Week 10: Temporal data

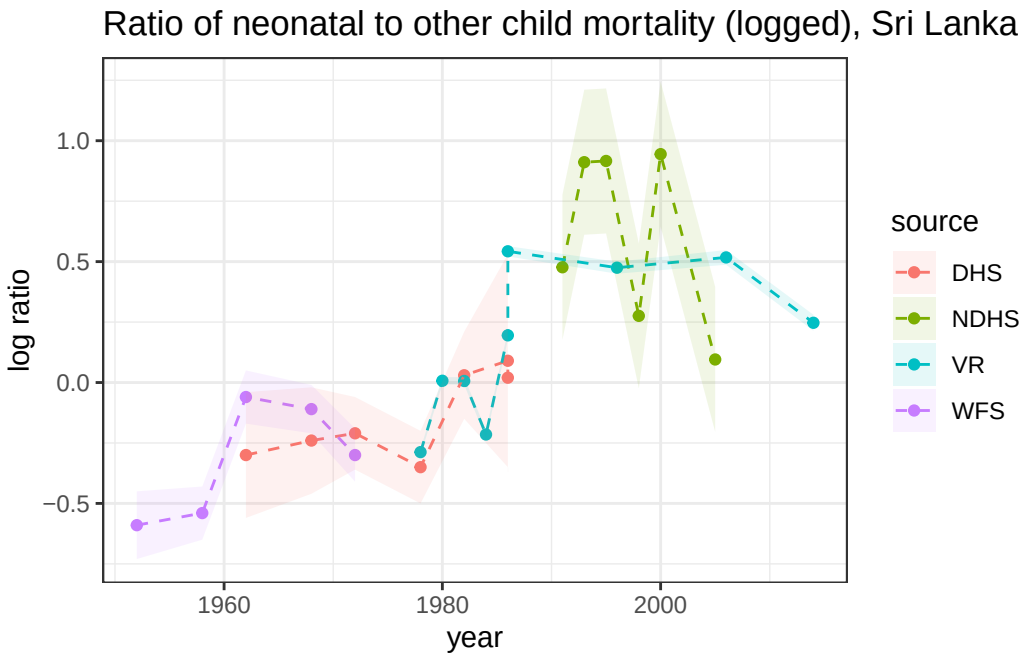
Xuanze Li, 1005854309

25/03/24

Child mortality in Sri Lanka

In this lab you will be fitting a couple of different models to the data about child mortality in Sri Lanka, which was used in the lecture. Here's the data and the plot from the lecture:

```
setwd("/Users/Charlie/Downloads")

library(tidyverse)
library(here)
library(rstan)
library(tidybayes)

lka <- read_csv(here("Downloads/lka.csv"))
ggplot(lka, aes(year, logit_ratio)) +
  geom_point(aes( color = source)) +
  geom_line(aes( color = source), lty = 2) +
  geom_ribbon(aes(ymin = logit_ratio - se,
                  ymax = logit_ratio + se,
                  fill =  source), alpha = 0.1) +
  theme_bw()+
  labs(title = "Ratio of neonatal to other child mortality (logged), Sri Lanka", y = "log
```



Fitting a linear model

Let's firstly fit a linear model in time to these data. Here's the code to do this:

```
observed_years <- lka$year
years <- min(observed_years):max(observed_years)
nyears <- length(years)

stan_data <- list(y = lka$logit_ratio, year_i = observed_years - years[1]+1,
                  T = nyears, years = years, N = length(observed_years),
                  mid_year = mean(years), se = lka$se)

mod <- stan(data = stan_data,
            file = here("Downloads/lka_linear_me.stan"))
```

```
Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
using C compiler: 'Apple clang version 15.0.0 (clang-1500.3.9.4)'
using SDK: 'MacOSX14.4.sdk'
clang -arch arm64 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG   -I"/Libra
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/St
```

```
In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/R
In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/R
/Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/
#include <cmath>
         ^~~~~~~
1 error generated.
make: *** [foo.o] Error 1

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 1.4e-05 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.14 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: 0.019 seconds (Warm-up)
Chain 1:                0.015 seconds (Sampling)
Chain 1:                0.034 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 2e-06 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.02 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: 0.017 seconds (Warm-up)
Chain 2:                0.013 seconds (Sampling)
Chain 2:                0.03 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
Chain 3:
Chain 3: Gradient evaluation took 2e-06 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.02 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: 0.018 seconds (Warm-up)
Chain 3:                0.014 seconds (Sampling)
Chain 3:                0.032 seconds (Total)
Chain 3:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
Chain 4:
Chain 4: Gradient evaluation took 2e-06 seconds
```

```
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.02 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: 0.017 seconds (Warm-up)
Chain 4:                0.014 seconds (Sampling)
Chain 4:                0.031 seconds (Total)
Chain 4:
```

