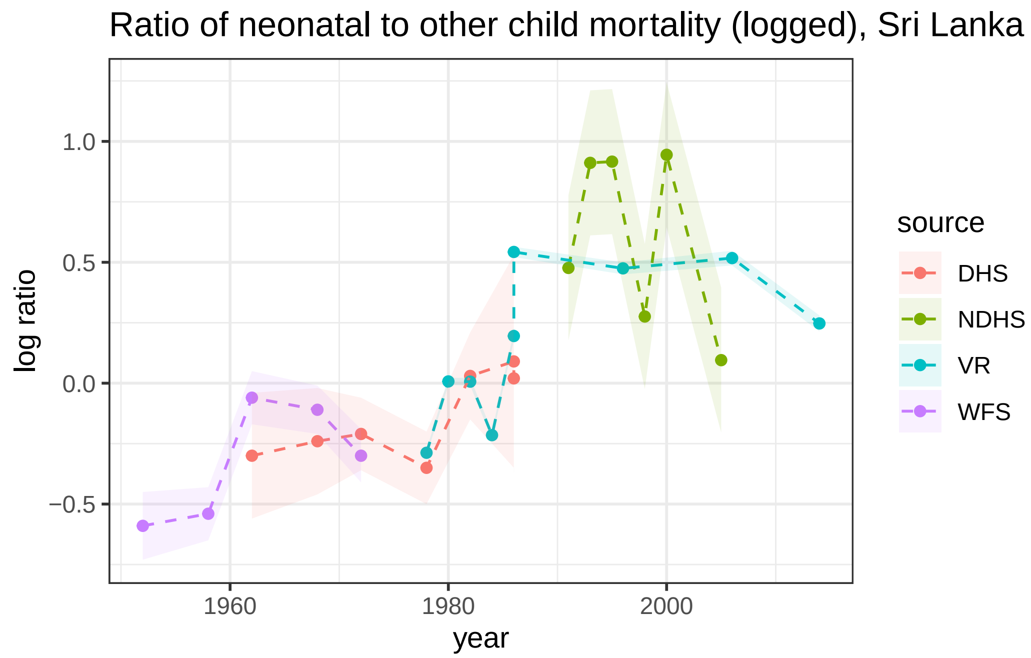
App stat 2 lab 10

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:

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

lka <- read_csv("data/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 = "code/models/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-1500.1.0.2.5)'
using SDK: 'MacOSX14.2.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/S
```

```
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/
namespace Eigen {
^
/Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/
namespace Eigen {
               ^
               ;
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/S
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 <complex>
         ^~~~~~~~~
3 errors generated.
make: *** [foo.o] Error 1

