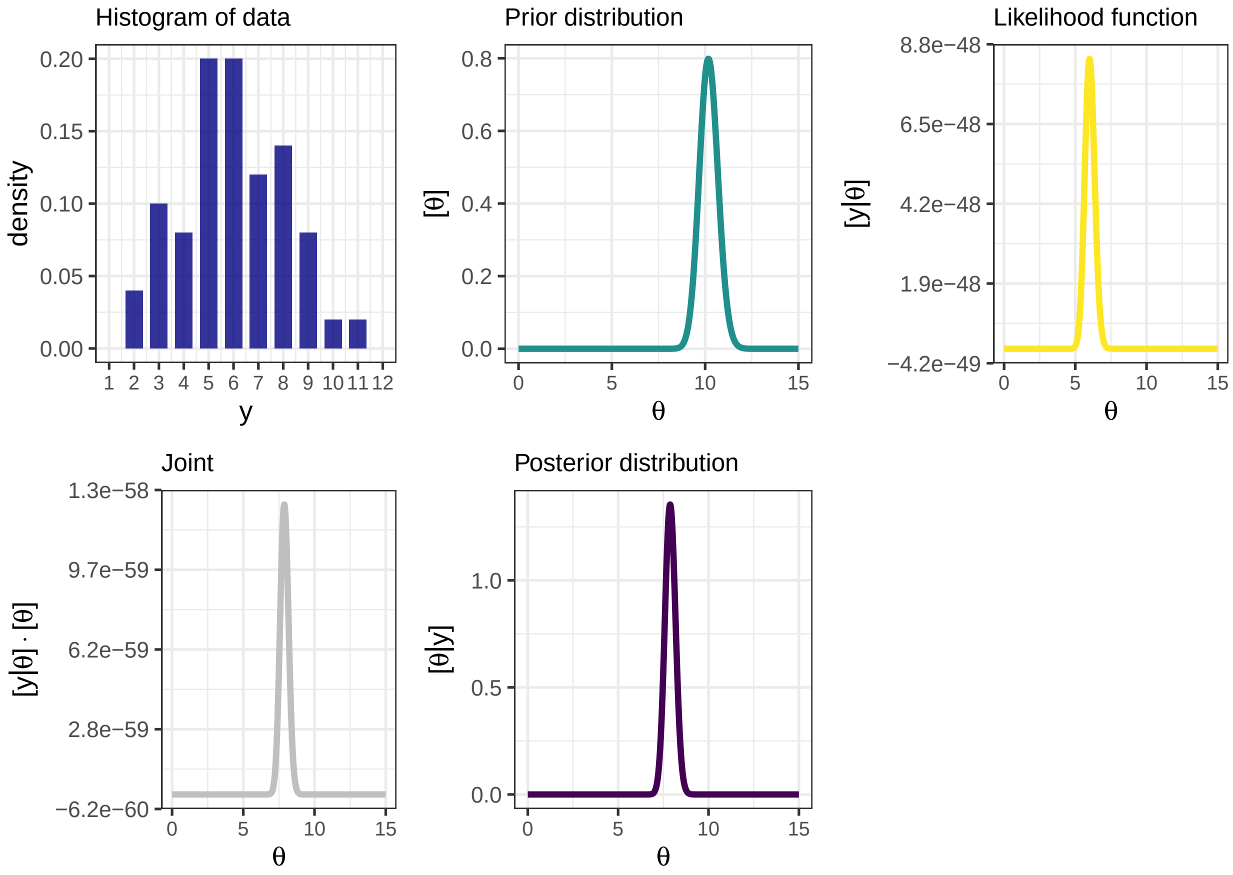


Putting it all together

13

Plot the prior, a histogram of the data, the likelihood, the joint, and the posterior in a six panel layout. Your results should be the same as the plot below:

```
# combine charts
  cowplot::plot_grid(
    plotlist =  list(plt_1, plt_2, plt_3, plt_4, plt_5)
      , nrow = 2
      , ncol = 3
    )
```



