Lab 6: Visualizing the Bayesian Workflow

23/02/23

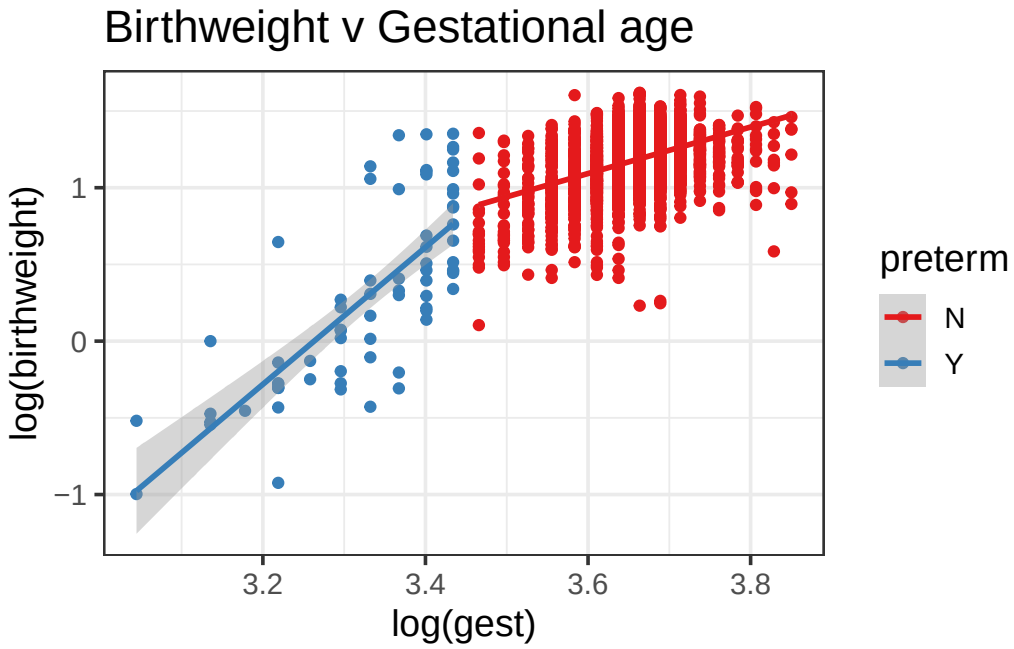
```
library(tidyverse)
library(here)
library(rstan)
library(bayesplot)
library(loo)
library(tidybayes)
library(fdrtool)

ds <- read_rds(here("births_2017_sample.RDS"))
```

```
ds = ds |>
  rename(birthweight = dbwt, gest = combgest) %>%
  mutate(preterm = ifelse(gest<32, "Y", "N")) %>%
  filter(ilive=="Y",gest< 99, birthweight<9.999)
```

Question 1

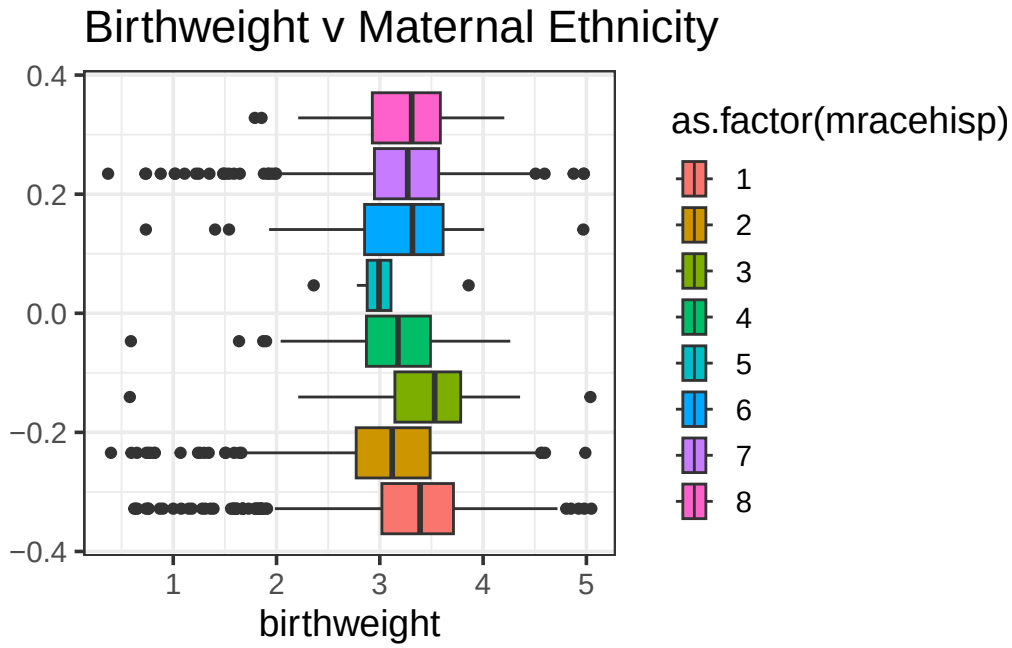
- Use plots or tables to show three interesting observations about the data

```
ds |>
  ggplot(aes(log(gest), log(birthweight), color = preterm)) +
  geom_point() + geom_smooth(method = "lm") +
  scale_color_brewer(palette = "Set1") +
  theme_bw(base_size = 14) +
  ggtitle("Birthweight v Gestational age")
```



