Week 10: Temporal data

today

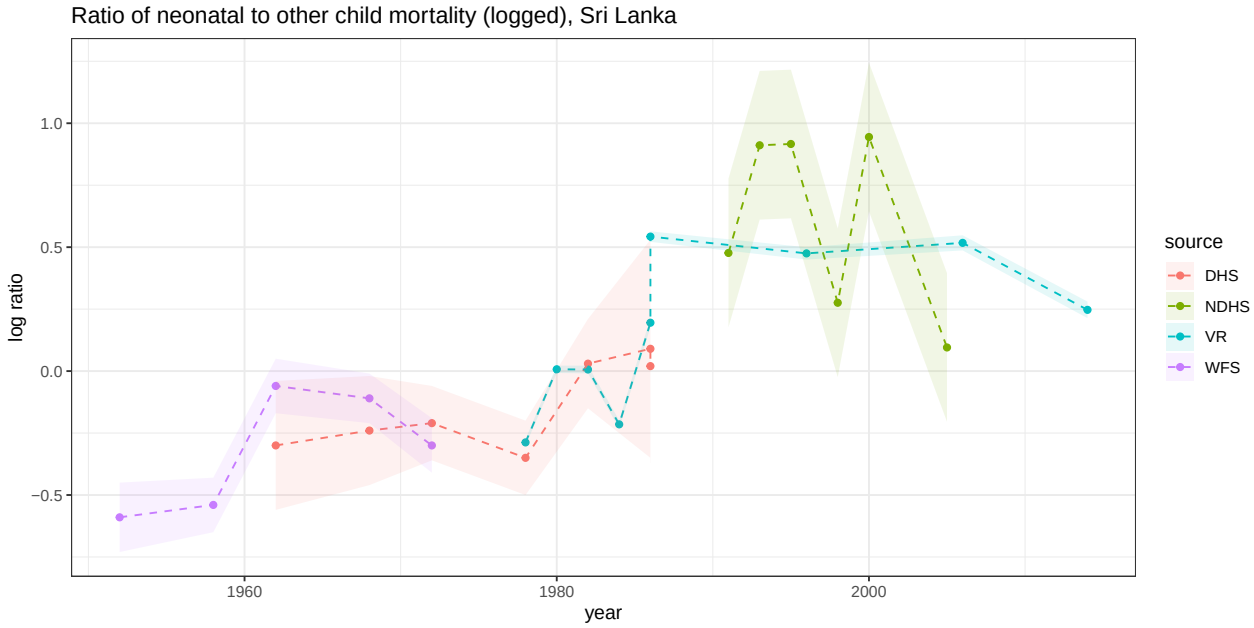
Child mortality in Sri Lanka

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:

```
# lka <- read_csv(here('./lka.csv'))
lka <- read_csv("./lka.csv")
```

```
## Rows: 27 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): country_name, country_code, source
## dbl (4): year, logit_ratio, se, ratio
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
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
    y = "log ratio")
```



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

# mod <- stan(data = stan_data, file = here('code/models/lka_linear_me.stan'))

mod1 <- stan(data = stan_data, file = "./lka_linear_me.stan")
```

```
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 8.7e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.87 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: 0.156 seconds (Warm-up)
## Chain 1:                0.122 seconds (Sampling)
## Chain 1:                0.278 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 1.8e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.18 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: 0.146 seconds (Warm-up)
## Chain 2:                0.122 seconds (Sampling)
## Chain 2:                0.268 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 2.8e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.28 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: 0.149 seconds (Warm-up)
## Chain 3:                0.139 seconds (Sampling)
## Chain 3:                0.288 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 1.9e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.19 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)
```

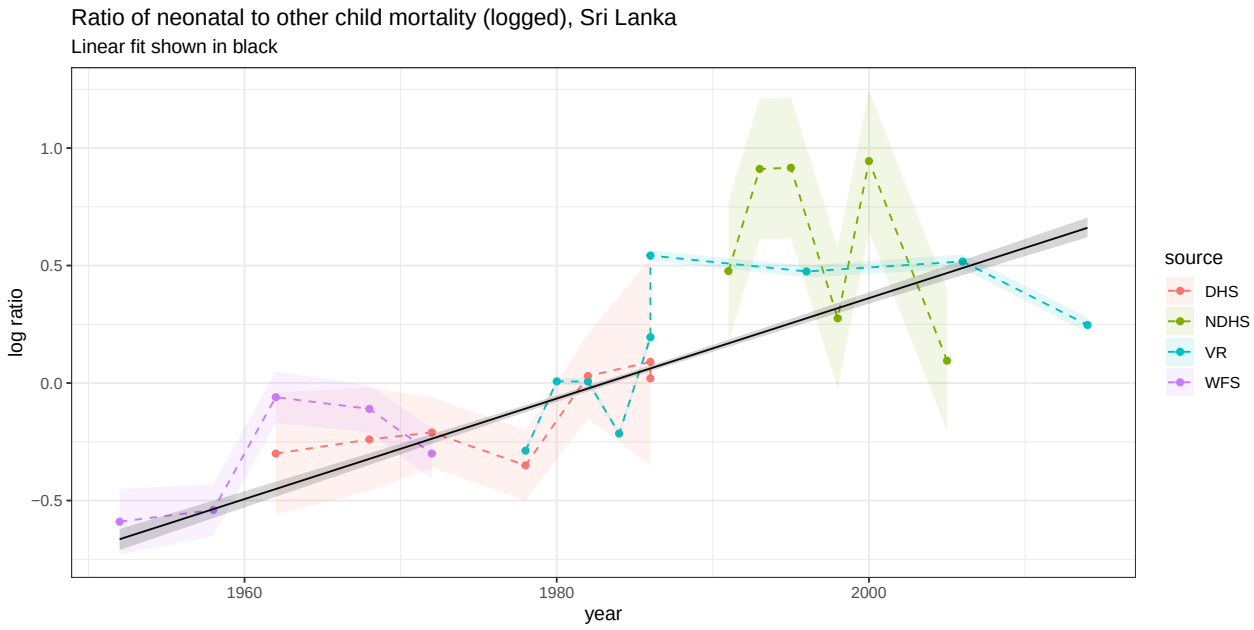
```
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.146 seconds (Warm-up)
## Chain 4:                0.121 seconds (Sampling)
## Chain 4:                0.267 seconds (Total)
## Chain 4:
```

Extract the results:

```
res1 <- mod1 %>%
    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 = res1, aes(year,
    .value)) + geom_ribbon(data = res1, aes(y = .value, ymin = .lower, ymax = .upper),
    alpha = 0.2) + theme_bw() + labs(title = "Ratio of neonatal to other child mortality (logged), Sri L
    y = "log ratio", subtitle = "Linear fit shown in black")
```



Question 1

Project the linear model above out to 2023 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.

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

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

```
        1, P = 9)
```

```
mod2 <- stan(data = stan_data, file = "./lka_linear_me_2.stan")
```

```
## Warning in readLines(file, warn = TRUE): incomplete final line found on 'E:
## \Code-archiv\R\STA2201_2023\Lab10\lka_linear_me_2.stan'
```

```
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 7.7e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.77 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: 0.161 seconds (Warm-up)
## Chain 1:                0.114 seconds (Sampling)
## Chain 1:                0.275 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 1.7e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.17 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: 0.157 seconds (Warm-up)
```

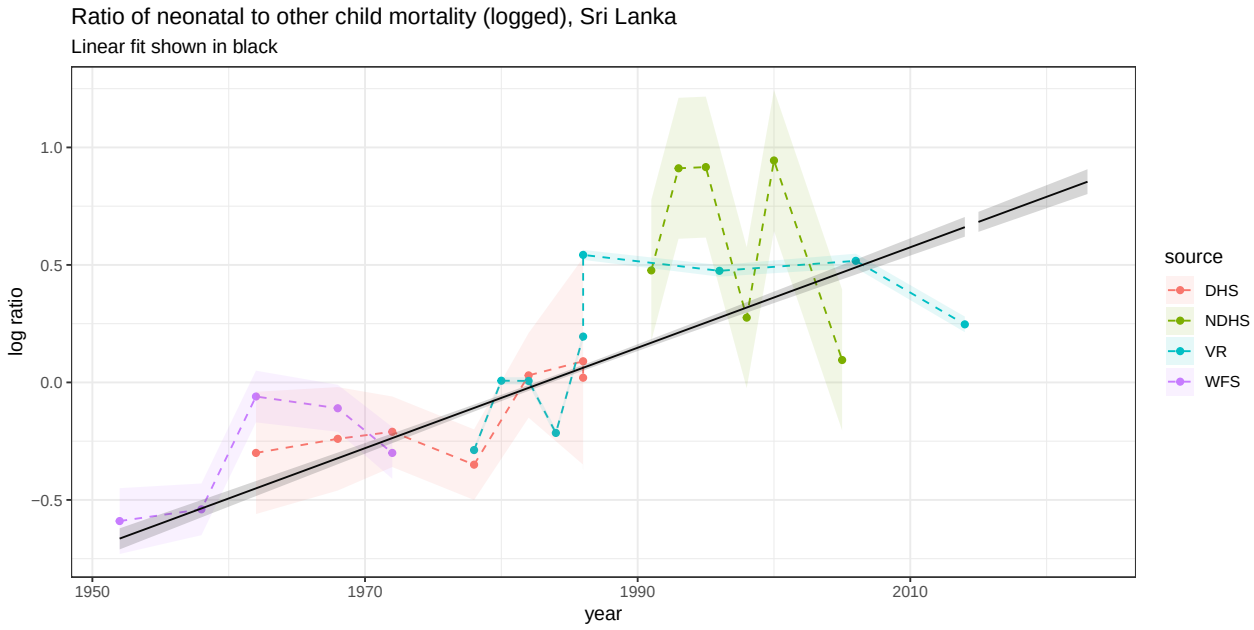
```
## Chain 2:                0.099 seconds (Sampling)
## Chain 2:                0.256 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 1.8e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.18 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: 0.165 seconds (Warm-up)
## Chain 3:                0.15 seconds (Sampling)
## Chain 3:                0.315 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 2.1e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.21 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: 0.162 seconds (Warm-up)
## Chain 4:                0.119 seconds (Sampling)
## Chain 4:                0.281 seconds (Total)
## Chain 4:
```

Extract the outcome

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

res_p <- mod2 |>
    gather_draws(mu_p[p]) |>
    median_qi() |>
    mutate(year = years[nyears] + p)
```

```
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 = res1, aes(year,
    .value)) + geom_ribbon(data = res1, aes(y = .value, ymin = .lower, ymax = .upper),
    alpha = 0.2) + geom_line(data = res_p, aes(year, .value)) + geom_ribbon(data = res_p,
    aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.2) + theme_bw() + labs(title = "Ratio of n
    y = "log ratio", subtitle = "Linear fit shown in black")
```