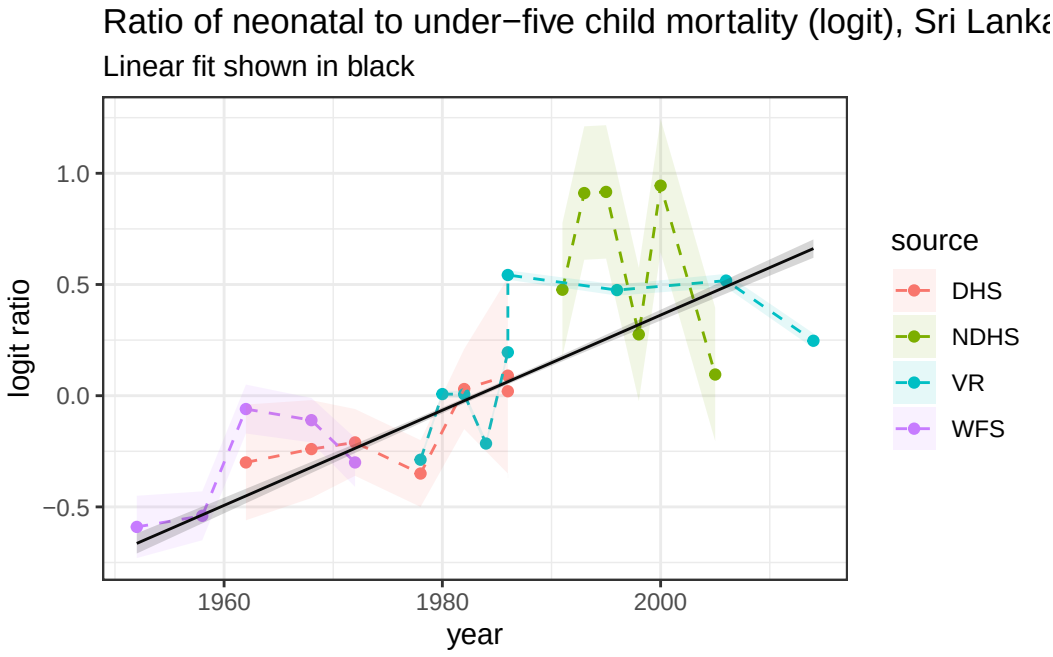
Extract the results:

```
res <- mod %>%
  gather_draws(mu[t]) %>%
  median_qi() %>%
  mutate(year = years[t])
```

Plot the results:

```
ggplot(lka, aes(year, logit_ratio)) +
  geom_point(aes( color = source)) +
  geom_line(aes( color = source), lty = 2) +
  geom_ribbon(aes(ymin = logit_ratio - se,
                  ymax = logit_ratio + se,
                  fill =  source), alpha = 0.1) +
  theme_bw()+
  geom_line(data = res, aes(year, .value)) +
  geom_ribbon(data = res, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.2)+
  theme_bw()+
  labs(title = "Ratio of neonatal to under-five child mortality (logit), Sri Lanka",
```

```
       y = "logit ratio", subtitle = "Linear fit shown in black")
```