Here we see the relationship between birth weight and the gestational age. We color the two sets of points based on a preterm indicator. The plot suggests that there exists some sort of interaction between gestational age and preterm-ness since the data indicates two different relationships when accounting for the indicator.

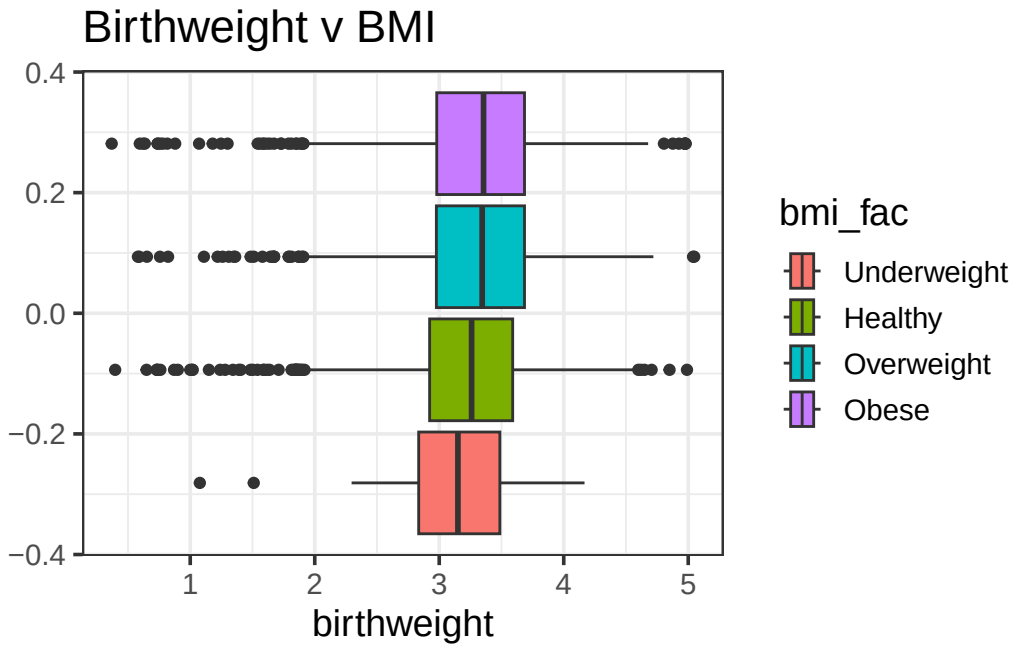
```
ds |>
  ggplot(aes(x= birthweight,fill=as.factor(mracehisp)))+
  geom_boxplot() +
  scale_color_brewer(palette = "Set1") +
  theme_bw(base_size = 14) +
  ggtitle("Birthweight v Maternal Ethnicity")
```



Here we explore the relationship between birth weight and the mother's ethnicity. We see varying distributions of birthweight amongst different ethnicities. They all appear to have the similar mass, but the spreads are very different. This may be attributed to the smaller sample sizes of some ethnicities.

We do not have an explicit preterm indicator in this plot, but from the first plot we note that many of the preterm points are exclusive to very small birthweights. It is interesting to see that the amount of low end birthweight outliers is not equal across ethnicity. Certain ethnicities have many more low birthweight points than others. These are likely preterms.

```
bmi_fac = cut(ds$bmi,breaks = c(-Inf,18.5,25,30,Inf),
              labels = c('Underweight','Healthy','Overweight','Obese'),
              include.lowest = T,right = F)

ds |>
  ggplot(aes(x= birthweight, fill = bmi_fac)) +
  geom_boxplot() +
  scale_color_brewer(palette = "Set1") +
  theme_bw(base_size = 14) +
  ggtitle("Birthweight v BMI")
```



