STA2201H Winter 2021 Assignment 2

Due: 11:59pm ET, March 29 2021

What to hand in: .Rmd file and the compiled pdf

How to hand in: Submit files via Quercus

Note that at the end of this document there are details about the research proposal.

1 Wells

This question uses data looking at the decision of households in Bangladesh to switch drinking water wells in response to their well being marked as unsafe or not. A full description from the Gelman Hill text book (page 87):

"Many of the wells used for drinking water in Bangladesh and other South Asian countries are contaminated with natural arsenic, affecting an estimated 100 million people. Arsenic is a cumulative poison, and exposure increases the risk of cancer and other diseases, with risks estimated to be proportional to exposure. Any locality can include wells with a range of arsenic levels. The bad news is that even if your neighbor's well is safe, it does not mean that yours is safe. However, the corresponding good news is that, if your well has a high arsenic level, you can probably find a safe well nearby to get your water from—if you are willing to walk the distance and your neighbor is willing to share. [In an area of Bangladesh, a research team] measured all the wells and labeled them with their arsenic level as well as a characterization as "safe" (below 0.5 in units of hundreds of micrograms per liter, the Bangladesh standard for arsenic in drinking water) or "unsafe" (above 0.5). People with unsafe wells were encouraged to switch to nearby private or community wells or to new wells of their own construction. A few years later, the researchers returned to find out who had switched wells."

The outcome of interest is whether or not household i switched wells:

$$y_i = \begin{cases} 1 & \text{if household } i \text{ switched to a new well} \\ 0 & \text{if household } i \text{ continued using its own well.} \end{cases}$$

The data we are using for this question are here: http://www.stat.columbia.edu/~gelman/arm/examples/arsenic/wells.dat and you can load them in directly using

The variables of interest for this questions are

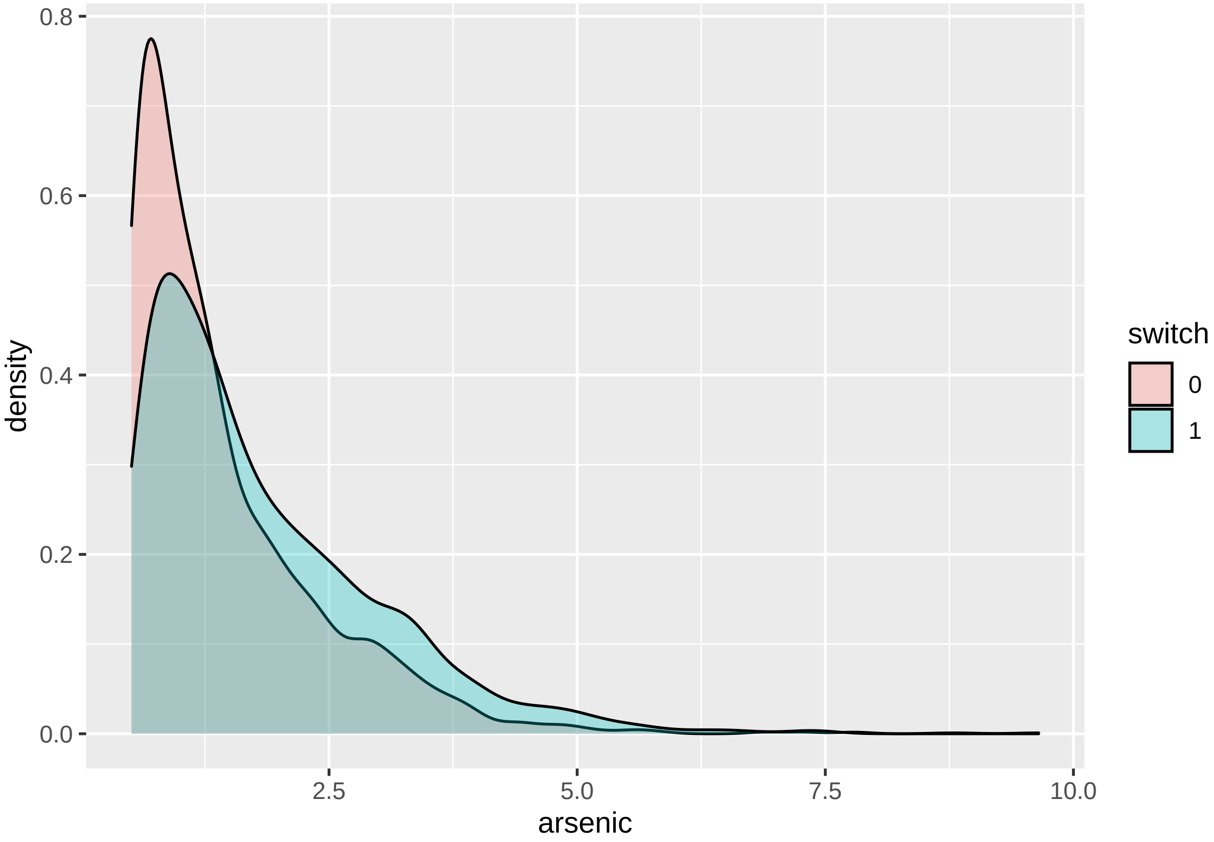
- `switch`, which is y_i above
- `arsenic`, the level of arsenic of the respondent's well
- `dist`, the distance (in metres) of the closest known safe well

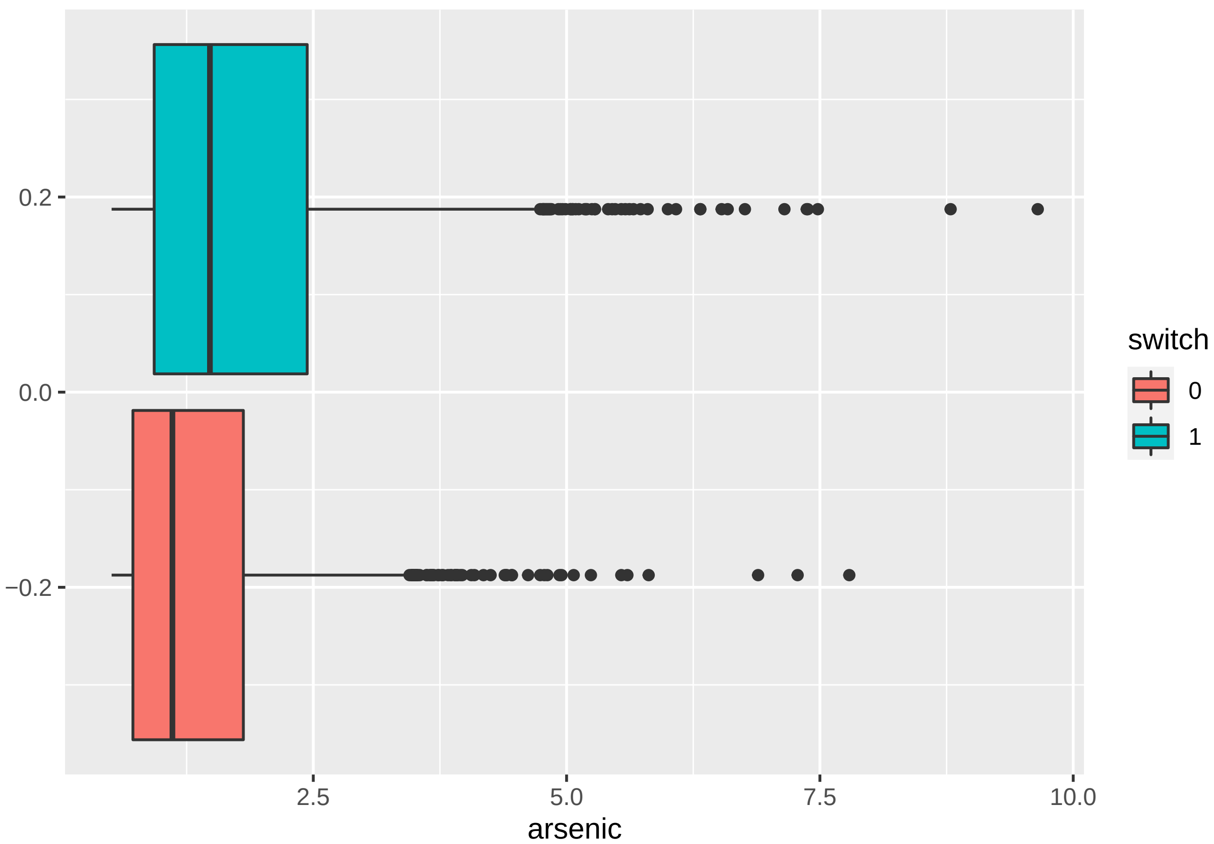
a) Do an exploratory data analysis illustrating the relationship between well-switching, distance and arsenic. Think about different ways of effectively illustrating the relationships given the binary outcome. As usual, a good EDA includes well-thought-out descriptions and analysis of any graphs and tables provided, well-labelled axes, titles etc.

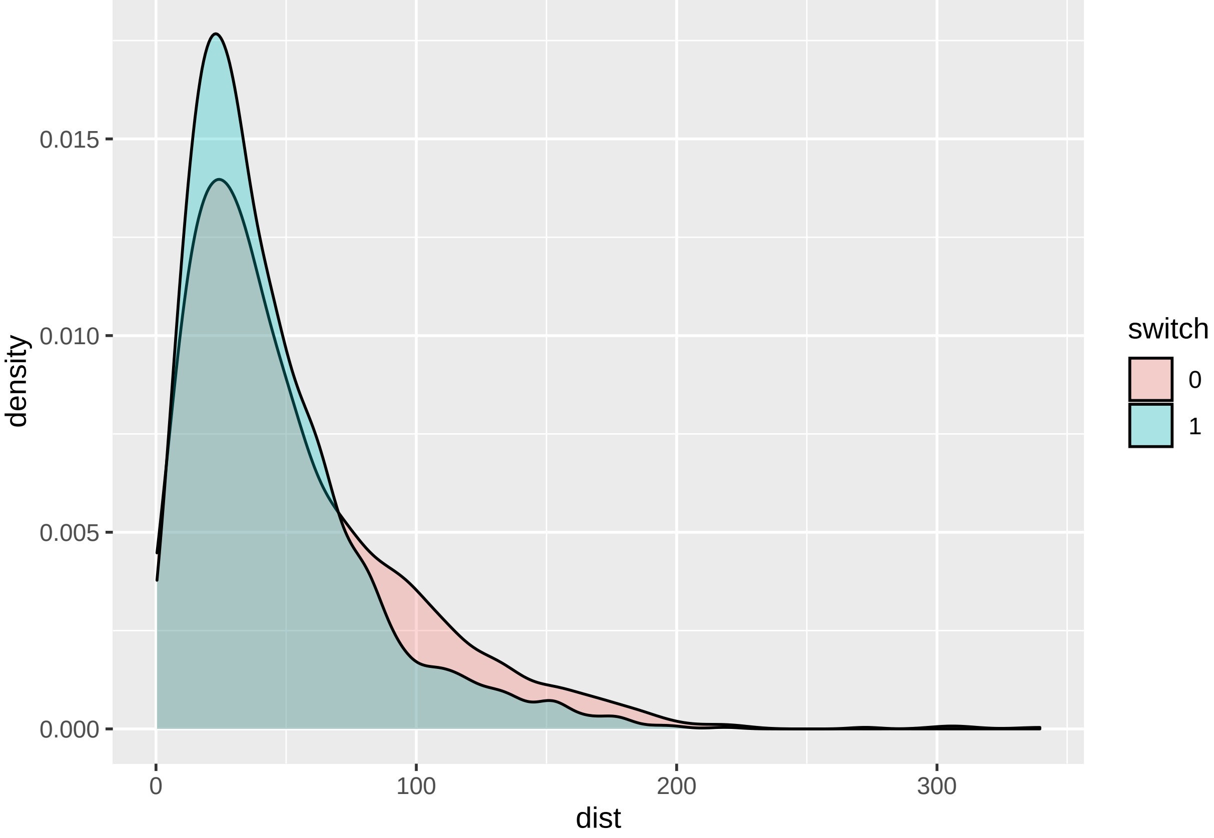
- Answer: In this dataset, we have 1737 people choose to switch their wells and 1283 people choose not to switch their wells. So the dataset is not very well balanced. The quantile for arsenic and distance can be displayed below. Both of their trends are right skewed. We will discuss their distribution by group next.