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 1.9e-05 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.19 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:                0.016 seconds (Sampling)
Chain 1:                0.035 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: 0.02 seconds (Warm-up)
Chain 2:                0.015 seconds (Sampling)
Chain 2:                0.035 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)
```

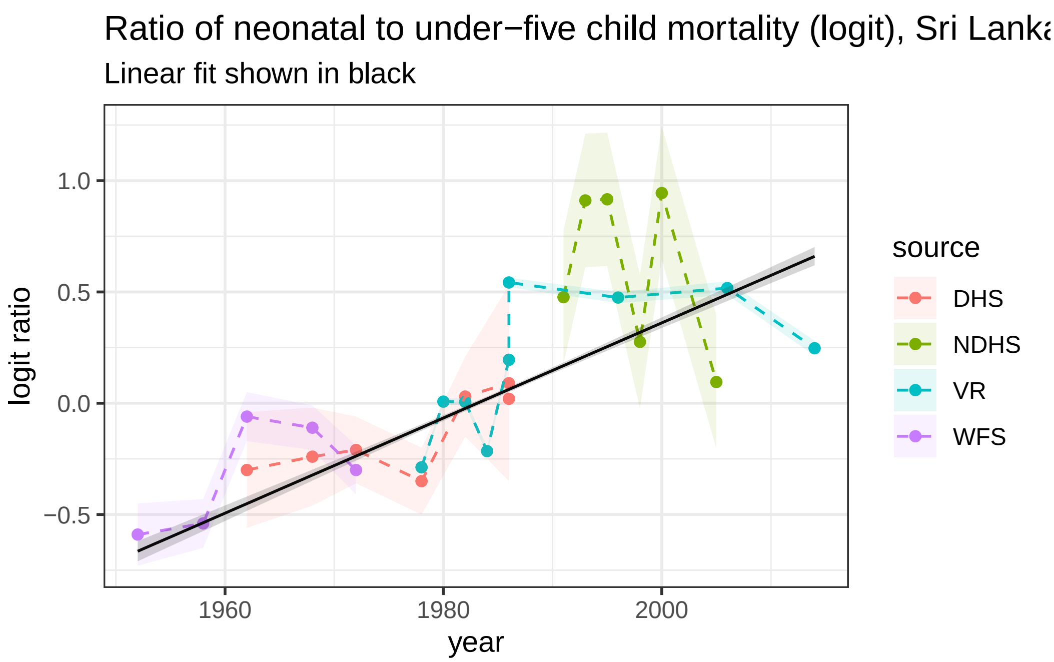
```
Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 3:
Chain 3:  Elapsed Time: 0.02 seconds (Warm-up)
Chain 3:                0.016 seconds (Sampling)
Chain 3:                0.036 seconds (Total)
Chain 3:

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: 0.018 seconds (Warm-up)
Chain 4:                0.015 seconds (Sampling)
Chain 4:                0.033 seconds (Total)
Chain 4:
```

Extract the results:

```r
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),
```

```r
                    mid_year = mean(years), se = lka$se, P = 8) # until 2022

mod2 <- stan(data = stan_data,
             file = "code/models/lka_linear_me2.stan")
```

```
Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
using C compiler: ‘Apple clang version 15.0.0 (clang-1500.1.0.2.5)’
using SDK: ‘MacOSX14.2.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/
namespace Eigen {
^
/Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/
namespace Eigen {
               ^
               ;
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
/Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/
#include <complex>
         ^~~~~~~~~
3 errors generated.
make: *** [foo.o] Error 1

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 1.7e-05 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.17 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: 0.019 seconds (Warm-up)
Chain 2:                0.016 seconds (Sampling)
Chain 2:                0.035 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:
```

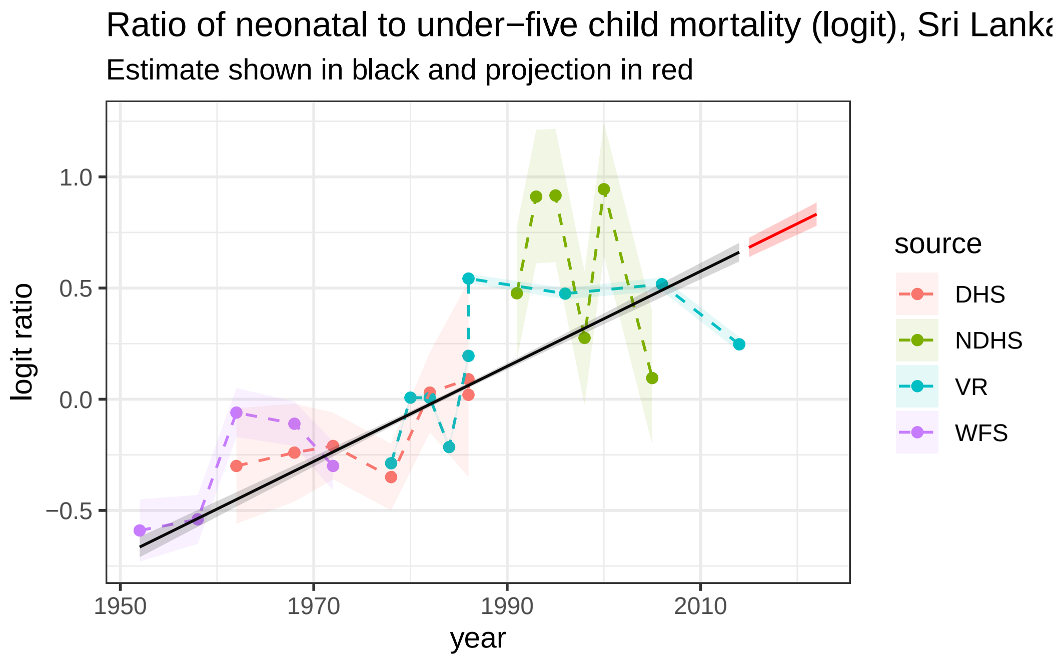
```
Chain 3:
Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
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: 0.02 seconds (Warm-up)
Chain 3:                0.015 seconds (Sampling)
Chain 3:                0.035 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: 0.018 seconds (Warm-up)
Chain 4:                0.015 seconds (Sampling)
Chain 4:                0.033 seconds (Total)
Chain 4:
```

Extract the results:

```
res2 <- mod2 %>%
  gather_draws(mu[t]) %>%
  median_qi() %>%
  mutate(year = years[t])

res2_p <- mod2 %>%
  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 = res2, aes(year, .value)) +
  geom_ribbon(data = res2, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.2)+
  geom_line(data = res2_p, aes(year, .value), col = 'red') +
  geom_ribbon(data = res2_p, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.2, f
  theme_bw()+
  labs(title = "Ratio of neonatal to under-five child mortality (logit), Sri Lanka",
       y = "logit ratio", subtitle = "Estimate shown in black and projection in red")
```



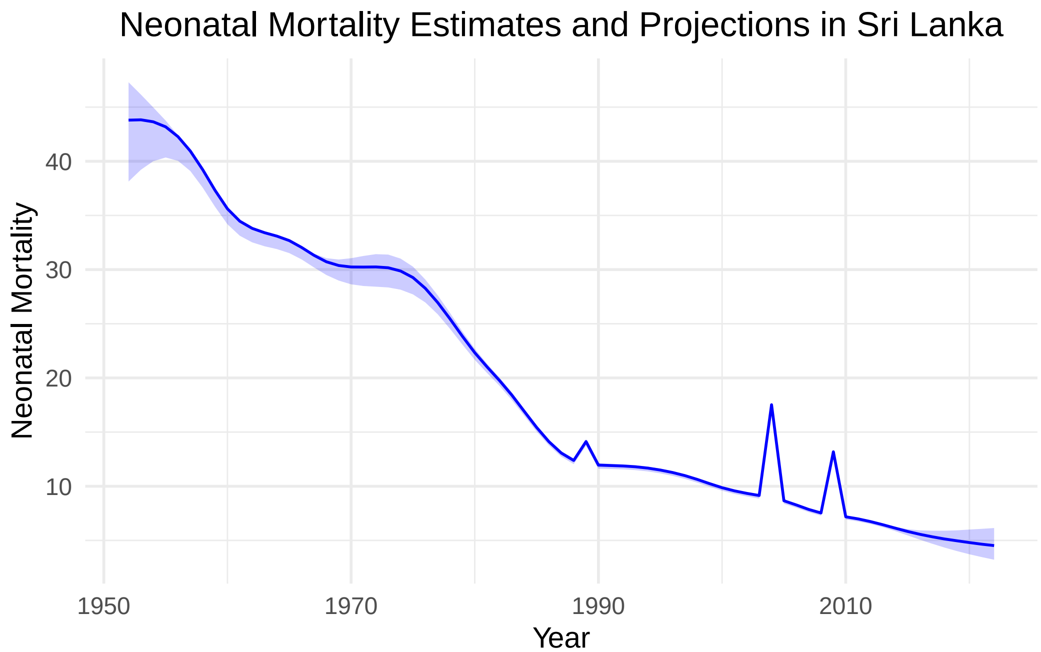
Question 2

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.

$$
logit(\pi) = \log\frac{\pi}{1-\pi}, where \quad \pi = \frac{neonatal}{u5mortality}
$$
$$
\pi = logit^{-1}(logit(\pi)) = \frac{1}{1+\exp(-logit(\pi))}
$$