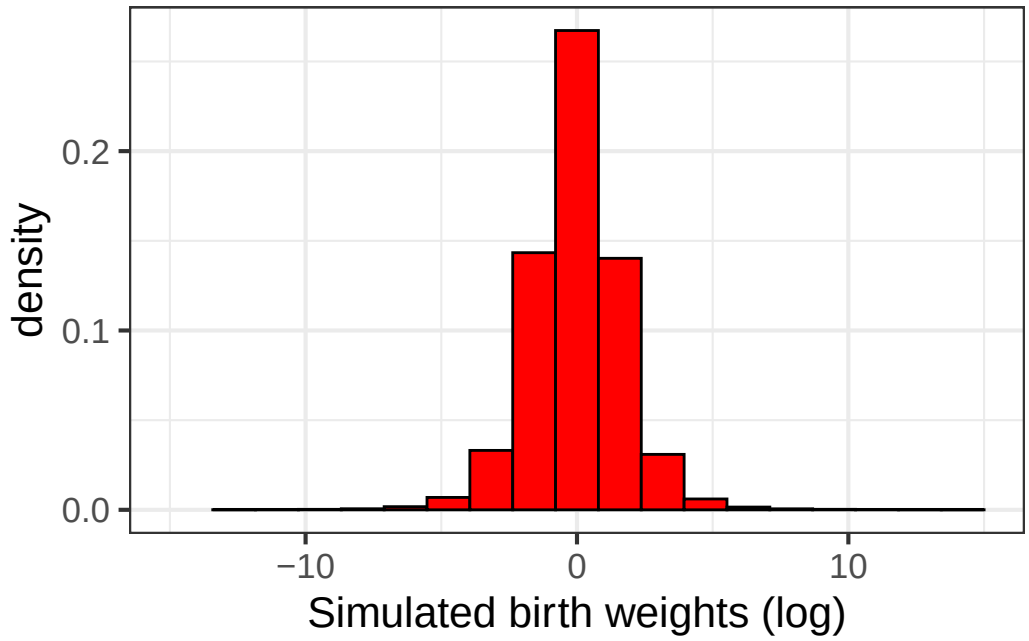
Here we explore any possible link between birthweight and the BMI of the mother. Using the typical BMI categorical splits we see that there are no significant deviations in birthweight for the different levels. The median birthweight does increase from being underweight to health and from healthy to being overweight, so there could be a small dependence, but it does appear to be anything substantial.

Question 2 (Prior Predictive Checks)

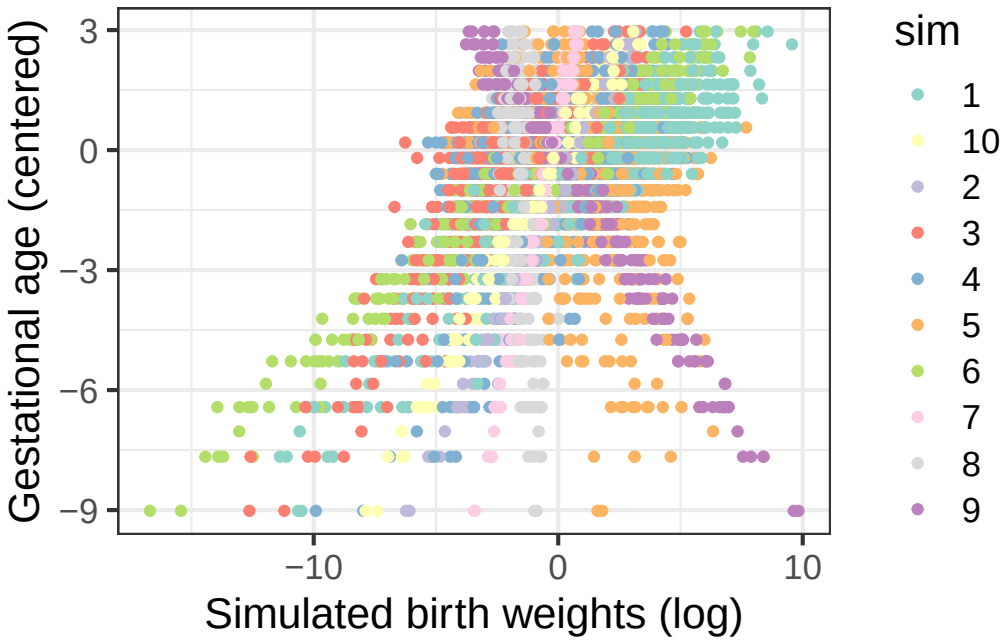
- Plot the resulting distribution of simulated (log) birth weights
- Plot ten simulations of (log) birthweights against gestational age

```
beta1s = rnorm(1000)
beta2s = rnorm(1000)
sigmas = rhalfnorm(1000)

dsims = tibble(log_gest_c = (log(ds$gest)-mean(log(ds$gest)))/sd(log(ds$gest)))

for(i in 1:1000){
  lin_predictor <- beta1s[i] + beta2s[i]*dsims$log_gest_c
  dsims[paste0(i)] <- lin_predictor + rnorm(nrow(dsims), 0, sigmas[i])
}
```

```r
all_sim = dsims |>
          pivot_longer(`1`:`1000`, names_to = "sim", values_to = "sim_weight")

all_sim |>
      ggplot(aes(sim_weight)) + geom_histogram(aes(y = ..density..),
      bins = 20, fill = "red", color = "black") +
      xlab("Simulated birth weights (log)") +
      xlim(c(-15, 15)) +
      theme_bw(base_size = 16)
```




```r
ten_sim = dsims[1:11] |>
          pivot_longer(`1`:`10`, names_to = "sim", values_to = "sim_weight")
ten_sim |>
        ggplot(aes(x = sim_weight, y = log_gest_c, color = sim)) +
        geom_point() +
        xlab("Simulated birth weights (log)") +
        ylab("Gestational age (centered)") +
        theme_bw(base_size = 16) +
        scale_color_brewer(palette = "Set3")
```



