Problem Set 2

Brennan Stout

2025-09-23

1. Bernoulli

Assume that the data [1,1,1,1,1,1,0,0,0,1,1,1,0,1,0,0,1,1,1,1] are produced from iid Bernoulli trials. Produce a $1-\alpha$ credible set for the unknown value of p using a uniform prior distribution.

$p \sim \beta(1,1)$ Uniform prior assumption

Let:

- Trials = 20 = n
- Successes (1s) = 14 = x
- Failures (0s) = 6 = n - x

Posterior distribution = $\binom{15}{7}$

$$p|data\ \beta(1+x, 1+n-x) = \beta\binom{15}{7}$$

Credible interval for a confidence interval of 95% on an interval of $[a, b]$.

$$P(a \leq p \leq b|data) = 0.95$$

Unknown Bernoulli parameter p at 95% credible interval = (0.4782, 0.8541)

$a = \beta^{-1}(0.025; 15, 7)$

$b = \beta^{-1}(0.025; 15, 7)$

2. Beta Distribution

The beta distribution,

$$f(x(\alpha, \beta)) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

for $0 < 1, \alpha > 0$, and $\beta > 0$, is often used to model the probability parameter in a binomial setup. If you were very unsure about the prior distribution of p, what values would you assign to α and β to make it relatively "flat"?

For relatively flat the exponent terms of $\alpha - 1$ and $\beta - 1$ close to 0

*$\alpha = 1$

*$\beta = 1$

Beta(1,1) distribution, being uniform over (0, 1) and max uncertainty about p and giving equal probability to all values in the interval.

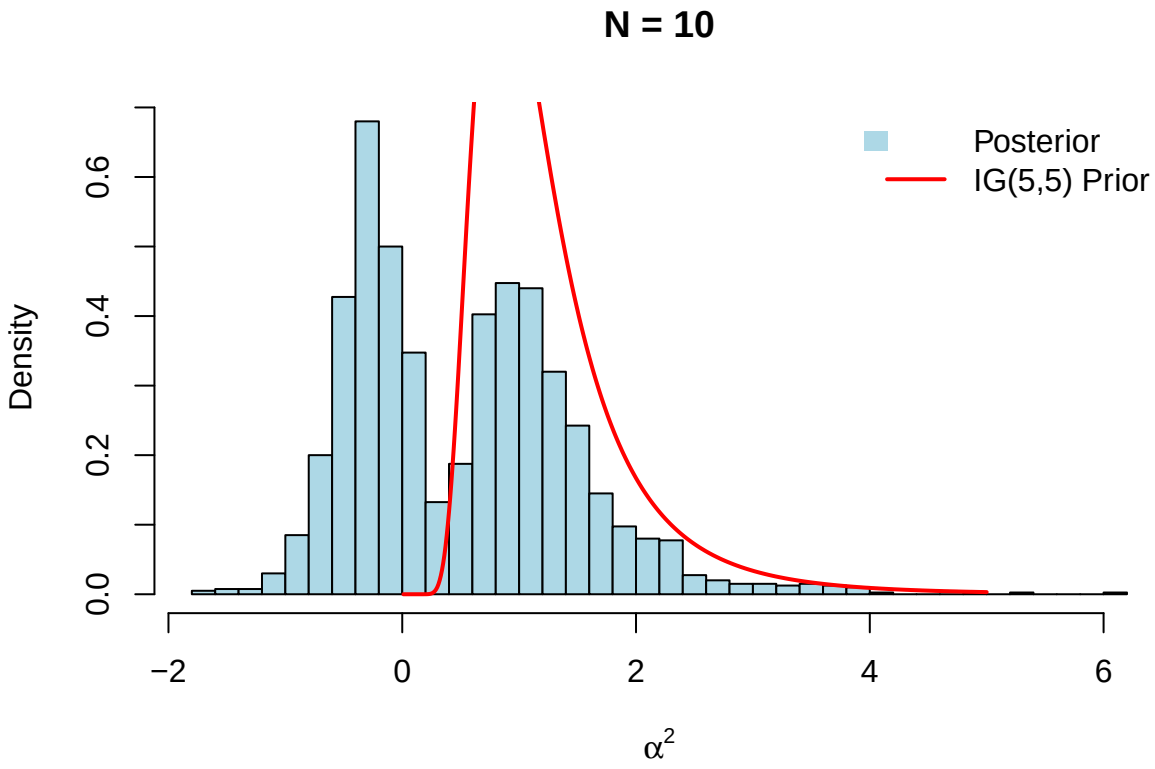
Standard non-informative prior.

