

# Week 6: Visualizing the Bayesian Workflow

27/02/23

## Introduction

This lab will be looking at trying to replicate some of the visualizations in the lecture notes, involving prior and posterior predictive checks, and LOO model comparisons.

The dataset is a 0.1% of all births in the US in 2017. I've pulled out a few different variables, but as in the lecture, we'll just focus on birth weight and gestational age.

## The data

Read it in, along with all our packages.

```
# A tibble: 6 x 8
  mager mracehisp meduc   bmi sex   combgest dbwt ilive
  <dbl>      <dbl> <dbl> <dbl> <chr>    <dbl> <dbl> <chr>
1    16         2    2   23    M        39  3.18 Y
2    25         7    2  43.6 M        40  4.14 Y
3    27         2    3  19.5 F        41  3.18 Y
4    26         1    3  21.5 F        36  3.40 Y
5    28         7    2  40.6 F        34  2.71 Y
6    31         7    3  29.3 M        35  3.52 Y
```

Brief overview of variables:

- **mager** mum's age
- **mracehisp** mum's race/ethnicity see here for codes: <https://data.nber.org/natality/2017/natl2017.pdf> page 15
- **meduc** mum's education see here for codes: <https://data.nber.org/natality/2017/natl2017.pdf> page 16

- `bmi` mum's bmi
- `sex` baby's sex
- `combgest` gestational age in weeks
- `dbwt` birth weight in kg
- `ilive` alive at time of report y/n/ unsure

I'm going to rename some variables, remove any observations with missing gestational age or birth weight, restrict just to babies that were alive, and make a preterm variable.

## Question 1

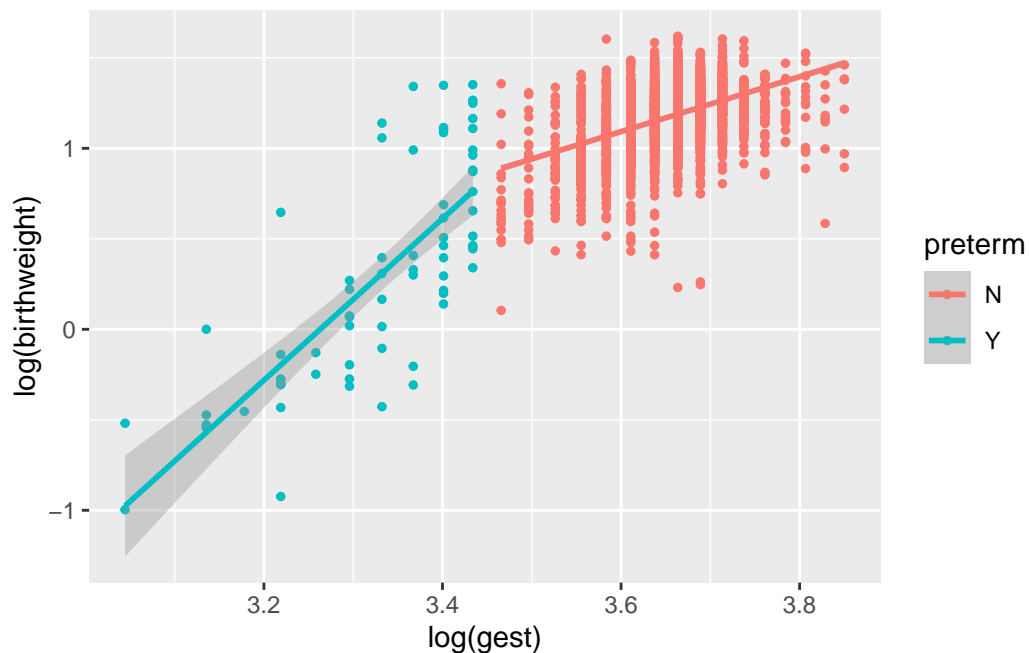
Use plots or tables to show three interesting observations about the data. Remember:

- Explain what your graph/ tables show
- Choose a graph type that's appropriate to the data type
- If you use `geom_smooth`, please also plot the underlying data

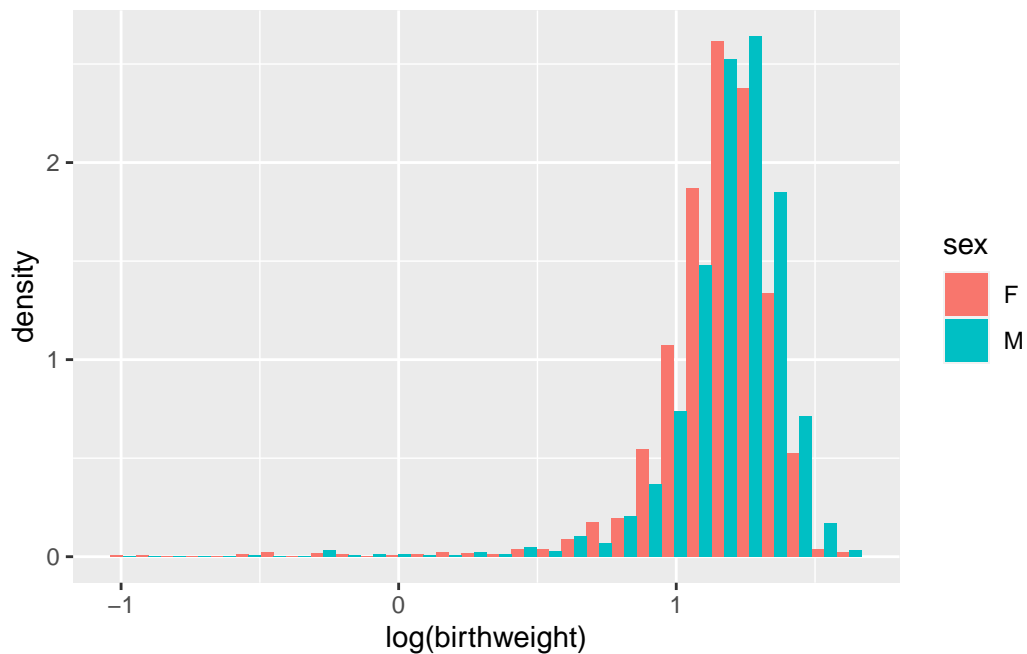
Feel free to replicate one of the scatter plots in the lectures as one of the interesting observations, as those form the basis of our models.

## Answer

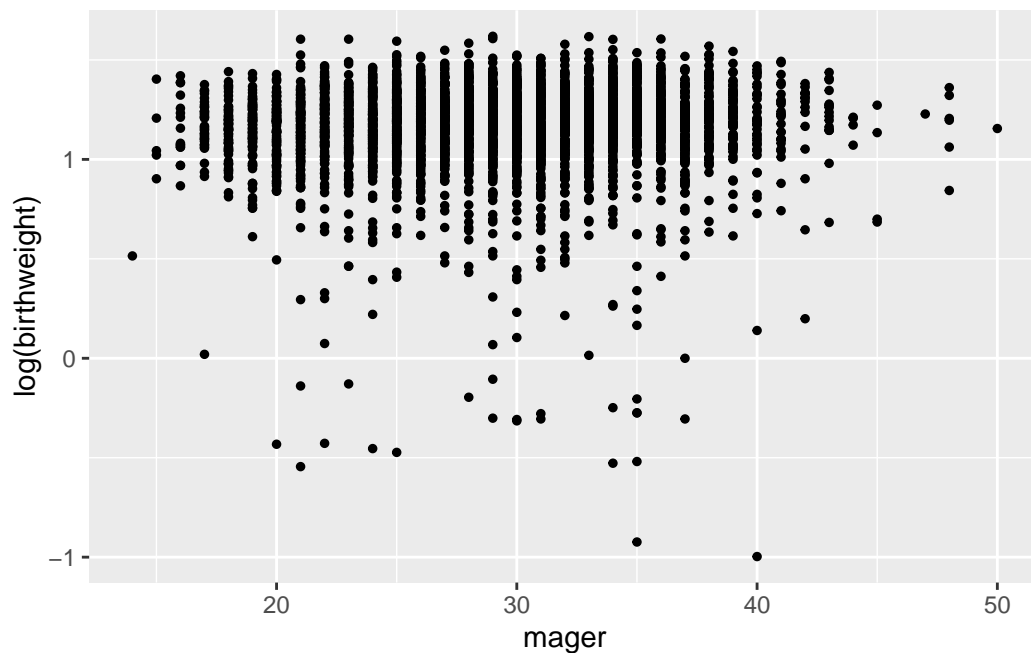
The following plots is the scatter plots in the lecture replicated. It indicates linear relation between  $(\log)\text{birthweight}$  and  $\log(\text{gest})$ . It also indicates that 'preterm' would be a significant covariate.