Question 1

Project the linear model above out to 2022 by adding a `generated quantities` block in Stan (do the projections based on the expected value μ). Plot the resulting projections on a graph similar to that above.

```
stan_data <- list(y = lka$logit_ratio, year_i = observed_years - years[1]+1,
                  T = nyears, years = years, N = length(observed_years),
                  mid_year = mean(years), se = lka$se, P = 8)
model1 <- stan(data = stan_data,
               file = "model1.stan")
```

```
Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
using C compiler: 'Apple clang version 15.0.0 (clang-1500.3.9.4)'
using SDK: 'MacOSX14.4.sdk'
clang -arch arm64 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG   -I"/Libra
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/St
In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/Rc
```

```
In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/R
/Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/
#include <cmath>
         ^~~~~~~
1 error generated.
make: *** [foo.o] Error 1

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 1.3e-05 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.13 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: 0.018 seconds (Warm-up)
Chain 1:                0.014 seconds (Sampling)
Chain 1:                0.032 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 2e-06 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.02 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: 0.017 seconds (Warm-up)
Chain 2:                0.015 seconds (Sampling)
Chain 2:                0.032 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
Chain 3:
Chain 3: Gradient evaluation took 2e-06 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.02 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: 0.017 seconds (Warm-up)
Chain 3:                0.014 seconds (Sampling)
Chain 3:                0.031 seconds (Total)
Chain 3:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
Chain 4:
Chain 4: Gradient evaluation took 1e-06 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.01 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: 0.018 seconds (Warm-up)
Chain 4:                0.017 seconds (Sampling)
Chain 4:                0.035 seconds (Total)
Chain 4:
```

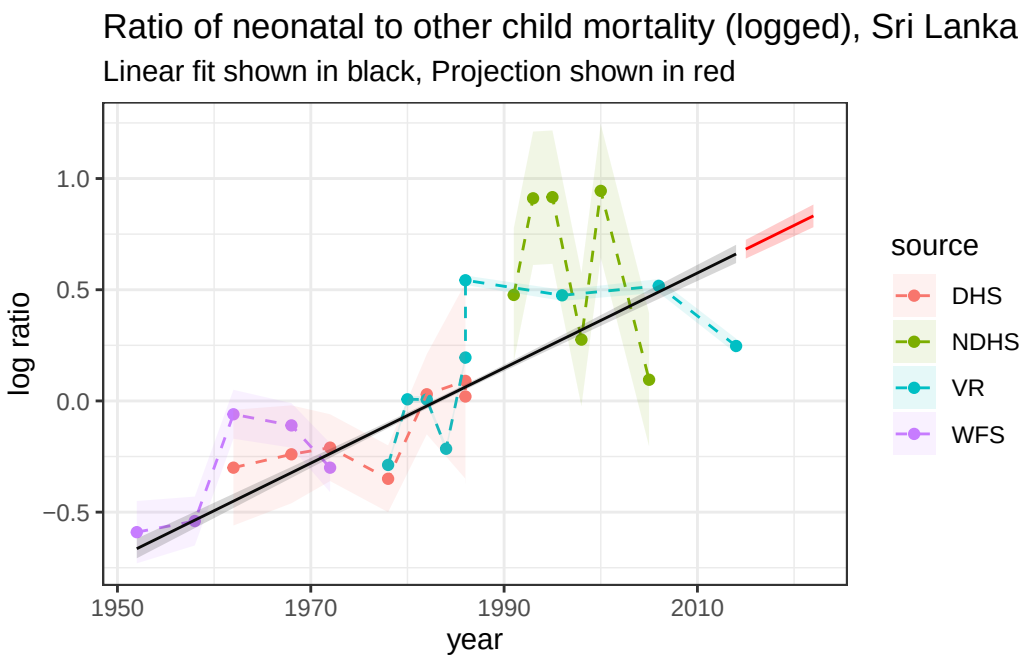
Extract the results:

```
res_model1 <- model1 %>%
  gather_draws(mu[t])%>%
  median_qi() %>%
  mutate(year = years[t])

res_p1 <-  model1 %>%
  gather_draws(mu_p[p])%>%
  median_qi() %>%
  mutate(year = years[nyears]+p)
```

Plot the results:

```
ggplot(lka, aes(year, logit_ratio)) +
  geom_point(aes( color = source)) +
  geom_line(aes( color = source), lty = 2) +
  geom_ribbon(aes(ymin = logit_ratio - se,
                  ymax = logit_ratio + se,
                  fill =  source), alpha = 0.1) +
  theme_bw()+
```

```
geom_line(data = res_model1, aes(year, .value)) +
geom_ribbon(data = res_model1, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.
geom_line(data = res_p1, aes(year, .value), col = "red") +
geom_ribbon(data = res_p1, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.2, f
theme_bw()+
labs(title = "Ratio of neonatal to other child mortality (logged), Sri Lanka",
     y = "log ratio", subtitle = "Linear fit shown in black, Projection shown in red")
```