Running Model 1 in Stan

```
ds$log_weight <- log(ds$birthweight)
ds$log_gest_c <- (log(ds$gest) - mean(log(ds$gest)))/sd(log(ds$gest))

# put into a list
stan_data <- list(N = nrow(ds),
                  log_weight = ds$log_weight,
                  log_gest = ds$log_gest_c)
```

```
mod1 <- stan(data = stan_data,
             file = here("simple_weight.stan"),
             iter = 500,
             seed = 243)
```

```
summary(mod1)$summary[c("beta[1]", "beta[2]", "sigma"),]
```

```
             mean      se_mean          sd      2.5%       25%       50%
beta[1] 1.1625308 8.937069e-05 0.002939494 1.1567671 1.1604564 1.1625497
beta[2] 0.1437732 8.218239e-05 0.002767627 0.1385299 0.1417673 0.1437258
sigma   0.1689611 1.007199e-04 0.001848675 0.1653767 0.1677087 0.1691049
```

```
              75%     97.5%     n_eff      Rhat
beta[1] 1.1645491 1.1681526 1081.8199 0.9972360
beta[2] 0.1455863 0.1491984 1134.1163 0.9986568
sigma   0.1701874 0.1725273  336.8921 1.0084718
```

Question 3

- Based on model 1, give an estimate of the expected birthweight of a baby who was born at a gestational age of 37 weeks.

```
new_data = (log(37)  - mean(log(ds$gest)))/sd(log(ds$gest))
new_lin_pred = 1.1625308 + 0.1437732 * new_data

new_pred = exp(new_lin_pred)
```

According to the model, an estimate of the expected birthweight, i.e. the mean, of a baby who was born at a gestational age of 37 weeks would be 2.9359931 kg.

Question 4

- Write a Stan model to run Model 2, and run it

```
stan_data <- list(N = nrow(ds),
                  log_weight = ds$log_weight,
                  log_gest = ds$log_gest_c,
                  preterm = ifelse(ds$preterm=="Y",1,0))

mymod2 <- stan(data = stan_data,
               file = here("6q4.stan"),
               iter = 500,
               seed = 243)
```

```
summary(mymod2)$summary[c("beta[1]", "beta[2]", "beta[3]", "beta[4]", "sigma"),]
```

```
             mean      se_mean          sd       2.5%        25%       50%
beta[1] 1.1695960 7.192111e-05 0.002723168 1.16447938 1.16774934 1.1694521
beta[2] 0.1017922 1.203250e-04 0.003664257 0.09436935 0.09947277 0.1017107
beta[3] 0.5579402 3.585589e-03 0.062249185 0.43782191 0.51517463 0.5575181
beta[4] 0.1975984 7.433239e-04 0.012744410 0.17399405 0.18841309 0.1980229
```

```
sigma   0.1611595 7.995345e-05 0.001870083 0.15756660 0.15990282 0.1611450
              75%     97.5%     n_eff      Rhat
beta[1] 1.1714748 1.1749684 1433.6272 0.9981153
beta[2] 0.1042566 0.1091657  927.3855 1.0042246
beta[3] 0.6032388 0.6740442  301.4022 1.0146412
beta[4] 0.2064166 0.2211302  293.9566 1.0172730
sigma   0.1624357 0.1647640  547.0753 0.9999611
```