The following histogram was created to check if 'sex' has something to with 'birthweight'. We can see from this plot that male are distributed slightly larger than female. Based on this observation, we'll choose 'sex' as the additional covariate in Question 8.



We also created the following plot to see if mother's age has something to do with birthweight. However, the plot does not indicate strong evidence that the covariate matters.



## The model

As in lecture, we will look at two candidate models

Model 1 has log birth weight as a function of log gestational age

$$\log(y_i) \sim N(\beta_1 + \beta_2 \log(x_i), \sigma^2)$$

Model 2 has an interaction term between gestation and prematurity

$$\log(y_i) \sim N(\beta_1 + \beta_2 \log(x_i) + \beta_2 z_i + \beta_3 \log(x_i) z_i, \sigma^2)$$

- $y_i$  is weight in kg
- $x_i$  is gestational age in weeks, CENTERED AND STANDARDIZED
- $z_i$  is preterm (0 or 1, if gestational age is less than 32 weeks)

## Prior predictive checks

Let's put some weakly informative priors on all parameters i.e. for the  $\beta$ s

$$\beta \sim N(0, 1)$$

and for  $\sigma$

$$\sigma \sim N^+(0, 1)$$

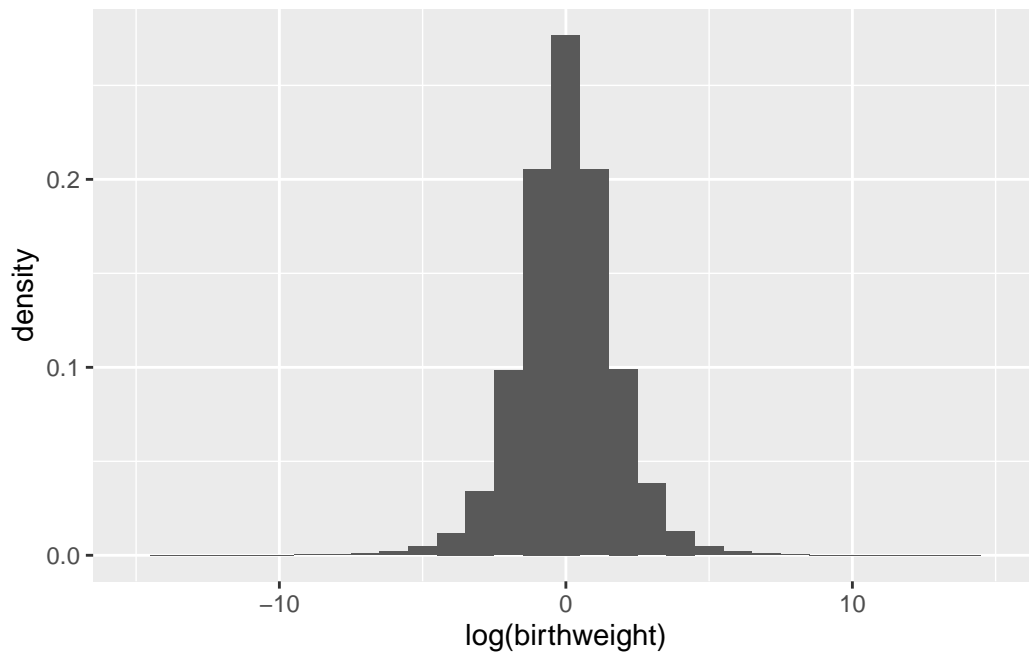
where the plus means positive values only i.e. Half Normal.

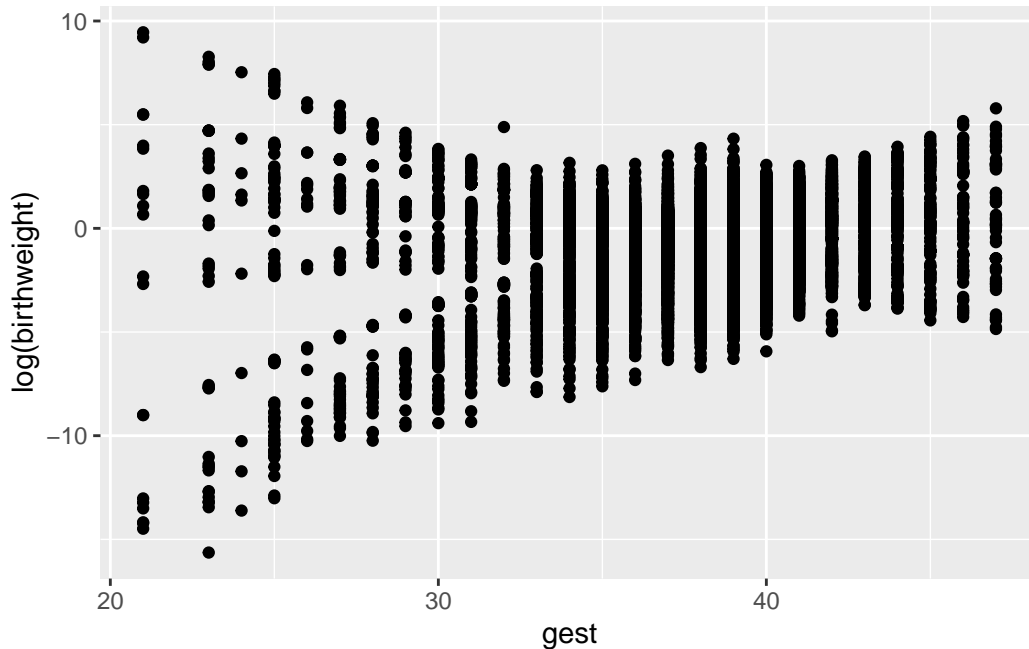
Let's check to see what the resulting distribution of birth weights look like given Model 1 and the priors specified above, assuming we had no data on birth weight (but observations of gestational age).

## Question 2

For Model 1, simulate values of  $\beta$ s and  $\sigma$  based on the priors above. Do 1000 simulations. Use these values to simulate (log) birth weights from the likelihood specified in Model 1, based on the set of observed gestational weights. **Remember the gestational weights should be centered and standardized.**

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





## Run the model

Now we're going to run Model 1 in Stan. The stan code is in the `code/models` folder.

First, get our data into right form for input into stan.