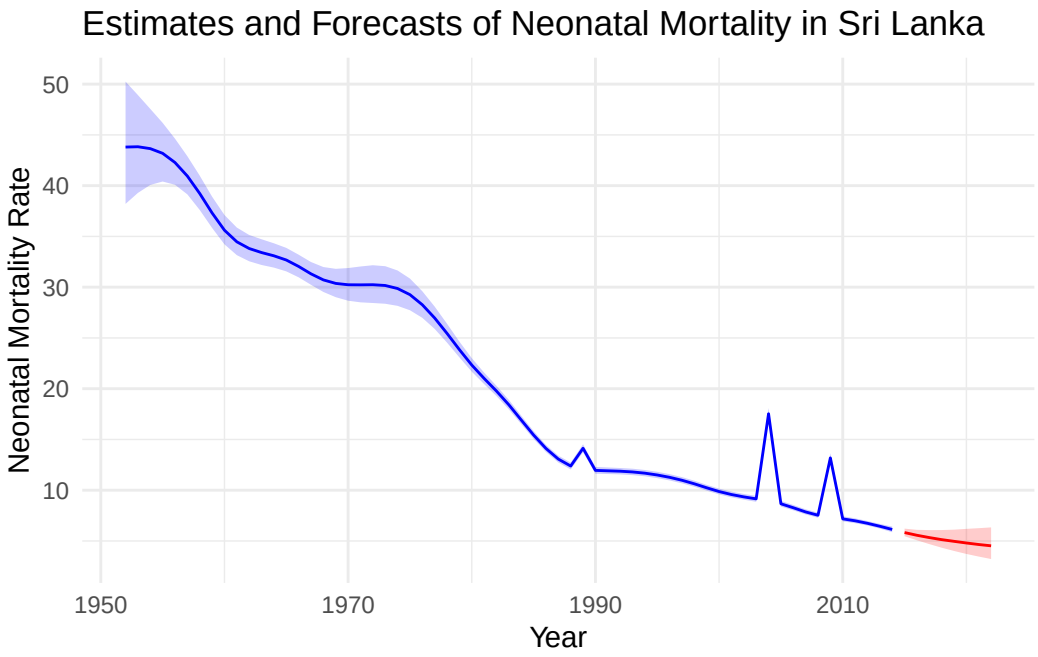


Question 2

The projections above are for the logit of the ratio of neonatal to under-five child mortality. You can download estimates of the under-five child mortality from 1951 to 2022 here: https://childmortality.org/all-cause-mortality/data/estimates?refArea=LKA. Use these data to get estimates and projections of neonatal mortality for Sri Lanka, and plot the results.

```
# Loading the dataset
Lka5_data <- read.csv("LK_data.csv")
# Renaming columns
colnames(Lka5_data)[colnames(Lka5_data) == "Lower bound"] <- "Lower_bound"
colnames(Lka5_data)[colnames(Lka5_data) == "Upper bound"] <- "Upper_bound"
```

```r
Lka5_data$year <- as.numeric(Lka5_data$Year)

# Defining the invlogit function
logit_transform <- function(p) {
  exp(p) / (1 + exp(p))
}

combined_estimates <- rbind(res_model1 %>% select(value = .value, lower = .lower, upper =
                            res_p1 %>% select(value = .value, lower = .lower, upper = .upp
  mutate(logit_value = logit_transform(value),
         logit_lower = logit_transform(lower),
         logit_upper = logit_transform(upper))

neonatal_mortality_estimates <- merge(Lka5_data, combined_estimates, by = "year")

neonatal_mortality_estimates <- neonatal_mortality_estimates %>%
  mutate(neonatal_estimate = Estimate * logit_value,
         neonatal_lower = Lower.bound * logit_lower,
         neonatal_upper = Upper.bound * logit_upper) %>%
  na.omit()

neonatal_mortality_estimates$year <- as.numeric(as.character(neonatal_mortality_estimates$

ggplot(neonatal_mortality_estimates, aes(x = year)) +
  geom_line(data = subset(neonatal_mortality_estimates, year <= 2014), aes(y = neonatal_es
  geom_ribbon(data = subset(neonatal_mortality_estimates, year <= 2014), aes(ymin = neonat
  labs(title = "Estimates and Forecasts of Neonatal Mortality in Sri Lanka",
       y = "Neonatal Mortality Rate",
       x = "Year") +
  geom_line(data = subset(neonatal_mortality_estimates, year > 2014), aes(y = neonatal_est
  geom_ribbon(data = subset(neonatal_mortality_estimates, year > 2014), aes(ymin = neonata
  theme_minimal()
```