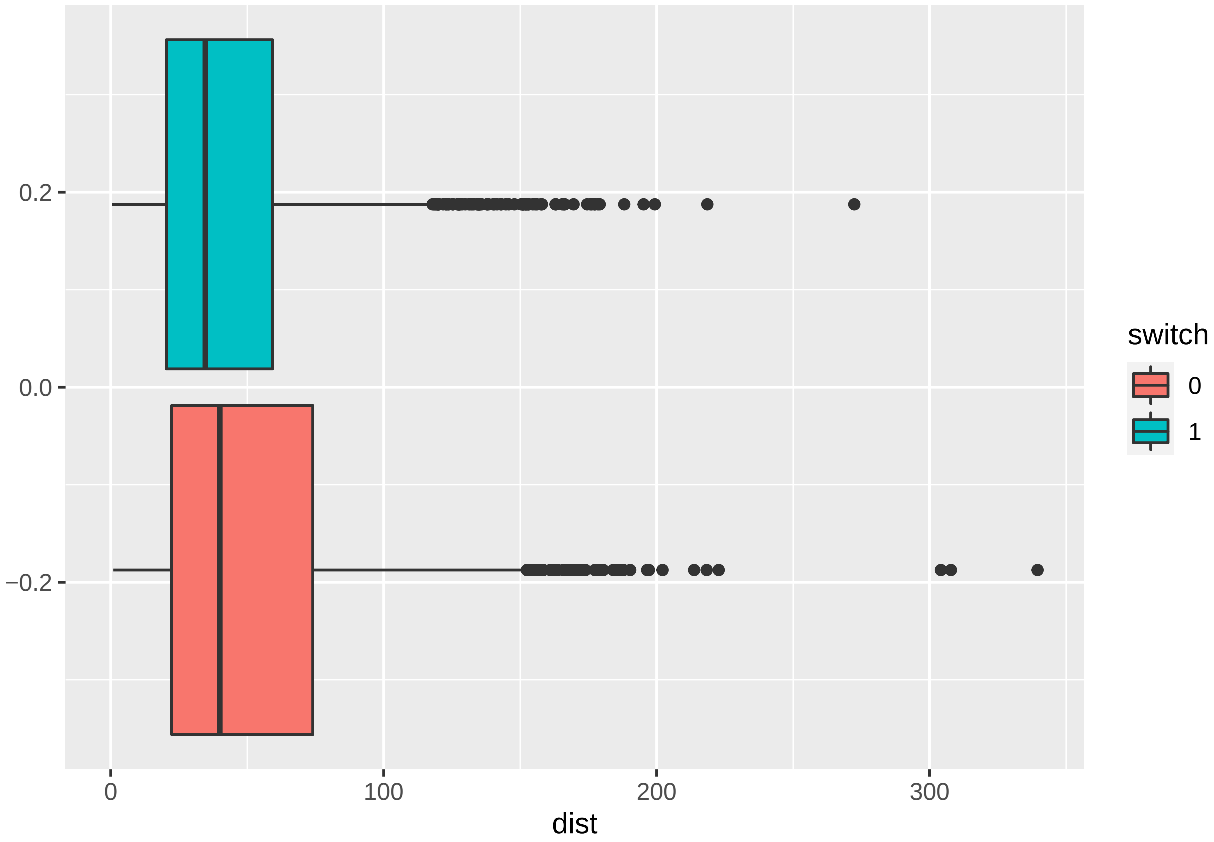
	min	25%	median	75%	max
arsenic	0.51	0.82	1.30	2.20	9.65
distance	0.387	21.117	36.761	64.04	339.531

We then want to explore the realtionship between switch well and arisenic. From the graph below we could see there is a postive relation between arsenic and switch, and a negative relation between distance and switch. Both of them has a quite long tails, which indicate the existance of some outliers. The peak of switch and not switch are both near 0.5. However, there are still more people prefer not to switch even a little above 0.5. This trend change when arsenic is close to 1.25. Where more people prefer to switch the well. However for the distance it is not the case. We could find the peak for switch and not to switch are both at 25, although people prefer to switch the well at 25. This trend changes when the distance is greater than 75. More people prefer not to switch at this point.




0.2
0.0
−0.2
2.5
5.0
7.5
10.0
arsenic
switch
0
1


density
0.015
0.010
0.005
0.000
0
100
200
300
dist
switch
0
1


0.2
0.0
−0.2
0
100
200
300
dist
switch
0
1

Assume $y_i \sim Bern(p_i)$, where p_i refers to the probability of switching. Consider two candidate models.

- Model 1:

$$\text{logit}\,(p_i) = \beta_0 + \beta_1 \cdot \left(d_i - \bar{d}\right) + \beta_2 \cdot (a_i - \bar{a}) + \beta_3 \cdot \left(d_i - \bar{d}\right)(a_i - \bar{a})$$

- Model 2:

$$\begin{aligned}\text{logit}\,(p_i) =& \beta_0 + \beta_1 \cdot \left(d_i - \bar{d}\right) + \beta_2 \cdot \left(\log(a_i) - \overline{\log(a)}\right) \\ &+ \beta_3 \cdot \left(d_i - \bar{d}\right)\left(\log(a_i) - \overline{\log(a)}\right)\end{aligned}$$

where d_i is distance and a_i is arsenic level.

b) Fit both of these models using Stan. Put $N(0,1)$ priors on all the βs. You should generate pointwise log likelihood estimates (to be used in later questions), and also samples from the posterior predictive distribution (unless you'd prefer to do it in R later on). For model 1, interpret each coefficient.

- Answer

beta_0	beta_1	beta_2	beta_3
0.3407	-0.0095	0.4700	-0.0018

First we show the coefficient in a table. Since we used a logit function during the transformation. We could use odds ratio to interpret the coefficients. We could use beta_1 as an example but the other coefficients follow the same logic. Since we could define $logit(p_i) = log(\frac{P(y=1|X)}{P(y=0|X)}) = \beta_0 + \beta_1(d_i - \bar{d}) + \beta_2(a_i - \bar{a}) + \beta_3(d_i - \bar{d})(a_i - \bar{a})$. For every one unit increase in $(d_i - \bar{d})$, which is equivalent to -0.0095 increase in the logit. There will be exp(-.0095) = 0.9905 increase in the odds $\frac{P(y=1|X)}{P(y=0|X)}$.

```r
N <- length(d$switch)
y <- as.numeric(d$switch)-1
arsenic <- d$arsenic - mean(d$arsenic)
dist <- d$dist - mean(d$dist)
covariate <- arsenic * dist

log_arsenic = log(d$arsenic) - mean(log(d$arsenic))
covariate2 <- log_arsenic * dist

data <- list(N = N,
             y = y,
             arsenic = arsenic,
             dist = dist,
             covariate = covariate)

data2 <- list(N = N,
              y = y,
              arsenic = log_arsenic,
              dist = dist,
              covariate = covariate2
              )

fit1 <- stan(file = "./q1b_stan.stan", data = data, iter = 500)
```

```
## Trying to compile a simple C file

## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and median
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess

## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and ta
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess
```

```
fit2 <- stan(file = "./q1b_stan.stan", data = data2, iter = 500)
```

```
## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and median
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess

## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and ta
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess
```

```
summary(fit1)$summary[c(1:4),]
```

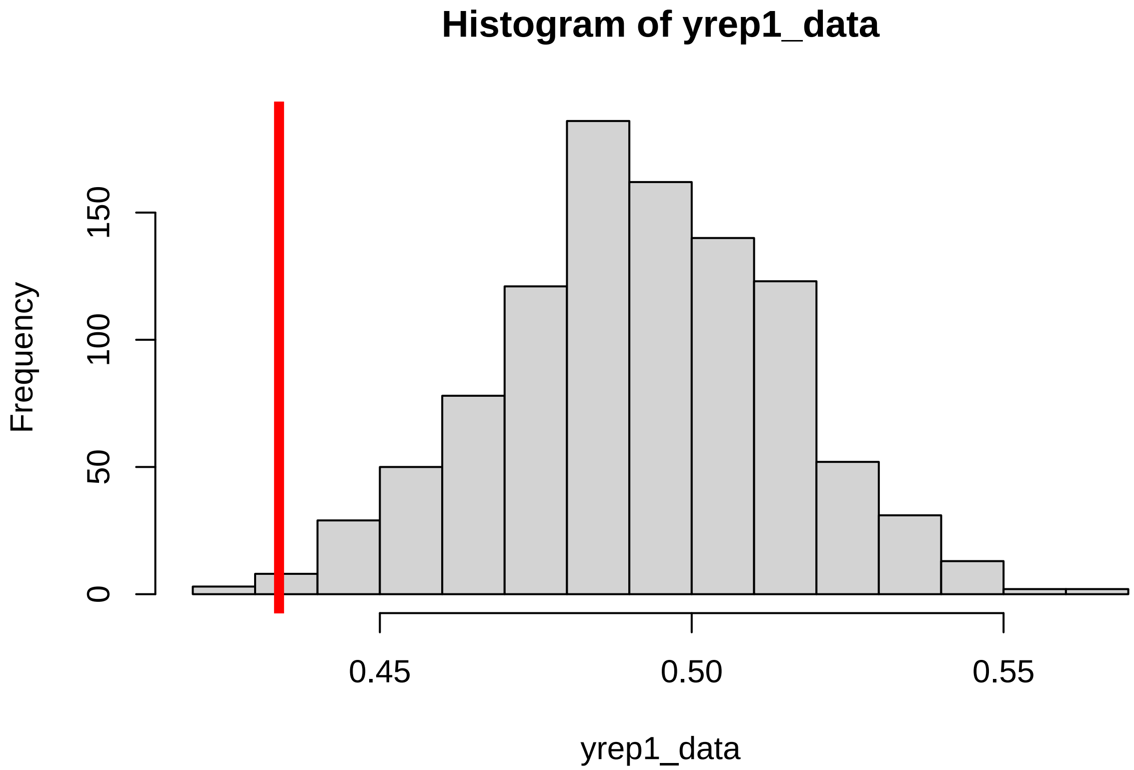
```
##               mean      se_mean          sd         2.5%          25%
## beta0  0.351307525 2.287823e-03 0.041309973  0.271418310  0.325489832
## beta1 -0.008785669 3.266757e-05 0.001071849 -0.010877816 -0.009506836
## beta2  0.474884764 2.449222e-03 0.042414549  0.390358035  0.443267510
## beta3 -0.001776696 3.257836e-05 0.001035220 -0.003738255 -0.002527081
##                50%          75%         97.5%     n_eff      Rhat
## beta0  0.350439946  0.379693197  0.4333690171  326.0357 1.0050562
## beta1 -0.008795347 -0.008033847 -0.0067795496 1076.5477 1.0000461
## beta2  0.476589249  0.505192377  0.5533523219  299.8979 1.0149901
## beta3 -0.001756788 -0.001065219  0.0001719283 1009.7346 0.9973899
```

c) Let $t(\boldsymbol{y}) = \sum_{i=1}^{n} 1\left(y_i = 1, a_i < 0.82\right) / \sum_{i=1}^{n} 1\left(a_i < 0.82\right)$ i.e. the proportion of households that switch with arsenic level less than 0.82. Calculate $t(\boldsymbol{y^{rep}})$ for each replicated dataset for each model, plot the resulting histogram for each model and compare to the observed value of $t(\boldsymbol{y})$. Calculate $P\left(t\left(\boldsymbol{y}^{rep}\right) < t(\boldsymbol{y})\right)$ for each model. Interpret your findings.