Now fit the model

```
Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
clang -mmacosx-version-min=10.13 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.2/Resources/library/StanHeaders/include/stan/headers/
In file included from /Library/Frameworks/R.framework/Versions/4.2/Resources/library/RcppEigen/include/Eigen/src/Core/
In file included from /Library/Frameworks/R.framework/Versions/4.2/Resources/library/RcppEigen/include/Eigen/src/Core/
/Library/Frameworks/R.framework/Versions/4.2/Resources/library/RcppEigen/include/Eigen/src/Core/
namespace Eigen {
~
/Library/Frameworks/R.framework/Versions/4.2/Resources/library/RcppEigen/include/Eigen/src/Core/
namespace Eigen {
~
~
;
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.2/Resources/library/StanHeaders/include/stan/headers/
```

```

In file included from /Library/Frameworks/R.framework/Versions/4.2/Resources/library/RcppEigen
/Library/Frameworks/R.framework/Versions/4.2/Resources/library/RcppEigen/include/Eigen/Core:
#include <complex>
      ~~~~~
3 errors generated.
make: *** [foo.o] Error 1

```

```

SAMPLING FOR MODEL 'simple_weight' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 0.000328 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 3.28 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration:   1 / 500 [  0%] (Warmup)
Chain 1: Iteration:  50 / 500 [ 10%] (Warmup)
Chain 1: Iteration: 100 / 500 [ 20%] (Warmup)
Chain 1: Iteration: 150 / 500 [ 30%] (Warmup)
Chain 1: Iteration: 200 / 500 [ 40%] (Warmup)
Chain 1: Iteration: 250 / 500 [ 50%] (Warmup)
Chain 1: Iteration: 251 / 500 [ 50%] (Sampling)
Chain 1: Iteration: 300 / 500 [ 60%] (Sampling)
Chain 1: Iteration: 350 / 500 [ 70%] (Sampling)
Chain 1: Iteration: 400 / 500 [ 80%] (Sampling)
Chain 1: Iteration: 450 / 500 [ 90%] (Sampling)
Chain 1: Iteration: 500 / 500 [100%] (Sampling)
Chain 1:
Chain 1: Elapsed Time: 0.528425 seconds (Warm-up)
Chain 1:                  0.471116 seconds (Sampling)
Chain 1:                  0.999541 seconds (Total)
Chain 1:

```

```

SAMPLING FOR MODEL 'simple_weight' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 0.00017 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 1.7 seconds.
Chain 2: Adjust your expectations accordingly!
Chain 2:
Chain 2:
Chain 2: Iteration:   1 / 500 [  0%] (Warmup)
Chain 2: Iteration:  50 / 500 [ 10%] (Warmup)
Chain 2: Iteration: 100 / 500 [ 20%] (Warmup)
Chain 2: Iteration: 150 / 500 [ 30%] (Warmup)

```

```

Chain 2: Iteration: 200 / 500 [ 40%] (Warmup)
Chain 2: Iteration: 250 / 500 [ 50%] (Warmup)
Chain 2: Iteration: 251 / 500 [ 50%] (Sampling)
Chain 2: Iteration: 300 / 500 [ 60%] (Sampling)
Chain 2: Iteration: 350 / 500 [ 70%] (Sampling)
Chain 2: Iteration: 400 / 500 [ 80%] (Sampling)
Chain 2: Iteration: 450 / 500 [ 90%] (Sampling)
Chain 2: Iteration: 500 / 500 [100%] (Sampling)
Chain 2:
Chain 2: Elapsed Time: 0.484584 seconds (Warm-up)
Chain 2: 0.510541 seconds (Sampling)
Chain 2: 0.995125 seconds (Total)
Chain 2:

```

SAMPLING FOR MODEL 'simple\_weight' NOW (CHAIN 3).

```

Chain 3:
Chain 3: Gradient evaluation took 0.000168 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 1.68 seconds.
Chain 3: Adjust your expectations accordingly!
Chain 3:
Chain 3:
Chain 3: Iteration: 1 / 500 [ 0%] (Warmup)
Chain 3: Iteration: 50 / 500 [ 10%] (Warmup)
Chain 3: Iteration: 100 / 500 [ 20%] (Warmup)
Chain 3: Iteration: 150 / 500 [ 30%] (Warmup)
Chain 3: Iteration: 200 / 500 [ 40%] (Warmup)
Chain 3: Iteration: 250 / 500 [ 50%] (Warmup)
Chain 3: Iteration: 251 / 500 [ 50%] (Sampling)
Chain 3: Iteration: 300 / 500 [ 60%] (Sampling)
Chain 3: Iteration: 350 / 500 [ 70%] (Sampling)
Chain 3: Iteration: 400 / 500 [ 80%] (Sampling)
Chain 3: Iteration: 450 / 500 [ 90%] (Sampling)
Chain 3: Iteration: 500 / 500 [100%] (Sampling)
Chain 3:
Chain 3: Elapsed Time: 0.572595 seconds (Warm-up)
Chain 3: 0.446531 seconds (Sampling)
Chain 3: 1.01913 seconds (Total)
Chain 3:

```

SAMPLING FOR MODEL 'simple\_weight' NOW (CHAIN 4).

```

Chain 4:
Chain 4: Gradient evaluation took 0.000196 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 1.96 seconds.

```



```
Chain 4: Adjust your expectations accordingly!
Chain 4:
Chain 4:
Chain 4: Iteration: 1 / 500 [ 0%] (Warmup)
Chain 4: Iteration: 50 / 500 [ 10%] (Warmup)
Chain 4: Iteration: 100 / 500 [ 20%] (Warmup)
Chain 4: Iteration: 150 / 500 [ 30%] (Warmup)
Chain 4: Iteration: 200 / 500 [ 40%] (Warmup)
Chain 4: Iteration: 250 / 500 [ 50%] (Warmup)
Chain 4: Iteration: 251 / 500 [ 50%] (Sampling)
Chain 4: Iteration: 300 / 500 [ 60%] (Sampling)
Chain 4: Iteration: 350 / 500 [ 70%] (Sampling)
Chain 4: Iteration: 400 / 500 [ 80%] (Sampling)
Chain 4: Iteration: 450 / 500 [ 90%] (Sampling)
Chain 4: Iteration: 500 / 500 [100%] (Sampling)
Chain 4:
Chain 4: Elapsed Time: 0.520697 seconds (Warm-up)
Chain 4: 0.497479 seconds (Sampling)
Chain 4: 1.01818 seconds (Total)
Chain 4:
```

	mean	se_mean	sd	2.5%	25%	50%
beta[1]	1.1626250	7.634607e-05	0.002583881	1.1575321	1.1609497	1.1626383
beta[2]	0.1436183	8.105504e-05	0.002791943	0.1380281	0.1417563	0.1436199
sigma	0.1689127	1.051837e-04	0.001979909	0.1650908	0.1676042	0.1688619

	75%	97.5%	n_eff	Rhat
beta[1]	1.1643919	1.1677313	1145.4383	0.9970543
beta[2]	0.1455075	0.1489575	1186.4598	0.9984953
sigma	0.1701148	0.1728405	354.3181	1.0046933

### 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.

### Answer

```
exp(1.1626250 + 0.1436183*(log(37)-m)/s)
```

```
[1] 2.93654
```

## Question 4

Write a stan model to run Model 2, and run it.