Random walks

Question 3

Code up and estimate a first order random walk model to fit to the Sri Lankan data, taking into account measurement error, and project out to 2022.

```
model2 <- stan(data = stan_data,
               file = "model2.stan")
```

```
Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
using C compiler: 'Apple clang version 15.0.0 (clang-1500.3.9.4)'
using SDK: 'MacOSX14.4.sdk'
clang -arch arm64 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG   -I"/Libra
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/St
In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/Rc
In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/Rc
/Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/
#include <cmath>
         ^~~~~~~
```

```
1 error generated.
make: *** [foo.o] Error 1

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 3.1e-05 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.31 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: 0.098 seconds (Warm-up)
Chain 1:                0.056 seconds (Sampling)
Chain 1:                0.154 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 3e-06 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.03 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: 0.1 seconds (Warm-up)
Chain 2:                0.066 seconds (Sampling)
Chain 2:                0.166 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
Chain 3:
Chain 3: Gradient evaluation took 4e-06 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.04 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: 0.099 seconds (Warm-up)
Chain 3:                0.085 seconds (Sampling)
Chain 3:                0.184 seconds (Total)
Chain 3:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
Chain 4:
Chain 4: Gradient evaluation took 3e-06 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.03 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: 0.106 seconds (Warm-up)
Chain 4:                0.077 seconds (Sampling)
Chain 4:                0.183 seconds (Total)
Chain 4:
```

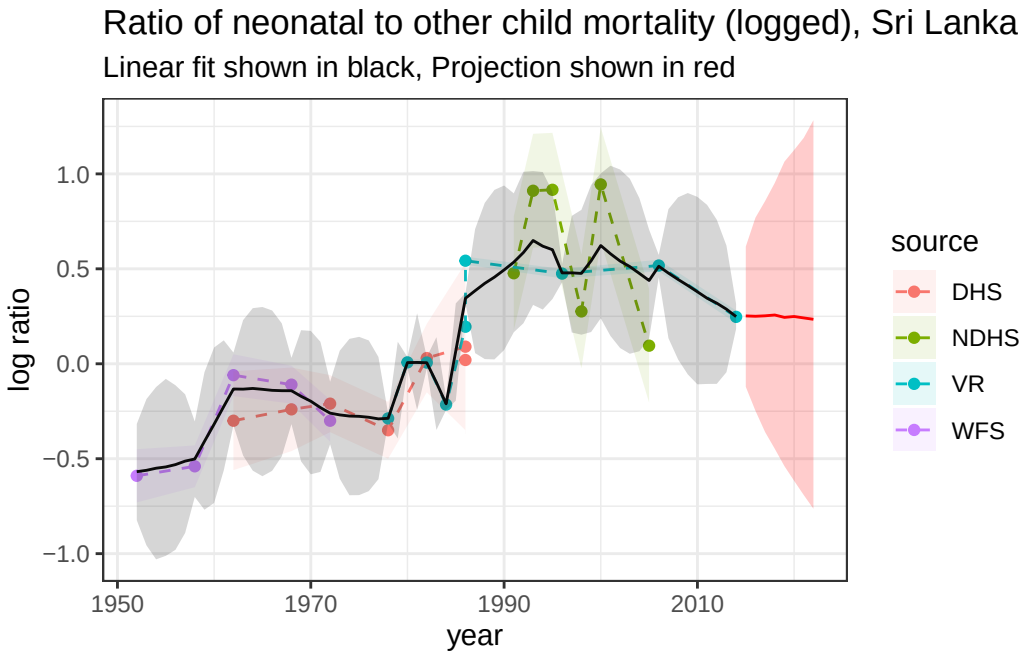
Extract the results:

```
result_2 <- model2 %>%
  gather_draws(mu[t])%>%
  median_qi() %>%
  mutate(year = years[t])

result_p2 <-  model2 %>%
  gather_draws(mu_p[p])%>%
  median_qi() %>%
  mutate(year = years[nyears]+p)
```

