Activity 6

Name here

Q1. Steps of Bayesian Data Analysis

Recall that for a Bayesian analysis we will follow these steps:

1. **Identify the data relevant to the research questions.** What are the measurement scales of the data? Which data variables are to be predicted, and which data variables are supposed to act as predictors?
2. **Define a descriptive model for the relevant data.** The mathematical form and its parameters should be meaningful and appropriate to the theoretical purposes of the analysis.
3. **Specify a prior distribution on the parameters.** The prior must pass muster with the audience of the analysis, such as skeptical scientists.
4. **Use Bayesian inference to re-allocate credibility across parameter values.** Interpret the posterior distribution with respect to theoretically meaningful issues (assuming that the model is a reasonable description of the data; see next step).
5. **Check that the posterior predictions mimic the data with reasonable accuracy (i.e., conduct a 'posterior predictive check').** If not, then consider a different descriptive model.

Q2. JAGS code modification

Recall the code form the weekly module.

```
model_string <- "model{
  # Likelihood
  z ~ dbinom(theta, N)

  # Prior
  theta ~ dbeta(alpha, beta)
  alpha <- 1 # prior successes
  beta <- 1 # prior failures
}"
```

Rewrite this in a way that alpha and beta can be inputed as data elements. Then re run the analyses. Recall that `z = 392` and `N = 869` when estimating the probability of a house in Seattle having more than two bedrooms.

```
model_string <- "model{
  # Likelihood
  z ~ dbinom(theta, N)

  # Prior
  theta ~ dbeta(alpha, beta)
}"
```

```
z <- 392
N <- 869
```

```r
dataList = list(z = z, N = N, alpha = 1, beta = 1)

model <- jags.model(file = textConnection(model_string), data = dataList)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 1
##    Unobserved stochastic nodes: 1
##    Total graph size: 5
##
## Initializing model
```

```r
update(model, n.iter = 5000) # warmup


# take samples
posterior_sample <- coda.samples(model,
                        variable.names = c("theta"),
                        n.iter = 10000)

summary(posterior_sample)
```

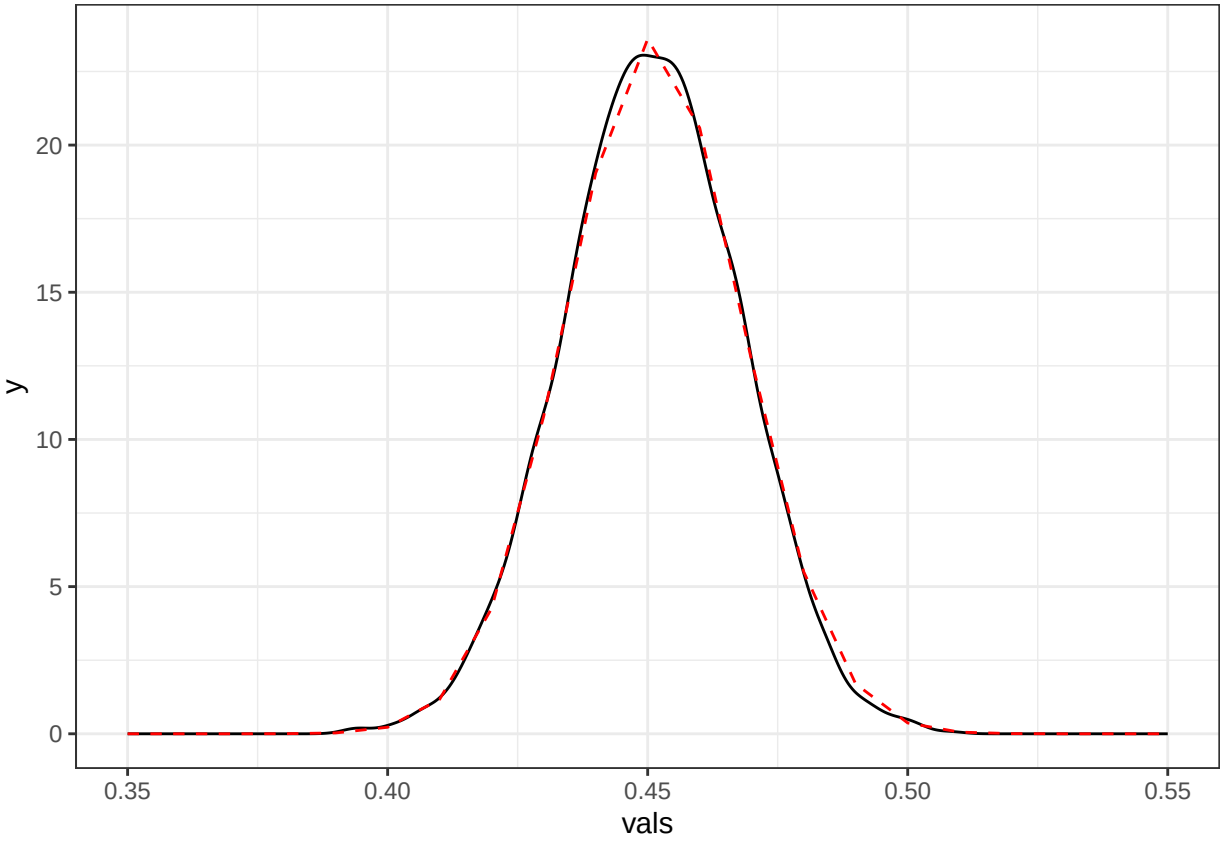
```
##
## Iterations = 6001:16000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean             SD       Naive SE Time-series SE
##      0.4508315      0.0168811      0.0001688      0.0002103
##
## 2. Quantiles for each variable:
##
##   2.5%    25%    50%    75%  97.5%
## 0.4176 0.4395 0.4508 0.4623 0.4836
```