- Answer

Below we could see the plot for two different models. We could see the mean of $t(y^{rep})$ from the second model is much closer to the real $t(y)$ than the first model. Then if we calculate $P(t(y^{rep}) < t(y))$, for the first model is 0.4% and for the second model is 27%. So from those two findings we could make our initial conclusion, the second model fit the data better than the first model.

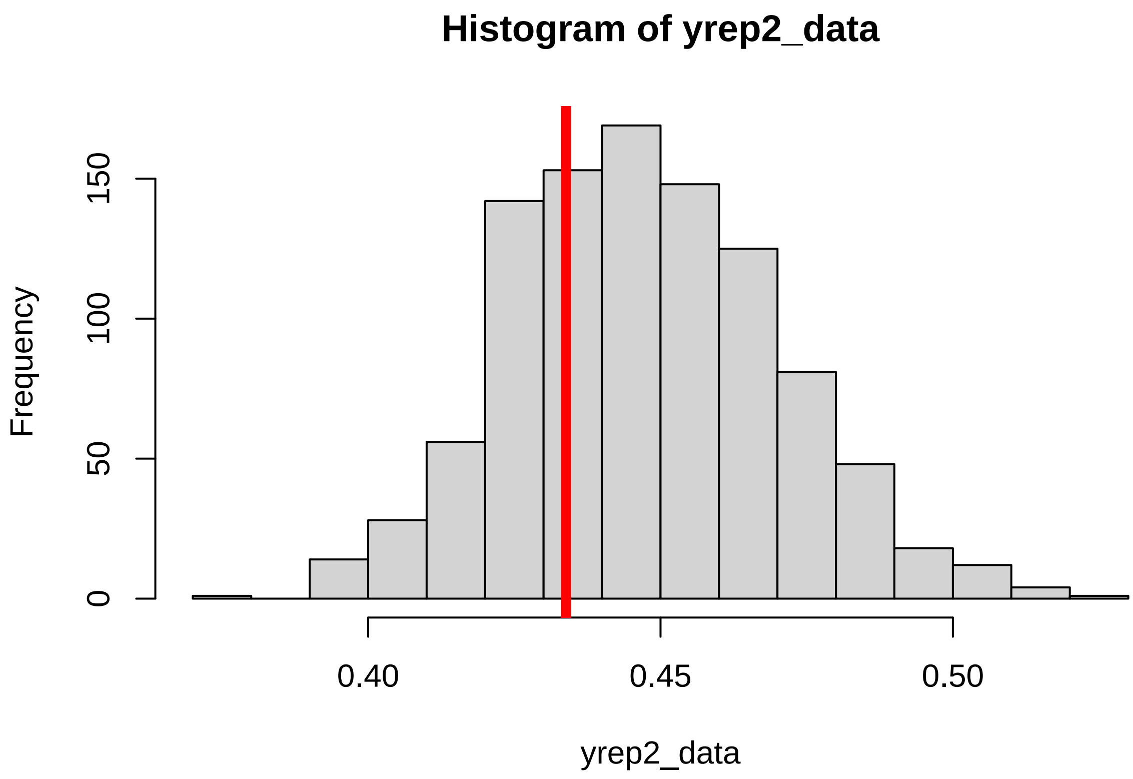
```
real_ty <- sum(d$switch == 1 & d$arsenic < 0.82)/sum(d$arsenic < 0.82)
ty <- function(X,Y){
  result = rep(0,nrow(X))
  for(j in c(1:nrow(X))){
    result[j] = sum(X[j,] == 1 & Y)/sum(Y)
  }
  return(result)
}
yrep1 <- rstan::extract(fit1)[["log_weight_rep"]]
yrep1_data <- ty(yrep1, d$arsenic < .82)
hist(yrep1_data)
abline(v = real_ty, col="red", lwd=5)
```



```
print(sum(yrep1_data < real_ty)/nrow(yrep1))
```

```
## [1] 0.004
```

```
yrep2 <- rstan::extract(fit2)[["log_weight_rep"]]
yrep2_data <- ty(yrep2, d$arsenic < .82)
hist(yrep2_data)
abline(v = real_ty, col="red", lwd=5)
```



```r
print(sum(yrep2_data < real_ty)/nrow(yrep2))
```

```
## [1] 0.272
```

d) Use the `loo` package to get estimates of the expected log pointwise predictive density for each point, $ELPD_i$. Based on $\sum_i ELPD_i$, which model is preferred?

- Answer: The elpd for model 1 is -1967.8 and the elpd for model 2 is -1952.4. We could find the ELPD for model 2 is 15.4 more than model 1. So we would prefer model 2.

```r
loglik1<- as.matrix(fit1, pars = "log_lik")
loo1 <- loo(loglik1)
```

```
## Warning: Relative effective sample sizes ('r_eff' argument) not specified.
## For models fit with MCMC, the reported PSIS effective sample sizes and
## MCSE estimates will be over-optimistic.
```

```r
print(loo1)
```

```
##
## Computed from 1000 by 3020 log-likelihood matrix
##
##           Estimate   SE
## elpd_loo   -1968.2 16.0
## p_loo          4.6  0.3
## looic       3936.4 31.9
## ------
## 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.
```

```
loglik2<- as.matrix(fit2, pars = "log_lik")
loo2 <- loo(loglik2)
```

```
## Warning: Relative effective sample sizes ('r_eff' argument) not specified.
## For models fit with MCMC, the reported PSIS effective sample sizes and
## MCSE estimates will be over-optimistic.
```

```
print(loo2)
```

```
##
## Computed from 1000 by 3020 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -1952.3 16.4
## p_loo         4.0  0.1
## looic      3904.7 32.8
## ------
## 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.
```

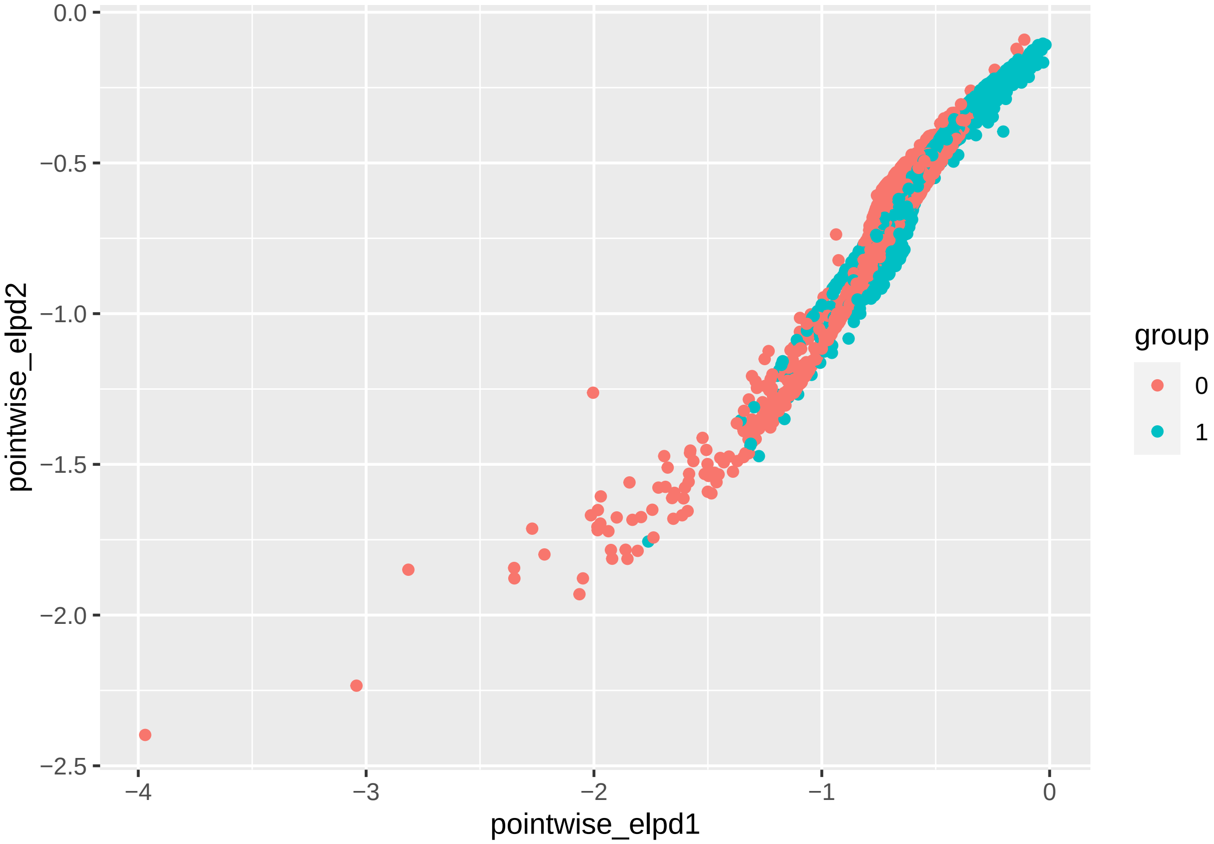
```
loo_compare(loo1, loo2)
```

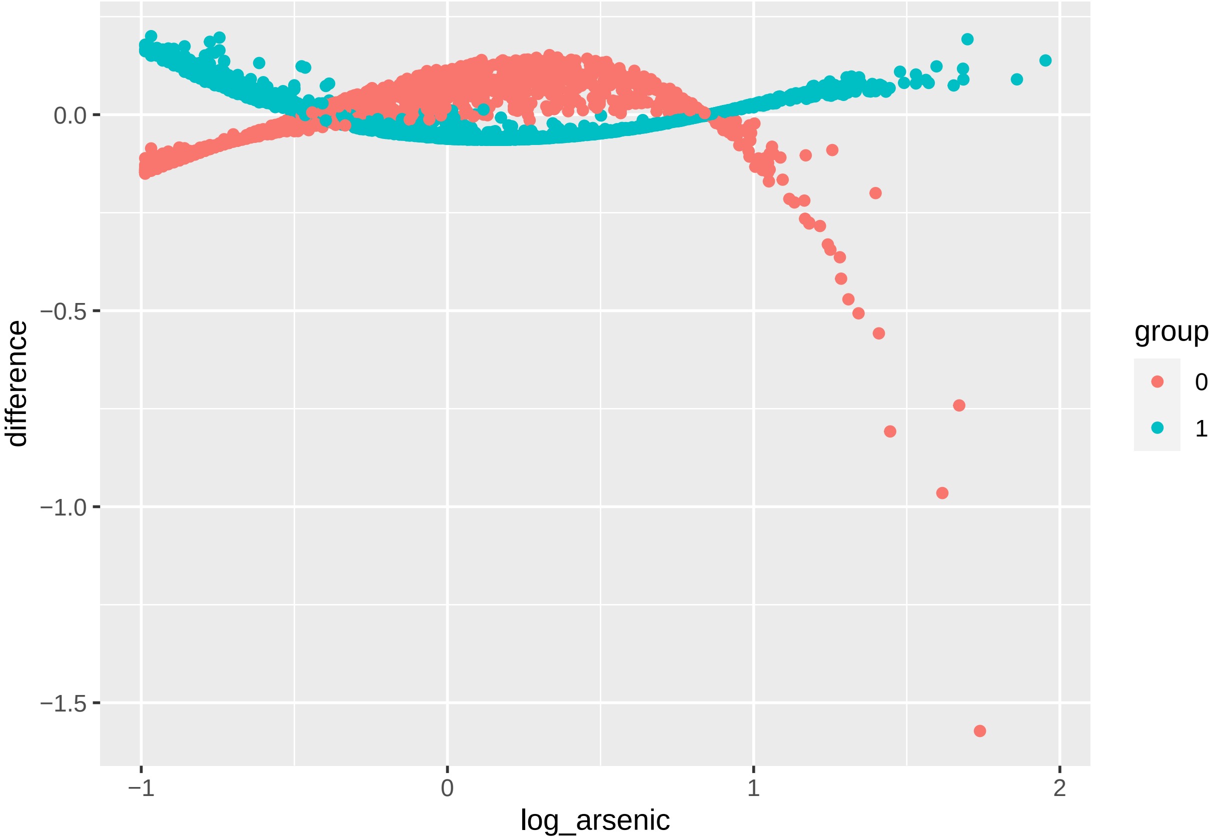
```
##        elpd_diff se_diff
## model2   0.0       0.0
## model1 -15.9       4.4
```

e) Create a scatter plot of the $ELPD_i$'s for Model 2 versus the $ELPD_i$'s for Model 1. Create another scatter plot of the difference in $ELPD_i$'s between the models versus log arsenic. In both cases, color the dots based on the value of y_i. Interpret both plots.

