

Lecture: Bayesian Fundamentals

Denis Cohen*

Bayesian Fundamentals

The punchline

In the Bayesian world the unobserved quantities are assigned distributional properties and, therefore, become random variables in the analysis.

These distributions come in two basic flavors. If the distribution of the unknown quantity is not conditioned on fixed data, it is called prior distribution because it describes knowledge prior to seeing data.

Alternatively, if the distribution is conditioned on data that we observe, it is clearly updated from the unconditioned state and, therefore, more informed. This distribution is called posterior distribution. [...]

The punchline is this: All likelihood-based models are Bayesian models in which the prior distribution is an appropriately selected uniform prior, and as the size of the data gets large they are identical given any finite appropriate prior. So such empirical researchers are really Bayesian; they just do not know it yet.

Gill, J., & Witko, C. (2013). Bayesian analytical methods: A methodological prescription for public administration. *Journal of Public Administration Research and Theory*, 23(2), 457–494.

Likelihood function

- Specification of a pdf or pmf: $p(\mathbf{y}|\theta)$.
- Also called the data generating process (or the generative model) for y .

*Mannheim Centre for European Social Research, University of Mannheim, 68131 Mannheim, Germany. denis.cohen@uni-mannheim.de.

- Logical inversion: “Which unknown θ most likely produces the known \mathbf{y} ?” $\rightarrow L(\theta|\mathbf{y})$.
- The notational distinction between $p(\mathbf{y}|\theta)$ and $L(\theta|\mathbf{y})$ is purely conceptual. $p(\mathbf{y}|\theta) = L(\theta|\mathbf{y})$.
- We will use $p(\mathbf{y}|\theta)$.
- Note that the likelihood function multiplies densities across *all* observations; e.g., a normal likelihood function is given by:

$$p(\mathbf{y}|\mu, \sigma) = \prod_{i=1}^N \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-0.5 \left((y_i - \mu_i)^2/\sigma\right)\right)$$

- This is what we mean mathematically when we use the shorthand
 - $\mathbf{y} \sim N(\mu, \sigma)$ or
 - $y_i \sim N(\mu_i, \sigma)$ for all $i = 1, \dots, N$.

Prior distribution

- A distributional characterization of our belief about an unknown quantity (i.e., a parameter) prior to seeing the data: $p(\theta)$
- This includes statements about *family*, *support*, and *density*.
 - *Family*: A pdf (continuous parameters) or pmf (discrete parameters) that can plausibly generate the parameter values.
 - *Support*: Some parameters have constrained support: Probability parameters must be inside $[0, 1]$; variance parameters must be ≥ 0 .
 - *Density*: A distributional characterization which values of the parameter we think are more or less likely to observe.
- The prior distribution can be
 - flat (i.e., uniformly distributed over the supported range – often improper)
 - purposefully very vague, and thus, rather uninformative
 - weakly informative
 - specific and substantively informed (e.g., by previous research or expert assessment)

Posterior distribution

- Updating our distributional belief about θ given the data, \mathbf{y} : $p(\theta|\mathbf{y})$
- Follows the proportional version of Bayes' Law: $p(\theta|\mathbf{y}) \propto p(\theta) \times p(\mathbf{y}|\theta)$
- Yields a weighted combination of likelihood and prior
- The prior pulls the posterior density toward the center of gravity of the prior distribution
- As the data grows large, the likelihood becomes more influential:
 - one factor for $p(\theta)$, N factors for $p(y_i|\theta_i)$
 - we will see this analytically and using simulations later on

Coin flip experiment

The experiment

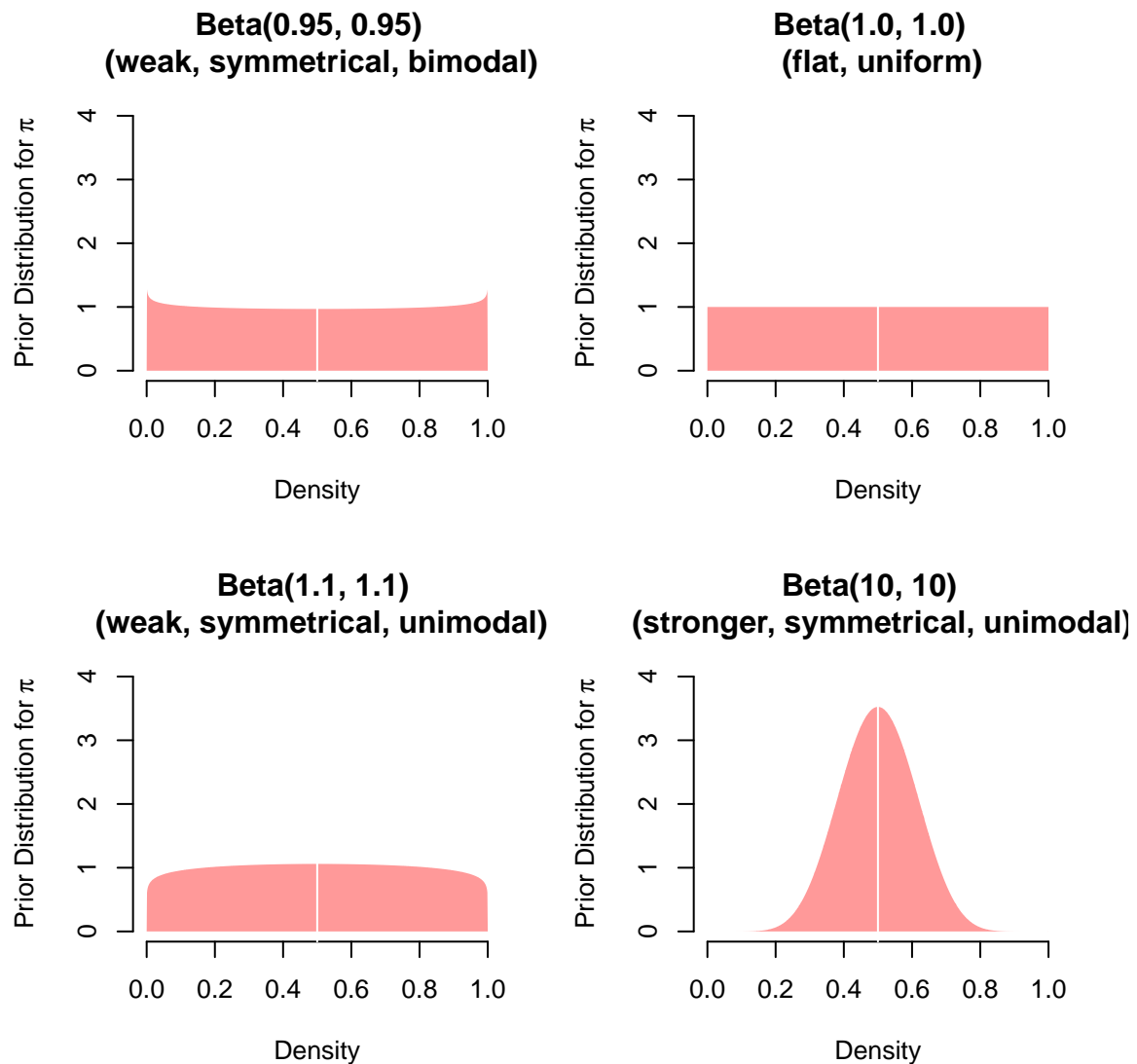
Suppose we flip a coin up to N times:

- The fairness of a coin can be expressed through a *probability parameter*, π , that governs the probability that a coin flip produces heads (1) has opposed to tails (0)
- We start out with the belief that the coin is fair – that is, we consider it more probable that the coin is fair ($\pi \approx 0.5$) and less probable that it systematically over-produces either heads or tails
- Unbeknownst to us, the coin is far from fair – it is 4 times as likely to produce heads as it is to produce tails (that is, $\pi = 0.8$)
- We slowly learn about this in the process of flipping the coin and keeping score of the number of flips n and the number of heads k ...