## Answer

Now fit the model

```
Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
clang -mmacosx-version-min=10.13 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.2/Resources/library/StanHeaders/include/stan/math/
In file included from /Library/Frameworks/R.framework/Versions/4.2/Resources/library/RcppEigen/include/Eigen/Core:1:
In file included from /Library/Frameworks/R.framework/Versions/4.2/Resources/library/RcppEigen/include/Eigen/src/Core/
/Library/Frameworks/R.framework/Versions/4.2/Resources/library/RcppEigen/include/Eigen/src/Core/Matrix.h:1:
namespace Eigen {
^
/Library/Frameworks/R.framework/Versions/4.2/Resources/library/RcppEigen/include/Eigen/src/Core/Matrix.h:1:
namespace Eigen {
^
;
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.2/Resources/library/StanHeaders/include/stan/math/
In file included from /Library/Frameworks/R.framework/Versions/4.2/Resources/library/RcppEigen/include/Eigen/Core:1:
/Library/Frameworks/R.framework/Versions/4.2/Resources/library/RcppEigen/include/Eigen/Core:1:
#include <complex>
^~~~~~
3 errors generated.
make: *** [foo.o] Error 1

SAMPLING FOR MODEL 'weight2' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 0.001226 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 12.26 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration: 1 / 500 [ 0%] (Warmup)
Chain 1: Iteration: 50 / 500 [ 10%] (Warmup)
Chain 1: Iteration: 100 / 500 [ 20%] (Warmup)
Chain 1: Iteration: 150 / 500 [ 30%] (Warmup)
Chain 1: Iteration: 200 / 500 [ 40%] (Warmup)
```

```

Chain 1: Iteration: 250 / 500 [ 50%] (Warmup)
Chain 1: Iteration: 251 / 500 [ 50%] (Sampling)
Chain 1: Iteration: 300 / 500 [ 60%] (Sampling)
Chain 1: Iteration: 350 / 500 [ 70%] (Sampling)
Chain 1: Iteration: 400 / 500 [ 80%] (Sampling)
Chain 1: Iteration: 450 / 500 [ 90%] (Sampling)
Chain 1: Iteration: 500 / 500 [100%] (Sampling)
Chain 1:
Chain 1: Elapsed Time: 2.56325 seconds (Warm-up)
Chain 1:                2.06526 seconds (Sampling)
Chain 1:                4.6285 seconds (Total)
Chain 1:

```

SAMPLING FOR MODEL 'weight2' NOW (CHAIN 2).

```

Chain 2:
Chain 2: Gradient evaluation took 0.000506 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 5.06 seconds.
Chain 2: Adjust your expectations accordingly!
Chain 2:
Chain 2:
Chain 2: Iteration:   1 / 500 [  0%] (Warmup)
Chain 2: Iteration:  50 / 500 [ 10%] (Warmup)
Chain 2: Iteration: 100 / 500 [ 20%] (Warmup)
Chain 2: Iteration: 150 / 500 [ 30%] (Warmup)
Chain 2: Iteration: 200 / 500 [ 40%] (Warmup)
Chain 2: Iteration: 250 / 500 [ 50%] (Warmup)
Chain 2: Iteration: 251 / 500 [ 50%] (Sampling)
Chain 2: Iteration: 300 / 500 [ 60%] (Sampling)
Chain 2: Iteration: 350 / 500 [ 70%] (Sampling)
Chain 2: Iteration: 400 / 500 [ 80%] (Sampling)
Chain 2: Iteration: 450 / 500 [ 90%] (Sampling)
Chain 2: Iteration: 500 / 500 [100%] (Sampling)
Chain 2:
Chain 2: Elapsed Time: 2.60051 seconds (Warm-up)
Chain 2:                2.78029 seconds (Sampling)
Chain 2:                5.38081 seconds (Total)
Chain 2:

```

SAMPLING FOR MODEL 'weight2' NOW (CHAIN 3).

```

Chain 3:
Chain 3: Gradient evaluation took 0.00046 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 4.6 seconds.
Chain 3: Adjust your expectations accordingly!

```

```

Chain 3:
Chain 3:
Chain 3: Iteration:   1 / 500 [  0%] (Warmup)
Chain 3: Iteration:  50 / 500 [ 10%] (Warmup)
Chain 3: Iteration: 100 / 500 [ 20%] (Warmup)
Chain 3: Iteration: 150 / 500 [ 30%] (Warmup)
Chain 3: Iteration: 200 / 500 [ 40%] (Warmup)
Chain 3: Iteration: 250 / 500 [ 50%] (Warmup)
Chain 3: Iteration: 251 / 500 [ 50%] (Sampling)
Chain 3: Iteration: 300 / 500 [ 60%] (Sampling)
Chain 3: Iteration: 350 / 500 [ 70%] (Sampling)
Chain 3: Iteration: 400 / 500 [ 80%] (Sampling)
Chain 3: Iteration: 450 / 500 [ 90%] (Sampling)
Chain 3: Iteration: 500 / 500 [100%] (Sampling)
Chain 3:
Chain 3: Elapsed Time: 2.84245 seconds (Warm-up)
Chain 3:                2.2888 seconds (Sampling)
Chain 3:                5.13125 seconds (Total)
Chain 3:

```

SAMPLING FOR MODEL 'weight2' NOW (CHAIN 4).

```

Chain 4:
Chain 4: Gradient evaluation took 0.000427 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 4.27 seconds.
Chain 4: Adjust your expectations accordingly!
Chain 4:
Chain 4:
Chain 4: Iteration:   1 / 500 [  0%] (Warmup)
Chain 4: Iteration:  50 / 500 [ 10%] (Warmup)
Chain 4: Iteration: 100 / 500 [ 20%] (Warmup)
Chain 4: Iteration: 150 / 500 [ 30%] (Warmup)
Chain 4: Iteration: 200 / 500 [ 40%] (Warmup)
Chain 4: Iteration: 250 / 500 [ 50%] (Warmup)
Chain 4: Iteration: 251 / 500 [ 50%] (Sampling)
Chain 4: Iteration: 300 / 500 [ 60%] (Sampling)
Chain 4: Iteration: 350 / 500 [ 70%] (Sampling)
Chain 4: Iteration: 400 / 500 [ 80%] (Sampling)
Chain 4: Iteration: 450 / 500 [ 90%] (Sampling)
Chain 4: Iteration: 500 / 500 [100%] (Sampling)
Chain 4:
Chain 4: Elapsed Time: 2.8805 seconds (Warm-up)
Chain 4:                1.93052 seconds (Sampling)
Chain 4:                4.81102 seconds (Total)

```

Chain 4:

## Question 5

For reference I have uploaded some model 2 results. Check your results are similar.

	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

## Answer

My results are similar to the one above except that 'beta[2]' and 'beta[3]' are flipped.

	mean	se_mean	sd	2.5%	25%	50%
beta[1]	1.1695474	7.775215e-05	0.002730748	1.16432439	1.16774539	1.1694545
beta[2]	0.1020646	1.333365e-04	0.003540319	0.09526538	0.09965927	0.1020458
beta[3]	0.5634542	4.555121e-03	0.066073353	0.42870273	0.52058743	0.5622696
beta[4]	0.1983429	9.280153e-04	0.013709344	0.17026957	0.18956716	0.1986937
sigma	0.1611931	7.237822e-05	0.001813104	0.15776604	0.16004710	0.1610828