- Answer

It is a little hard to interpret the first plot. But we could see a trend that the first elpd model are often less than the second model elpd. The second graph is much clearer. We could see for the not switch group. There is a concave down trends as the log(arsenic). Especially when log(arsenic) gets larger where the second model has a much larger elpd than the first model. Then for the switch group there is a concave up trend, but the elpd difference between two models are quite close to 0 all the time.


0.0
−0.5
−1.0
−1.5
−2.0
−2.5
pointwise_elpd2
−4
−3
−2
−1
0
pointwise_elpd1
group
0
1


0.0
−0.5
−1.0
−1.5
difference
−1
0
1
2
log_arsenic
group
0
1

f) Given the outcome in this case is discrete, we can directly interpret the $ELPD_i$s. In particular, what is $\exp(ELPD_i)$?

- Answer

Mathematically, the definition of $\exp(ELPD_i)$ is $p(y_i|y_{-i})$. Which is the expected predictive density of y_i given all Y except y_i.

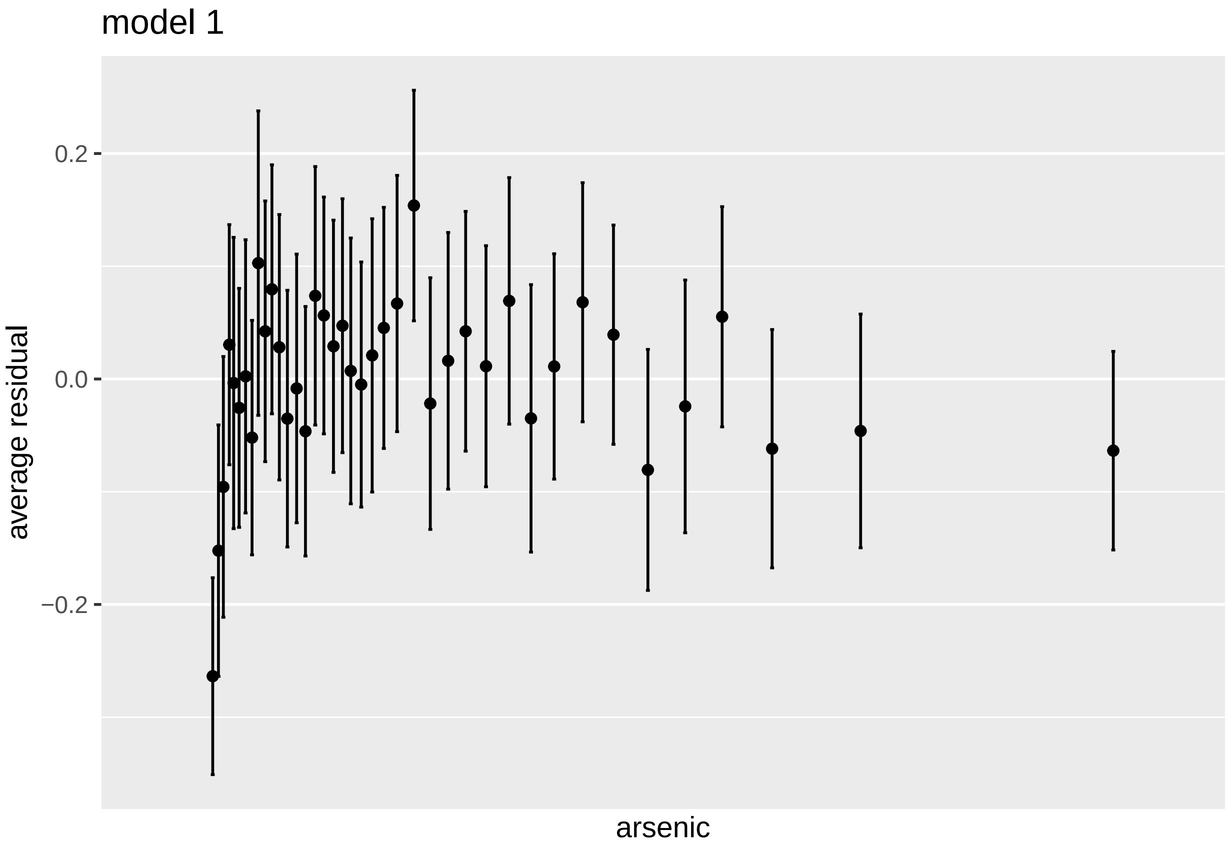
g) For each model recode the $ELPD_i$'s to get $\hat{y}_i = E\left(Y_i|\boldsymbol{y}_{-i}\right)$. Create a binned residual plot, looking at the average residual $y_i - \hat{y}_i$ by arsenic for Model 1 and by log(arsenic) for Model 2. Split the data such that there are 40 bins. On your plots, the average residual should be shown with a dot for each bin. In addition, add in a line to represent +/- 2 standard errors for each bin. Interpret the plots for both models.

- Answer

For both of the redisual plot. We could see the confidence interval gets narrower when the log(arsenic) gets larger. The residual from the second model is quite similar as the first model regardless the log transform. For the first model binned residual plot, we could see the average residual for different bins stay close to 0 most of the time. Although there is a little shift on the first few bins. For the second model binned residual plot. We could see its residual estimation are closer to 0 compare to model 1, also it has a smaller confidence intervals. These conclusions confirm our hypothesis from previous questions. The second model fits better the the first model.

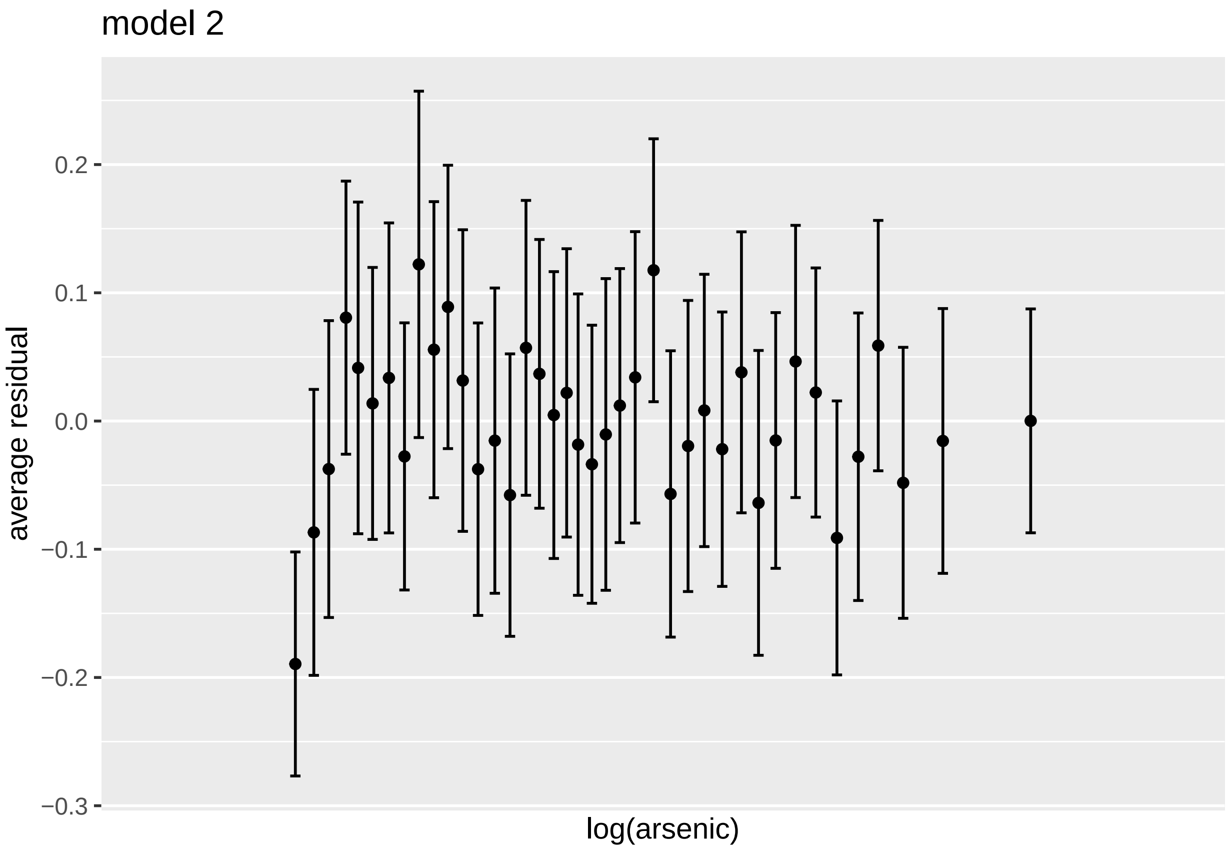
```r
#Create a dataframe we want
residual = data.frame(residual1 = ifelse(y == 1, y - exp(pointwise_elpd1), y - 1 + exp(pointwi
                      residual2 = ifelse(y == 1, y - exp(pointwise_elpd2), y - 1 + exp(pointwi
                      split = as.numeric(cut_number(d$arsenic, 40)),
                      arsenic = d$arsenic,
                      log_arsenic = log(d$arsenic), 40)

se <- function(x){
  return(sd(x)/sqrt(length(x)))
}

model_1 <- residual %>%
  select(residual1, arsenic, split) %>%
  group_by(split) %>%
  summarize(residual = mean(residual1), sd = se(residual1), arsenic = mean(arsenic))

model_2 <- residual %>%
  select(residual2, log_arsenic, split) %>%
  group_by(split) %>%
  summarize(residual = mean(residual2), sd = se(residual2), log_arsenic = mean(log_arsenic))

p <- ggplot(model_1, aes(x=arsenic, y=residual)) +
   geom_point(position=position_dodge()) +
  geom_errorbar(aes(ymin=residual-2*sd, ymax=residual+2*sd)) +
  scale_x_discrete(guide = guide_axis(check.overlap = TRUE)) +
  labs(y = "average residual", x = "arsenic", title = "model 1")
p
```

```
## Warning: Width not defined. Set with `position_dodge(width = ?)`
```



```
p <- ggplot(model_2, aes(x=log_arsenic, y=residual)) +
   geom_point(position=position_dodge()) +
  geom_errorbar(aes(ymin=residual-2*sd, ymax=residual+2*sd))+
  scale_x_discrete(guide = guide_axis(check.overlap = TRUE))+
  labs(y = "average residual", x = "log(arsenic)", title = "model 2")
p
```

```
## Warning: Width not defined. Set with `position_dodge(width = ?)`
```


model 2
average residual
0.2
0.1
0.0
−0.1
−0.2
−0.3
log(arsenic)

2 Maternal mortality

This question relates to estimating the maternal mortality for countries worldwide. A maternal death is defined by the World Health Organization as "the death of a woman while pregnant or within 42 days of termination of pregnancy, irrespective of the duration and site of the pregnancy, from any cause related to or aggravated by the pregnancy or its management but not from accidental or incidental causes". The indicator we are interested in is the (non-AIDS) maternal mortality ratio (MMR) which is defined as the number of non-AIDS maternal deaths divided by the number of live births.

In the data folder of the class repo there are two files relevant to this question. `mmr_data` contains information on, for a range of countries over a range of years:

- Observations of the proportion of non-AIDS deaths that are maternal (PM^{NA})
- Data source, most commonly from Vital Registration systems (VR)
- The Gross Domestic Product (GDP)
- The General Fertility Rate (GFR)
- The average number of skilled attendants at birth (SAB)
- The geographical region of the country
- The total number of women, births, deaths to women of reproductive age (WRA), and the estimated proportion of all WRA deaths that are due to HIV/AIDS

The `mmr_data` file will be used for fitting. Note that data on PM^{NA} is not available for every country.