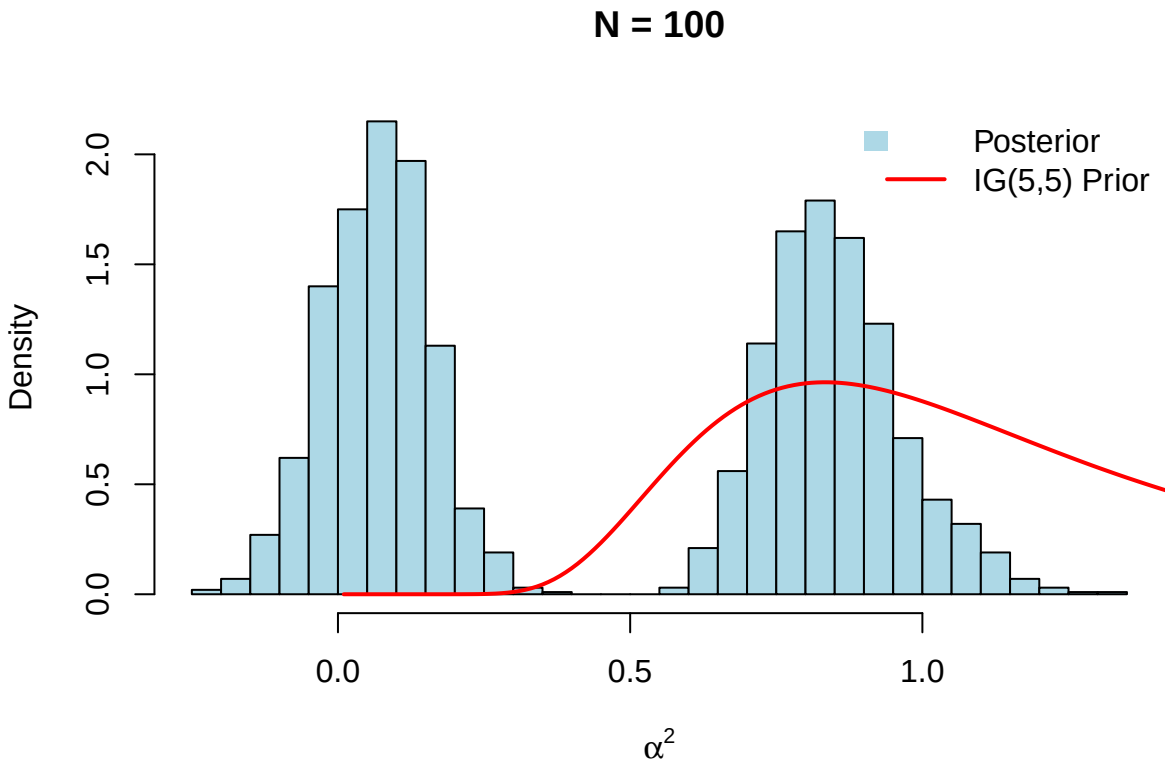
3. Unknown normal pdf inference problem

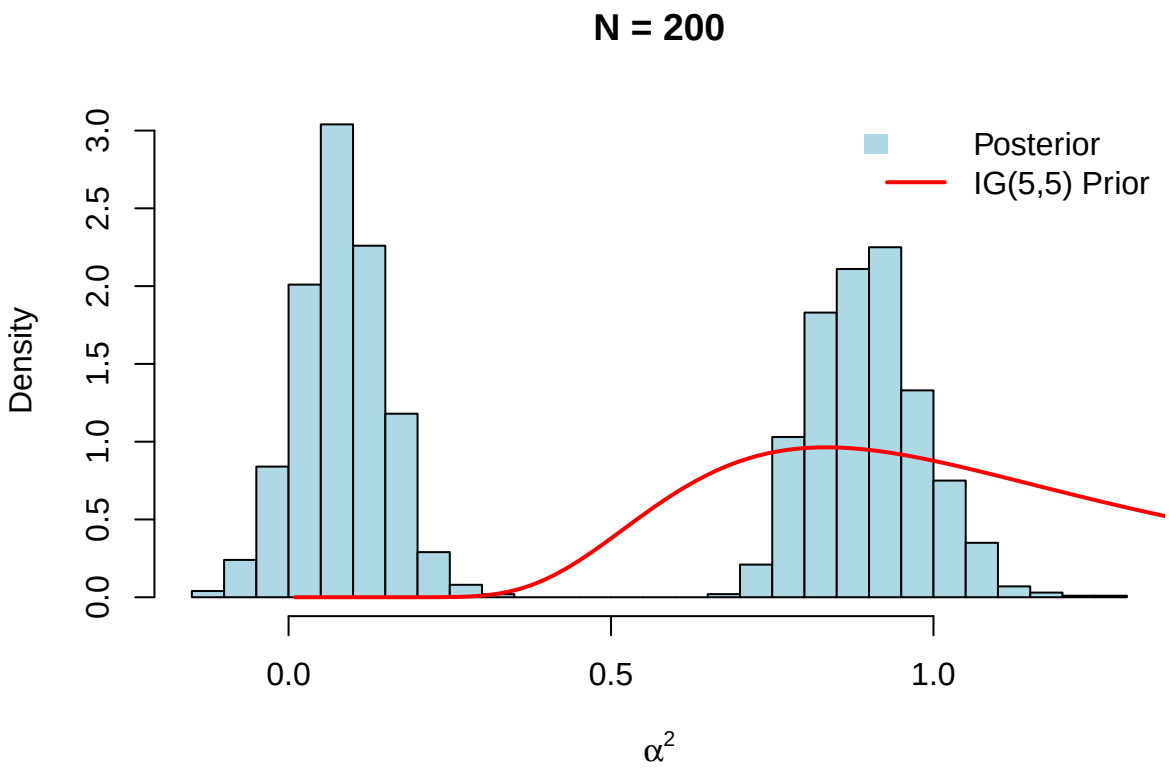
Start with the mean-variance unknown normal pdf inference problem (we covered the code for this in class). Now use this code to simulate (via a Gibbs sampler) the posterior for α^2. Then plot and compare it to the IG(5,5) case. Do this for samples of size $N = \{10,100,200,500\}$.