Random walks

Question 2

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 2023.

```
mod_rw1 <- stan(data = stan_data, file = "./lka_linear_me_rw.stan")
```

```
## Warning in readLines(file, warn = TRUE): incomplete final line found on 'E:
## \Code-archiv\R\STA2201_2023\Lab10\lka_linear_me_rw.stan'
```

```
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
```

```
## Chain 1:
## Chain 1: Gradient evaluation took 5.7e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.57 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: 1.226 seconds (Warm-up)
## Chain 1:                0.834 seconds (Sampling)
## Chain 1:                2.06 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 1.9e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.19 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: 1.061 seconds (Warm-up)
## Chain 2:                0.745 seconds (Sampling)
## Chain 2:                1.806 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 2e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.2 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: 0.994 seconds (Warm-up)
## Chain 3:                0.738 seconds (Sampling)
## Chain 3:                1.732 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 1.9e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.19 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: 0.983 seconds (Warm-up)
## Chain 4:                0.655 seconds (Sampling)
## Chain 4:                1.638 seconds (Total)
## Chain 4:
```

```
mod_rw1
```

```
## Inference for Stan model: anon_model.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##          mean se_mean    sd   2.5%    25%   50%   75% 97.5% n_eff Rhat
## mu[1]   -0.57    0.00  0.13  -0.83  -0.66 -0.57 -0.48 -0.32  3889 1.00
## mu[2]   -0.56    0.00  0.19  -0.94  -0.68 -0.55 -0.43 -0.19  3193 1.00
## mu[3]   -0.54    0.00  0.22  -0.99  -0.68 -0.54 -0.40 -0.11  2862 1.00
```