The `mmr_pred` file contains information on GDP, GFR, SAB, total number of births, deaths and women, and proportion of deaths that are due to HIV/AIDS, for every country at different time points (every five years from mid 1985 to mid 2015). Information in this file is used for producing estimates of MMR for countries without data, and for producing estimates centered at a particular time point.

Consider the following model

$$
\begin{aligned}
y_i|\eta_{c[i]}^{\text{country}}, \eta_{r[i]}^{\text{region}} &\sim N\left(\beta_0 + \eta_{c[i]}^{\text{country}} + \eta_{r[i]}^{\text{region}} + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \beta_3 x_{i,3}, \sigma_y^2\right) \\
\eta_c^{\text{country}} &\sim N\left(0, \left(\sigma_\eta^{\text{country}}\right)^2\right), \text{ for } c = 1, 2, \dots, C \\
\eta_r^{\text{region}} &\sim N\left(0, \left(\sigma_\eta^{\text{region}}\right)^2\right), \text{ for } r = 1, 2, \dots, R
\end{aligned}
$$

where

- y_i is the ith observed $\log PM^{NA}$ in country $c[i]$ in region $r[i]$
- C is total number of countries and R is total number of regions
- $x_{i,1}$ is log(GDP)
- $x_{i,2}$ is log(GFR)
- $x_{i,3}$ is SAB

a) Turn this model into a Bayesian model by specifying appropriate prior distributions for the hyper-parameters and fit the Bayesian model in Stan. Report the full model specification as well as providing the Stan model code.

- Answer

Since the country name has different specifications on pred and raw datasets. I choose to use the iso as their country specification. The data specification could be view below and the model code are included in the folder.

```
#Read and factorize the data
q2_data <- read_csv("./mmr_data.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   iso = col_character(),
##   Country = col_character(),
##   mid.date = col_double(),
##   source.detail = col_character(),
##   data.type = col_character(),
##   GDP = col_double(),
##   GFR = col_double(),
##   SAB = col_double(),
##   Births = col_double(),
##   Women = col_double(),
##   Deaths = col_double(),
##   PM_na = col_double(),
##   region = col_character(),
##   prop.AIDS = col_double()
## )
```

```
q2_validate <- read_csv("./mmr_pred.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   iso = col_character(),
##   Country = col_character(),
##   mid.date = col_double(),
##   GDP = col_double(),
##   GFR = col_double(),
##   SAB = col_double(),
##   Births = col_double(),
##   Women = col_double(),
##   Deaths = col_double(),
##   region = col_character(),
##   prop.AIDS = col_double()
## )
```

```r
#Extract the data from the training dataset
N = nrow(q2_data)
C = length(unique(q2_validate$iso))
R = length(unique(q2_validate$region))

y = log(q2_data$PM_na)
x1 = log(q2_data$GDP)
x2 = log(q2_data$GFR)
x3 = q2_data$SAB

#Index the country, region of the training dataset using prediction dataset.
q2_validate$Cindex = as.numeric(as.factor(q2_validate$iso))
q2_validate$Rindex = as.numeric(as.factor(q2_validate$region))

#Get the matching dataframe
q2_partial = unique(q2_validate[,c(1,10,12,13)])

#Get the matched index
q2_data = left_join(q2_data, q2_partial, by=c("iso", "region"))

country = q2_data$Cindex
region = q2_data$Rindex

data = list(N = N,
            C = C,
            R = R,
            y = y,
            x1 = x1,
            x2 = x2,
            x3 = x3,
            country = country,
            region = region)

q2_fit = stan(file = "./q2a_stan.stan", data = data, iter = 500)
```

```
## Trying to compile a simple C file
```

```
## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and mediar
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess
```

```
## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and ta
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#tail-ess
```

```r
summary(q2_fit)$summary[c("beta_0", "beta_1", "beta_2", "beta_3"),]
```

Hint: I would recommend indexing countries and regions, and calculating C and R based on the full set of countries contained in `mmr_pred`, rather than the subset contained in `mmr_data`. This will mean you will automatically get estimates for η for every country and region, even the missing ones, which will help later on.

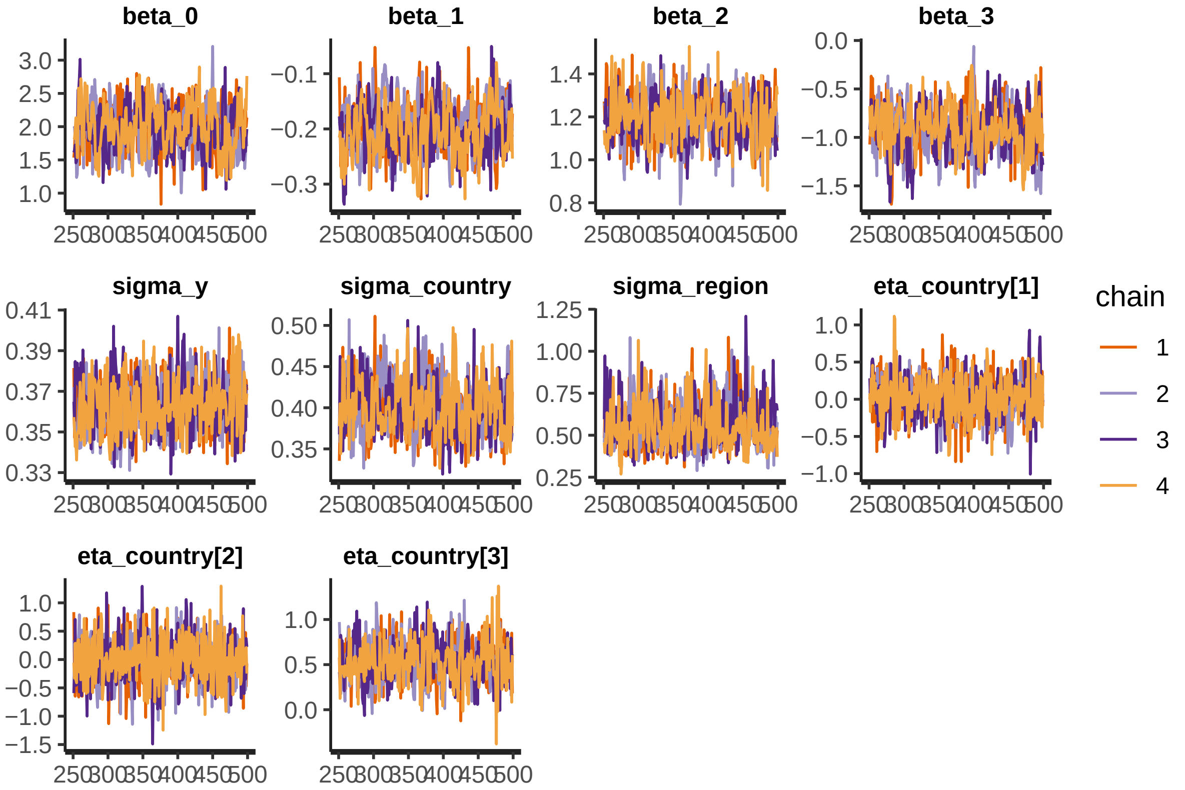
b) Check the trace plots and effective sample size to check convergence and mixing. Summarize your findings using a few example trace plots and effective sample sizes.

- Answer

From the trace plot, we suspect sigma region is not mixing very well. However, when we check the effective sample size of sigma_region. It is as goog as the other parameters from the reports. So our simulation should be good. To include more details. Sigma y has more effective samples. The other parameters are roughly the same, which is about (250 ~ 400). Compare to 2000 are good enough.

```r
traceplot(q2_fit)
```

```
## 'pars' not specified. Showing first 10 parameters by default.
```



```r
summary(q2_fit)$summary[c(1:7), "n_eff"] %>% t() %>% as.data.frame() %>% gt()
```

beta_0	beta_1	beta_2	beta_3	sigma_y	sigma_country	sigma_region
286.9977	348.3327	394.7802	257.6214	745.1813	381.8421	361.2863

c) Plot (samples of the) prior and posterior distributions for $\beta_0, \sigma_y, \sigma_\eta^{\text{country}}$ and $\sigma_\eta^{\text{region}}$. Interpret the estimates of β_1 and β_3.

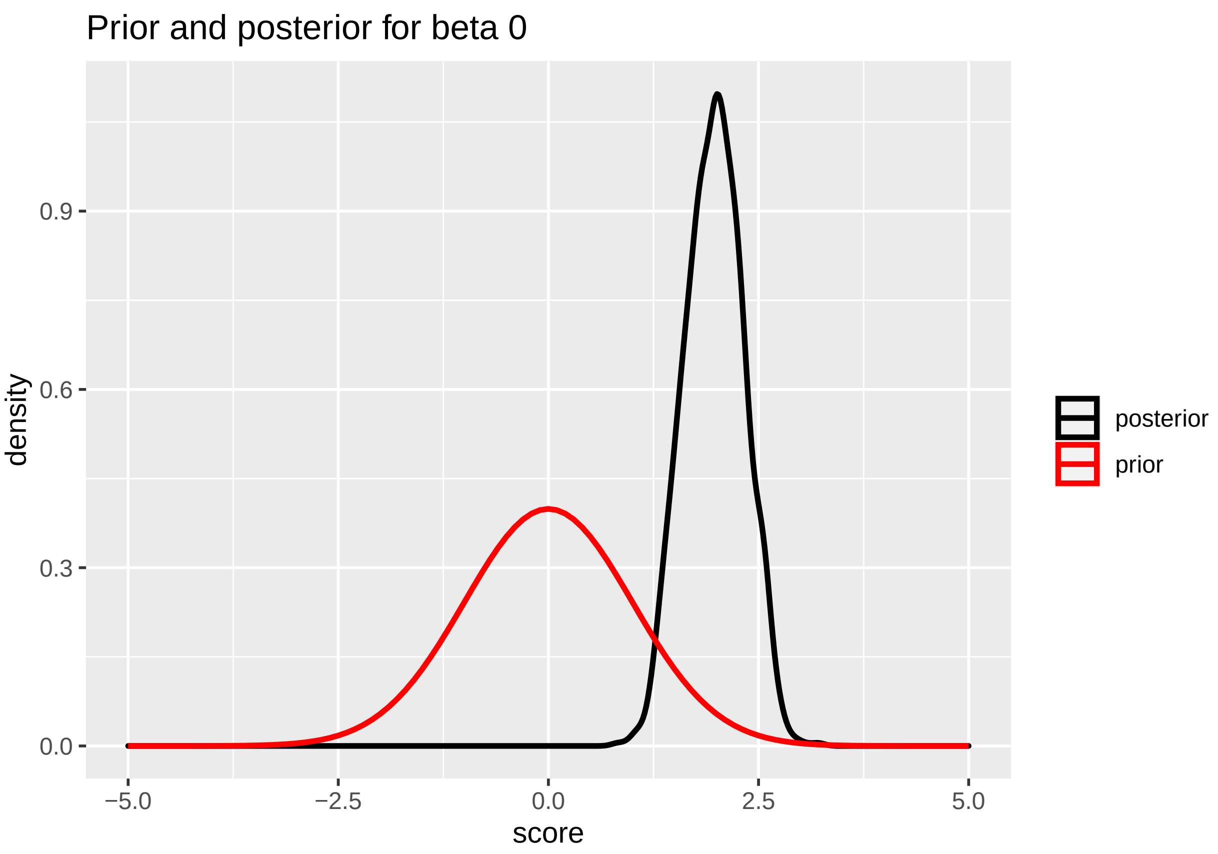
- Answer

For the coefficient for β_1. Intuitively we could think there is a linear negative effect between log(GDP) and log(Proportion of the non AIDS deaths that are maternal). Since both the predicator and the response variable are log transformed. We could interpret the coefficient as the percent increase in the response variable for evert 1% increase in the predicator variable. In our case, β_1 is about -0.20. Which represents for every 1% increase in the GDP, there will be a 0.2% decrease in the PM_NA.