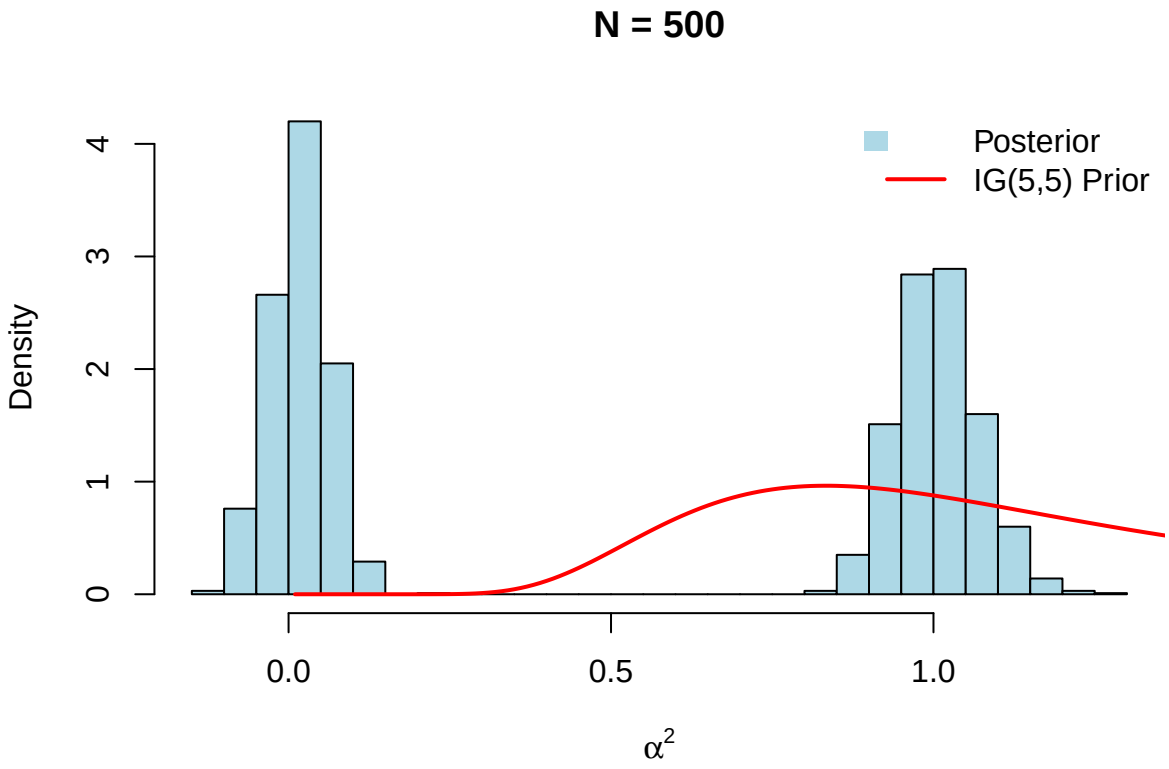
```
library(coda)
library(invgamma)
library(ggplot2)

sample.mean <- function(n, meanx, m, s0, sigma2)
{
    ns0 <- n+s0
    mean <- (n*meanx + m*s0)/ns0
    var <- sigma2/ns0
    return(rnorm(1, mean=mean, sd=sqrt(var)))
}

# Function to sample the variance
sample.variance <- function(alpha, beta, n, varx)
{
    shape <- alpha + 0.5*n - 0.5
    scale <- beta + 0.5*n*varx
    return(1/rgamma(1, shape, scale))
}
# Gibbs sampler for the blocks
gibbs.sampler <- function(N1, N2, x, m, s0, alpha, beta)
{
    # Setup storage
    output <- matrix(0, N2, 2)
    # Compute constants we need
    n <- length(x)
    varx <- var(x)
    meanx <- mean(x)
    # Loop for the burn-in iterations
    for (i in 1:N1)
    {
        sigma2 <- sample.variance(alpha, beta, n, varx)
        mu <- sample.mean(n, meanx, m, s0, sigma2)
    }
    # Loop for the final posterior sample
    for (i in 1:N2)
    {
        sigma2 <- sample.variance(alpha, beta, n, varx)
        mu <- sample.mean(n, meanx, m, s0, sigma2)

        output[i,] <- c(mu, sigma2)
```

```
    }
    colnames(output) <- c("mu", "sigma2")
    return(output)
}
# Now use this to sample the posterior
set.seed(1234)
N <- c(10,100,200,500)          # Generate data
m <- 1              # Parameters for the prior
alpha <- 1
beta <- 1
s0 <- 1

#par(mfrow = c(2, 2))
x <- seq(0.01, 5, length.out = 500)
ig_pdf <- dinvgamma(x, shape = 5, rate = 5)

for (n in N) {
  X <- rnorm(n)
  sigma2_samples <- gibbs.sampler(N1=100, N2=1000, x=X, m=m, s0=s0,
                            alpha=alpha, beta=beta)
  hist(sigma2_samples, breaks = 30, probability = TRUE,
       main = paste("N =", n), xlab = expression(alpha^2),
       col = "lightblue", border = "black")
  lines(x, ig_pdf, col = "red", lwd = 2)
  legend("topright", legend = c("Posterior", "IG(5,5) Prior"),
         bty = "n", fill = c("lightblue", NA),
         border = NA,
         lty = 1, col = c(NA, "red"),
         lwd = 2)
}
```


N = 10
Posterior
IG(5,5) Prior
Density
0.0
0.2
0.4
0.6
−2
0
2
4
6
α²


N = 100
Posterior
IG(5,5) Prior
Density
0.0
0.5
1.0
1.5
2.0
0.0
0.5
1.0
α^2


N = 200
Posterior
IG(5,5) Prior
Density
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0.0
0.5
1.0
α^2



