QPM II

Problem Set 3

Daniel Cher

2024-10-25

Problem 10

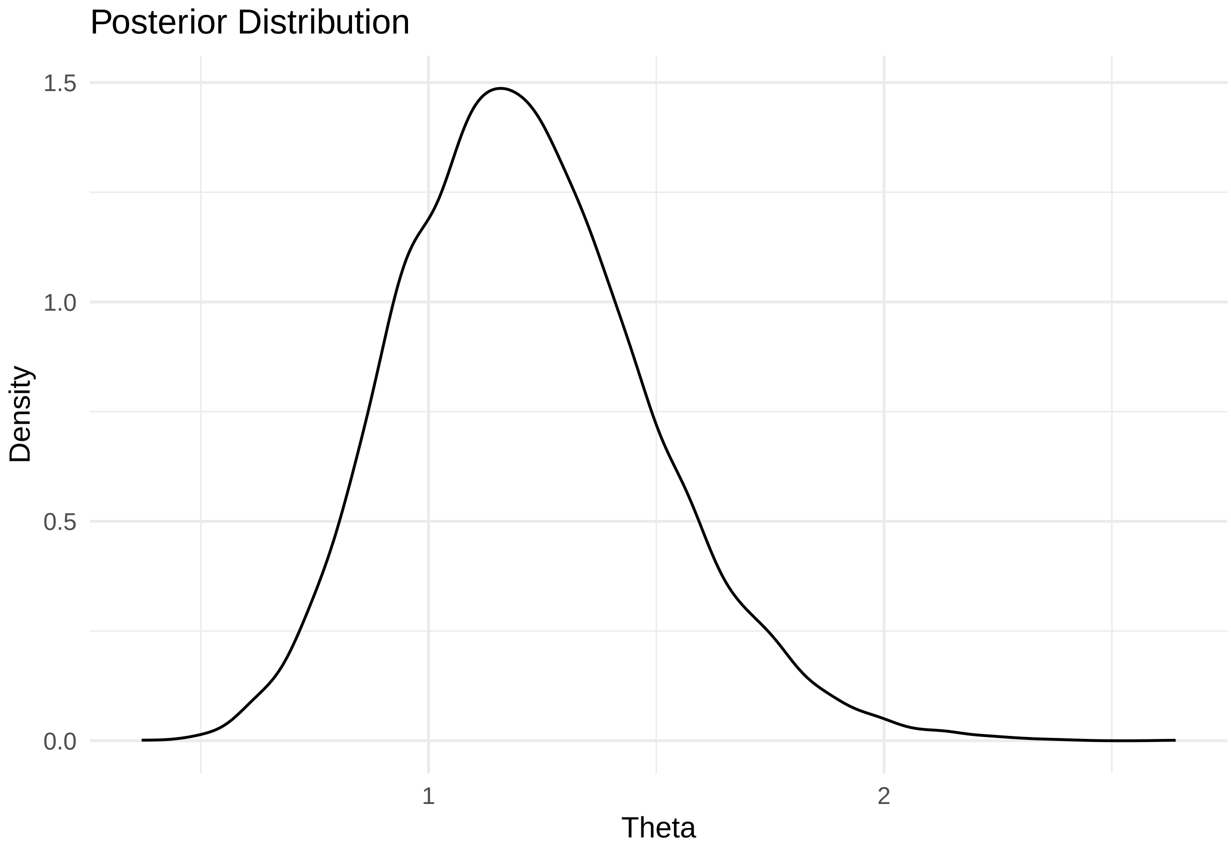
```r
# Parameters
alpha <- 15
beta <- 15

data <- c(3.59, 1.23, 1.47, 0.32, 2.09)

# Posterior
n <- length(data)
sum_log_data <- sum(log(data))

# Samples from posterior
posterior_samples <- rgamma(10000, shape = n + alpha, rate = beta + sum_log_data)

# Calculate 95% credible interval
credible_interval <- quantile(posterior_samples, probs = c(0.025, 0.975))
print(paste("95% Credible Interval: [", credible_interval[1], ", ", credible_interval[2], "]", sep = ""
```

```
## [1] "95% Credible Interval: [0.735377920763131, 1.80412651640224]"
```

```r
# Calculate 95% HPD interval
hpd_interval <- bayestestR::hdi(posterior_samples, ci = 0.95)
print(paste("95% HPD Interval: [", hpd_interval$CI_low, ", ", hpd_interval$CI_high, "]", sep = ""))
```

```
## [1] "95% HPD Interval: [0.704458038277993, 1.76115203362067]"
```

```r
# Posterior dist plot
ggplot() +
  geom_density(aes(x = posterior_samples)) +
  theme_minimal() +
  labs(title = "Posterior Distribution", x = "Theta", y = "Density")
```