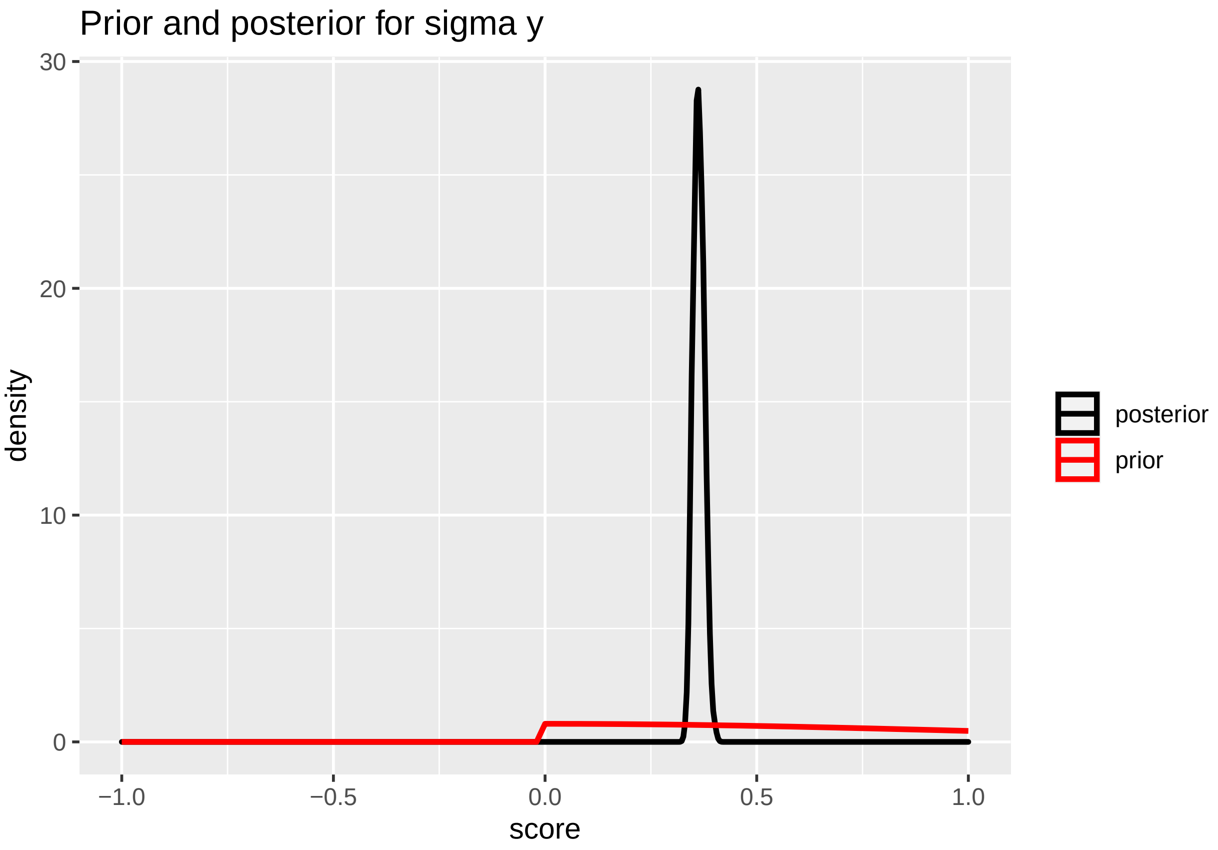
For the coefficient for β_3. Only PM_NA is log transformed. Since only the response variable is log transformed. We could exponentiate the coefficient and subtract one to give the percent increase in the response variable for every one unit increase in the predicator variable. In our case, β_3 is about -0.7597753. Since exp(-0.76) - 1 = -0.53. It represents for every one unit increase in the SAB, there will be 53% decrease in the PM_NA.

Reference: https://data.library.virginia.edu/interpreting-log-transformations-in-a-linear-model/

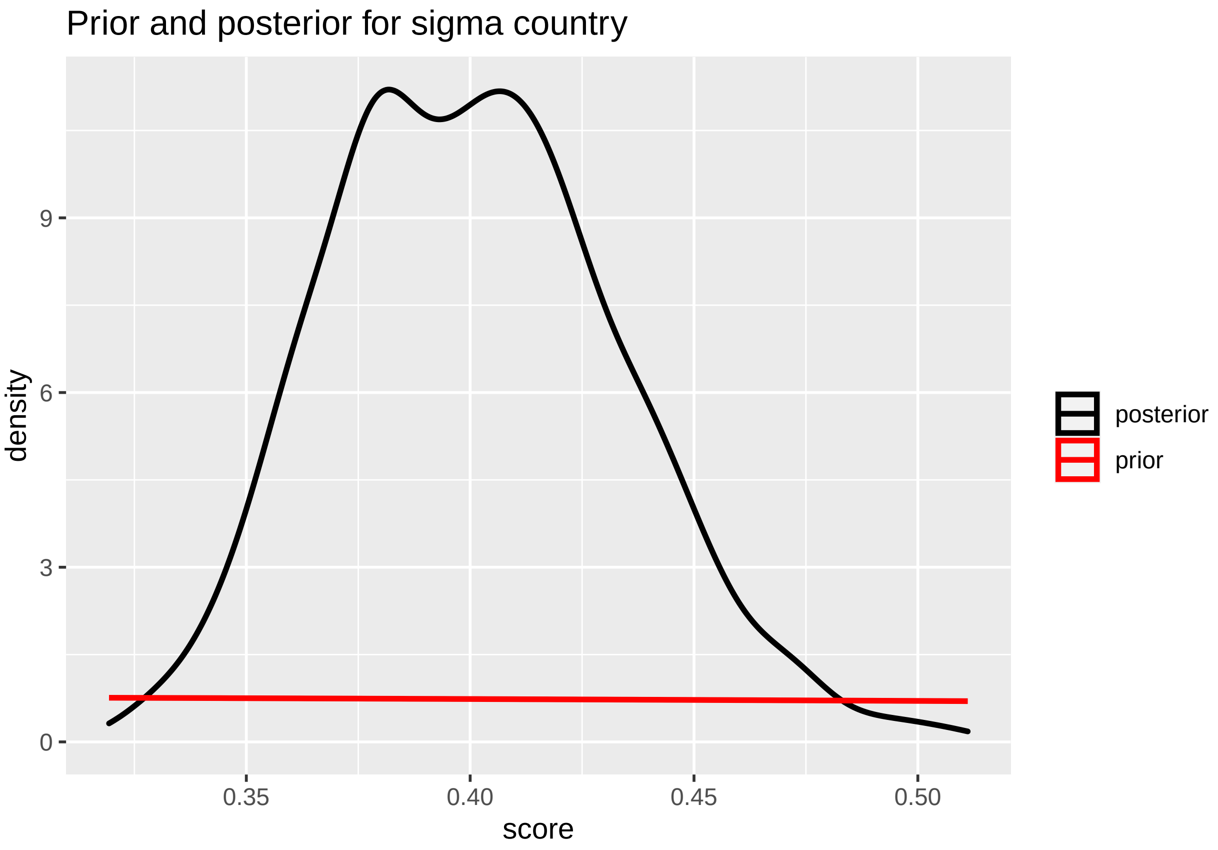
```
samples <- q2_fit %>% gather_draws(beta_0)
samples %>%
  filter(.variable == "beta_0") %>%
  ggplot(aes(.value, color = "posterior")) + geom_density(size = 1) +
  xlim(c(-5, 5)) +
  stat_function(fun = dnorm,
        args = list(mean = 0,
                    sd = 1),
        aes(colour = 'prior'), size = 1) +
  scale_color_manual(name = "", values = c("prior" = "red", "posterior" = "black")) +
  ggtitle("Prior and posterior for beta 0") +
  xlab("score")
```



```
samples <- q2_fit %>% gather_draws(sigma_y)
samples %>%
  filter(.variable == "sigma_y") %>%
  ggplot(aes(.value, color = "posterior")) + geom_density(size = 1) +
  xlim(c(-1, 1)) +
  stat_function(fun = dhnorm,
        args = list(sigma = 1),
        aes(colour = 'prior'), size = 1) +
  scale_color_manual(name = "", values = c("prior" = "red", "posterior" = "black")) +
  ggtitle("Prior and posterior for sigma y") +
  xlab("score")
```

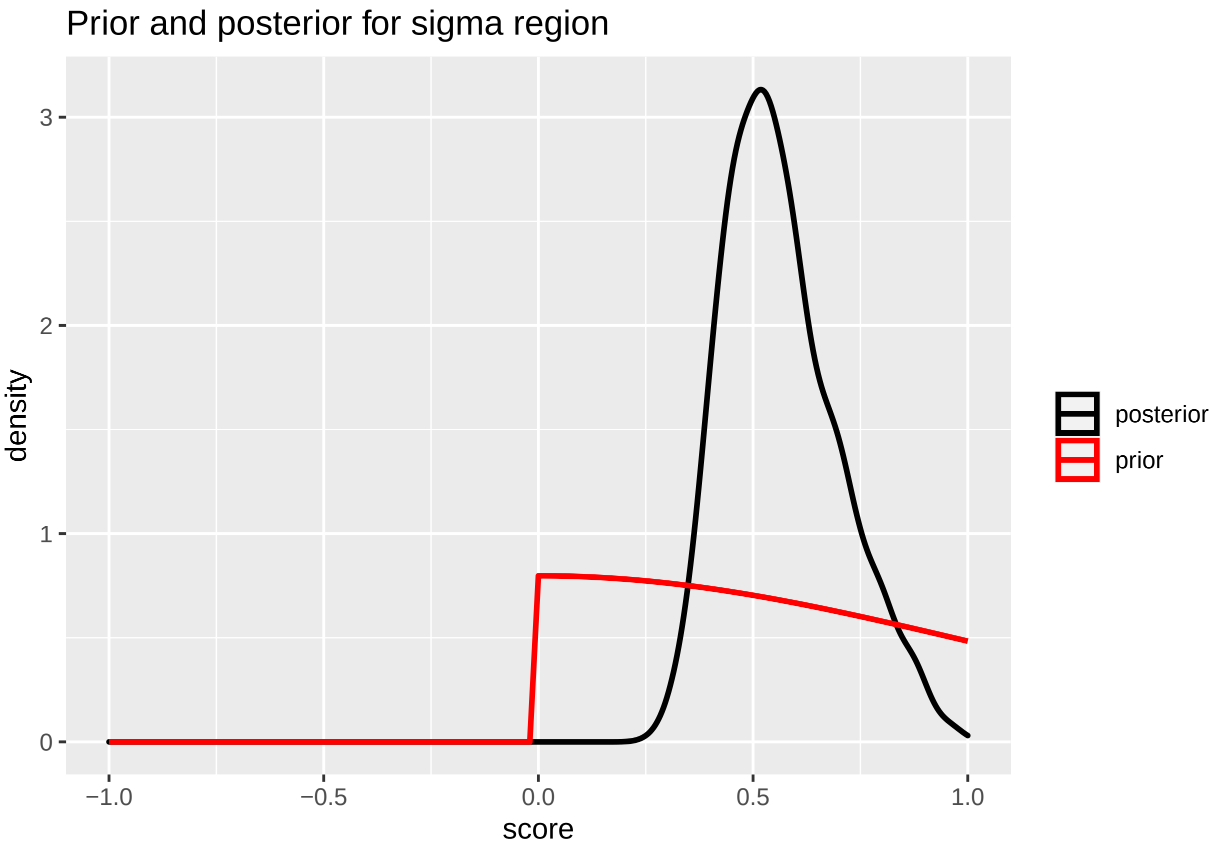


```
samples <- q2_fit %>% gather_draws(sigma_country)
samples %>%
  filter(.variable == "sigma_country") %>%
  ggplot(aes(.value, color = "posterior")) + geom_density(size = 1) +
  stat_function(fun = dhnorm,
        args = list(sigma = 1),
        aes(colour = 'prior'), size = 1) +
  scale_color_manual(name = "", values = c("prior" = "red", "posterior" = "black")) +
  ggtitle("Prior and posterior for sigma country") +
  xlab("score")
```



```
samples <- q2_fit %>% gather_draws(sigma_region)
samples %>%
  filter(.variable == "sigma_region") %>%
  ggplot(aes(.value, color = "posterior")) + geom_density(size = 1) +
  xlim(c(-1, 1)) +
  stat_function(fun = dhnorm,
        args = list(sigma = 1),
        aes(colour = 'prior'), size = 1) +
  scale_color_manual(name = "", values = c("prior" = "red", "posterior" = "black")) +
  ggtitle("Prior and posterior for sigma region") +
  xlab("score")
```

```
## Warning: Removed 7 rows containing non-finite values (stat_density).
```



```r
summary(q2_fit)$summary[c(2,4),]
```

```
##              mean     se_mean         sd       2.5%        25%        50%
## beta_1 -0.2007466 0.002504085 0.04673542 -0.2949248 -0.2327927 -0.2010592
## beta_3 -0.9315596 0.015107884 0.24249046 -1.4082077 -1.0840250 -0.9320836
##               75%      97.5%    n_eff     Rhat
## beta_1 -0.1697588 -0.1078821 348.3327 1.005359
## beta_3 -0.7698877 -0.4334424 257.6214 1.020910
```