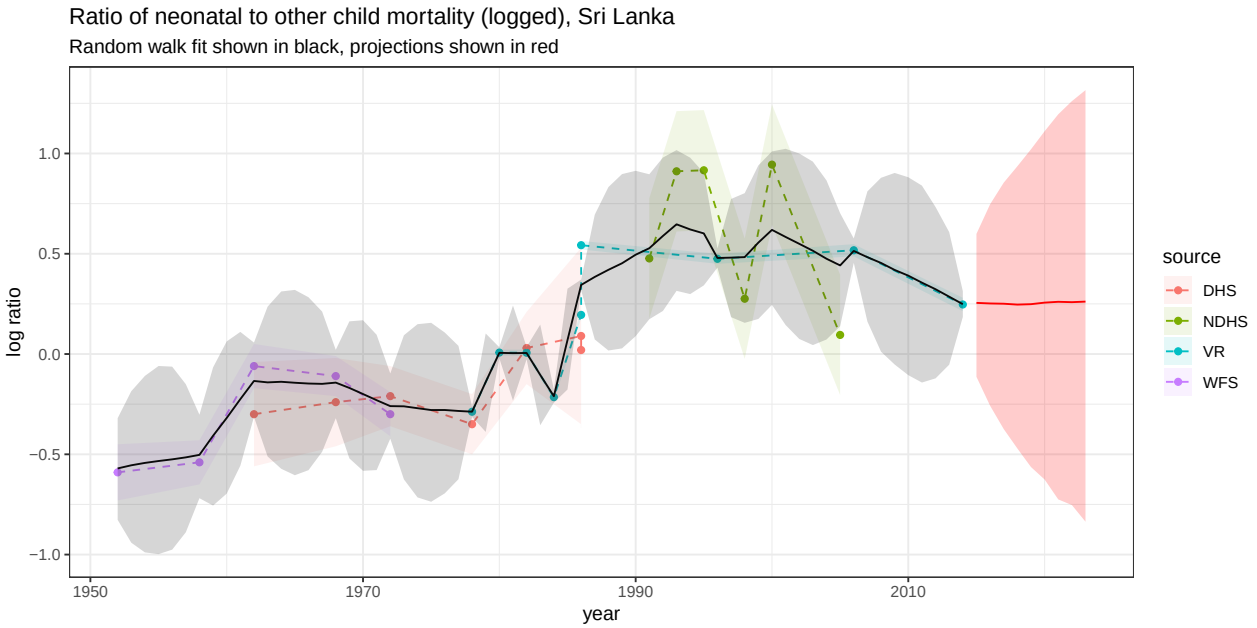
```
## mu[4]   -0.53    0.00  0.23  -1.00  -0.67 -0.53 -0.38 -0.06  2874 1.00
## mu[5]   -0.52    0.00  0.23  -0.98  -0.66 -0.53 -0.38 -0.06  2761 1.00
## mu[6]   -0.51    0.00  0.19  -0.89  -0.63 -0.52 -0.39 -0.15  3221 1.00
## mu[7]   -0.50    0.00  0.10  -0.72  -0.57 -0.50 -0.43 -0.30  3187 1.00
## mu[8]   -0.41    0.00  0.17  -0.76  -0.52 -0.41 -0.30 -0.07  3745 1.00
## mu[9]   -0.32    0.00  0.19  -0.70  -0.44 -0.32 -0.19  0.06  3206 1.00
## mu[10]  -0.22    0.00  0.17  -0.56  -0.33 -0.22 -0.11  0.11  3774 1.00
## mu[11]  -0.13    0.00  0.10  -0.31  -0.20 -0.13 -0.07  0.06  3652 1.00
## mu[12]  -0.14    0.00  0.18  -0.51  -0.25 -0.14 -0.02  0.24  3290 1.00
## mu[13]  -0.14    0.00  0.22  -0.57  -0.28 -0.14  0.00  0.31  2870 1.00
## mu[14]  -0.14    0.00  0.23  -0.60  -0.29 -0.14  0.01  0.32  2371 1.00
## mu[15]  -0.14    0.00  0.22  -0.58  -0.28 -0.15 -0.01  0.28  2476 1.00
## mu[16]  -0.14    0.00  0.18  -0.49  -0.26 -0.15 -0.03  0.21  2786 1.00
## mu[17]  -0.14    0.00  0.09  -0.32  -0.20 -0.14 -0.09  0.02  3422 1.00
## mu[18]  -0.17    0.00  0.17  -0.52  -0.28 -0.17 -0.06  0.16  4136 1.00
## mu[19]  -0.20    0.00  0.19  -0.58  -0.32 -0.20 -0.08  0.17  3365 1.00
## mu[20]  -0.23    0.00  0.17  -0.58  -0.34 -0.23 -0.12  0.10  3386 1.00
## mu[21]  -0.26    0.00  0.08  -0.42  -0.32 -0.26 -0.20 -0.10  4314 1.00
## mu[22]  -0.26    0.00  0.18  -0.63  -0.38 -0.26 -0.15  0.09  3279 1.00
## mu[23]  -0.27    0.00  0.21  -0.71  -0.41 -0.27 -0.13  0.15  2731 1.00
## mu[24]  -0.28    0.00  0.22  -0.74  -0.42 -0.28 -0.14  0.16  2979 1.00
## mu[25]  -0.28    0.00  0.21  -0.69  -0.41 -0.28 -0.15  0.11  2953 1.00
## mu[26]  -0.29    0.00  0.17  -0.62  -0.39 -0.28 -0.18  0.04  3812 1.00
## mu[27]  -0.29    0.00  0.01  -0.31  -0.30 -0.29 -0.28 -0.26  5107 1.00
## mu[28]  -0.14    0.00  0.12  -0.39  -0.22 -0.14 -0.06  0.10  5731 1.00
## mu[29]   0.01    0.00  0.01  -0.02   0.00  0.01  0.02  0.03  5967 1.00
## mu[30]   0.00    0.00  0.12  -0.23  -0.07  0.00  0.08  0.24  5705 1.00
## mu[31]   0.01    0.00  0.02  -0.03  -0.01  0.01  0.02  0.04  5731 1.00
## mu[32]  -0.10    0.00  0.13  -0.36  -0.18 -0.10 -0.02  0.15  6452 1.00
## mu[33]  -0.21    0.00  0.02  -0.24  -0.22 -0.21 -0.20 -0.18  5990 1.00
## mu[34]   0.07    0.00  0.12  -0.18  -0.01  0.07  0.15  0.33  5194 1.00
## mu[35]   0.34    0.00  0.01   0.32   0.33  0.34  0.35  0.37  5808 1.00
## mu[36]   0.38    0.00  0.16   0.07   0.28  0.38  0.48  0.69  3231 1.00
## mu[37]   0.42    0.00  0.20   0.02   0.29  0.42  0.55  0.83  2795 1.00
## mu[38]   0.45    0.00  0.22   0.03   0.32  0.45  0.59  0.90  2728 1.00
## mu[39]   0.49    0.00  0.21   0.09   0.36  0.50  0.63  0.91  2593 1.00
## mu[40]   