Analytical form: Prior distribution

- The *beta distribution* is a suitable candidate for characterizing our prior beliefs: $\pi \sim \text{beta}(a, b)$
- Characterized by two shape parameters, a and b
- a and b are *hyperparameters*: Known (or chosen) parameters that characterize a prior distribution.
- Constrained support: $\pi \in [0, 1]$

- pdf: $p(\pi) = \frac{\pi^{a-1}(1-\pi)^{b-1}}{B(a,b)}$



Analytical form: Likelihood

- Flipping one and the same coin n times is a series of Bernoulli trials
- The *binomial distribution* describes the corresponding data generating process: $k \sim \text{Binomial}(n, \pi)$
- pmf: $p(k|n, \pi) = \binom{n}{k} \pi^k (1 - \pi)^{(n-k)}$

Analytical form: Posterior distribution

Remember:

$$p(\theta|\mathbf{y}) \propto p(\theta) \times p(\mathbf{y}|\theta)$$

So what does this mean in the present example?

$$\begin{aligned} p(\pi|n, k) &\propto p(\pi) \times p(k|n, \pi) \\ p(\pi|n, k) &\propto \frac{\pi^{a-1}(1-\pi)^{b-1}}{B(a, b)} \times \binom{n}{k} \pi^k (1-\pi)^{(n-k)} \end{aligned}$$

Note that since we use the proportional version of Bayes' Law (i.e., we do not stipulate exact equality), we can drop any constant terms that do not involve our parameter of interest, π :

$$p(\pi|n, k) \propto \pi^{a-1}(1-\pi)^{b-1} \times \pi^k (1-\pi)^{(n-k)}$$

The rest, then, is easy: Following the rules of exponentiation, we add exponents for identical bases.

This gives us our posterior distribution for π :

$$p(\pi|n, k) \propto \pi^{a+k-1}(1-\pi)^{b+n-k-1}$$

As you see, our posterior has the exact same form as our prior. It is a beta distribution with updated parameters

- $a' = a + k - 1$
- $b' = b + n - k - 1$

This property is called *conjugacy*: Prior and posterior are of the same family.

Now, take a moment to think about our analytical solution for the updated parameter:

- What does it take for the data to dominate the prior?

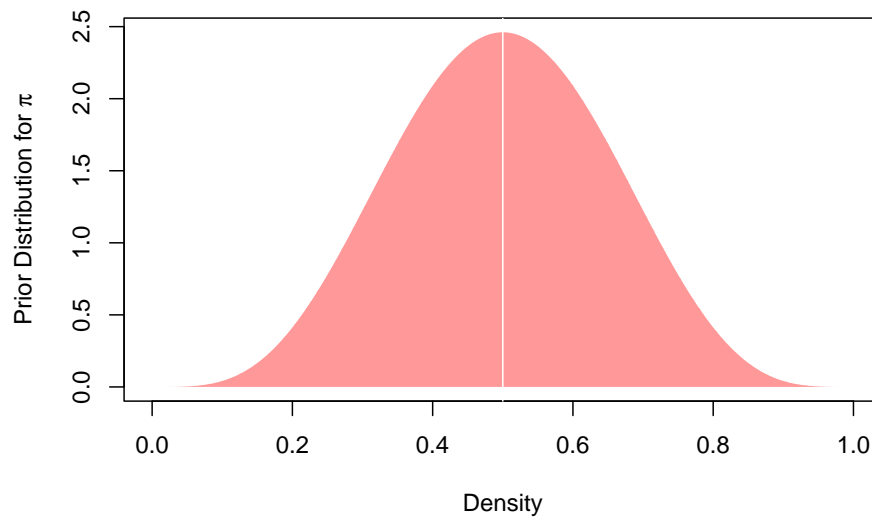
- What if the prior is weak (e.g., $\pi \sim \text{beta}(1, 1)$)?
- What if the prior is strong (e.g., $\pi \sim \text{beta}(100, 100)$)?

Simulation

Prior distribution Code: Defining and plotting the prior distribution

```
len_pi <- 1001L                                     ### number of candidate values for pi
pi <- seq(0, 1, length.out = len_pi) ### candidate values for pi
a <- b <- 5                                         ### hyperparameters
prior <- dbeta(pi, a, b)                           ### prior distribution

## Plot
plot(                                                ### set up empty plot, specify labels
  pi, prior,
  type = 'n',
  xlab = "Density",
  ylab = expression(paste("Prior Distribution for ", pi))
)
polygon(                                             ### draw density distribution
  c(rep(0, length(pi)), pi),
  c(prior, rev(prior)),
  col = adjustcolor('red', alpha.f = .4),
  border = NA
)
abline(                                             ### add vertical at pi = 0.5
  v = .5,
  col = 'white'
)
```



Posterior distribution Code: Simulating the experiment

```

set.seed(20210329)                ### set seed for replicability
len_pi <- 1001L                   ### number of candidate values for pi
pi <- seq(0, 1, length.out = len_pi) ### candidate values for pi
a <- b <- 5                       ### hyperparameters
n <- 300                          ### num. of coin flips
pi_true <- .8                     ### true parameter
data <- rbinom(n, 1, pi_true)     ### n coin flips
posterior <- matrix(NA, 3L, n)    ### matrix container for posterior

for (i in seq_len(n)) {
  current_sequence <- data[1:i]   ### sequence up until ith draw
  k <- sum(current_sequence)      ### number of heads in current sequence

  ##### Updating
  a_prime <- a + k
  b_prime <- b + i - k