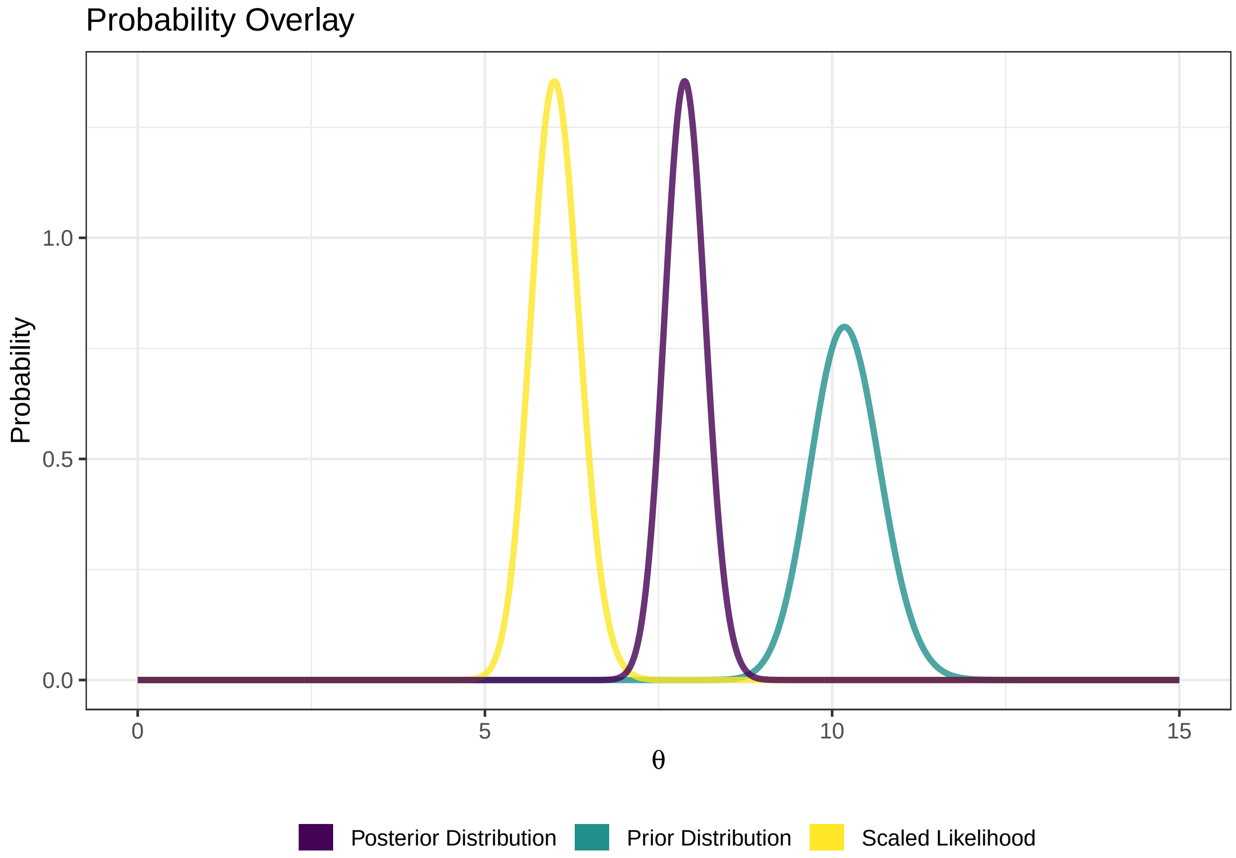
14

Overlay the prior, the likelihood, and the posterior on a single plot. To do this, you will need to rescale the likelihood, which of course is OK because it is defined up to an arbitrary, multiplicative constant, i.e., $[y \mid \theta] = cL(\theta \mid y)$. It doesn't matter what c is. We can rescale the likelihood to any value we like and the inference doesn't change because all evidence is relative in the likelihood framework. Do the following to obtain a scaled likelihood that can be plotted in a revealing way next to the posterior distribution. Divide each element in the likelihood vector by the maximum likelihood (thus, the maximum becomes 1). Multiply the resulting vector by the maximum value of the posterior density.

```
# scale likelihood
scaled_likelihood <- (likelihood_y_theta/max(likelihood_y_theta)) * max(posterior_dist)

#plot
plt_overlay <-
data.frame(
  theta
  , scaled_likelihood
  , posterior_dist
  , prior_theta
```

```r
) %>%
ggplot(., mapping = aes(x = theta)) +
  geom_line(mapping = aes(y = prior_theta, color = "Prior Distribution"), lwd = 1.2) +
  geom_line(mapping = aes(y = scaled_likelihood, color = "Scaled Likelihood"), lwd = 1.2) +
  geom_line(mapping = aes(y = posterior_dist, color = "Posterior Distribution"), lwd = 1.2) +
  xlab(latex2exp::TeX("$\\theta$")) +
  ylab("Probability") +
  labs(
    title = "Probability Overlay"
  ) +
  scale_color_viridis_d(alpha = 0.8) +
  theme_bw() +
  theme(
    legend.title = element_blank()
    , legend.position = "bottom"
    , legend.direction = "horizontal"
    , legend.justification = "center"
    , legend.box.just = "bottom"
  ) +
  guides(color = guide_legend(override.aes = list(size = 5)))
plt_overlay
```