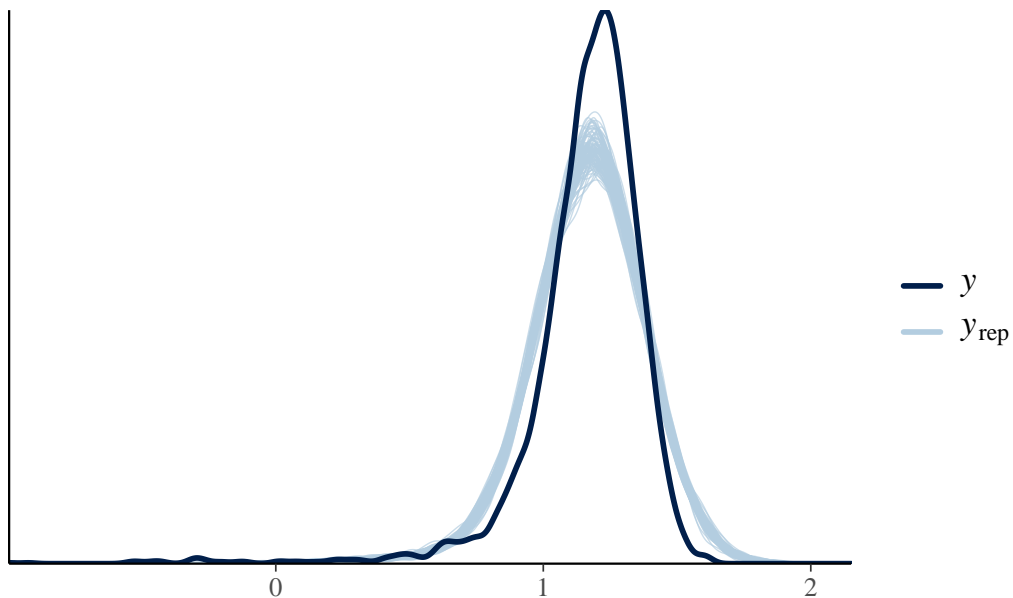
	75%	97.5%	n_eff	Rhat
beta[1]	1.1714191	1.1746801	1233.4982	1.000121
beta[2]	0.1044228	0.1088863	704.9956	1.008594
beta[3]	0.6072238	0.6902591	210.4035	1.031391
beta[4]	0.2076363	0.2246584	218.2343	1.027843
sigma	0.1624286	0.1648346	627.5230	1.000475

## PPCs

Now we've run two candidate models let's do some posterior predictive checks. The `bayesplot` package has a lot of inbuilt graphing functions to do this. For example, let's plot the distribution of our data ( $y$ ) against 100 different datasets drawn from the posterior predictive distribution:

```
[1] 1000 3842
```

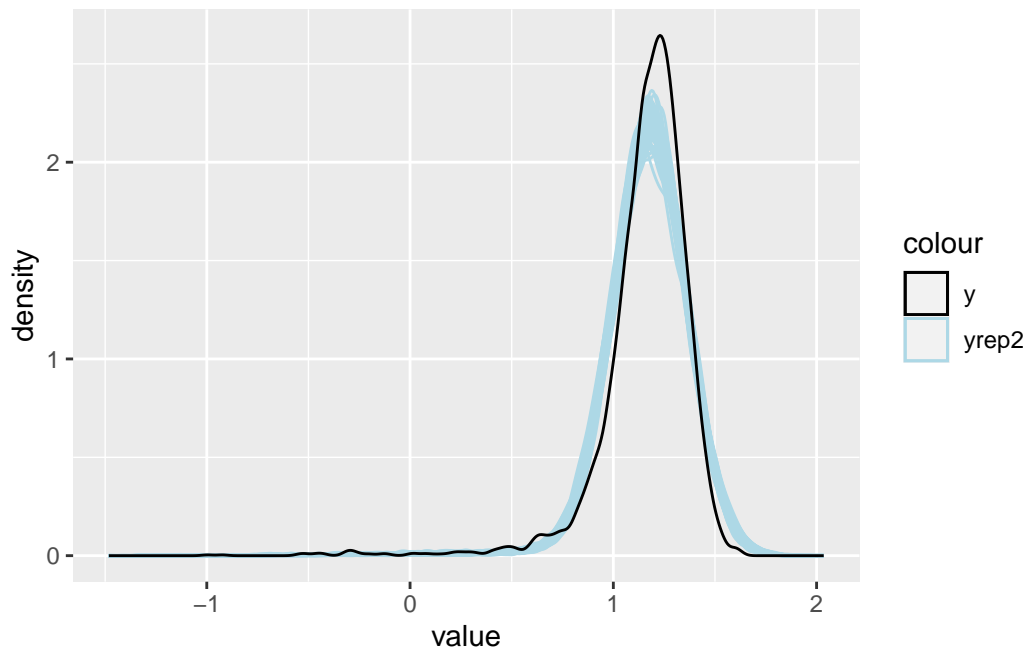
distribution of observed versus predicted birthweights



### Question 6

Make a similar plot to the one above but for model 2, and **not** using the bayes plot in built function (i.e. do it yourself just with `geom_density`)

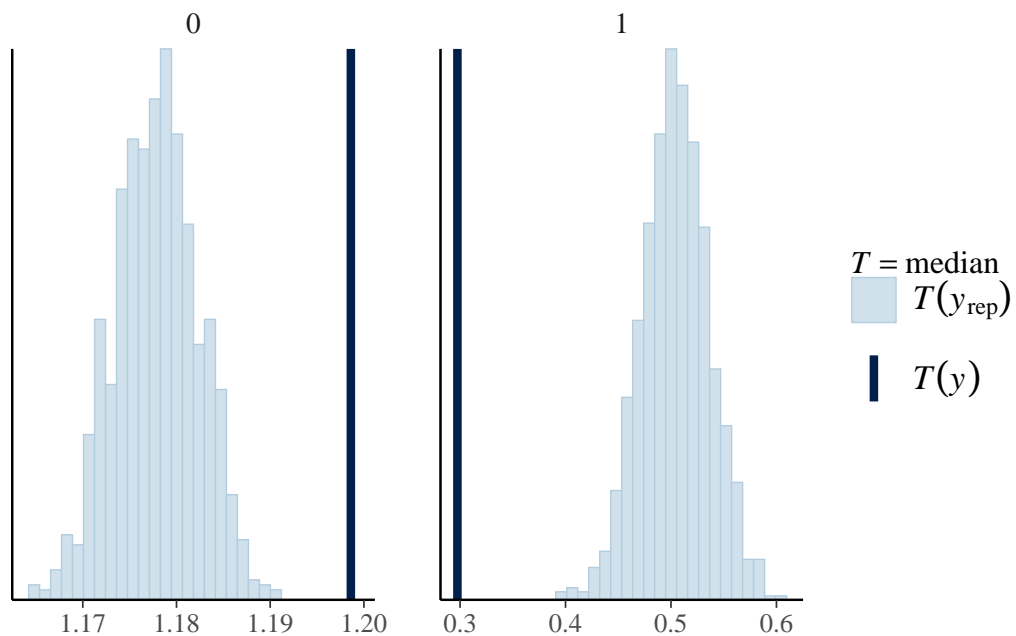
## Answer



## Test statistics

We can also look at some summary statistics in the PPD versus the data, again either using `bayesplot` – the function of interest is `ppc_stat` or `ppc_stat_grouped` – or just doing it ourselves using `ggplot`.