```

```

### Analytical means and credible intervals
posterior[1, i] <- a_prime / (a_prime + b_prime)
posterior[2, i] <- qbeta(0.025, a_prime, b_prime)
posterior[3, i] <- qbeta(0.975, a_prime, b_prime)
}

## Plot
plot(
  1:n, 1:n,
  type = 'n',
  xlab = "Number of Coin Flips",
  ylab = expression(paste("Posterior Means of ",
                           pi,
                           sep = " ")),
  ylim = c(0, 1),
  xlim = c(1, n)
)
abline(
  h = c(.5, .8),
  col = "gray80"
)
rect(-.5, qbeta(0.025, 5, 5),
     0.5, qbeta(0.975, 5, 5),
     col = adjustcolor('red', .4),
     border = adjustcolor('red', .2))
segments(-.5, .5,
         0.5, .5,
         col = adjustcolor('red', .9),
         lwd = 1.5)
polygon(
  c(seq_len(n), rev(seq_len(n))),
  c(posterior[2, ], rev(posterior[3, ])),
  ### posterior means + intervals

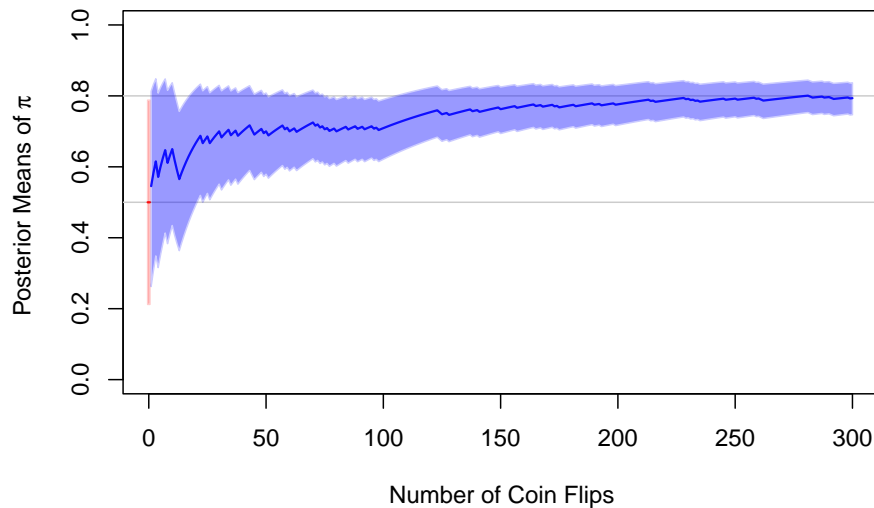
```



```

col = adjustcolor('blue', .4),
border = adjustcolor('blue', .2)
)
lines(
  seq_len(n),
  posterior[1, ],
  col = adjustcolor('blue', .9),
  lwd = 1.5
)

```



Note: After 300 coin flips, we have observed 241 heads, which is a proportion of 0.803. The posterior median is 0.794; the 95% credible interval is [0.747, 0.837].

MCMC algorithms

Analytical (classical) Bayesian inference

- As you may have noticed: Our coin flip example did *not* involve *any* numerical estimation algorithms.
- We simply observed the data, applied Bayes' Law, and analytically updated our parameters.

- This allowed us to retrieve a distributional characterization of our parameter of interest at each iteration of the coin flip series.
- The reasons why we could do this with ease is that this simple Binomial problem involved a single parameter π ; i.e, we were dealing with a uni-dimensional *parameter space*.

The limits of analytical Bayesian inference

- Even in only slightly more intricate applications, Bayesian inference involves finding a *joint* posterior for *all* parameters in a model, i.e., finding a *multi-dimensional* parameter space.
- Inference on single parameters from a joint multi-dimensional parameter space requires that we retrieve the marginal posterior distribution from the joint posterior distribution.
- Marginalizing the joint multidimensional posterior distribution w.r.t. to a given a parameter gives the posterior distribution for that parameter. This requires *integrating* out all other parameters.
- For instance, when our joint posterior in a three-dimensional parameter space is $p(\alpha, \beta, \gamma)$, we need to obtain each marginal posterior akin to $p(\alpha) = \int_{\beta} \int_{\gamma} p(\alpha, \beta, \gamma) d\beta d\gamma$
- For complex multi-dimensional posterior distributions, finding analytical solutions through integration becomes cumbersome, if not outright impossible.

Numerical approximation via MCMC

That's where numerical approximation through Markov Chain Monte Carlo (MCMC) algorithms comes in:

- MCMC are iterative computational processes that explore and describe a posterior distribution.
- Developed in the 1980s and popularized in the 1990s, MCMC algorithms quickly eliminated the need for analytical marginalizations of single parameters from joint multi-dimensional posteriors.
- The core idea:
 - *Markov Chains* wander through, and take samples from, the parameter space. Following an initial warmup period, the Markov Chains will converge to high-density regions of the underlying posterior distribution (ergodicity).

- The proportion of “steps” in a given region of multidimensional parameter space gives a stochastic simulation of the posterior probability density.
- This yields a numerical approximation of the underlying posterior distribution, much like Monte Carlo simulations of MLE parameters yield numerical approximations of the underlying sampling distribution.

(Some) MCMC Algorithms

1. **Gibbs:** Draws iteratively and alternatively from the conditional conjugate distribution of each parameter.
2. **Metropolis-Hastings:** Considers a single multidimensional move on each iteration depending on the quality of the proposed candidate draw.
3. **Hamiltonian Monte Carlo (HMC),** used in Stan:

The Hamiltonian Monte Carlo algorithm starts at a specified initial set of parameters θ ; in Stan, this value is either user-specified or generated randomly. Then, for a given number of iterations, a new momentum vector is sampled and the current value of the parameter θ is updated using the leapfrog integrator with discretization time ϵ and number of steps L according to the Hamiltonian dynamics. Then a Metropolis acceptance step is applied, and a decision is made whether to update to the new state (θ^*, ρ^*) or keep the existing state.

Source: Stan Reference Manual, Section 14.1

Implementing a Gibbs sampler

In a nutshell

- We want to perform inference on a variable y , of which we have N observations.
- We stipulate that the data-generating process that produces y is normal: $\mathbf{y} \sim N(\mu, \sigma^2)$. This yields a two-dimensional parameter space.
- For reasons of convenience, we parameterize the variance of this normal distribution in terms of its precision $\tau = \frac{1}{\sigma^2}$, not in terms of its standard deviation or variance.
- Note, however, that `rnorm()` in R uses the standard deviation, which is `sqrt(1 / tau)`.

- We will use a *Gibbs sampler*. Remember that Gibbs draws iteratively and alternatively from the conditional conjugate distribution of each parameter.
- We thus need some analytical preliminaries: Namely, analytical forms for the posterior distributions of the two parameters from whose marginal posteriors we would like to sample.
- This will *not* involve marginalizing out the “unwanted” parameters; instead, we will derive the posteriors of μ and τ as conditional functions of τ and μ , respectively

Application

- Specifically, we will focus on the variable `sup_afd` from the data set `gles`.
- Let’s pretend our prior belief is completely naive:
 - We don’t know how (un)popular the AfD is in the German electorate
 - But we know that individual support is measured on a -5 to 5 scale
 - Our prior belief for μ should thus be agnostic as to whether people like or dislike the AfD and sufficiently vague to allow for the possibility that we may be wrong: $\mu \sim N(\theta = 0, \omega^{-1} = 10)$ (mean θ and precision ω are hyperparameters for the prior of μ)
 - Our prior belief for τ will also be vague: $\tau \sim \Gamma(\alpha = 20, \beta = 200)$ (shape α and rate β are hyperparameters for the prior of τ)
 - We have no prior belief about the dependence of both parameters and hence specify independent prior distributions

Analytical preliminaries: μ

Our prior belief is that μ is distributed normal with mean $\theta = 0$ and precision $\omega = .1$:

$$\mu \sim N(0, 10)$$

The prior pdf is given by:

$$p(\mu|\theta, \omega) = \sqrt{\frac{\omega}{2\pi}} \exp\left(-\frac{\omega(\mu - \theta)^2}{2}\right)$$

while the likelihood for the data \mathbf{y} is given by

$$p(\mathbf{y}|\mu, \tau) = \prod_i^N \sqrt{\frac{\tau}{2\pi}} \exp\left(-\frac{\tau(y_i - \mu)^2}{2}\right)$$

Multiplying prior and likelihood and performing some algebraic transformations, we see that our conditional posterior density will be

$$p(\mu|\theta, \omega, \tau, \mathbf{y}) \propto \exp\left(-\frac{\omega + N\tau}{2} \left(\mu - \frac{\omega\theta + N\tau\bar{y}}{\omega + N\tau}\right)^2\right)$$

We recognize this as the normal pdf with updated mean parameter $\theta^* = \frac{\omega\theta + N\tau\bar{y}}{\omega + N\tau}$ and updated precision parameter $\omega^* = \omega + N\tau$.