3. Bayesian linear model

Develop a Bayesian linear model for the following data that describe the average weekly household spending on tobacco and alcohol (in pounds) for the eleven regions of the United Kingdom (Moore and McCabe 1989, originally from Family Expenditure Survey, Department of Employment, 1981, British Official Statistics). Specify both an informed conjugate and uninformed prior using the level for alcoholic beverages as the outcome variable and the level for tobacco products as the explanatory variable. Do you notice a substantial difference in the resulting posteriors? Describe. Start with the R code for the regression data:

From the dianostics, both regressions show good fits for the small sample size. From the plots, it is apparent that the uniformed prior is a much more symetric curve with a generally slightly lower mean than the informed prior (0.3 to 0.4). For the informed prior, the plot has a much more constrained curve but is less symmetric around the mean. The informed is probably the better model with its less extensive distribution.

```
# Set up data

Region <- c("Northern Ireland", "East Anglia", "Southwest",
  "East Midlands", "Wales", "West Midlands", "Southeast",
  "Scotland", "Yorkshire", "Northeast", "North")
d <- matrix(c(4.02, 4.56, 4.52, 2.92, 4.79, 2.71, 4.89, 3.34, 5.27,
  3.53, 5.63, 3.47, 5.89, 3.20, 6.08, 4.51, 6.13, 3.76,
  6.19, 3.77, 6.47, 4.03), ncol=2, byrow=TRUE)
# Now make the dataframe
d <- as.data.frame(list(Region=Region, Alcohol = d[,1],
  Tobacco = d[,2]))
```

```
#################
library(rstanarm)
```

```
## Loading required package: Rcpp
```

```
## This is rstanarm version 2.32.1
```

```
## - See https://mc-stan.org/rstanarm/articles/priors for changes to default priors!
```

```
## - Default priors may change, so it's safest to specify priors, even if equivalent to the defaults.
```

```
## - For execution on a local, multicore CPU with excess RAM we recommend calling
```

```
##   options(mc.cores = parallel::detectCores())
```

```
library(bayesplot)
```

```
## This is bayesplot version 1.14.0
```

```
## - Online documentation and vignettes at mc-stan.org/bayesplot
```

```
## - bayesplot theme set to bayesplot::theme_default()
```

```
##    * Does _not_ affect other ggplot2 plots
```

```
##    * See ?bayesplot_theme_set for details on theme setting
```

```
library(ggplot2)
set.seed(123)

# Fit Bayesian linear regression with null priors
model_uninformed <- stan_glm(Alcohol ~ Tobacco, data = d,
                             prior = NULL, prior_intercept = NULL,
                             chains = 4, iter = 2000, refresh = 0)
# Posterior summary
print(summary(model_uninformed))
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      Alcohol ~ Tobacco
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 11
##  predictors:   2
##
## Estimates:
```