Question 5

- Check your results are similar to the reference model

```
load(here("mod2.Rda"))
summary(mod2)$summary[c(paste0("beta[", 1:4, "]"), "sigma"),]
```

```
             mean      se_mean          sd       2.5%        25%       50%
beta[1] 1.1697241 1.385590e-04 0.002742186 1.16453578 1.16767109 1.1699278
beta[2] 0.5563133 5.835253e-03 0.058054991 0.43745504 0.51708255 0.5561553
beta[3] 0.1020960 1.481816e-04 0.003669476 0.09459462 0.09997153 0.1020339
beta[4] 0.1967671 1.129799e-03 0.012458398 0.17164533 0.18817091 0.1974114
sigma   0.1610727 9.950037e-05 0.001782004 0.15784213 0.15978020 0.1610734
              75%     97.5%     n_eff      Rhat
beta[1] 1.1716235 1.1750167 391.67359 1.0115970
beta[2] 0.5990427 0.6554967  98.98279 1.0088166
beta[3] 0.1044230 0.1093843 613.22428 0.9978156
beta[4] 0.2064079 0.2182454 121.59685 1.0056875
sigma   0.1623019 0.1646189 320.75100 1.0104805
```

```
tab <- matrix(c(1.1695960 - 1.1697241,
                0.5579402 - 0.5563133,
                0.1017922-0.1020960,
                0.1975984-0.1967671,
                0.1611450-0.1610727), ncol=5, byrow=TRUE)
colnames(tab) <- c('Beta_1','Beta_2','Beta_3','Beta_4', 'Sigma')
rownames(tab) <- c('Difference')
tab <- as.table(tab)
tab
```

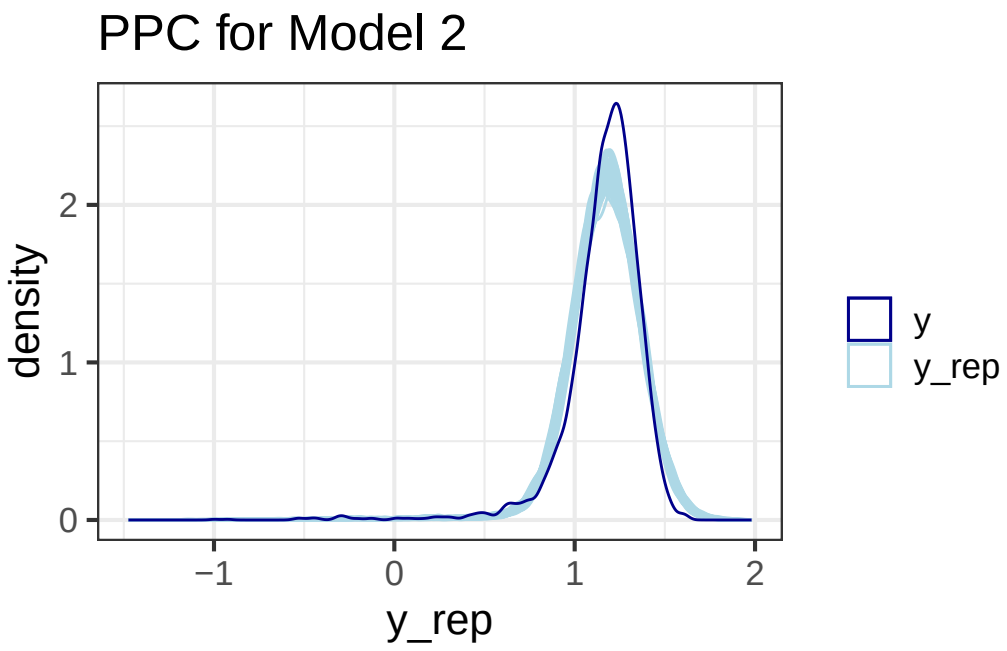
```
              Beta_1     Beta_2     Beta_3     Beta_4      Sigma
Difference -0.0001281  0.0016269 -0.0003038  0.0008313  0.0000723
```

We see from this table of difference between the two models that the estimates are nearly identical for all of the coefficients.