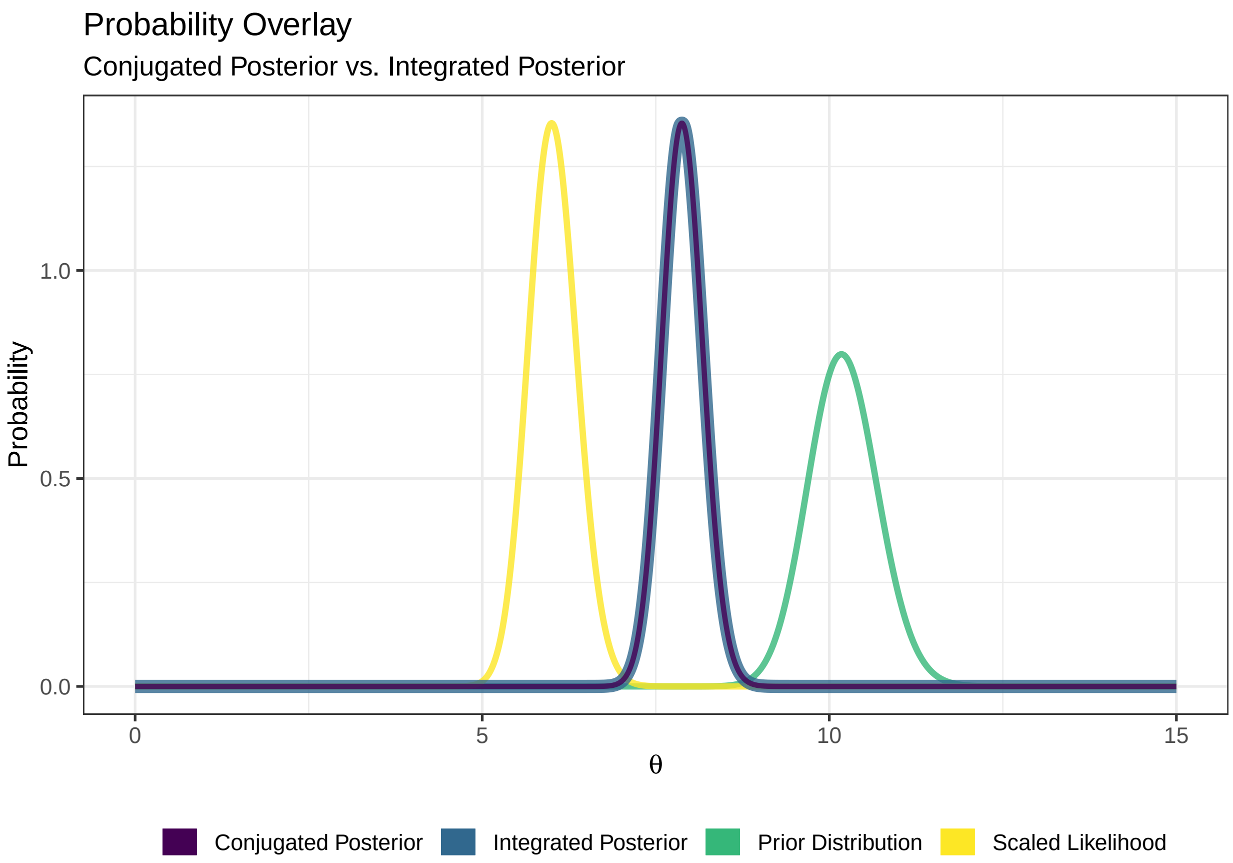


15

Check to be sure that everything is correct using the gamma-Poisson *conjugate* relationship. A gamma distribution is the conjugate for the Poisson likelihood, which means if we have a gamma prior and a Poisson likelihood, then the posterior is a gamma distribution with parameters $\alpha + \sum_{i=1}^{n} y_i$ and $\beta + n$, where α and β are the parameters of the gamma prior, and we have n observations (y_i) of new data. Overlay a plot of the posterior obtained using the conjugate on the plot of the posterior obtained by integration. Your plot should look like the one below. Take a look at your scaled overlay of the posterior, the likelihood, and the prior. The likelihood profile for θ is based on the data but it shows much less dispersion than the distribution of the data shown in the histogram you constructed. Why?

```r
# if we have a gamma prior and a Poisson likelihood
  # , then the posterior is a gamma distribution with the parameters...
  ## calculate the shape alpha for gamma dist
  alpha <- ( (mu.prior^2)/(sigma.prior^2) ) + sum(y)
  ## calculate the rate beta for gamma dist
  beta <- ( (mu.prior)/(sigma.prior^2) ) + length(y)
  ## probability density f'n
  posterior_gamma_dist <- dgamma(x = theta, shape = alpha, rate = beta)

#plot
plt_overlay2 <-
data.frame(
  theta
  , scaled_likelihood
  , posterior_dist
  , prior_theta
  , posterior_gamma_dist
) %>%
ggplot(., mapping = aes(x = theta)) +
  geom_line(mapping = aes(y = prior_theta, color = "Prior Distribution"), lwd = 1.2) +
  geom_line(mapping = aes(y = scaled_likelihood, color = "Scaled Likelihood"), lwd = 1.2) +
  geom_line(mapping = aes(y = posterior_dist, color = "Integrated Posterior"), lwd = 2.5) +
  geom_line(mapping = aes(y = posterior_gamma_dist, color = "Conjugated Posterior"), lwd = 1) +
  xlab(latex2exp::TeX("$\\theta$")) +
  ylab("Probability") +
  labs(
    title = "Probability Overlay"
    , subtitle = "Conjugated Posterior vs. Integrated Posterior"