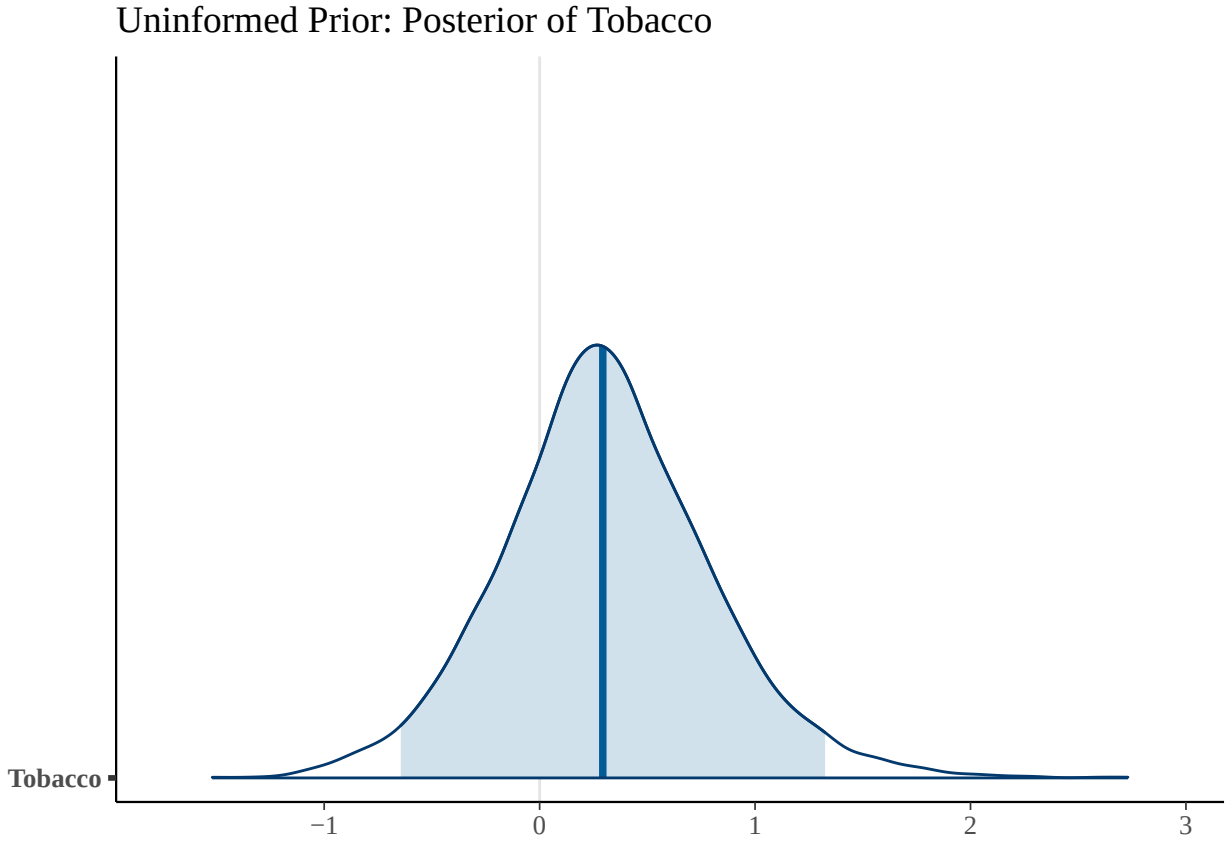
```
##               mean   sd   10%   50%   90%
## (Intercept)  4.3    1.8  2.1   4.4   6.5
## Tobacco      0.3    0.5 -0.3   0.3   0.9
## sigma        0.9    0.2  0.6   0.8   1.2
##
## Fit Diagnostics:
##           mean   sd   10%   50%   90%
## mean_PPD 5.4    0.4  5.0   5.4   5.9
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.0  1.0  2658
## Tobacco       0.0  1.0  2661
## sigma         0.0  1.0  2225
## mean_PPD      0.0  1.0  3427
## log-posterior 0.0  1.0  1602
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```r
# Use normal priors for slope and intercept, and exponential prior for sigma
model_informed <- stan_glm(Alcohol ~ Tobacco, data = d,
                           prior = normal(location = 0.5, scale = 0.2,
                            autoscale = TRUE),
                           prior_intercept = normal(location = 4, scale = 1,
                            autoscale = TRUE),
                           prior_aux = exponential(rate = 1),
                           chains = 4, iter = 2000, refresh = 0)
# Posterior summary
print(summary(model_informed))
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      Alcohol ~ Tobacco
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 11
##  predictors:   2
##
## Estimates:
##               mean   sd   10%   50%   90%
## (Intercept) 3.7    0.9  2.6   3.7   4.8
## Tobacco     0.4    0.2  0.2   0.4   0.7
## sigma       0.9    0.2  0.6   0.8   1.1
##
## Fit Diagnostics:
##           mean   sd   10%   50%   90%
## mean_PPD 5.3    0.4  4.8   5.3   5.8
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
```

```
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.0  1.0  3204
## Tobacco       0.0  1.0  3278
## sigma         0.0  1.0  2356
## mean_PPD      0.0  1.0  3607
## log-posterior 0.0  1.0  1672
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```r
# Extract posterior samples
posterior_uninformed <- as.matrix(model_uninformed)
posterior_informed <- as.matrix(model_informed)

# Plot comparison of slope (Tobacco coefficient)
mcmc_areas(posterior_uninformed, pars = "Tobacco", prob = 0.95) +
  ggtitle("Uninformed Prior: Posterior of Tobacco")
```




```r
mcmc_areas(posterior_informed, pars = "Tobacco", prob = 0.95) +
  ggtitle("Informed Prior: Posterior of Tobacco")
```

Informed Prior: Posterior of Tobacco