```
estimate <- read.csv("data/LKA-Under-five mortality rate-Total-estimates-download.csv", sk


# We get estimate and projection from 1952 to 2022
u5_estimate <- estimate %>%
  filter(Year > 1951) %>%
  mutate(year = Year)

# Get ratio estimate using inverse logit function
inv_logit <- function(x) {
```

```r
  1 / (1 + exp(-x))
}

ratio_estimate <- rbind(res2 %>% select(.value, .lower, .upper, year),
                        res2_p %>% select(.value, .lower, .upper, year)) %>%
                        mutate(ratio_est = inv_logit(.value),
                               ratio_lower = inv_logit(.lower),
                               ratio_upper = inv_logit(.lower)
                               )

# Get neonatal mortality estimate and projections multiplying raitio and u5 estimate
neo_estimate <- left_join(u5_estimate, ratio_estimate, by = "year") %>%
                mutate(neo_est = Estimate * ratio_est,
                       neo_lower = Lower.bound * ratio_lower,
                       neo_upper = Upper.bound * ratio_upper)

# Plot neonatal mortality estimates and projections
ggplot(neo_estimate, aes(x = year)) +
  geom_line(aes(y = neo_est), color = "blue") +
  geom_ribbon(aes(ymin = neo_lower, ymax = neo_upper), fill = "blue", alpha = 0.2) +
  labs(title = "Neonatal Mortality Estimates and Projections in Sri Lanka",
       y = "Neonatal Mortality",
       x = "Year") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```



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

```
stan_data <- list(y = lka$logit_ratio, year_i = observed_years - years[1]+1,
                  T = nyears, years = years, N = length(observed_years),
                  se = lka$se, P = 8) # until 2022

mod3 <- stan(data = stan_data,
             file = "code/models/lka_mod3.stan")
```

```
Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
using C compiler: 'Apple clang version 15.0.0 (clang-1500.1.0.2.5)'
using SDK: 'MacOSX14.2.sdk'
clang -arch arm64 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG   -I"/Librar
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/Re
In file included from /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/Re
```

```
/Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/
namespace Eigen {
^
/Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/
namespace Eigen {
               ^
               ;
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
/Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/
#include <complex>
         ^~~~~~~~~
3 errors generated.
make: *** [foo.o] Error 1

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 4e-05 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.4 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: 0.105 seconds (Warm-up)
Chain 1:                0.095 seconds (Sampling)
Chain 1:                0.2 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: 0.108 seconds (Warm-up)
Chain 2:                0.074 seconds (Sampling)
Chain 2:                0.182 seconds (Total)
Chain 2:

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

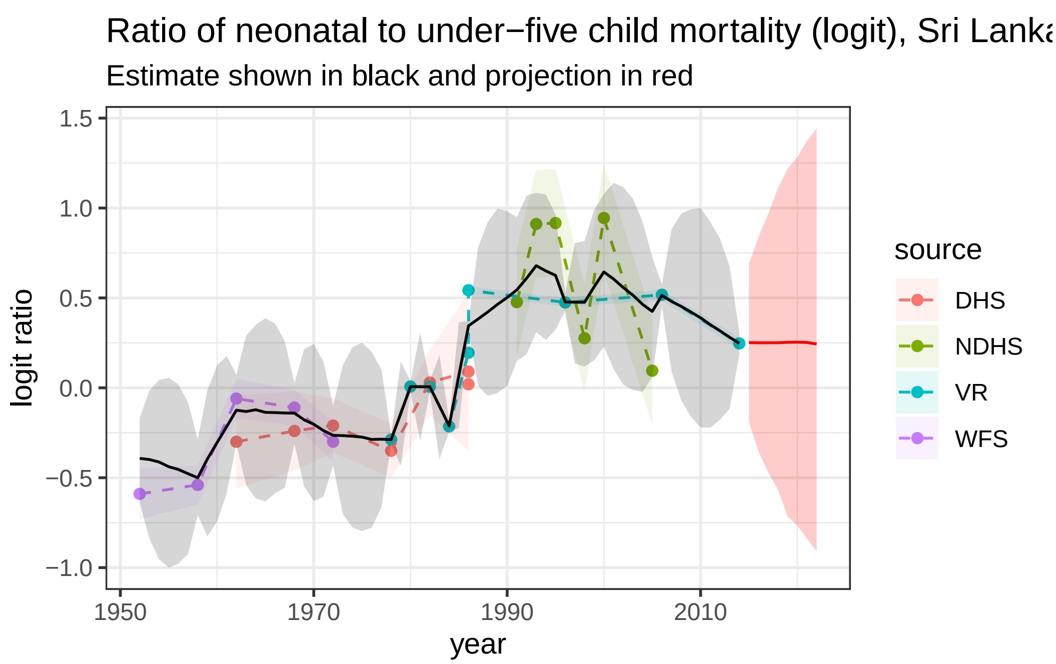
```
Chain 3:  Elapsed Time: 0.109 seconds (Warm-up)
Chain 3:                0.074 seconds (Sampling)
Chain 3:                0.183 seconds (Total)
Chain 3: 