0.53    0.00  0.18   0.17   0.41  0.53  0.66  0.90  2619 1.00
## mu[41]   0.59    0.00  0.20   0.21   0.46  0.59  0.72  0.98  2575 1.00
## mu[42]   0.65    0.00  0.18   0.32   0.53  0.65  0.76  1.02  2427 1.00
## mu[43]   0.63    0.00  0.18   0.30   0.51  0.62  0.74  0.98  3130 1.00
## mu[44]   0.60    0.00  0.14   0.34   0.51  0.60  0.69  0.90  3645 1.00
## mu[45]   0.48    0.00  0.02   0.43   0.46  0.48  0.50  0.53  6067 1.00
## mu[46]   0.48    0.00  0.15   0.18   0.39  0.48  0.58  0.77  4562 1.00
## mu[47]   0.48    0.00  0.17   0.16   0.37  0.48  0.59  0.80  3601 1.00
## mu[48]   0.55    0.00  0.19   0.17   0.43  0.56  0.68  0.94  3259 1.00
## mu[49]   0.62    0.00  0.19   0.25   0.49  0.62  0.75  1.01  2486 1.00
## mu[50]   0.58    0.00  0.22   0.14   0.44  0.58  0.73  1.02  2835 1.00
## mu[51]   0.55    0.00  0.23   0.08   0.40  0.55  0.69  1.00  2590 1.00
## mu[52]   0.51    0.00  0.23   0.04   0.36  0.51  0.66  0.96  2664 1.00
## mu[53]   0.47    0.00  0.20   0.07   0.35  0.48  0.60  0.87  2805 1.00
## mu[54]   0.44    0.00  0.14   0.14   0.34  0.44  0.53  0.70  3310 1.00
## mu[55]   0.51    0.00  0.03   0.45   0.50  0.51  0.53  0.58  4941 1.00
## mu[56]   0.48    0.00  0.17   0.16   0.38  0.48  0.59  0.81  3273 1.00
## mu[57]   0.45    0.00  0.22   0.01   0.31  0.45  0.59  0.88  2572 1.00
```

```
## mu[58]   0.42    0.00  0.24  -0.05   0.26  0.42  0.57  0.90  2405 1.00
## mu[59]   0.39    0.01  0.25  -0.11   0.23  0.39  0.55  0.88  2415 1.00
## mu[60]   0.35    0.01  0.25  -0.14   0.20  0.36  0.51  0.84  2383 1.00
## mu[61]   0.32    0.00  0.22  -0.12   0.18  0.32  0.46  0.73  2695 1.00
## mu[62]   0.29    0.00  0.17  -0.05   0.18  0.29  0.39  0.61  3773 1.00
## mu[63]   0.25    0.00  0.03   0.18   0.23  0.25  0.27  0.31  6184 1.00
## sigma    0.17    0.00  0.04   0.11   0.14  0.16  0.19  0.27   496 1.01
## mu_p[1]  0.25    0.00  0.18  -0.11   0.14  0.26  0.36  0.60  4240 1.00
## mu_p[2]  0.25    0.00  0.25  -0.26   0.09  0.25  0.41  0.75  4065 1.00
## mu_p[3]  0.25    0.01  0.31  -0.37   0.06  0.25  0.45  0.85  3704 1.00
## mu_p[4]  0.24    0.01  0.35  -0.47   0.03  0.25  0.47  0.94  3725 1.00
## mu_p[5]  0.25    0.01  0.40  -0.56   0.00  0.25  0.51  1.02  3967 1.00
## mu_p[6]  0.25    0.01  0.44  -0.63  -0.03  0.26  0.53  1.11  3961 1.00
## mu_p[7]  0.25    0.01  0.47  -0.72  -0.05  0.26  0.56  1.20  4000 1.00
## mu_p[8]  0.25    0.01  0.50  -0.75  -0.05  0.26  0.57  1.26  3968 1.00
## mu_p[9]  0.26    0.01  0.53  -0.84  -0.06  0.26  0.60  1.32  3994 1.00
## lp__    -7.31    0.57 12.23 -33.23 -14.77 -6.37  1.38 13.90   458 1.01
##
## Samples were drawn using NUTS(diag_e) at Mon Mar 27 18:12:18 2023.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