  ) +
  scale_color_viridis_d(alpha = 0.8) +
  theme_bw() +
  theme(
    legend.title = element_blank()
    , legend.position = "bottom"
    , legend.direction = "horizontal"
    , legend.justification = "center"
    , legend.box.just = "bottom"
  ) +
  guides(color = guide_legend(override.aes = list(size = 5)))
plt_overlay2
```



16

Now that you have these lovely functions working and plots emerging from them, do some experiments to understand the effect of prior information on the posterior distribution of θ relative to the effect of the data. Increase the variance of the prior distribution to 2.5. Reduce it to .1. Increase the number of observations from 50 to 100. Examine the overlaid plots you produced above for each case.

Simulation function

```
sim_function <- function(n_sim, lambda_sim, theta_min, theta_max, theta_step, mu_prior, sigma_prior){
  # define theta
  theta_temp <- seq(theta_min, theta_max, theta_step)
  # simulate the y data
  y_temp <- rpois(n = n_sim, lambda = lambda_sim)
  # prior distribution
  prior_temp <- prior(theta = theta_temp, mu = mu_prior, sigma = sigma_prior)
  # likelihood
  likelihood_temp <- like(theta = theta_temp, y = y_temp)
  # joint distribution
  joint_temp <- joint(theta = theta_temp, y = y_temp, mu = mu_prior, sigma = sigma_prior)
  #  normalizing constant (i.e marginal probability of the data)
  marginal_p_temp <- sum(joint_temp * theta_step)
  # posterior distribution
  posterior_temp <- joint_temp / marginal_p_temp
  # scaled likelihood
  scaled_likelihood_temp <- (likelihood_temp/max(likelihood_temp)) * max(posterior_temp)
  # combine together in data frame
  dta_temp <- data.frame(
      theta = theta_temp
      , prior = prior_temp
      , likelihood = scaled_likelihood_temp
      , posterior = posterior_temp
    ) %>%
    dplyr::mutate(
      n = n_sim
      , lambda = lambda_sim
      , mu_prior = mu_prior
      , sigma_prior = sigma_prior
    )
  return(dta_temp)
}
```