Plot the model:

```
ggplot(lka, aes(year, logit_ratio)) +
  geom_point(aes( color = source)) +
  geom_line(aes( color = source), lty = 2) +
  geom_ribbon(aes(ymin = logit_ratio - se,
                  ymax = logit_ratio + se,
                  fill =  source), alpha = 0.1) +
  theme_bw()+
  geom_line(data = result_2, aes(year, .value)) +
  geom_ribbon(data = result_2, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.2)
  geom_line(data = result_p2, aes(year, .value), col = "red") +
  geom_ribbon(data = result_p2, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.2
```

```
  theme_bw()+
  labs(title = "Ratio of neonatal to other child mortality (logged), Sri Lanka",
       y = "log ratio", subtitle = "Linear fit shown in black, Projection shown in red")
```



Question 4

Now alter your model above to estimate and project a second-order random walk model (RW2).

```
  model3 <- stan(data = stan_data,
                 file = "model3.stan")
```

```
Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
using C compiler: 'Apple clang version 15.0.0 (clang-1500.3.9.4)'
using SDK: 'MacOSX14.4.sdk'
clang -arch arm64 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG   -I"/Libra
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/St
In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/Rc
In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/Rc
/Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/
```

```
#include <cmath>
         ^~~~~~~
1 error generated.
make: *** [foo.o] Error 1

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 3.9e-05 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.39 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: 0.285 seconds (Warm-up)
Chain 1:                0.276 seconds (Sampling)
Chain 1:                0.561 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 4e-06 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.04 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: 0.306 seconds (Warm-up)
Chain 2:                0.28 seconds (Sampling)
Chain 2:                0.586 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
Chain 3:
Chain 3: Gradient evaluation took 5e-06 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.05 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: 0.272 seconds (Warm-up)
Chain 3:                0.217 seconds (Sampling)
Chain 3:                0.489 seconds (Total)
Chain 3:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
Chain 4:
Chain 4: Gradient evaluation took 5e-06 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.05 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: 0.283 seconds (Warm-up)
Chain 4:                0.232 seconds (Sampling)
Chain 4:                0.515 seconds (Total)
Chain 4:
```

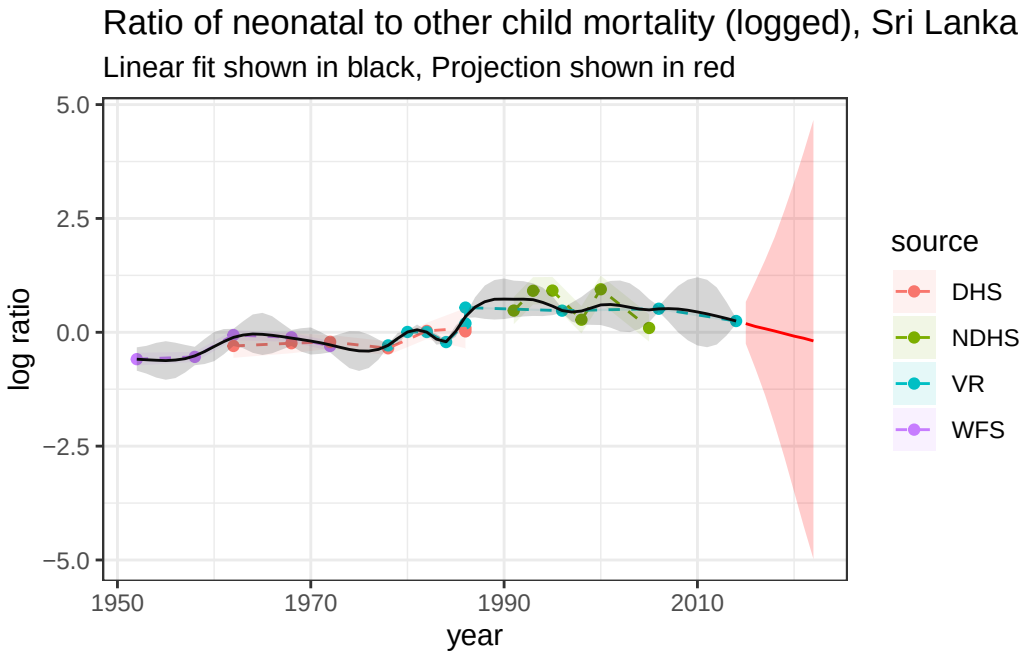
Extract the results:

```
result_3 <- model3 %>%
  gather_draws(mu[t])%>%
  median_qi() %>%
  mutate(year = years[t])

result_p3 <- model3 %>%
  gather_draws(mu_p[p])%>%
  median_qi() %>%
  mutate(year = years[nyears]+p)
```