d) Use the MCMC samples to construct 95% credible intervals for the PM^{NA} for 5-year periods from 1985.5 to 2015.5 for one country with data and one country without any observed PM^{NA} values. Provide point estimates and CIs in a table and a nice plot. Add the observed data to the plot as well (for the country that has it).

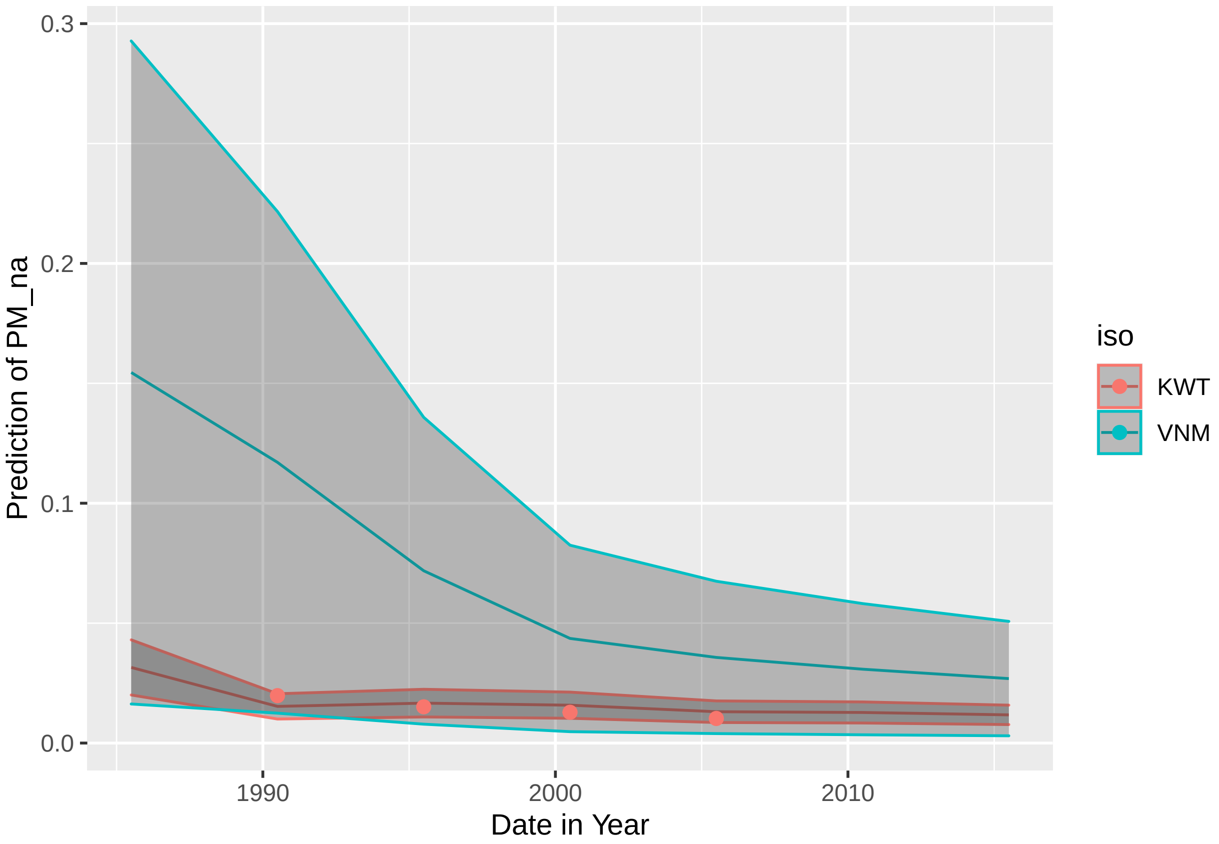
- Answer: We choose "VNM" Vietnem to be the country that is not in the dataset. Its country index is 175 and region index is 11. We also choose "KWT" Kuwait to be the country that in the dataset. Its country index is 90 and region index is 16. Then we retrive the raw data from the prediction dataset and plot it on the graph, which has been shown below. We also attached the table which include the 95% CI and the point estimates. The plot aligns with our intuition. The condifence interval for unobserved data is large and for the observed data is small.

	1985	1990	1995	2000	2005	2010	2015
VNM	0.155	0.118	0.072	0.044	0.036	0.031	0.027
Lower CI	0.023	0.018	0.011	0.007	0.006	0.005	0.004

	1985	1990	1995	2000	2005	2010	2015
Upper CI	0.287	0.217	0.132	0.080	0.066	0.057	0.049

	1985	1990	1995	2000	2005	2010	2015
KWT	0.032	0.015	0.017	0.016	0.013	0.013	0.012
observed_KWT		0.0197	0.0151	0.0128	0.0102		
Lower CI	0.021	0.010	0.011	0.011	0.009	0.009	0.008
Upper CI	0.043	0.020	0.022	0.021	0.017	0.017	0.016

```r
parameters = rstan::extract(q2_fit)
beta0 = parameters$beta_0
beta1 = parameters$beta_1
beta2 = parameters$beta_2
beta3 = parameters$beta_3

eta_ckmt = parameters$eta_country[,90]
eta_rkmt = parameters$eta_region[,16]

eta_cvnm = parameters$eta_country[,175]
eta_rvnm = parameters$eta_region[,11]

kwt = q2_validate[q2_validate$iso == "KWT",]
kwt$pred = 0; kwt$sd = 0
vnm = q2_validate[q2_validate$iso == "VNM",]
vnm$pred = 0; vnm$sd = 0
obs_kwt = q2_data[q2_data$iso == "KWT", ]

for(j in c(1:nrow(kwt))){
  pred = exp(beta0 + eta_ckmt + eta_rkmt + beta1 * log(kwt$GDP[j]) + beta2*log(kwt$GFR[j]) + be
  kwt$pred[j] = mean(pred)
  kwt$sd[j] = sd(pred)
}

for(i in c(1:nrow(vnm))){
  pred = exp(beta0 + eta_cvnm + eta_rvnm + beta1 * log(vnm$GDP[i]) + beta2*log(vnm$GFR[i]) + be
  vnm$pred[i] = mean(pred)
  vnm$sd[i] = sd(pred)
}

q2d_data = rbind(kwt, vnm)

ggplot(data = q2d_data, aes(x=mid.date, y=pred, color = iso)) + geom_line() +
  geom_ribbon(aes(ymin=pred - 2*sd,ymax=pred+2*sd),alpha=0.3) +
  geom_point(data = obs_kwt, aes(x=mid.date, y=PM_na), size=2) +
  xlab("Date in Year") + ylab("Prediction of PM_na")
```



e) The non-AIDS MMR is given by

$$
\begin{aligned}
MMR^{NA} &= \frac{\#\text{ Non-AIDS maternal deaths}}{\#\text{ Births}} \\
&= \frac{\#\text{ Non-AIDS maternal deaths}}{\#\text{ Non-AIDS deaths}} \cdot \frac{\#\text{ Non-AIDS deaths}}{\#\text{ Births}} \\
&= PM^{NA} \cdot \frac{\#\text{ Deaths } * (1 - \text{ prop AIDS})}{\text{Births}}
\end{aligned}
$$

where deaths and births are to all women of reproductive age in the country-period of interest. Use this formula, your answers from d) and the data in `mmr_pred` to obtain point estimates and CIs for the non-AIDS MMR for the two countries you chose in (d) in the year 2010.5.

- Answer We follow the similar process to obtain the point estimates and confidence intervals for KWT and VNM. The confidence intervals could be viewed as below

	Point Estimate	Upper CI	Lower CI
KWT	7.069312e-05	9.432219e-05	4.706405e-05
VNM	0.0006301146	0.0011581717	0.0001020575

```
pmna1 = exp(beta0 + eta_ckmt + eta_rkmt + beta1 * log(kwt$GDP[6]) +
              beta2*log(kwt$GFR[6]) + beta3*kwt$SAB[6])
```

```
pred = pmna1 * (kwt$Deaths[6] * (1-kwt$prop.AIDS[6]))/kwt$Births[6]
kwt_mmr_mean = mean(pred)
kwt_mmr_sd = sd(pred)

pmna2 = exp(beta0 + eta_cvnm + eta_rvnm + beta1 * log(vnm$GDP[6]) +
             beta2*log(vnm$GFR[6]) + beta3*vnm$SAB[6])
pred = pmna2 * (vnm$Deaths[6] * (1-vnm$prop.AIDS[6]))/vnm$Births[6]
vnm_mmr_mean = mean(pred)
vnm_mmr_sd = sd(pred)

print(c(kwt_mmr_mean - 2*kwt_mmr_sd, kwt_mmr_mean + 2*kwt_mmr_sd))
```

```
## [1] 4.618732e-05 9.478057e-05
```

```
print(c(vnm_mmr_mean - 2*vnm_mmr_sd, vnm_mmr_mean + 2*vnm_mmr_sd))
```

```
## [1] 7.061399e-05 1.191031e-03
```

f) In the model used so far, we assume that error variance σ_y^2 is the same for all observations but this is probably not a very realistic assumption. Let's explore if the model fit changes if we would estimate two variance parameters: one for VR data (denoted by σ^2_{VR}) and one for non-VR data (denoted by $\sigma^2_{\text{non-VR}}$). Write out the model specification for this extended model, give the Stan model code, and fit the model. Show priors and posteriors for σ_{VR} and $\sigma_{\text{non-VR}}$ and construct a plot with data for a country with VR data, with point estimates and CIs from the models with and without equal variance.

- Answer First we create the vr data with 1(non vr) and 2(vr data). Then we could retrieve the following prior posterior plot from the model. We could see the sigma for VR data is larger than the sigma for non VR data. We then select the same country KWT in 2(d), which is VR data. Then the point estimates and confidence intervals with and without equal variance could be viewed as below. I also plot it in a graph. Where we could see there is not huge difference.