```
res_rw1 <- mod_rw1 |>
    gather_draws(mu[t]) |>
    median_qi() |>
    mutate(year = years[t])

res_prw1 <- mod_rw1 |>
    gather_draws(mu_p[p]) |>
    median_qi() |>
    mutate(year = years[nyears] + p)
```

```
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_rw1, aes(year,
    .value)) + geom_ribbon(data = res_rw1, aes(y = .value, ymin = .lower, ymax = .upper),
    alpha = 0.2) + geom_line(data = res_prw1, aes(year, .value), col = "red") + geom_ribbon(data = res_
    aes(y = .value, ymin = .lower, ymax = .upper), fill = "red", alpha = 0.2) + theme_bw() +
    labs(title = "Ratio of neonatal to other child mortality (logged), Sri Lanka",
        y = "log ratio", subtitle = "Random walk fit shown in black, projections shown in red")
```



Question 3

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

```
mod_rw2 <- stan(data = stan_data, file = "./lka_linear_me_rw2.stan")
```

```
## Warning in readLines(file, warn = TRUE): incomplete final line found on 'E:
## \Code-archiv\R\STA2201_2023\Lab10\lka_linear_me_rw2.stan'
```

```
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.000108 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 1.08 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: 2.169 seconds (Warm-up)
## Chain 1:                1.92 seconds (Sampling)
## Chain 1:                4.089 seconds (Total)
## Chain 1:
##
```

```
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 4.4e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.44 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: 2.279 seconds (Warm-up)
## Chain 2:                2.064 seconds (Sampling)
## Chain 2:                4.343 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 3.7e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.37 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: 2.426 seconds (Warm-up)
## Chain 3:                2.004 seconds (Sampling)
## Chain 3:                4.43 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 2.5e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.25 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: 2.503 seconds (Warm-up)
## Chain 4:                1.96 seconds (Sampling)
## Chain 4:                4.463 seconds (Total)
## Chain 4:
```

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