E.g. medians by prematurity for Model 1



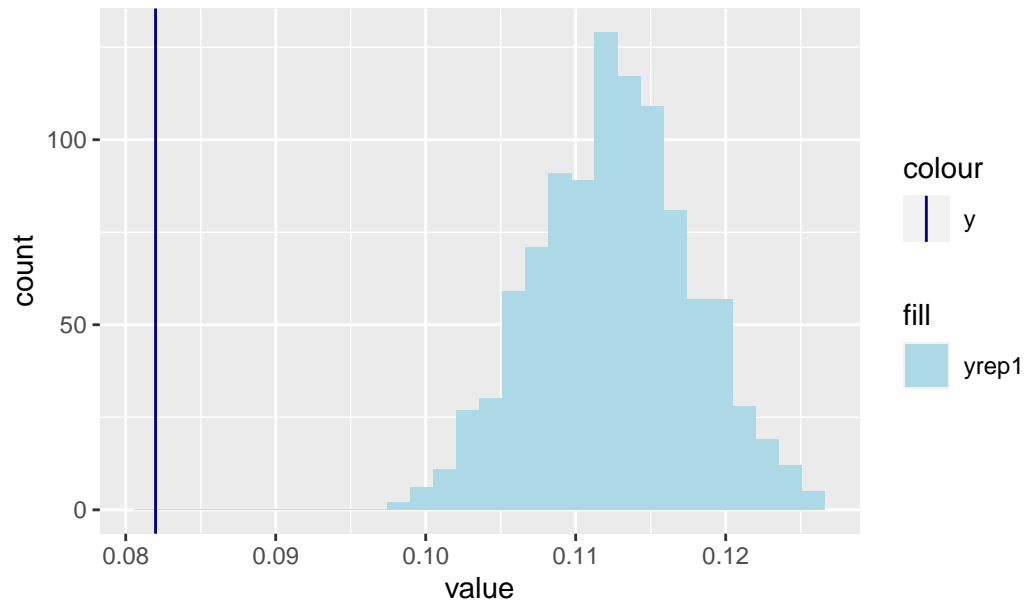
### Question 7

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).

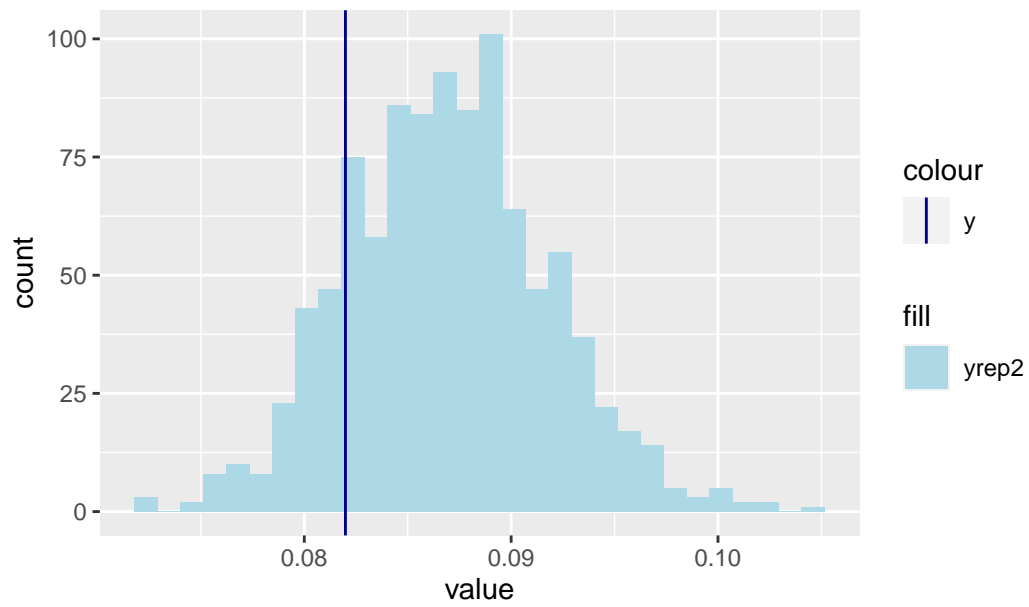


## Answer

Model 1



Model 2



## LOO

Finally let's calculate the LOO elpd for each model and compare. The first step of this is to get the point-wise log likelihood estimates from each model:

And then we can use these in the `loo` function to get estimates for the elpd. Note the `save_psis = TRUE` argument saves the calculation for each simulated draw, which is needed for the LOO-PIT calculation below.

Look at the output:

Computed from 1000 by 3842 log-likelihood matrix

	Estimate	SE
elpd_loo	1377.2	72.6
p_loo	9.6	1.5
looic	-2754.5	145.2

-----

Monte Carlo SE of elpd\_loo is 0.1.

All Pareto k estimates are good ( $k < 0.5$ ).  
See `help('pareto-k-diagnostic')` for details.

Computed from 1000 by 3842 log-likelihood matrix

	Estimate	SE
elpd_loo	1552.3	69.9
p_loo	15.7	2.4
looic	-3104.7	139.7

-----

Monte Carlo SE of elpd\_loo is 0.2.

Pareto k diagnostic values:

		Count	Pct.	Min. n_eff
(-Inf, 0.5]	(good)	3841	100.0%	220
(0.5, 0.7]	(ok)	1	0.0%	462
(0.7, 1]	(bad)	0	0.0%	<NA>
(1, Inf)	(very bad)	0	0.0%	<NA>

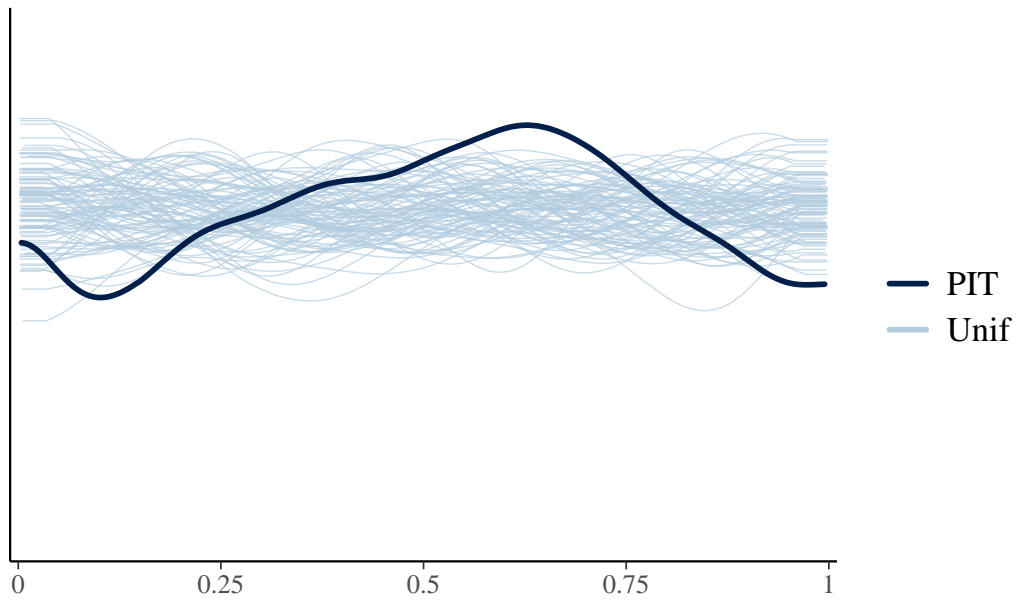
All Pareto k estimates are ok ( $k < 0.7$ ).  
See `help('pareto-k-diagnostic')` for details.