Problem 11

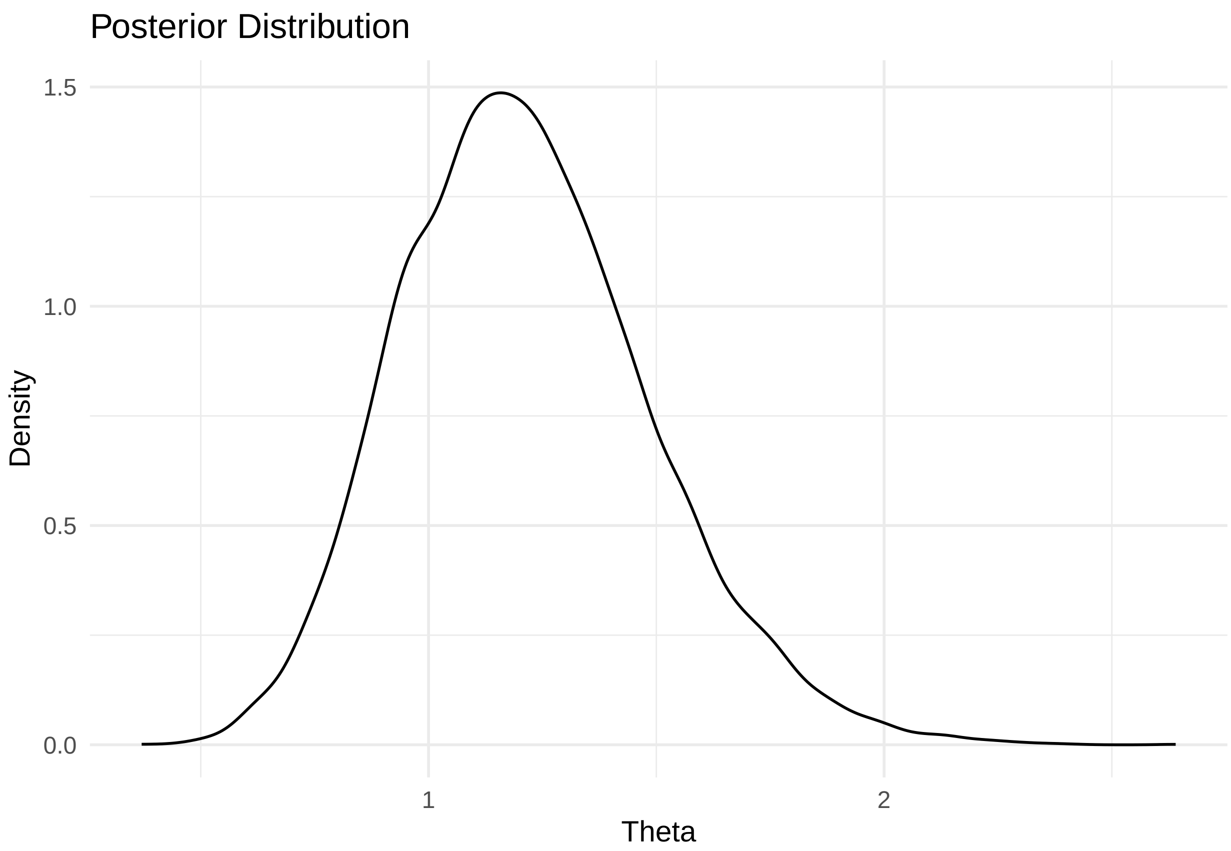
```
# Take samples from uninformed posterior
posterior_samples_un <- rgamma(10000, shape = n, rate = sum_log_data)

# Calculate 95% credible interval
credible_interval_un <- quantile(posterior_samples_un, probs = c(0.025, 0.975))
print(paste("95% Credible Interval: [", credible_interval_un[1], ", ", credible_interval_un[2], "]", sep
```

```
## [1] "95% Credible Interval: [1.10820694135715, 7.07290364946621]"
```

```
# Calculate 95% HPD interval
hpd_interval_un <- bayestestR::hdi(posterior_samples_un, ci = 0.95)
print(paste("95% HPD Interval: [", hpd_interval_un$CI_low, ", ", hpd_interval_un$CI_high, "]", sep = ""
```

```
## [1] "95% HPD Interval: [0.776059038505162, 6.4564941520768]"
```

```
# Posterior dist plot
ggplot() +
  geom_density(aes(x = posterior_samples)) +
  theme_minimal() +
  labs(title = "Posterior Distribution", x = "Theta", y = "Density")
```