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
Chain 4: 
Chain 4: Gradient evaluation took 4e-06 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.04 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: 0.116 seconds (Warm-up)
Chain 4:                0.072 seconds (Sampling)
Chain 4:                0.188 seconds (Total)
Chain 4: 
```

Extract the results:

```r
res3 <- mod3 %>%
  gather_draws(mu[t]) %>%
  median_qi() %>%
  mutate(year = years[t])

res3_p <- mod3 %>%
  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 = res3, aes(year, .value)) +
  geom_ribbon(data = res3, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.2)+
  geom_line(data = res3_p, aes(year, .value), col = 'red') +
  geom_ribbon(data = res3_p, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.2, f
  theme_bw()+
  labs(title = "Ratio of neonatal to under-five child mortality (logit), Sri Lanka",
       y = "logit ratio", subtitle = "Estimate shown in black and projection in red")
```



Question 4

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

```r
stan_data <- list(y = lka$logit_ratio, year_i = observed_years - years[1]+1,
                  T = nyears, years = years, N = length(observed_years),
                  se = lka$se, P = 8) # until 2022

mod4 <- stan(data = stan_data,
             file = "code/models/lka_mod4.stan")
```

```
Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
using C compiler: 'Apple clang version 15.0.0 (clang-1500.1.0.2.5)'
using SDK: 'MacOSX14.2.sdk'
clang -arch arm64 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG   -I"/Librar
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/
namespace Eigen {
^
/Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/
namespace Eigen {
               ^
               ;
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
/Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library/RcppEigen/include/Eigen/
#include <complex>
         ^~~~~~~~~
3 errors generated.
make: *** [foo.o] Error 1

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 5.2e-05 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.52 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: 0.298 seconds (Warm-up)
Chain 1:                0.267 seconds (Sampling)
Chain 1:                0.565 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 5e-06 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.05 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: 0.274 seconds (Warm-up)
Chain 2:                0.25 seconds (Sampling)
Chain 2:                0.524 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: 0.287 seconds (Warm-up)
Chain 3:                0.266 seconds (Sampling)
Chain 3:                0.553 seconds (Total)
Chain 3: 