This gives us the required analytical solutions for the normal parameters that characterize the posterior density of μ .

Analytical preliminaries: τ

Furthermore, for our prior knowledge about the precision, we assume that τ is Gamma-distributed with shape $\alpha = 20$ and rate $\beta = 200$: $\tau \sim \Gamma(20, 200)$ which yields the prior pdf:

$$p(\tau|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \tau^{\alpha-1} \exp(-\beta\tau)$$

while the likelihood for the data is still given by

$$p(\mathbf{y}|\mu, \tau) = \prod_i^N \sqrt{\frac{\tau}{2\pi}} \exp\left(-\frac{\tau(y_i - \mu)^2}{2}\right)$$

Once again taking the product and rearranging, we find that the conditional posterior pdf of τ is given by

$$p(\tau|\alpha, \beta, \mu, \mathbf{y}) \propto \tau^{\alpha + \frac{N}{2} - 1} \exp\left(-\left(\beta + \sum_{i=1}^N \frac{(y_i - \mu)^2}{2}\right)\tau\right)$$

This is a gamma distribution with updated parameters $\alpha^* = \alpha + \frac{N}{2}$ and $\beta^* = \beta + \sum_{i=1}^N \frac{(y_i - \mu)^2}{2}$. Thus, we also have analytical solutions for the Gamma parameters that characterize the posterior density of τ .

Simulating the independent prior distributions

Code: Function for simulating the priors

```
# Function
draw_from_prior <-
  function(theta,
            omega,
            alpha,
            beta,
            n_draws,
            seed = 20210329) {
    # Set seed
    set.seed(seed)

    # Take draws
    mu <- rnorm(n_draws, theta, 1 / sqrt(omega))
    tau <- rgamma(n_draws, alpha, beta)

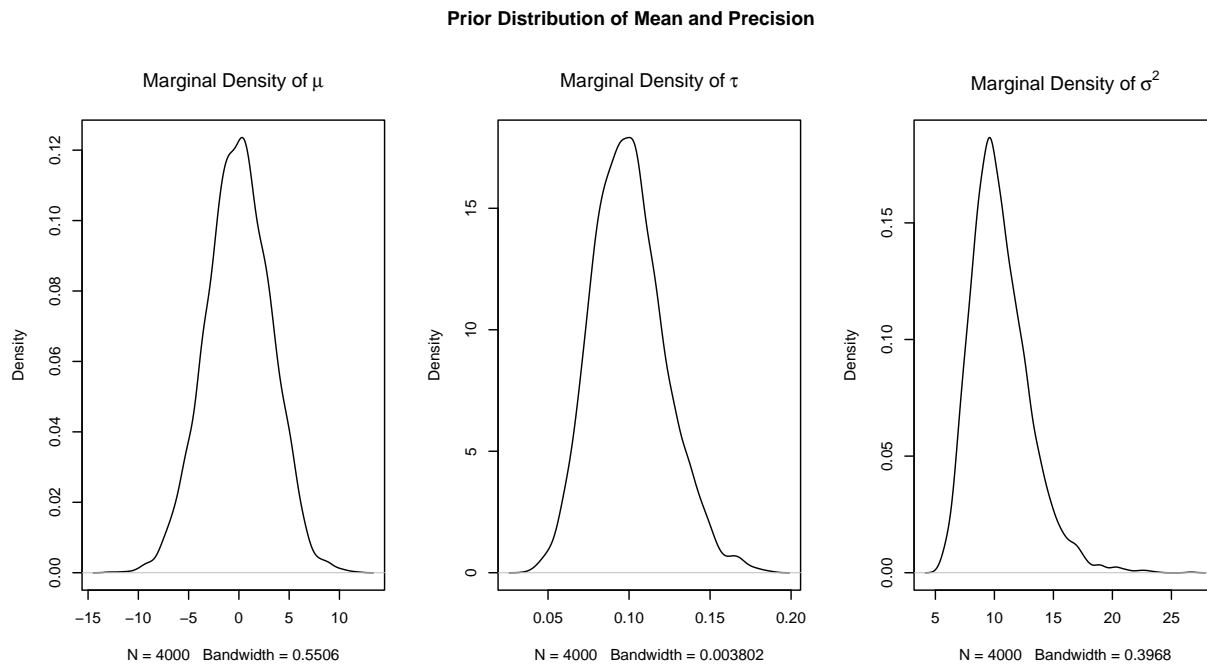
    ## Return output
    return(list(mu = mu,
                tau = tau))
  }
```

```
# Apply function
draws_prior <-
  draw_from_prior(
    theta = 0,
    omega = .1,
    alpha = 20,
    beta = 200,
    n_draws = 4000
  )
```

```

# Plots of Marginal Densities
par(mfrow = c(1, 3), oma = c(0, 0, 3, 0))
plot(density(draws_prior$mu),
     main = expression("Marginal Density of" ~ mu))
plot(density(draws_prior$tau),
     main = expression("Marginal Density of" ~ tau))
plot(density(1 / draws_prior$tau),
     main = expression("Marginal Density of" ~ sigma^2))
title("Prior Distribution of Mean and Precision", outer = T)

```



Implementing the Gibbs sampler for the posterior

Code: Gibbs sampler for the posterior

```

# Define function
draw_from_posterior <- function(theta,
                                omega,

```

```

        alpha,
        beta,
        n_warmup,
        n_draws,
        data,
        seed = 20210329,
        keep_warmup = TRUE) {

# Set seed
set.seed(seed)

# Length of chain
len_chain <- n_warmup + n_draws

# Data characteristics
n_data <- length(data)
mean_data <- mean(data)

# Initialize containers
mu <- rep(NA, len_chain)
tau <- rep(NA, len_chain)

# Run Gibbs sampler
for (i in seq_len(len_chain)) {
  if (i == 1) {
    ## Iteration 1: Initialize from prior
    alpha_star <- alpha
    beta_star <- beta
  } else {
    ## Iterations 2+: Update alpha and beta
    alpha_star <- alpha + n_data / 2
    beta_star <- beta + sum(((data - mu[i - 1]) ^ 2) / 2)
  }
}