Comparing the two models tells us Model 2 is better:

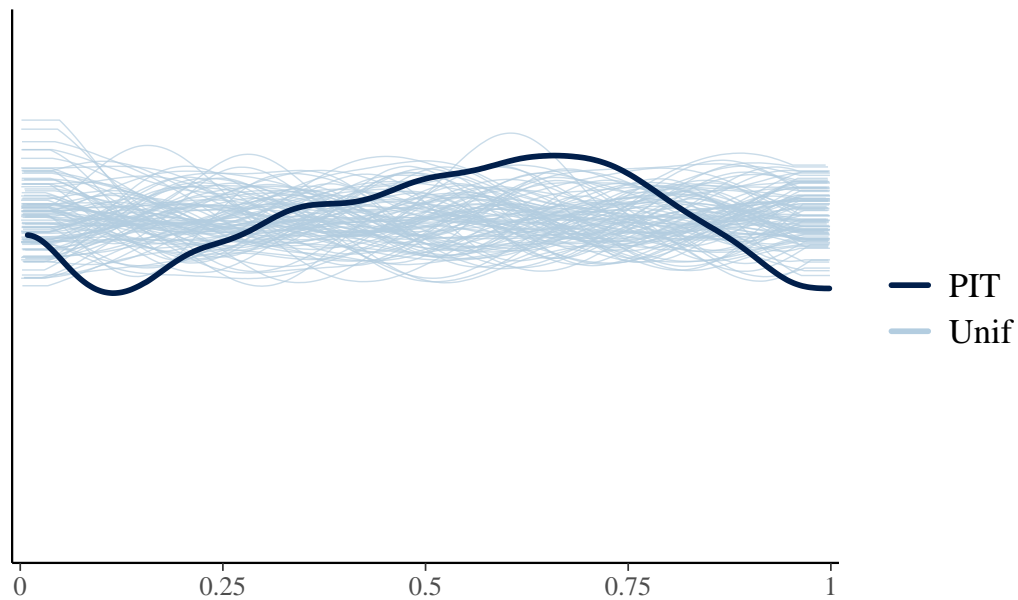
	elpd_diff	se_diff
model2	0.0	0.0
model1	-175.1	36.5

We can also compare the LOO-PIT of each of the models to standard uniforms. The both do pretty well.

**Model1**



## Model2



### Bonus question (not required)

Create your own PIT histogram “from scratch” for Model 2.

### 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.

### Answer

We chose ‘sex’ as a covariate and build a model as follows:

$$\log(y_i) \sim N(\beta_1 + \beta_2 \log(x_i) + \beta_3 z_i + \beta_4 \log(x_i) z_i + \beta_5 w_i, \sigma^2),$$

where  $w_i$  is sex (0 or 1, if male).

Now fit the model

```

Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
clang -mmacosx-version-min=10.13 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.2/Resources/library/StanHeaders/include/stan/math/
In file included from /Library/Frameworks/R.framework/Versions/4.2/Resources/library/RcppEigen/include/Eigen/src/Core/
In file included from /Library/Frameworks/R.framework/Versions/4.2/Resources/library/RcppEigen/include/Eigen/src/Core/
namespace Eigen {
~

/Library/Frameworks/R.framework/Versions/4.2/Resources/library/RcppEigen/include/Eigen/src/Core/
namespace Eigen {
~
;
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.2/Resources/library/StanHeaders/include/stan/math/
In file included from /Library/Frameworks/R.framework/Versions/4.2/Resources/library/RcppEigen/include/Eigen/Core:
#include <complex>
~~~~~~
3 errors generated.
make: *** [foo.o] Error 1

```

```

SAMPLING FOR MODEL 'weight3' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 0.001462 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 14.62 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration: 1 / 500 [ 0%] (Warmup)
Chain 1: Iteration: 50 / 500 [ 10%] (Warmup)
Chain 1: Iteration: 100 / 500 [ 20%] (Warmup)
Chain 1: Iteration: 150 / 500 [ 30%] (Warmup)
Chain 1: Iteration: 200 / 500 [ 40%] (Warmup)
Chain 1: Iteration: 250 / 500 [ 50%] (Warmup)
Chain 1: Iteration: 251 / 500 [ 50%] (Sampling)
Chain 1: Iteration: 300 / 500 [ 60%] (Sampling)
Chain 1: Iteration: 350 / 500 [ 70%] (Sampling)
Chain 1: Iteration: 400 / 500 [ 80%] (Sampling)
Chain 1: Iteration: 450 / 500 [ 90%] (Sampling)
Chain 1: Iteration: 500 / 500 [100%] (Sampling)
Chain 1:
Chain 1: Elapsed Time: 3.50429 seconds (Warm-up)

```

Chain 1: 3.8366 seconds (Sampling)  
Chain 1: 7.34089 seconds (Total)  
Chain 1:

SAMPLING FOR MODEL 'weight3' NOW (CHAIN 2).

Chain 2:  
Chain 2: Gradient evaluation took 0.000583 seconds  
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 5.83 seconds.  
Chain 2: Adjust your expectations accordingly!  
Chain 2:  
Chain 2:  
Chain 2: Iteration: 1 / 500 [ 0%] (Warmup)  
Chain 2: Iteration: 50 / 500 [ 10%] (Warmup)  
Chain 2: Iteration: 100 / 500 [ 20%] (Warmup)  
Chain 2: Iteration: 150 / 500 [ 30%] (Warmup)  
Chain 2: Iteration: 200 / 500 [ 40%] (Warmup)  
Chain 2: Iteration: 250 / 500 [ 50%] (Warmup)  
Chain 2: Iteration: 251 / 500 [ 50%] (Sampling)  
Chain 2: Iteration: 300 / 500 [ 60%] (Sampling)  
Chain 2: Iteration: 350 / 500 [ 70%] (Sampling)  
Chain 2: Iteration: 400 / 500 [ 80%] (Sampling)  
Chain 2: Iteration: 450 / 500 [ 90%] (Sampling)  
Chain 2: Iteration: 500 / 500 [100%] (Sampling)  
Chain 2:  
Chain 2: Elapsed Time: 3.67053 seconds (Warm-up)  
Chain 2: 3.00095 seconds (Sampling)  
Chain 2: 6.67147 seconds (Total)  
Chain 2:

SAMPLING FOR MODEL 'weight3' NOW (CHAIN 3).

Chain 3:  
Chain 3: Gradient evaluation took 0.000553 seconds  
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 5.53 seconds.  
Chain 3: Adjust your expectations accordingly!  
Chain 3:  
Chain 3:  
Chain 3: Iteration: 1 / 500 [ 0%] (Warmup)  
Chain 3: Iteration: 50 / 500 [ 10%] (Warmup)  
Chain 3: Iteration: 100 / 500 [ 20%] (Warmup)  
Chain 3: Iteration: 150 / 500 [ 30%] (Warmup)  
Chain 3: Iteration: 200 / 500 [ 40%] (Warmup)  
Chain 3: Iteration: 250 / 500 [ 50%] (Warmup)  
Chain 3: Iteration: 251 / 500 [ 50%] (Sampling)

```

Chain 3: Iteration: 300 / 500 [ 60%] (Sampling)
Chain 3: Iteration: 350 / 500 [ 70%] (Sampling)
Chain 3: Iteration: 400 / 500 [ 80%] (Sampling)
Chain 3: Iteration: 450 / 500 [ 90%] (Sampling)
Chain 3: Iteration: 500 / 500 [100%] (Sampling)
Chain 3:
Chain 3: Elapsed Time: 4.24582 seconds (Warm-up)
Chain 3:           2.76068 seconds (Sampling)
Chain 3:           7.0065 seconds (Total)
Chain 3:

```

SAMPLING FOR MODEL 'weight3' NOW (CHAIN 4).