	1985	1990	1995	2000	2005	2010	2015
(equal)KWT	0.032	0.015	0.017	0.016	0.013	0.013	0.012
(equal)Lower CI	0.021	0.010	0.011	0.011	0.009	0.009	0.008
(equal)Upper CI	0.043	0.020	0.022	0.021	0.017	0.017	0.016
(unequal)KWT	0.032	0.015	0.017	0.016	0.013	0.013	0.012
(unequal)Lower CI	0.020	0.010	0.011	0.010	0.009	0.008	0.008
(unequal)Upper CI	0.044	0.021	0.023	0.022	0.018	0.017	0.016

```
vr_data = as.numeric(q2_data$data.type == "VR") + 1
data_2f = list(N = N,
            C = C,
            R = R,
            y = y,
            x1 = x1,
            x2 = x2,
            x3 = x3,
            country = country,
            region = region,
            VR = vr_data)

fit_2f = stan(file = "./q2f_stan.stan", data = data_2f)
```

```
## Trying to compile a simple C file
```

```
## 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" -DNDI
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHea
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEig
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEig
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/C
## namespace Eigen {
## ^
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/C
## namespace Eigen {
##                ^
##                ;
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHea
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEig
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/Core
## #include <complex>
##          ^~~~~~~~~
## 3 errors generated.
## make: *** [foo.o] Error 1
##
## SAMPLING FOR MODEL 'q2f_stan' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.000164 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 1.64 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
```

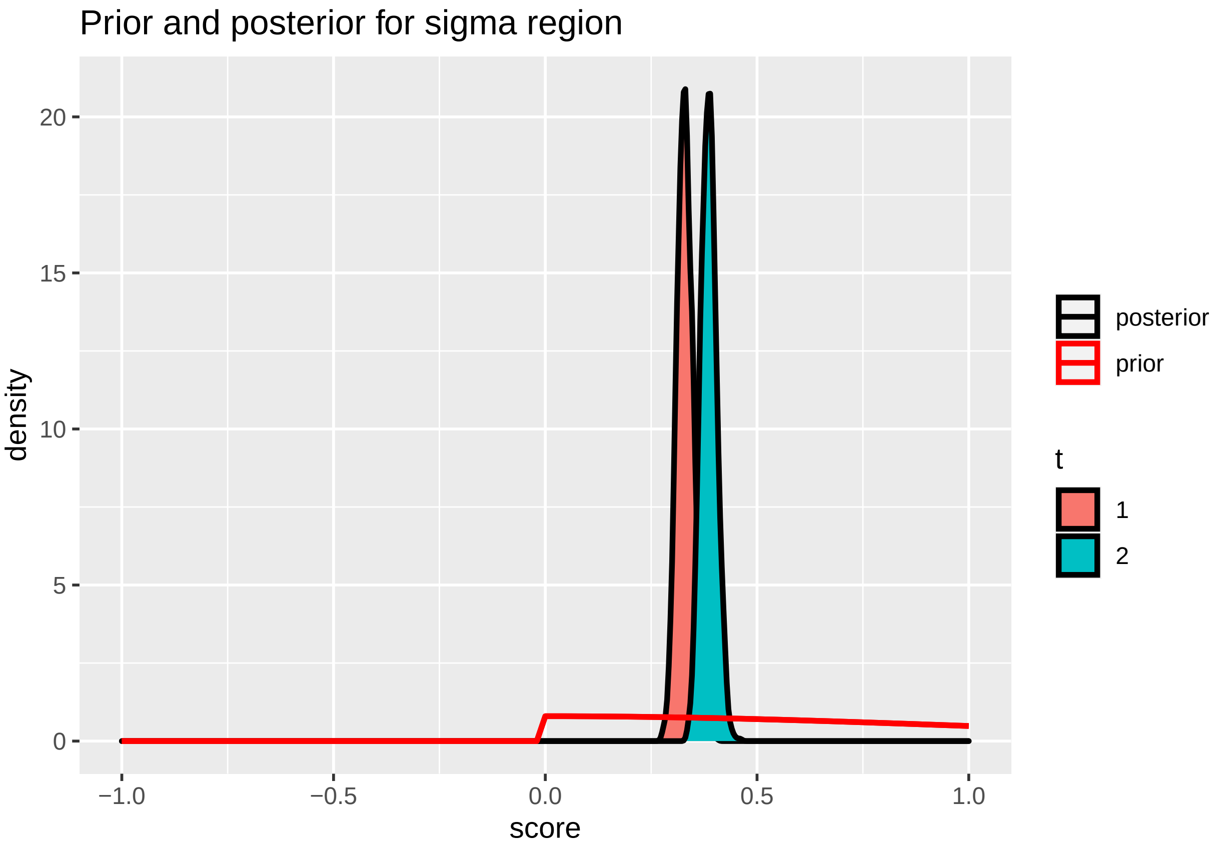
```
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 16.6376 seconds (Warm-up)
## Chain 1:                19.5633 seconds (Sampling)
## Chain 1:                36.2009 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'q2f_stan' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 9.5e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.95 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 17.2003 seconds (Warm-up)
## Chain 2:                19.3089 seconds (Sampling)
## Chain 2:                36.5092 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'q2f_stan' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 9.5e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.95 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 16.6103 seconds (Warm-up)
## Chain 3:                13.2207 seconds (Sampling)
## Chain 3:                29.831 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'q2f_stan' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 9.1e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.91 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 16.0786 seconds (Warm-up)
## Chain 4:                9.8407 seconds (Sampling)
## Chain 4:                25.9193 seconds (Total)
## Chain 4:
```

```
## Warning: There were 8 transitions after warmup that exceeded the maximum treedepth. Increase
## http://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

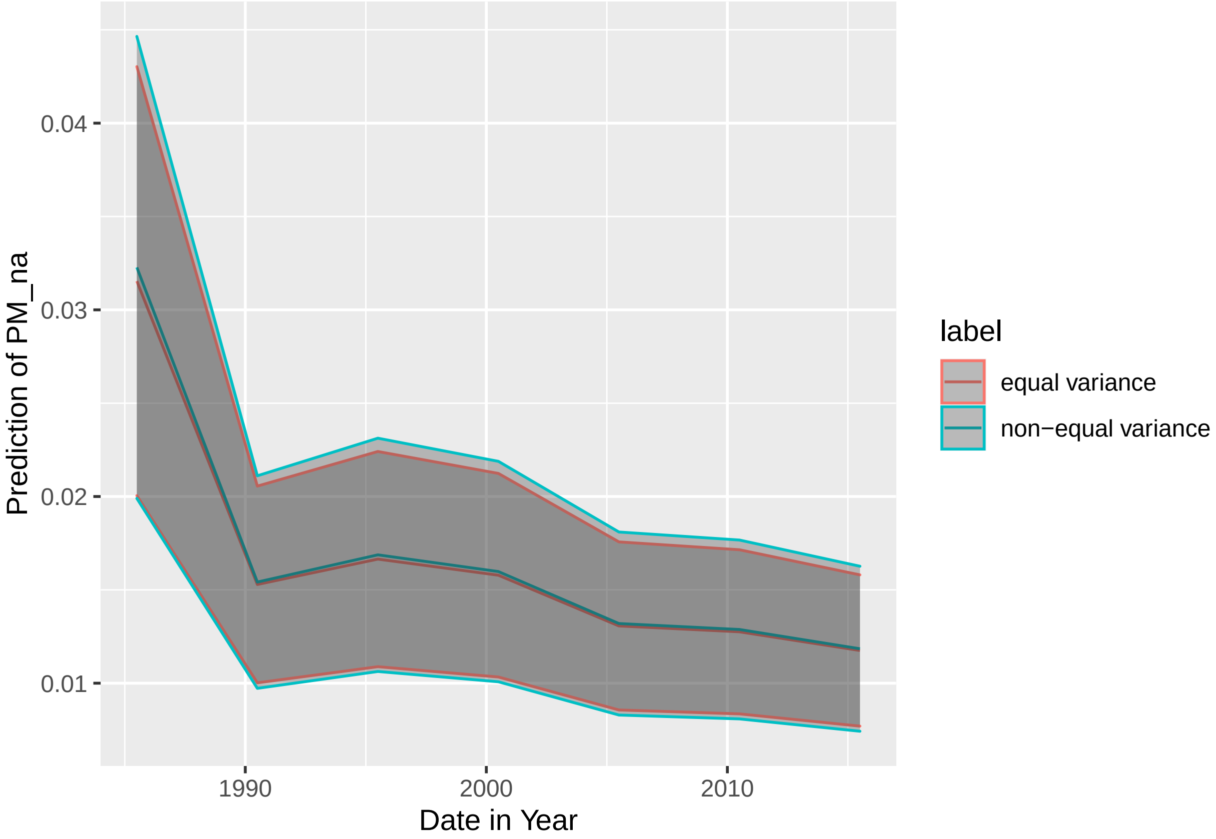
```r
samples <- fit_2f %>% gather_draws(sigma_y[t])
samples$t <- as.factor(samples$t)

samples %>%
  ggplot(aes(.value, fill = t, color = "posterior")) + geom_density(size = 1) +
  xlim(c(-1, 1)) +
  stat_function(fun = dhnorm,
        args = list(sigma = 1),
        aes(colour = 'prior'), size = 1) +
  scale_color_manual(name = "", values = c("prior" = "red", "posterior" = "black")) +
  ggtitle("Prior and posterior for sigma region") +
  xlab("score")
```

```
## Warning: Multiple drawing groups in `geom_function()`. Did you use the correct
## `group`, `colour`, or `fill` aesthetics?
```



```
#Extract the parameters from the model to get the point estimation
parameters = rstan::extract(fit_2f)
beta0 = parameters$beta_0
beta1 = parameters$beta_1
beta2 = parameters$beta_2
beta3 = parameters$beta_3

eta_ckmt = parameters$eta_country[,90]
eta_rkmt = parameters$eta_region[,16]

eta_cvnm = parameters$eta_country[,175]
eta_rvnm = parameters$eta_region[,11]

kwt2 = q2_validate[q2_validate$iso == "KWT",]
kwt2$pred = 0; kwt2$sd = 0; kwt2$label = "non-equal variance"
kwt$label = "equal variance"
```

```
for(j in c(1:nrow(kwt2))){
  pred = exp(beta0 + eta_ckmt + eta_rkmt + beta1 * log(kwt2$GDP[j]) + beta2*log(kwt2$GFR[j]) +
  kwt2$pred[j] = mean(pred)
  kwt2$sd[j] = sd(pred)
}

q2f_data = rbind(kwt, kwt2)
q2f_data$label <- as.factor(q2f_data$label)

ggplot(data = q2f_data, aes(x=mid.date, y=pred, color = label)) + geom_line() +
  geom_ribbon(aes(ymin=pred - 2*sd,ymax=pred+2*sd),alpha=0.3) +
  xlab("Date in Year") + ylab("Prediction of PM_na")
```



3 Research proposal

The final project for this class involves exploring a research question that you are interested in using a dataset of your choice. For the research proposal, I'm interested in finding out about your topic, and seeing some EDA based on your dataset of choice. Please describe

- your research question(s) of interest, and why they are of interest (if there's an obvious literature, feel free to cite a few papers)
- the dataset you plan to use
- your main dependent variable of interest
- your main independent variables of interest (including control variables)

3.1 Exploratory data analysis

As part of your research proposal please undertake some basic EDA to illustrate the characteristics of your dataset, patterns in the raw data, and to present descriptive statistics related to your data and your research question.

There is no set format, but here are a few pointers of things to look at

- **General characteristics of dataset** and **Summary statistics of variables of interest**: for example, how many observations, how were the data collected (is the dataset representative of the population of interest?); you could present a table of summary statistics of main variables, including things like number of observations, mean/median/sd (if a continuous variable), proportions by group, etc…
- **Missing data**: If your dataset does not have any missing observations, then fine to just say this (don't need to do EDA graphs or discuss). If you have missing observations, summarize what is missing, and give a brief discussion about whether or not you think missingness may be a problem (e.g. is there more likely to be missing data for some groups compared to others?)
- **Graphs showing both univariate and bivariate patterns**: likely to be interested in both univariate patterns (e.g., the distribution of continuous variables, proportions for categorical outcomes…) and bivariate patterns (e.g. scatterplots, proportions/boxplots by group, trends over time…).

3.2 What to submit

It is expected that you present and write up your findings in Rmd. You should submit:

- your R Markdown file; and
- the knitted PDF resulting from your R Markdown file.

If your dataset is reasonably small (and publicly available), then it would be great if you could submit that, too.

Please submit files via Quercus.