```



```

## Sample tau
tau[i] <- rgamma(1, alpha_star, beta_star)

## Update theta and omega
theta_star <-
  (omega * theta + n_data * tau[i] * mean_data) /
  (omega + n_data * tau[i])
omega_star <- omega + n_data * tau[i]

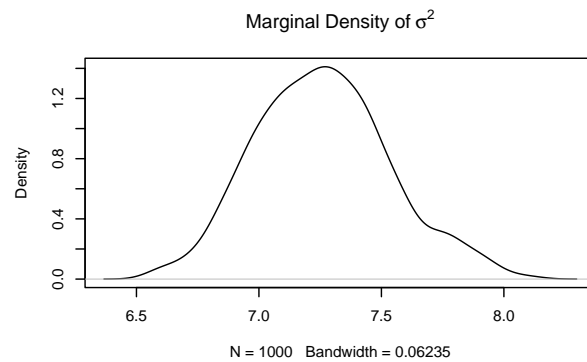
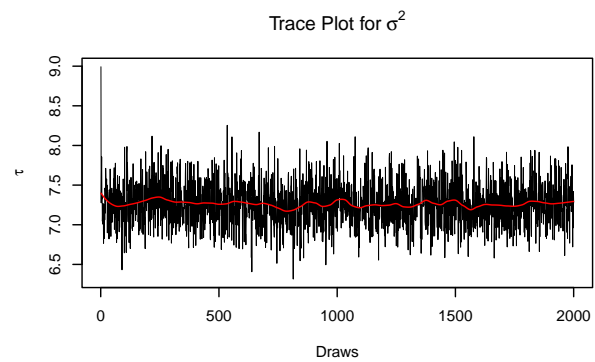
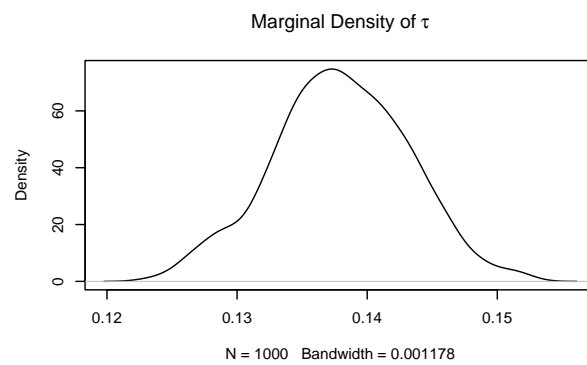
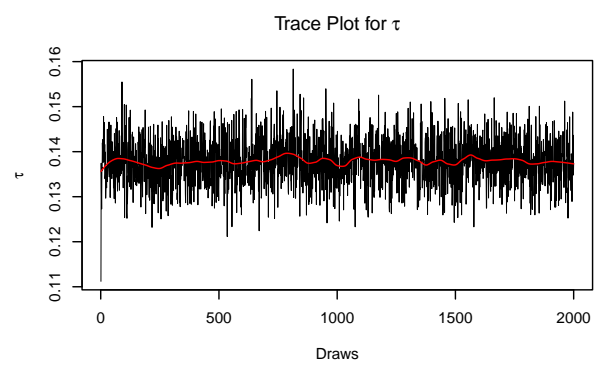
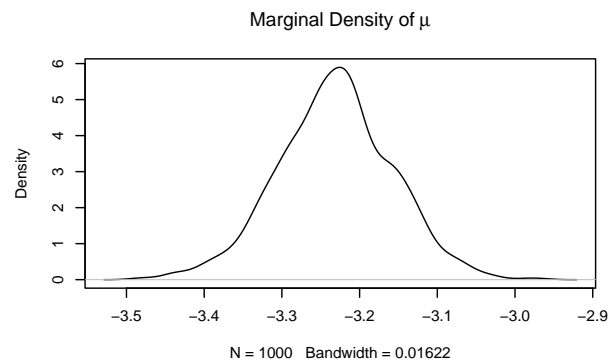
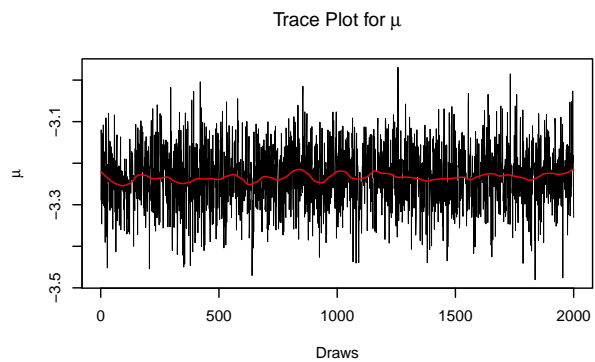
## Sample mu
mu[i] <- rnorm(1, theta_star, 1 / sqrt(omega_star))
}

## Conditionally discard warmup-draws
if (!keep_warmup) {
  tau <- tau[(n_warmup + 1):len_chain]
  mu <- mu[(n_warmup + 1):len_chain]
}

## Return output
return(list(mu = mu,
            tau = tau))
}

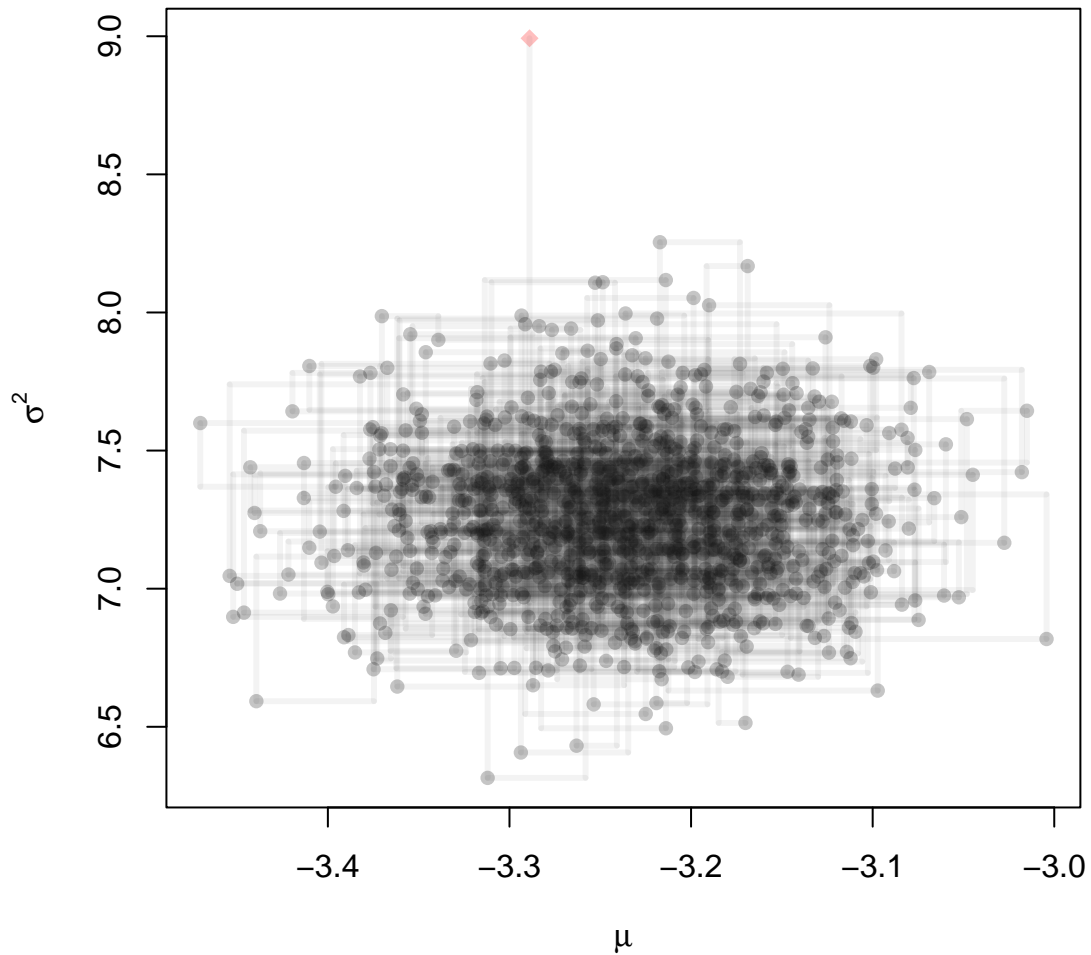
```

Posterior Distribution of Mean and Precision



How the sampler explores the joint posterior density

Exploration of the joint posterior



Convergence diagnostics

Why diagnose?