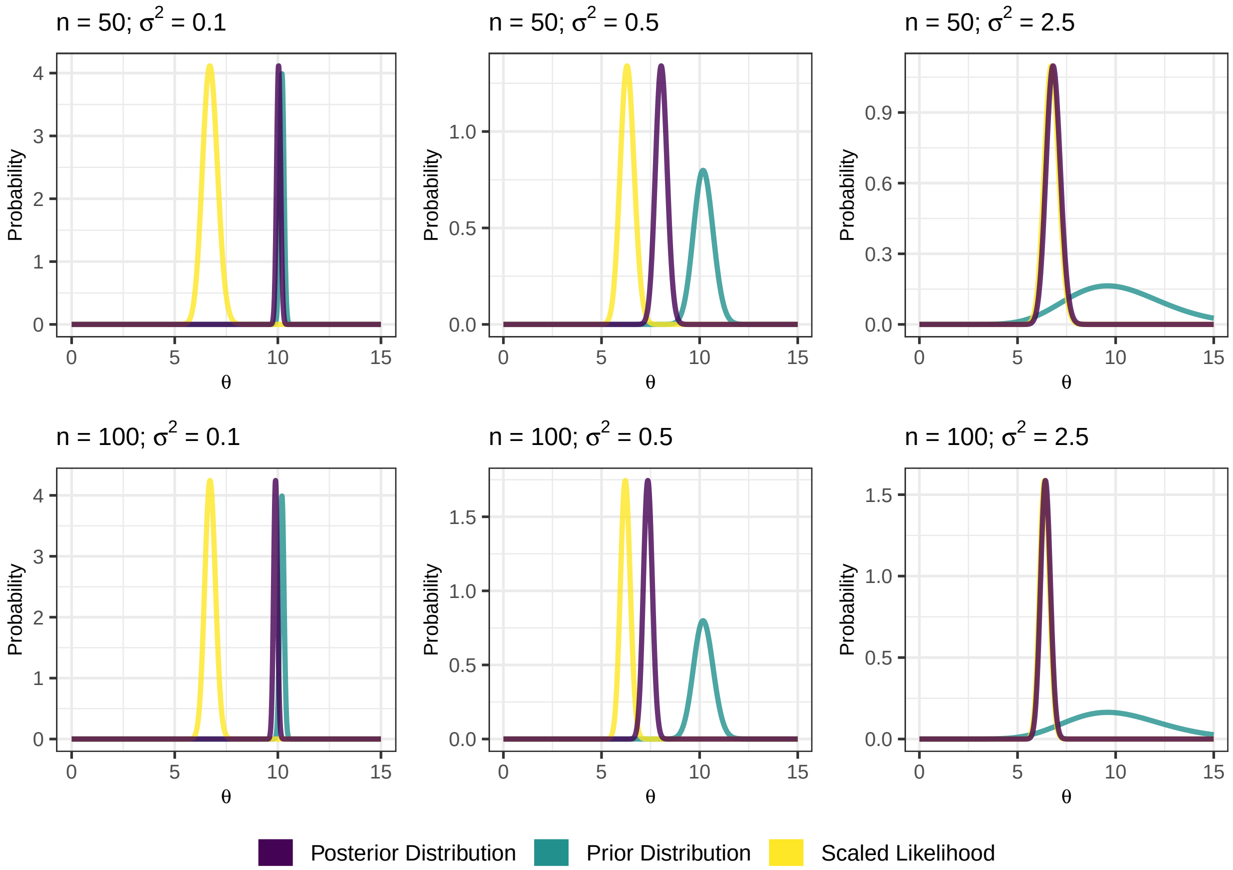
Simulation Plots

```
# data frame for sim
sim_dta <- data.frame(n = c(50, 100)) %>% # n
  dplyr::full_join(data.frame(sigma = c(0.1, 0.5, 2.5)), by = character()) %>%  # sigma
  dplyr::arrange(n, sigma)

# set up plot grid
plts <- list()
# create plot for each
for (i in 1:nrow(sim_dta)) {
  plts[[i]] <-
    sim_function(
      n_sim = sim_dta$n[i]
      , lambda_sim = 6.4
      , theta_min = 0
      , theta_max = 15
      , theta_step = step
      , mu_prior = mu.prior
      , sigma_prior = sim_dta$sigma[i]
    ) %>%
    ggplot(., mapping = aes(x = theta)) +
      geom_line(mapping = aes(y = prior, color = "Prior Distribution"), lwd = 1) +
      geom_line(mapping = aes(y = likelihood, color = "Scaled Likelihood"), lwd = 1) +
      geom_line(mapping = aes(y = posterior, color = "Posterior Distribution"), lwd = 1) +
      xlab(latex2exp::TeX("$\\theta$")) +
      ylab("Probability") +
      labs(
        title = bquote(paste(
          "n = "
          , .(sim_dta$n[i])
          , "; "
          , sigma^2
          , " = "
          , .(sim_dta$sigma[i])
        ))
      ) +
      scale_color_viridis_d(alpha = 0.8) +
      theme_bw() +
      theme(
        legend.title = element_blank()
        , legend.position="none"
        , plot.title = element_text(size = 10)
        , axis.text = element_text(size = 8)
        , axis.title = element_text(size = 8)
      ) +
      guides(color = guide_legend(override.aes = list(size = 5)))

}

# extract the legend from one of the plots
plts[[nrow(sim_dta)+1]] <- NA
plts[[nrow(sim_dta)+2]] <- get_legend(plt_overlay)
cowplot::plot_grid(plotlist =  plts, nrow = 3, rel_heights = c(1, 1, 0.15))
```


n = 50; σ^2 = 0.1
n = 50; σ^2 = 0.5
n = 50; σ^2 = 2.5
n = 100; σ^2 = 0.1
n = 100; σ^2 = 0.5
n = 100; σ^2 = 2.5
Probability
θ
■ Posterior Distribution
■ Prior Distribution
■ Scaled Likelihood