Plot the model fit and projection to 2023 in a similar style:

```
ggplot(lka, aes(year, logit_ratio)) +
  geom_point(aes( color = source)) +
  geom_line(aes( color = source), lty = 2) +
  geom_ribbon(aes(ymin = logit_ratio - se,
                  ymax = logit_ratio + se,
                  fill =  source), alpha = 0.1) +
  theme_bw()+
  geom_line(data = result_3, aes(year, .value)) +
  geom_ribbon(data = result_3, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.2)
```

```
  geom_line(data = result_p3, aes(year, .value), col = "red") +
  geom_ribbon(data = result_p3, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.2
  theme_bw()+
  labs(title = "Ratio of neonatal to other child mortality (logged), Sri Lanka",
       y = "log ratio", subtitle = "Linear fit shown in black, Projection shown in red")
```



Question 5

Run the first order and second order random walk models, including projections out to 2022. Compare these estimates with the linear fit by plotting everything on the same graph.

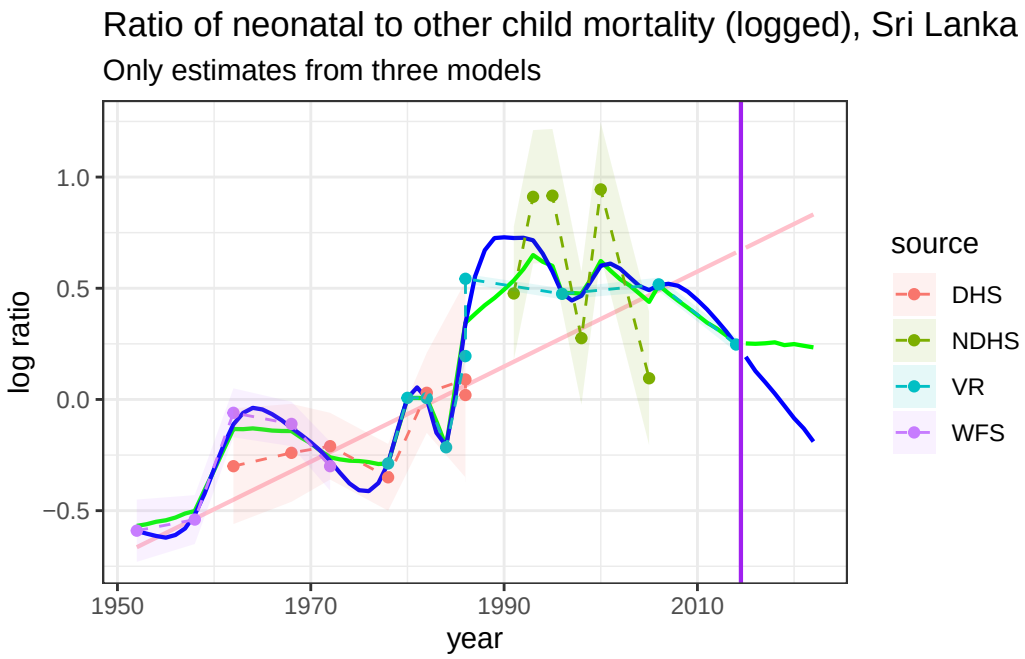
Linear: pink.

RW1: green.

RW2: blue.

```
ggplot(lka, aes(year, logit_ratio)) +
  theme_bw()+
  geom_line(data = res_model1, aes(year, .value), col = "pink", lwd = 0.75) +
  geom_line(data = res_p1, aes(year, .value), col = "pink", lwd = 0.75) +
  geom_line(data = result_2, aes(year, .value), col = "green", lwd = 0.75) +
```