MCMC algorithms use iterative algorithms to explore posterior distributions and to produce numerical approximations thereof.

However, even with appropriately specified models and algorithms, we can never know a priori if and when a chain has converged to its target distribution. We must thus rely on *convergence diagnostics*.

Important: Convergence diagnostics cannot show or prove convergence. They can only show signs of non-convergence!

To conclude that the post-warmup draws of our sampler in fact explore the target distribution, we want to show at least two things:

1. Every chain is in a stationary state (i.e., does not “wander off” the target distribution)
2. Multiple independent chains are in the same stationary state (i.e., no convergence to different target distributions given identical data)

Generic diagnostics

Generic diagnostics (see Gill 2015, Ch. 14.3) include:

1. **Potential scale reduction statistic** \hat{R} (aka Gelman-Rubin convergence diagnostic)

$$\widehat{Var}(\theta) = \left(1 - \frac{1}{n_{\text{iter}}}\right) \underbrace{\left(\frac{1}{n_{\text{chains}}(n_{\text{iter}} - 1)} \sum_{j=1}^{n_{\text{chains}}} \sum_{i=1}^{n_{\text{iter}}} (\theta_{ij} - \bar{\theta}_j)^2\right)}_{\text{Within chain var}} + \frac{1}{n_{\text{iter}}} \underbrace{\left(\frac{n_{\text{iter}}}{n_{\text{chains}} - 1} \sum_{j=1}^{n_{\text{chains}}} (\bar{\theta}_j - \bar{\bar{\theta}})^2\right)}_{\text{Between chain var}}$$

- low values indicate that chains are stationary (convergence to target distribution within chains)
 - low values indicate that chains mix (convergence to same target distribution across chains)
2. **Geweke Time-Series Diagnostic:** Compare non-overlapping post-warmup portions of each chain to test within-convergence
 3. **Heidelberger and Welch Diagnostic:** Compare early post-warmup portion of each chain with late portion to test within-convergence
 4. **Raftery and Lewis Integrated Diagnostic:** Evaluates the full chain of a pilot run (requires that `save_warmup = TRUE`) to estimate minimum required length of warmup and sampling

These are implemented as part of the `coda` package (Output Analysis and Diagnostics for MCMC).

Visual diagnostics

The most widespread visual diagnostics are:

1. **Traceplots:** Visually inspect if chains are stationary and have converged to the same distribution
2. **Autocorrelation plots:** Visually inspect if the chain is sluggish in exploring the parameter space.

Application

In the following, we will use multiple chain runs of our sampler in conjunction with the `coda` package to check for signs of non-convergence.

Note that `coda` functions require that we combine our chains into `mcmc.list` objects.

Raftery and Lewis Integrated Diagnostic

The Raftery-Lewis diagnostic takes a single chain, including warm-up draws, to estimate the minimum required length of warmup and sampling runs:

```
##
## Quantile (q) = 0.5
## Accuracy (r) = +/- 0.0125
## Probability (s) = 0.95
##
##      Burn-in  Total Lower bound  Dependence
##      (M)      (N)   (Nmin)      factor (I)
## mu  2        6278  6147         1.020
## tau 2        5906  6147         0.961
```

Gelman-Rubin, Geweke, and Heidelberger-Welch diagnostics

We will use the recommended run-length from the Raftery-Lewis diagnostic for four independent runs of our sampler.

We will ensure that our chains run independently (i.e., using different starting values and different random number sequences) by setting different seed:

```
seeds <- sample(10001:99999, 4)
draws_multiple_chains <- lapply(seeds,
                                function(seed) {
                                  as.mcmc(simplify2array(
                                    draw_from_posterior(
                                      theta = 0,
                                      omega = .1,
                                      alpha = 20,
                                      beta = 200,
                                      n_warmup = 200,
                                      n_draws = 6147,
                                      data = gles$sup_afd,
                                      keep_warmup = FALSE,
                                      seed = seed
                                    )
                                  ))
                                })

# Save as mcmc.list
draws_multiple_chains <- as.mcmc.list(draws_multiple_chains)
```

```
## Potential scale reduction factors:
```

```
##
```

```
##      Point est. Upper C.I.
```

```
## mu          1          1
```

```
## tau         1          1
```

```
##
```

```
## Multivariate psrf
```

```
##
```

```
## 1
```

```

## [[1]]
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##      mu      tau
## 0.9199 0.7017
##
##
## [[2]]
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##      mu      tau
## -0.1502 0.2656
##
##
## [[3]]
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##      mu      tau
## -0.1191 -0.9184
##
##
## [[4]]
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##

```

```

##      mu      tau
## 1.194 1.114

## [[1]]
##
##      Stationarity start      p-value
##      test          iteration
## mu  passed          1          0.204
## tau passed          1          0.695
##
##      Halfwidth Mean    Halfwidth
##      test
## mu  passed    -3.231 0.00182
## tau passed     0.138 0.00013
##
## [[2]]
##
##      Stationarity start      p-value
##      test          iteration
## mu  passed          1          0.771
## tau passed          1          0.332
##
##      Halfwidth Mean    Halfwidth
##      test
## mu  passed    -3.230 0.001855
## tau passed     0.138 0.000134
##
## [[3]]
##
##      Stationarity start      p-value
##      test          iteration
## mu  passed          1          0.388