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
Chain 4: 
Chain 4: Gradient evaluation took 4e-06 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.04 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: 0.288 seconds (Warm-up)
Chain 4:                0.241 seconds (Sampling)
```

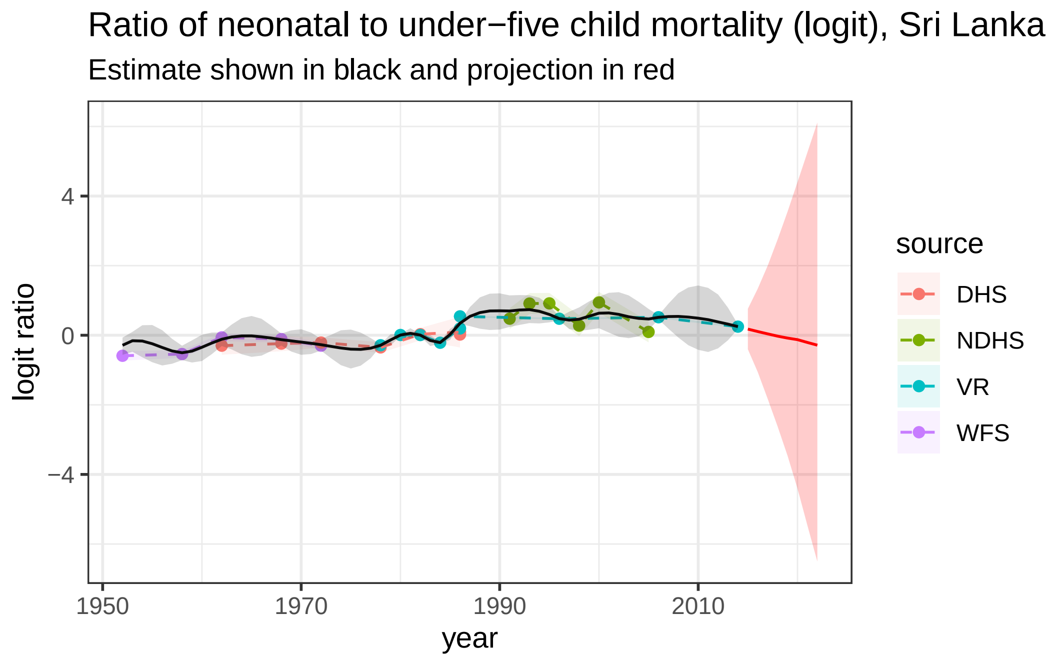
```
Chain 4:                0.529 seconds (Total)
Chain 4:
```

Extract the results:

```
res4 <- mod4 %>%
  gather_draws(mu[t]) %>%
  median_qi() %>%
  mutate(year = years[t])

res4_p <- mod4 %>%
  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 = res4, aes(year, .value)) +
  geom_ribbon(data = res4, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.2)+
  geom_line(data = res4_p, aes(year, .value), col = 'red') +
  geom_ribbon(data = res4_p, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.2, f
  theme_bw()+
  labs(title = "Ratio of neonatal to under-five child mortality (logit), Sri Lanka",
       y = "logit ratio", subtitle = "Estimate shown in black and projection in red")
```

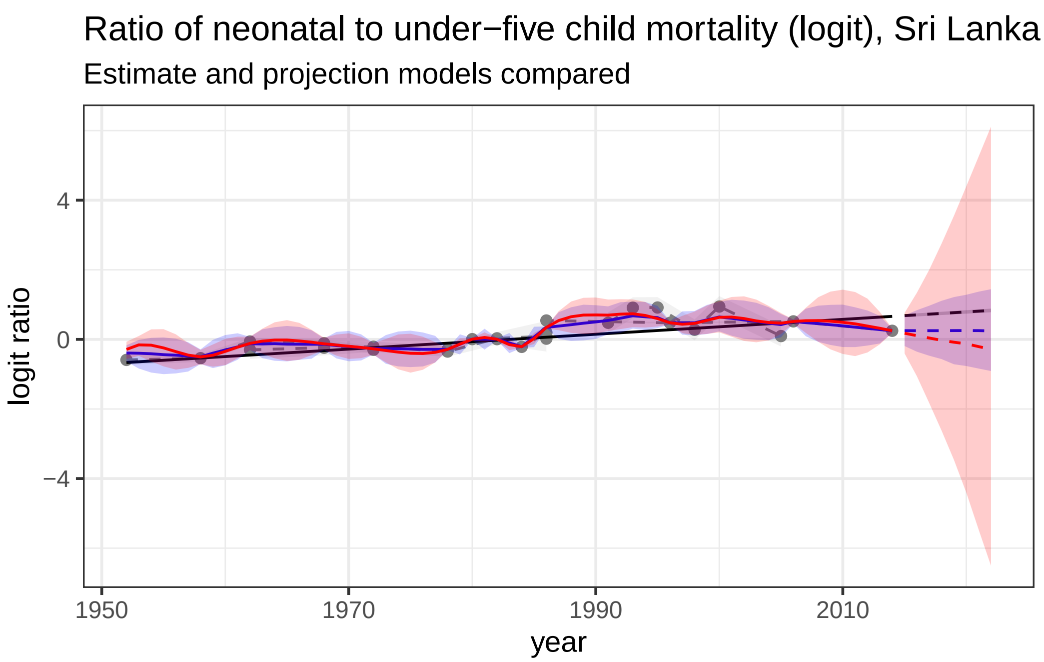


Question 5

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.