```

Chain 4:
Chain 4: Gradient evaluation took 0.000602 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 6.02 seconds.
Chain 4: Adjust your expectations accordingly!
Chain 4:
Chain 4:
Chain 4: Iteration:   1 / 500 [  0%] (Warmup)
Chain 4: Iteration:  50 / 500 [ 10%] (Warmup)
Chain 4: Iteration: 100 / 500 [ 20%] (Warmup)
Chain 4: Iteration: 150 / 500 [ 30%] (Warmup)
Chain 4: Iteration: 200 / 500 [ 40%] (Warmup)
Chain 4: Iteration: 250 / 500 [ 50%] (Warmup)
Chain 4: Iteration: 251 / 500 [ 50%] (Sampling)
Chain 4: Iteration: 300 / 500 [ 60%] (Sampling)
Chain 4: Iteration: 350 / 500 [ 70%] (Sampling)
Chain 4: Iteration: 400 / 500 [ 80%] (Sampling)
Chain 4: Iteration: 450 / 500 [ 90%] (Sampling)
Chain 4: Iteration: 500 / 500 [100%] (Sampling)
Chain 4:
Chain 4: Elapsed Time: 3.74266 seconds (Warm-up)
Chain 4:           3.43875 seconds (Sampling)
Chain 4:           7.18141 seconds (Total)
Chain 4:

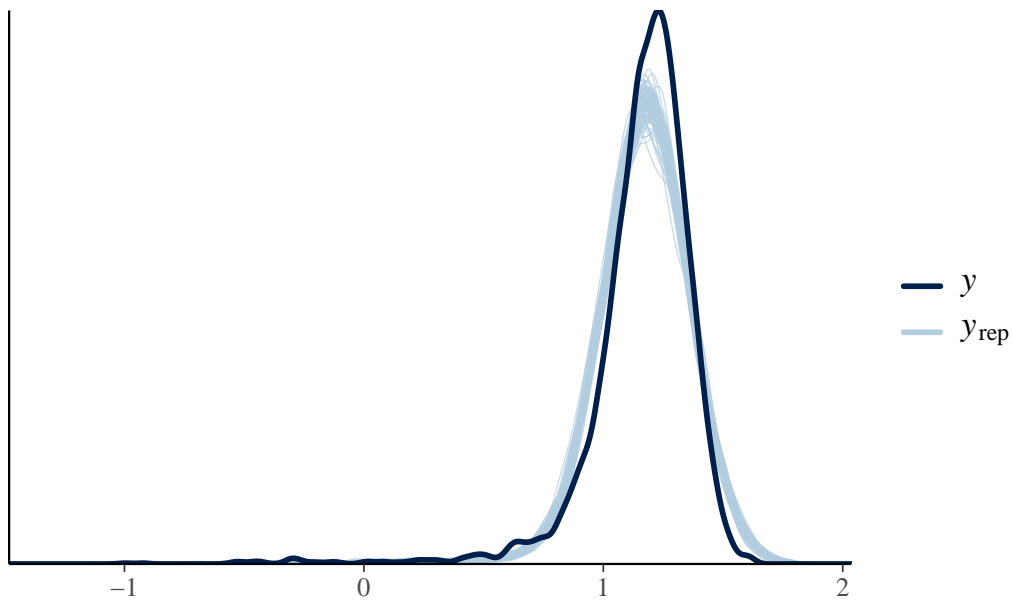
```

	mean	se_mean	sd	2.5%	25%	50%
beta[1]	1.14844358	0.0001291497	0.003692184	1.14087258	1.14614805	1.14849077
beta[2]	0.10244761	0.0001087468	0.003497191	0.09601436	0.09993039	0.10226833
beta[3]	0.55433633	0.0032287618	0.065870173	0.42585550	0.51003456	0.55391326
beta[4]	0.19599569	0.0006725433	0.013724689	0.16955548	0.18636853	0.19611854
beta[5]	0.04224789	0.0001877414	0.005139763	0.03203777	0.03879210	0.04226883

sigma	0.15990533	0.0000813047	0.001858884	0.15652342	0.15864097	0.15975673
	75%	97.5%	n_eff	Rhat		
beta[1]	1.15099808	1.15561356	817.2963	0.9992439		
beta[2]	0.10491610	0.10930858	1034.2030	0.9994427		
beta[3]	0.59698866	0.68957203	416.2034	1.0093294		
beta[4]	0.20501423	0.22307944	416.4516	1.0059028		
beta[5]	0.04575562	0.05199171	749.4905	0.9985962		
sigma	0.16111271	0.16378550	522.7251	0.9974357		

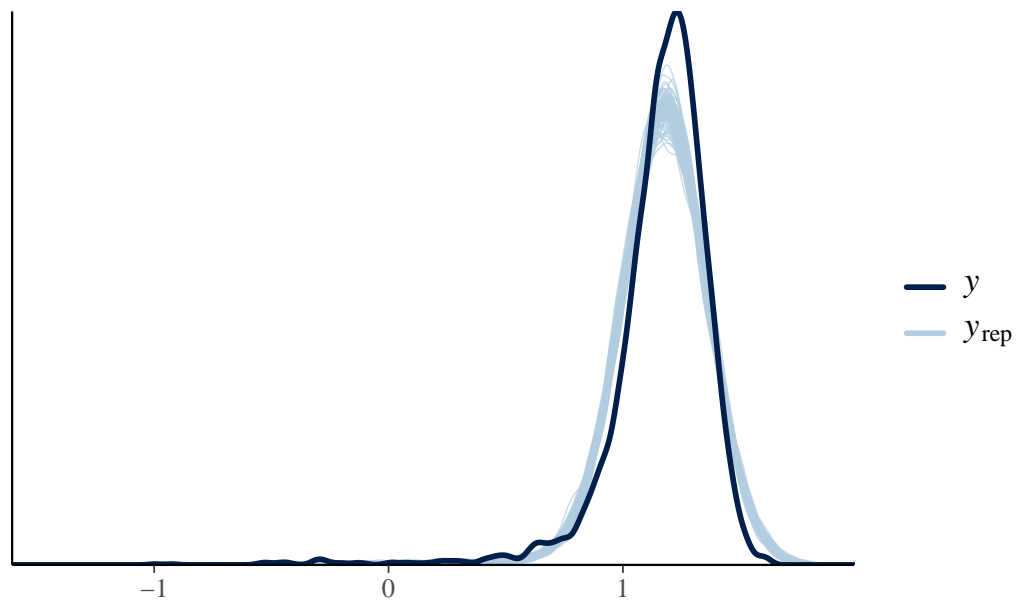
We now compare distribution of predicted birthweights with model 2 and model 3. From the following plots, we cannot see evident difference between them.

### Model 2: distribution of observed versus predicted birthweights





### Model 3: distribution of observed versus predicted birthweights



Next, we use leave one out method and compare  $ELPD_{loo}$  with model 2.

Computed from 1000 by 3842 log-likelihood matrix

	Estimate	SE
elpd_loo	1552.3	69.9
p_loo	15.7	2.4
looic	-3104.7	139.7

-----  
Monte Carlo SE of elpd\_loo is 0.2.

Pareto k diagnostic values:

		Count	Pct.	Min. n_eff
(-Inf, 0.5]	(good)	3841	100.0%	220
(0.5, 0.7]	(ok)	1	0.0%	462
(0.7, 1]	(bad)	0	0.0%	<NA>
(1, Inf)	(very bad)	0	0.0%	<NA>

All Pareto k estimates are ok ( $k < 0.7$ ).  
See `help('pareto-k-diagnostic')` for details.

Computed from 1000 by 3842 log-likelihood matrix

	Estimate	SE
elpd_loo	1584.2	70.4
p_loo	17.0	2.5
looic	-3168.5	140.7

-----

Monte Carlo SE of elpd\_loo is 0.2.

All Pareto k estimates are good ( $k < 0.5$ ).  
 See `help('pareto-k-diagnostic')` for details.

Comparing these two, model 3 has slightly larger elpd\_loo and thus we conclude model 3 is better. However, the difference is small. So, it may also be reasonable to choose model 2 for simplicity.