```

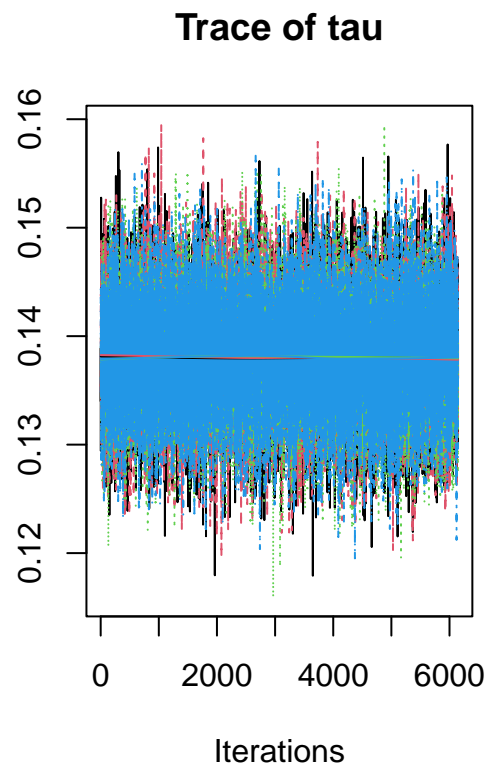
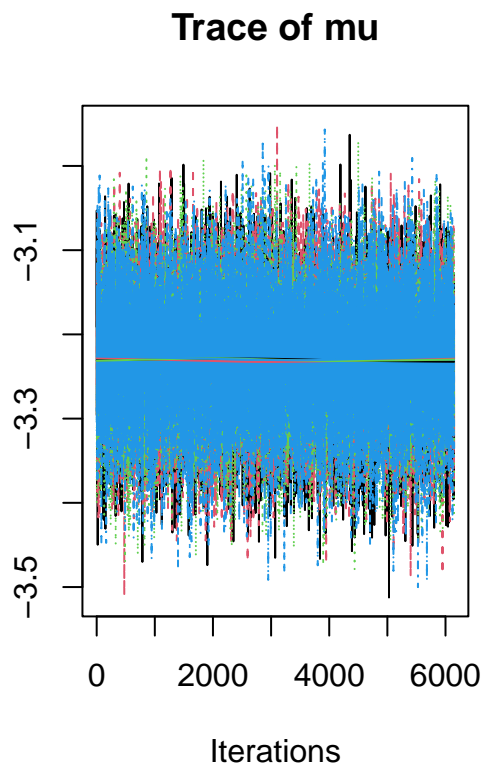


```

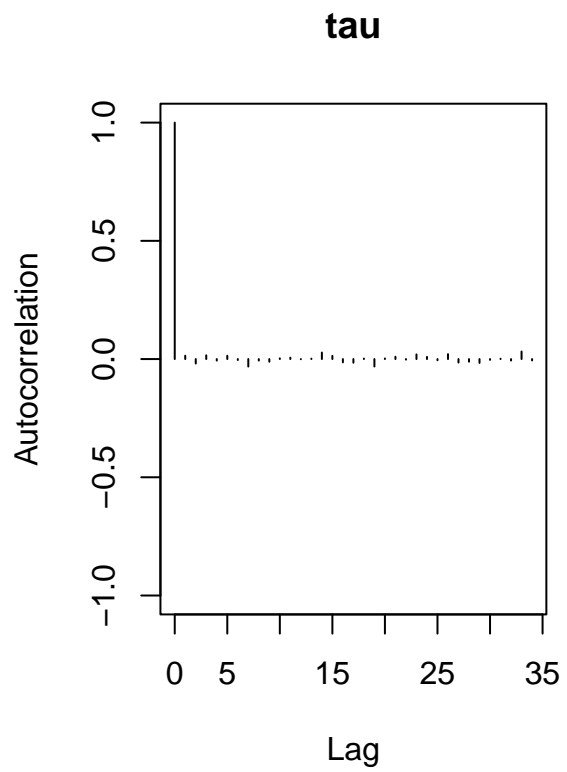
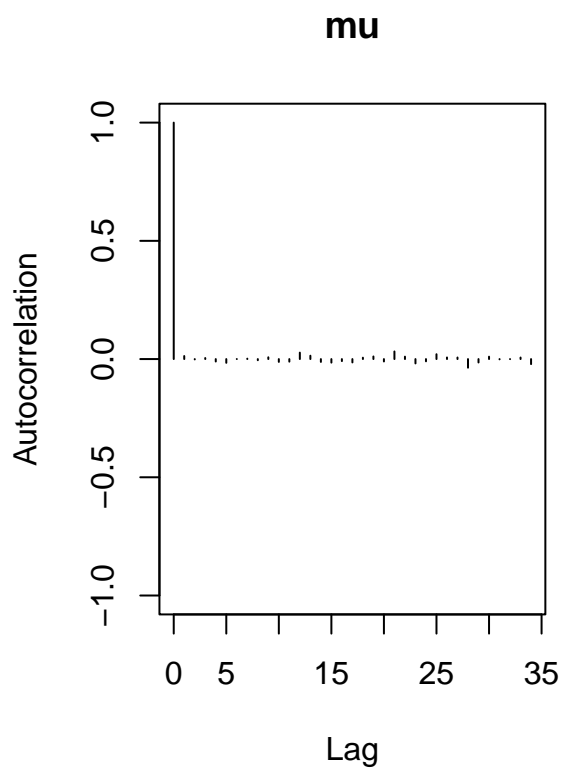
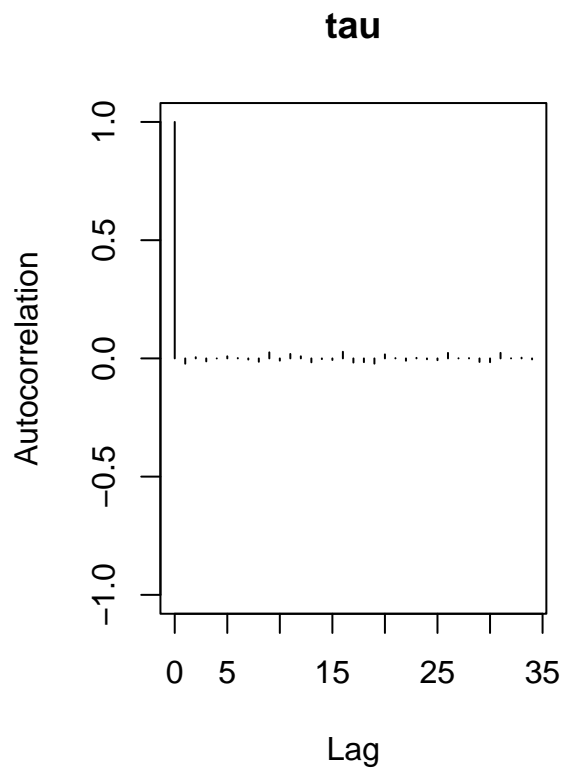
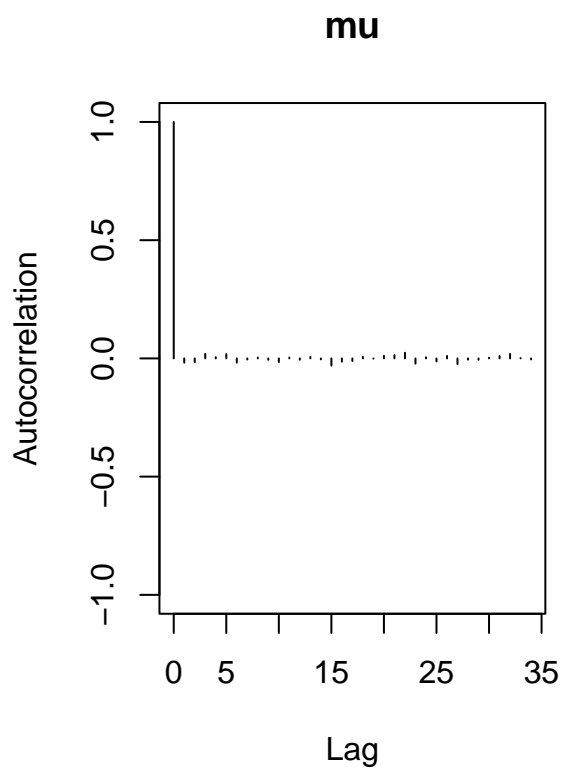
## tau passed      1      0.552
##
##      Halfwidth Mean      Halfwidth
##      test
## mu  passed      -3.230 0.001855
## tau passed      0.138 0.000132
##
## [[4]]
##
##      Stationarity start      p-value
##      test      iteration
## mu  passed      1      0.652
## tau passed      1      0.340
##
##      Halfwidth Mean      Halfwidth
##      test
## mu  passed      -3.229 0.00186
## tau passed      0.138 0.00014

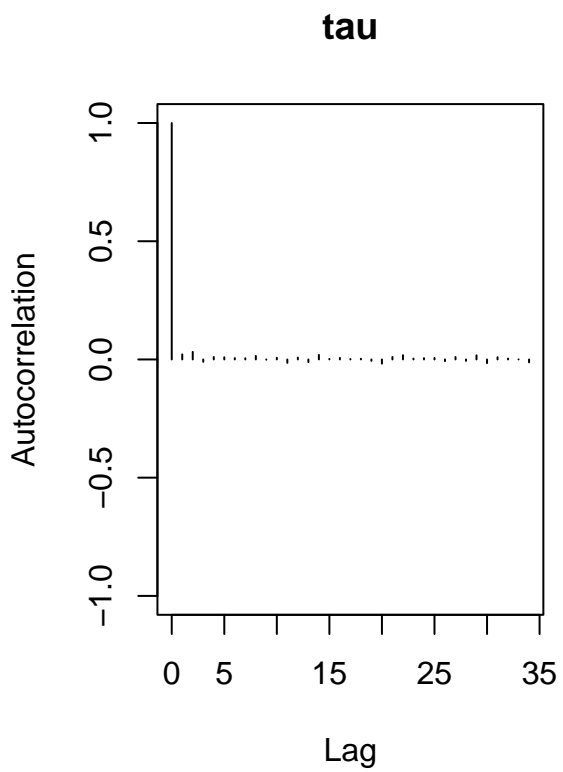
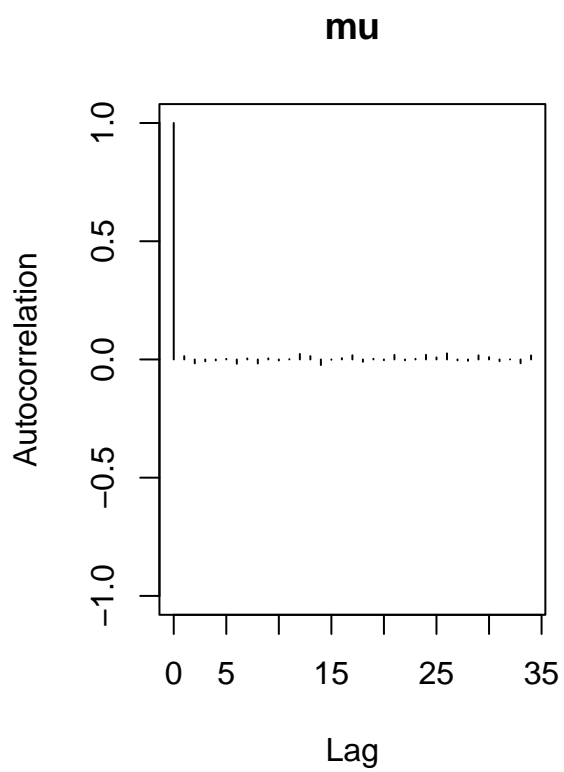
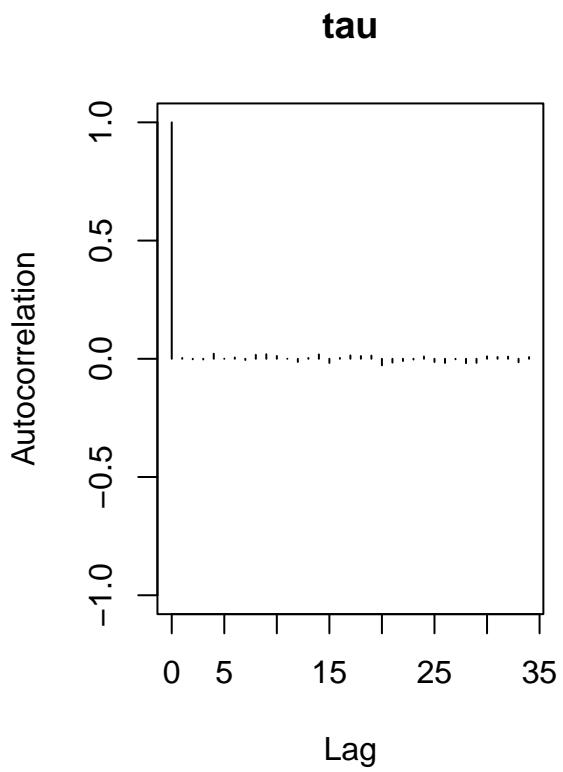
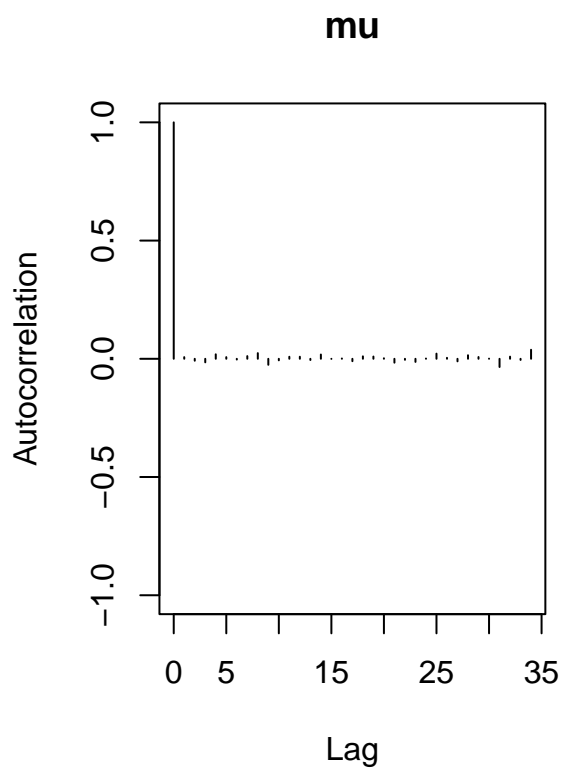
```