```
mod_rw2
```

```
## Inference for Stan model: anon_model.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##          mean se_mean    sd   2.5%   25%   50%   75% 97.5% n_eff Rhat
## mu[1]   -0.58    0.00  0.14  -0.86 -0.67 -0.58 -0.49 -0.32  5164 1.00
## mu[2]   -0.61    0.00  0.19  -1.00 -0.73 -0.61 -0.48 -0.24  2455 1.00
## mu[3]   -0.63    0.01  0.26  -1.15 -0.79 -0.63 -0.47 -0.13  2195 1.00
## mu[4]   -0.63    0.01  0.27  -1.18 -0.81 -0.63 -0.46 -0.09  2036 1.00
## mu[5]   -0.62    0.01  0.24  -1.12 -0.77 -0.62 -0.47 -0.15  2145 1.00
## mu[6]   -0.59    0.00  0.17  -0.93 -0.70 -0.59 -0.47 -0.25  2545 1.00
## mu[7]   -0.52    0.00  0.11  -0.74 -0.60 -0.52 -0.45 -0.32  4089 1.00
## mu[8]   -0.43    0.00  0.14  -0.70 -0.52 -0.43 -0.33 -0.14  3958 1.00
## mu[9]   -0.31    0.00  0.17  -0.64 -0.42 -0.31 -0.20  0.02  3591 1.00
## mu[10]  -0.20    0.00  0.14  -0.47 -0.29 -0.20 -0.10  0.07  3582 1.00
## mu[11]  -0.11    0.00  0.10  -0.31 -0.18 -0.11 -0.04  0.09  3687 1.00
## mu[12]  -0.06    0.00  0.16  -0.38 -0.16 -0.06  0.05  0.24  3204 1.00
## mu[13]  -0.04    0.00  0.22  -0.46 -0.18 -0.04  0.10  0.38  3019 1.00
## mu[14]  -0.04    0.00  0.24  -0.51 -0.20 -0.05  0.11  0.43  2884 1.00
## mu[15]  -0.06    0.00  0.21  -0.47 -0.20 -0.07  0.08  0.36  3056 1.00
## mu[16]  -0.09    0.00  0.15  -0.38 -0.19 -0.10  0.00  0.20  3551 1.00
## mu[17]  -0.13    0.00  0.09  -0.30 -0.19 -0.13 -0.07  0.04  5037 1.00
## mu[18]  -0.16    0.00  0.13  -0.43 -0.25 -0.16 -0.08  0.10  4095 1.00
## mu[19]  -0.19    0.00  0.16  -0.51 -0.29 -0.19 -0.09  0.13  3387 1.00
## mu[20]  -0.23    0.00  0.13  -0.50 -0.32 -0.23 -0.14  0.04  3557 1.00
## mu[21]  -0.27    0.00  0.09  -0.45 -0.33 -0.27 -0.22 -0.10  4620 1.00
## mu[22]  -0.33    0.00  0.15  -0.62 -0.43 -0.33 -0.23 -0.04  3547 1.00
## mu[23]  -0.38    0.00  0.21  -0.78 -0.51 -0.38 -0.24  0.02  3107 1.00
```

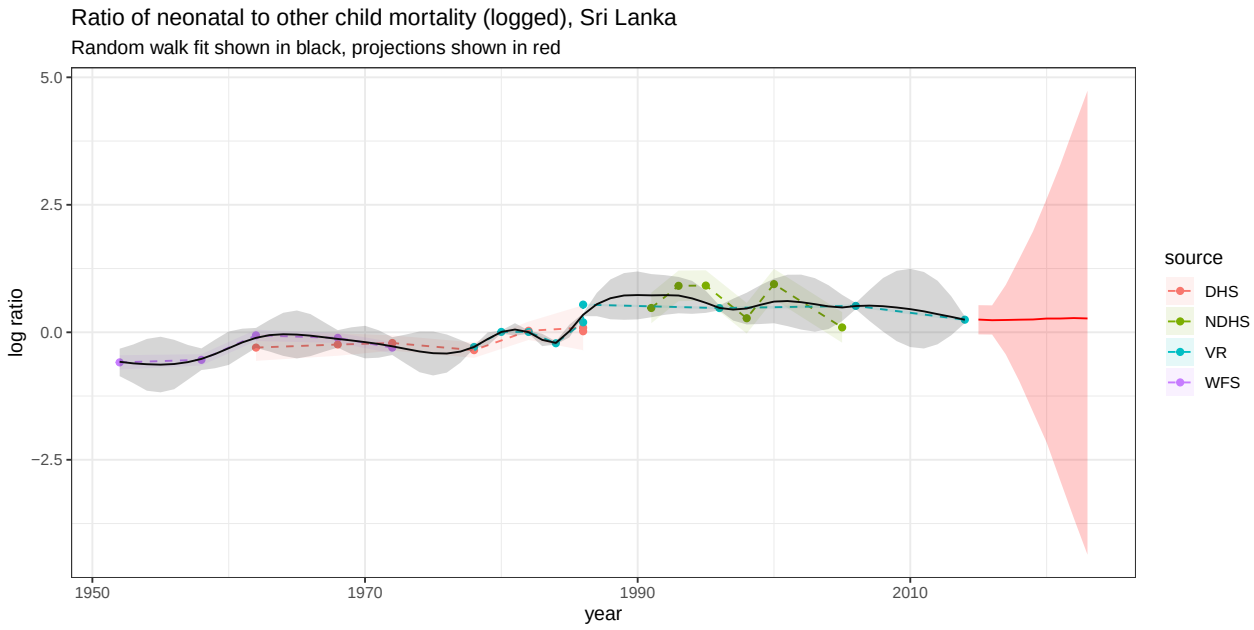
```
## mu[24]  -0.41    0.00  0.22  -0.85 -0.56 -0.41 -0.27  0.02  3052 1.00
## mu[25]  -0.41    0.00  0.19  -0.79 -0.54 -0.42 -0.29 -0.04  3158 1.00
## mu[26]  -0.38    0.00  0.11  -0.60 -0.45 -0.38 -0.30 -0.15  3744 1.00
## mu[27]  -0.29    0.00  0.01  -0.31 -0.30 -0.29 -0.28 -0.26  7164 1.00
## mu[28]  -0.13    0.00  0.06  -0.26 -0.17 -0.13 -0.09  0.00  5698 1.00
## mu[29]   0.01    0.00  0.01  -0.02  0.00  0.01  0.02  0.04  6997 1.00
## mu[30]   0.05    0.00  0.06  -0.07  0.02  0.05  0.09  0.18  6284 1.00
## mu[31]   0.00    0.00  0.02  -0.03 -0.01  0.00  0.01  0.04  6893 1.00
## mu[32]  -0.15    0.00  0.06  -0.27 -0.19 -0.15 -0.11 -0.03  8111 1.00
## mu[33]  -0.21    0.00  0.02  -0.24 -0.22 -0.21 -0.20 -0.18  5741 1.00
## mu[34]   0.02    0.00  0.06  -0.10 -0.02  0.02  0.06  0.15  5828 1.00
## mu[35]   0.34    0.00  0.01   0.32  0.33  0.34  0.35  0.37  6780 1.00
## mu[36]   0.55    0.00  0.11   0.32  0.48  0.55  0.62  0.77  3597 1.00
## mu[37]   0.66    0.00  0.19   0.26  0.54  0.67  0.79  1.04  2985 1.00
## mu[38]   0.71    0.00  0.23   0.24  0.57  0.72  0.86  1.16  2706 1.00
## mu[39]   0.73    0.00  0.24   0.25  0.57  0.73  0.88  1.19  2493 1.00
## mu[40]   0.72    0.00  0.21   0.29  0.58  0.72  0.87  1.14  2580 1.00
## mu[41]   0.73    0.00  0.20   0.34  0.59  0.73  0.87  1.12  2812 1.00
## mu[42]   0.72    0.00  0.18   0.37  0.60  0.72  0.85  1.09  2730 1.00