Question 6

- Make a similar plot to the one above but for model 2 (a PPC), and **not** using the bayes plot in built function

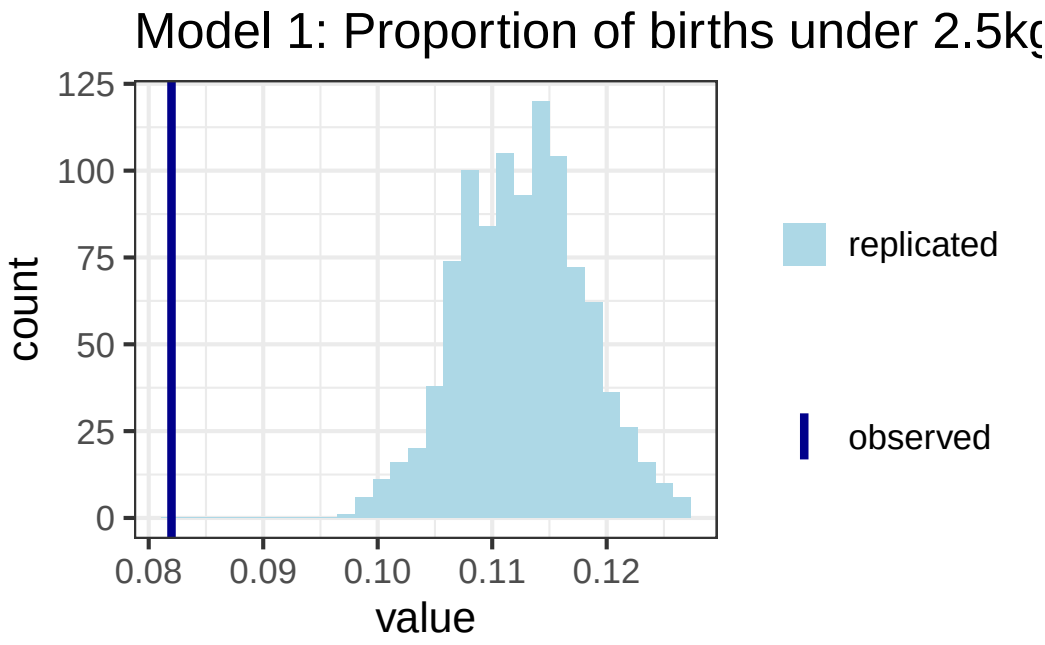
```
set.seed(1856)
y <- ds$log_weight
yrep1 <- extract(mod1)[["log_weight_rep"]]
yrep2 <- extract(mymod2)[["log_weight_rep"]]
samp100 <- sample(nrow(yrep2), 100)
N = length(ds$birthweight)

rownames(yrep2) <- 1:nrow(yrep2)
dr = as_tibble(t(yrep2))
dr = dr |> bind_cols(i = 1:N, log_weight_obs = log(ds$birthweight))

dr = dr |>
  pivot_longer(-(i:log_weight_obs), names_to = "sim", values_to ="y_rep")

dr |>
  filter(sim %in% samp100) |>
  ggplot(aes(y_rep, group = sim)) +
  geom_density(alpha = 0.2, aes(color = "y_rep")) +
  geom_density(data = ds |> mutate(sim = 1),
               aes(x = log(birthweight), col = "y")) +
  scale_color_manual(name = "",
                     values = c("y" = "darkblue",
                                "y_rep" = "lightblue")) +
  ggtitle("PPC for Model 2") +
  theme_bw(base_size = 16)
```