```
# Define colors for the estimates and projections
color_palette <- c("Linear Estimate" = "black",
                   "Linear Projection" = "black",
                   "RW1 Estimate" = "blue",
                   "RW1 Projection" = "blue",
                   "RW2 Estimate" = "red",
                   "RW2 Projection" = "red")  # Replace "Source Color" with the actual sou

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
  geom_line(data = res2, aes(year, .value), color = color_palette["Linear Estimate"]) +
  geom_ribbon(data = res2, aes(year, .value, ymin = .lower, ymax = .upper), alpha = 0.2, f
  geom_line(data = res2_p, aes(year, .value), color = color_palette["Linear Projection"],
  geom_ribbon(data = res2_p, aes(year, .value, ymin = .lower, ymax = .upper), alpha = 0.2,
  geom_line(data = res3, aes(year, .value), color = color_palette["RW1 Estimate"]) +
```

```
geom_ribbon(data = res3, aes(year, .value, ymin = .lower, ymax = .upper), alpha = 0.2, f
geom_line(data = res3_p, aes(year, .value), color = color_palette["RW1 Projection"], lin
geom_ribbon(data = res3_p, aes(year, .value, ymin = .lower, ymax = .upper), alpha = 0.2,
geom_line(data = res4, aes(year, .value), color = color_palette["RW2 Estimate"]) +
geom_ribbon(data = res4, aes(year, .value, ymin = .lower, ymax = .upper), alpha = 0.2, f
geom_line(data = res4_p, aes(year, .value), color = color_palette["RW2 Projection"], lin
geom_ribbon(data = res4_p, aes(year, .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 = "Estimate and projection models compared",
  color = "Legend",
  fill = "Legend"
) +
scale_color_manual(values = color_palette) +
scale_fill_manual(values = color_palette)
```



Question 6

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

Among the models considered for projecting the ratio of neonatal to under-five child mortality in Sri Lanka, the second-order random walk model seems the most appropriate. This preference is based on the model's ability to take into account more of the data's historical trend than the first-order random walk and the linear model. The linear model doesn't catch non-linear historical trend at all, and the first-order model relies heavily on the most recent observation, that is, projections are based solely on the 2014 observation. On the other hand, the second-order model considers both the 2013 and 2014 data points, which may provide a more accurate reflection of the trend and potentially capture any acceleration or deceleration in the mortality rate changes.

Moreover, from the provided graph, it is evident that the second-order random walk model offers a more nuanced projection, bending with the historical data's trajectory rather than projecting linearly from the last point. This capacity to 'bend' allows the model to adjust to recent changes in the data, which could reflect important shifts in the underlying factors affecting neonatal and under-five mortality. Such flexibility makes the second-order random walk model potentially more reliable for forecasting in this context, where recent trends can significantly influence future outcomes.