Trace plots



Autocorrelation plots





Contrasting Bayesian and frequentist approaches

Priors

- Choice of priors allows us to explicitly incorporate prior beliefs about parameters...
- ...but also comes with the obligation to be transparent and responsible with respect to the subjectivity this brings into our analyses

Inference

Interpretation

- Bayesian inference does not presume large (quasi-infinite) streams of independent identically distributed (IID) data; data are considered fixed, parameters random.
- This allows for straightforward interpretations of inferential uncertainty:
 - *Bayesian*: “Given the data, we can conclude that there is a 95% probability that the mean is between 8 and 12, with highest probability density at a value of 10”.
 - *Frequentist*: “If we took a large number of independent random samples from the same population and constructed a 95% confidence interval around the sample for each of them, these confidence intervals would contain the *true* population mean 95% of the time. Given this long-run frequency, we are 95% confident that the specific 95% confidence intervals from our singular sample contains the true population parameter...”.

Finite-sample and asymptotic properties

- Bayesian inference allows for exact inference in finite-sample applications where the asymptotic properties of MLE estimators are implausible (normal approximation, etc.)...
- ...yet, posterior distribution often asymptotically converge to the sampling distribution of MLE estimators (Bernstein-von-Mises Theorem)

Flexibility and computational reliability

- The use of MCMC algorithms for probabilistic approximate inference makes Bayesian approaches incredibly flexible and allows for computationally reliable estimation of complex,

analytically intractable marginal likelihoods (avoids integration of super high-dimensional integrals)...

- ...but sometimes comes with the necessity of high computational resources and/or long computation times, and always necessitates convergence diagnosis and model checking