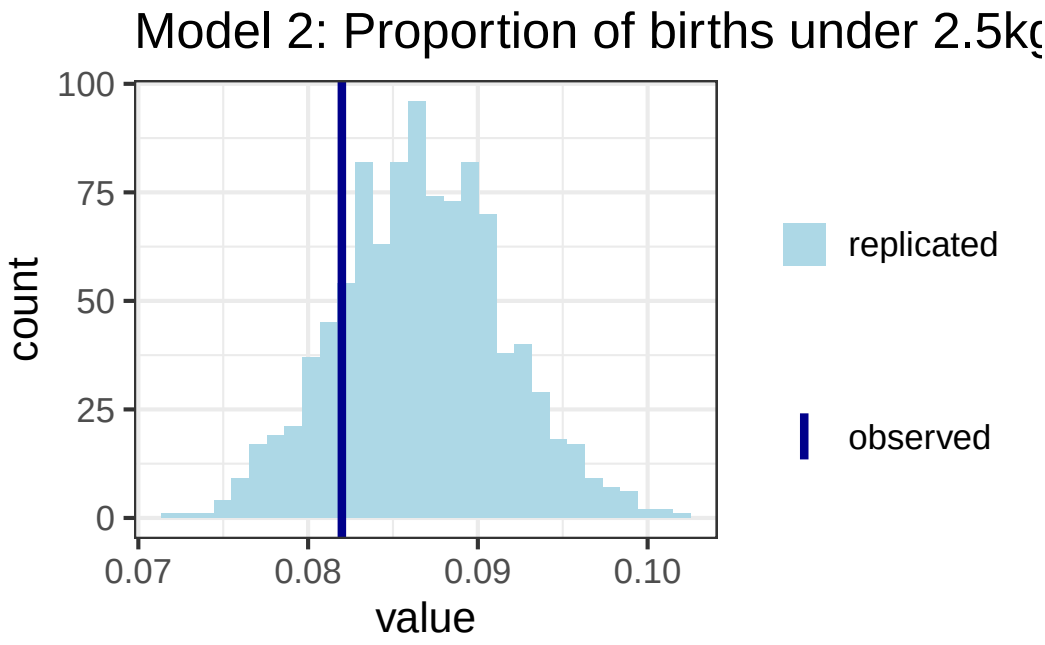
Question 7

- Use a test statistic of the proportion of births under 2.5kg. Calculate the test statistic for the data, and the posterior predictive samples for both models, and plot the comparison (one plot per model).

```
t_y <- mean(y<=log(2.5))
t_y_rep <- sapply(1:nrow(yrep1), function(i) mean(yrep1[i,]<=log(2.5)))
t_y_rep_2 <- sapply(1:nrow(yrep2), function(i) mean(yrep2[i,]<=log(2.5)))

ggplot(data = as_tibble(t_y_rep), aes(value)) +
    geom_histogram(aes(fill = "replicated")) +
    geom_vline(aes(xintercept = t_y, color = "observed"), lwd = 1.5) +
  ggtitle("Model 1: Proportion of births under 2.5kg") +
  theme_bw(base_size = 16) +
  scale_color_manual(name = "",
                     values = c("observed" = "darkblue"))+
  scale_fill_manual(name = "",
                     values = c("replicated" = "lightblue"))
```




```
ggplot(data = as_tibble(t_y_rep_2), aes(value)) +
    geom_histogram(aes(fill = "replicated")) +
    geom_vline(aes(xintercept = t_y, color = "observed"), lwd = 1.5) +
  ggtitle("Model 2: Proportion of births under 2.5kg") +
  theme_bw(base_size = 16) +
  scale_color_manual(name = "",
                     values = c("observed" = "darkblue"))+
  scale_fill_manual(name = "",
                     values = c("replicated" = "lightblue"))
```



We see that model 2 does a much better job in this case.

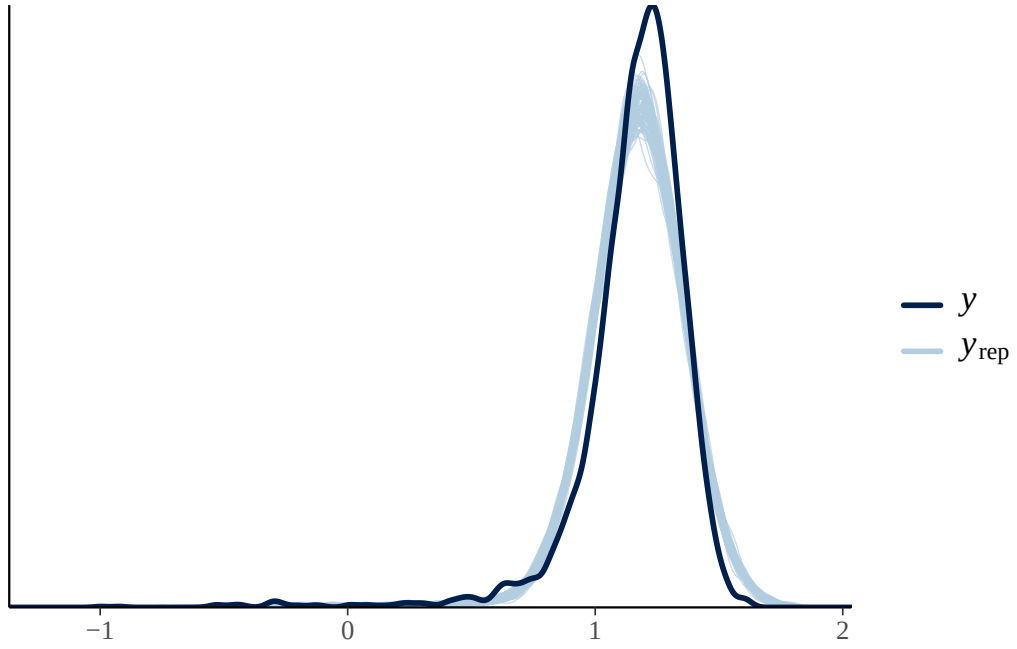
Question 8

- Based on the original dataset, choose one (or more) additional covariates to add to the linear regression model. Run the model in Stan, and compare with Model 2 above on at least 2 posterior predictive checks