Problem 12

```
# MLE
theta_mle <- n / sum_log_data

# SE
se_theta <- theta_mle / sqrt(n)

# 95% confidence interval
z_value <- 1.96
ci_lower <- theta_mle - z_value * se_theta
ci_upper <- theta_mle + z_value * se_theta

# Print results
cat("MLE of theta:", theta_mle, "\n")
```

```
## MLE of theta: 3.405627
```

```
cat("Standard Error:", se_theta, "\n")
```

```
## Standard Error: 1.523043
```

```
cat("95% Confidence Interval: [", ci_lower, ",", ci_upper, "]\n")
```

```
## 95% Confidence Interval: [ 0.4204633 , 6.39079 ]
```

The informed bayesian prior leads to a tighter distribution compared to the uninformed prior with the incorporation of prior knowledge , especially with a stronger prior (larger alpha & beta). In contrast, the MLE-based confidence interval is derived solely from the observed data, and with a relatively small sample size ($n = 5$), it's distribution is more comparable to the uninformed Bayesian distribution. As the sample size n increases, we would expect the MLE-based interval to narrow, but given current conditions, it's wide confidence-interval is a product of small sample size.

Problem 14

```
# Parameters
set.seed(42)
beta_true <- 5
theta_true <- 7

# Random draws from Weibull distribution
n <- 1000
data <- rweibull(n, shape = beta_true, scale = theta_true)

# Score f(n)
score_function <- function(beta, x) {
  n <- length(x)
  term1 <- sum(x^beta * log(x)) / sum(x^beta)
  term2 <- sum(log(x)) / n
  score <- ((1/(term1 - term2)) - beta)
  return(score)
}

# uniroot
output <- uniroot(score_function, lower = 0.1, upper = 10, x = data)
print(output$root)
```

```
## [1] 4.996324
```

Problem 15

```
# Data
x <- c(2, 4, 5, 3, 8, 10, -2, 1, -1, 2, 5, 5)
set.seed(42)

# Parameters
replications <- 2000

# Bootstrap
bootstrap_medians <- rep(NA, replications)
for (i in 1:replications) {
```

```r
  sample_x <- sample(x, length(x), replace = TRUE)
  bootstrap_medians[i] <- median(sample_x)
}

# Calculate Bootstrap SE
bootstrap_se <- sd(bootstrap_medians)

cat("Bootstrap SE for the median:", bootstrap_se, "\n")
```

```
## Bootstrap SE for the median: 1.202393
```

Problem 16

```r
# Jackknife: Leave-one-out approach to calculate the median
n <- length(x)
jackknife_medians <- rep(NA, n)

for (i in 1:n) {
  jackknife_sample <- x[-i]
  jackknife_medians[i] <- median(jackknife_sample)
}

# Calculate the Jackknife estimate of the SE
jackknife_mean <- mean(jackknife_medians)
jackknife_se <- sqrt((n - 1) * mean((jackknife_medians - jackknife_mean)^2))

cat("Jackknife SE for the median:", jackknife_se, "\n")
```

```
## Jackknife SE for the median: 1.658312
```