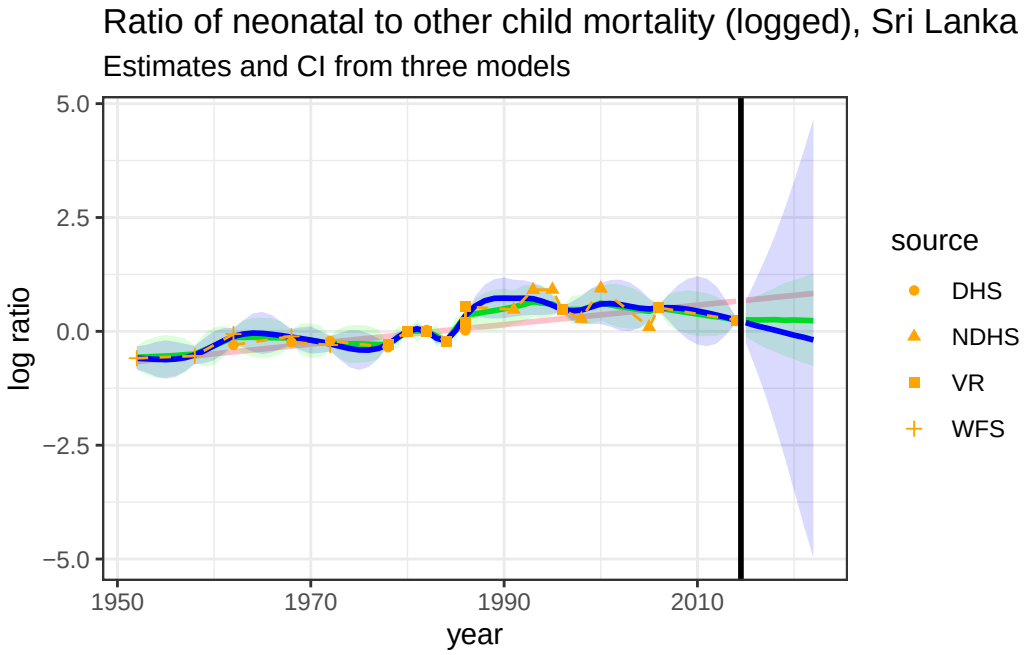
```
  geom_line(data = result_p2, aes(year, .value), col = "green", lwd = 0.75) +
  geom_line(data = result_3, aes(year, .value), col = "blue", lwd = 0.75) +
  geom_line(data = result_p3, aes(year, .value), col = "blue", lwd = 0.75) +
  geom_point(aes( color = source)) +
  geom_line(aes( color = source), lty = 2) +
  geom_ribbon(aes(ymin = logit_ratio - se,
                  ymax = logit_ratio + se,
                  fill =  source), alpha = 0.1) +
  geom_vline(xintercept = 2014.5, color = "purple", lwd = 0.75)+
  labs(title = "Ratio of neonatal to other child mortality (logged), Sri Lanka",
       y = "log ratio", subtitle = "Only estimates from three models")
```



In this plot, we include the credible interval into the plot and the green dots represents the data with different sources (different shape).

```
ggplot(lka, aes(year, logit_ratio)) +
  theme_bw() +
  geom_line(data = res_model1, aes(year, .value), col = "pink", lwd = 1) +
  geom_ribbon(data = res_model1, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.
  geom_line(data = res_p1, aes(year, .value), col = "pink", lwd = 1) +
  geom_ribbon(data = res_p1, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.2, f
```

```
geom_line(data = result_2, aes(year, .value), col = "green", lwd = 1) +
geom_ribbon(data = result_2, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.15
geom_line(data = result_p2, aes(year, .value), col = "green", lwd = 1) +
geom_ribbon(data = result_p2, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.1
geom_line(data = result_3, aes(year, .value), col = "blue", lwd = 1) +
geom_ribbon(data = result_3, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.15
geom_line(data = result_p3, aes(year, .value), col = "blue", lwd = 1) +
geom_ribbon(data = result_p3, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.1
geom_point(aes(shape = source), color = "orange") +
geom_line(lty = 2, color = "orange") +
geom_vline(xintercept = 2014.5, color = "black", lwd = 1)+
labs(title = "Ratio of neonatal to other child mortality (logged), Sri Lanka",
     y = "log ratio", subtitle = "Estimates and CI from three models")
```



RW2 has a much better and smooth fit than RW1 and linear fit. But it also has a large confidence interval. RW2 seems to have a better prediction ability than RW1 and linear as well.

Question 6

Briefly comment on which model you think is most appropriate, or an alternative model that would be more appropriate in this context.

Based on my observation, it seems that RW2 is definitely the one that is better. It seems that RW2's fit is more smooth and have a better predictive ability than RW1 and linear. But both RW2 and RW1 offer better fit than the linear model. RW2 has a better long-term trend estimation as well, seemingly capturing long-term trend better than the other two. RW2 shows a decreasing trend for future estimates which seems to be more convincing than the other two model as well.