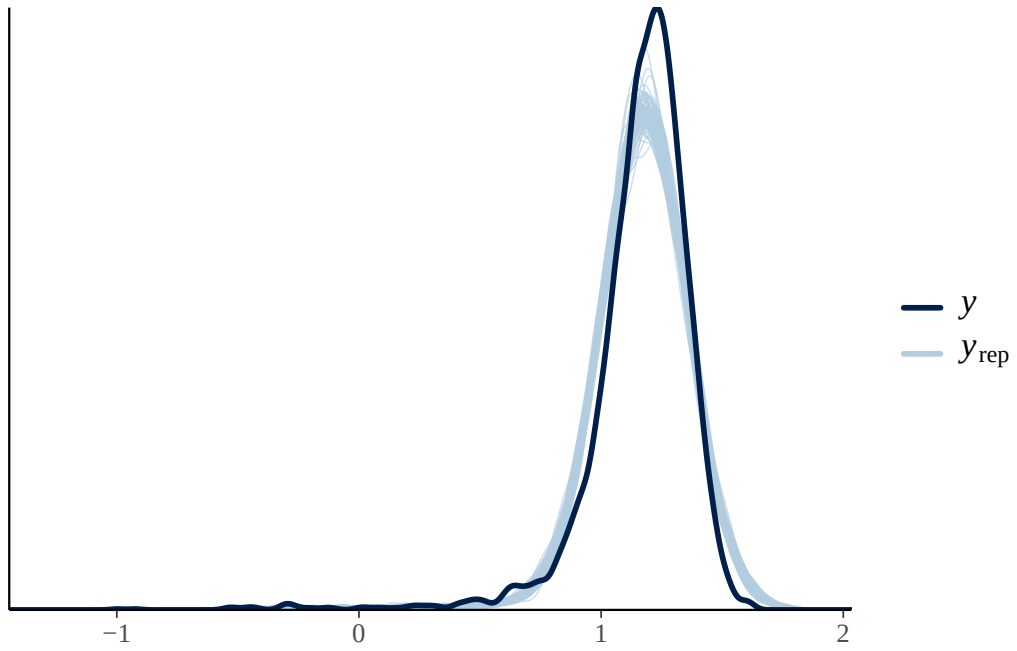
```
stan_data <- list(N = nrow(ds),
                  log_weight = ds$log_weight,
                  log_gest = ds$log_gest_c,
                  preterm = ifelse(ds$preterm=="Y",1,0),
                  ethnicity = ds$mracehisp)

mymod3 <- stan(data = stan_data,
               file = here("6q8.stan"),
               iter = 500,
               seed = 243)
```

We first define our new model and run it above. Lets first check out the PPC with Stan output, i.e using bayesplot.

```
y <- ds$log_weight
yrep2 <- extract(mymod2)[["log_weight_rep"]]
yrep3 <- extract(mymod3)[["log_weight_rep"]]
samp100 <- sample(nrow(yrep2), 100)

par(mfrow=c(1,2))
ppc_dens_overlay(y, yrep2[samp100, ])
```




```
ppc_dens_overlay(y, yrep3[samp100, ])
```



Its very hard to see any major differences. It is safe to say that the new model is not worse than model 2. However, it does look to be significantly better if it is indeed better.

Lets instead try LOO-CV:

```
loglik2 <- extract(mymod2)[["log_lik"]]
loglik3 <- extract(mymod3)[["log_lik"]]
loo2 <- loo(loglik2, save_psis = TRUE)
loo3 <- loo(loglik3, save_psis = TRUE)
loo_compare(loo2, loo3)
```

```
       elpd_diff se_diff
model2  0.0       0.0
model1 -1.2       3.4
```

We see that between the two model, there is a difference of 1.2 ELPD units. Comparing this to the SE, really shows how insignificant this difference is. Our old model is technically worse on this ELPD metric, however our new model is really not much better.