Q3. JAGS Code object

Following the previous question, use the posterior samples `posterior_sample[[1]]` to create a density plot of the posterior distribution and overlay the true posterior density.

```r
xseq <- seq(.35, .55, by =.01)
dens_df <- tibble(x = xseq,
                  y = dbeta(xseq, z + 1, N -z + 1))
tibble(vals = posterior_sample[[1]]) %>%
  ggplot(aes(x = vals)) + geom_density() +
  theme_bw() +
  geom_line(data = dens_df, inherit.aes = F, aes(x =x, y =y), color = 'red', linetype = 2)
```

```
## Don't know how to automatically pick scale for object of type <mcmc>.
## Defaulting to continuous.
```



Q4. Synthetic Data

a. Simulate data from a normal process (mean .75, sd = 10) for 1000 trials.

```
N <- 1000
y <- rnorm(N, mean = 75, sd = 10)
```

b. State priors for μ and σ

With minimal information, assume $\mu \sim N(0, 1000^2)$ and $\sigma \sim Gamma(.01, .01)$

c. Given this data and priors, run jags code to estimate posterior distributions for μ and σ

```
model_normal<- "model{
  # Likelihood
  for (i in 1:n){
    y[i] ~ dnorm(mu, 1/sigma^2)
  }

  # Prior
  mu ~ dnorm(mu0, 1/sigma0^2)
  sigma ~ dgamma(a, b)
}"
```

```
dataList = list(y = y, n = N, mu0 = 0,
                sigma0 = 1000, a = .1, b = .1)

model <- jags.model(file = textConnection(model_normal), data = dataList)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 1000
##    Unobserved stochastic nodes: 2
##    Total graph size: 1013
##
## Initializing model
```

```
update(model, n.iter = 5000) # warmup


# take samples
posterior_sample <- coda.samples(model,
                        variable.names = c("mu", 'sigma'),
                        n.iter = 10000)

summary(posterior_sample)
```

```
##
## Iterations = 6001:16000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##        Mean     SD Naive SE Time-series SE
## mu    75.01 0.3106 0.003106       0.003216
## sigma  9.74 0.2165 0.002165       0.002760
##
## 2. Quantiles for each variable:
##
##         2.5%    25%    50%    75% 97.5%
## mu    74.404 74.804 75.017 75.224 75.62
## sigma  9.328  9.592  9.736  9.885 10.17
```

d. Compare your results from part c with what you'd expect. The mean of both parameters are very close to the generative values.

Q5. Regression

Assume we will use the Seattle housing dataset, but will now focus on housing price and use `sqft_living` as a predictor in a regression model.

a. Identify a descriptive statistical model for the relevant data. Then interpret the statistical parameters in that model.

Our general regression model can be written as

$$y = \beta_0 + \beta_1 x + \epsilon \quad \epsilon \sim N(0, \sigma^2)$$

,

let y be housing price and x be living square footage. Here we have identified a normal model is appropriate for the residuals in our model.

With properties of the normal model, this assumption of the residuals, implies $y \sim N(\beta_0 + \beta_1 x, \sigma^2)$. Thus there are three parameters in the model: β_0, β_1, and σ^2.

b. Specify a prior distribution for all parameters in the model.

Similar to previous models we might specify that

- $\beta_0 \sim N(0, 1000^2)$
- $\beta_1 \sim N(0, 1000^2)$,
- $\sigma \sim Gamma(.001, .001)$

Note this are uniformative priors and we can likely do better with some data transformations and additional thought.