## mu[43]   0.67    0.00  0.16   0.36  0.56  0.66  0.77  1.01  2741 1.00
## mu[44]   0.58    0.00  0.11   0.38  0.51  0.58  0.65  0.81  2460 1.00
## mu[45]   0.48    0.00  0.02   0.43  0.46  0.48  0.49  0.53  6861 1.00
## mu[46]   0.44    0.00  0.11   0.23  0.37  0.45  0.51  0.66  3544 1.00
## mu[47]   0.47    0.00  0.16   0.15  0.37  0.47  0.57  0.78  3002 1.00
## mu[48]   0.54    0.00  0.19   0.16  0.41  0.54  0.66  0.92  2660 1.00
## mu[49]   0.61    0.00  0.22   0.18  0.46  0.60  0.76  1.05  2358 1.00
## mu[50]   0.62    0.01  0.25   0.12  0.44  0.61  0.79  1.13  2395 1.00
## mu[51]   0.59    0.01  0.28   0.05  0.40  0.59  0.77  1.13  2366 1.00
## mu[52]   0.54    0.01  0.26   0.01  0.36  0.55  0.72  1.06  2301 1.00
## mu[53]   0.50    0.00  0.22   0.06  0.35  0.51  0.65  0.92  2239 1.00
## mu[54]   0.49    0.00  0.13   0.22  0.40  0.49  0.58  0.73  2344 1.00
## mu[55]   0.52    0.00  0.03   0.45  0.49  0.51  0.54  0.58  6572 1.00
## mu[56]   0.52    0.00  0.15   0.22  0.42  0.52  0.62  0.83  2192 1.00
## mu[57]   0.51    0.01  0.27   0.01  0.34  0.51  0.67  1.06  1996 1.00
## mu[58]   0.49    0.01  0.35  -0.17  0.26  0.49  0.71  1.21  1921 1.00
## mu[59]   0.46    0.01  0.39  -0.29  0.20  0.45  0.70  1.25  1861 1.00
## mu[60]   0.41    0.01  0.38  -0.32  0.17  0.41  0.64  1.18  1864 1.00
## mu[61]   0.36    0.01  0.31  -0.24  0.16  0.36  0.55  1.01  2057 1.00
## mu[62]   0.31    0.00  0.19  -0.06  0.18  0.30  0.42  0.71  2458 1.00
## mu[63]   0.25    0.00  0.03   0.18  0.22  0.25  0.27  0.31  8669 1.00
## sigma    0.14    0.00  0.03   0.09  0.11  0.13  0.16  0.22   406 1.01
## mu_p[1]  0.25    0.00  0.14  -0.04  0.16  0.25  0.34  0.53  4152 1.00
## mu_p[2]  0.24    0.00  0.15  -0.04  0.15  0.24  0.34  0.53  3971 1.00
## mu_p[3]  0.24    0.01  0.35  -0.43  0.01  0.24  0.46  0.93  3734 1.00
## mu_p[4]  0.24    0.01  0.60  -0.96 -0.14  0.25  0.62  1.44  3779 1.00
## mu_p[5]  0.24    0.01  0.89  -1.56 -0.31  0.25  0.81  1.98  3630 1.00
## mu_p[6]  0.24    0.02  1.20  -2.16 -0.52  0.27  0.99  2.61  3601 1.00
## mu_p[7]  0.24    0.03  1.53  -2.91 -0.73  0.27  1.17  3.28  3618 1.00
## mu_p[8]  0.24    0.03  1.88  -3.65 -0.94  0.28  1.39  4.02  3626 1.00
## mu_p[9]  0.25    0.04  2.26  -4.36 -1.15  0.27  1.65  4.74  3606 1.00
## lp__     3.77    0.66 12.72 -24.28 -4.05  4.75 12.77 25.78   376 1.01
##
## Samples were drawn using NUTS(diag_e) at Mon Mar 27 18:15:25 2023.
## For each parameter, n_eff is a crude measure of effective sample size,
```

```
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

```
res_rw2 <- mod_rw2 |>
    gather_draws(mu[t]) |>
    median_qi() |>
    mutate(year = years[t])

res_prw2 <- mod_rw2 |>
    gather_draws(mu_p[p]) |>
    median_qi() |>
    mutate(year = years[nyears] + p)
```

```
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_rw2, aes(year,
    .value)) + geom_ribbon(data = res_rw2, aes(y = .value, ymin = .lower, ymax = .upper),
    alpha = 0.2) + geom_line(data = res_prw2, aes(year, .value), col = "red") + geom_ribbon(data = res_
    aes(y = .value, ymin = .lower, ymax = .upper), fill = "red", alpha = 0.2) + theme_bw() +
    labs(title = "Ratio of neonatal to other child mortality (logged), Sri Lanka",
        y = "log ratio", subtitle = "Random walk fit shown in black, projections shown in red")
```



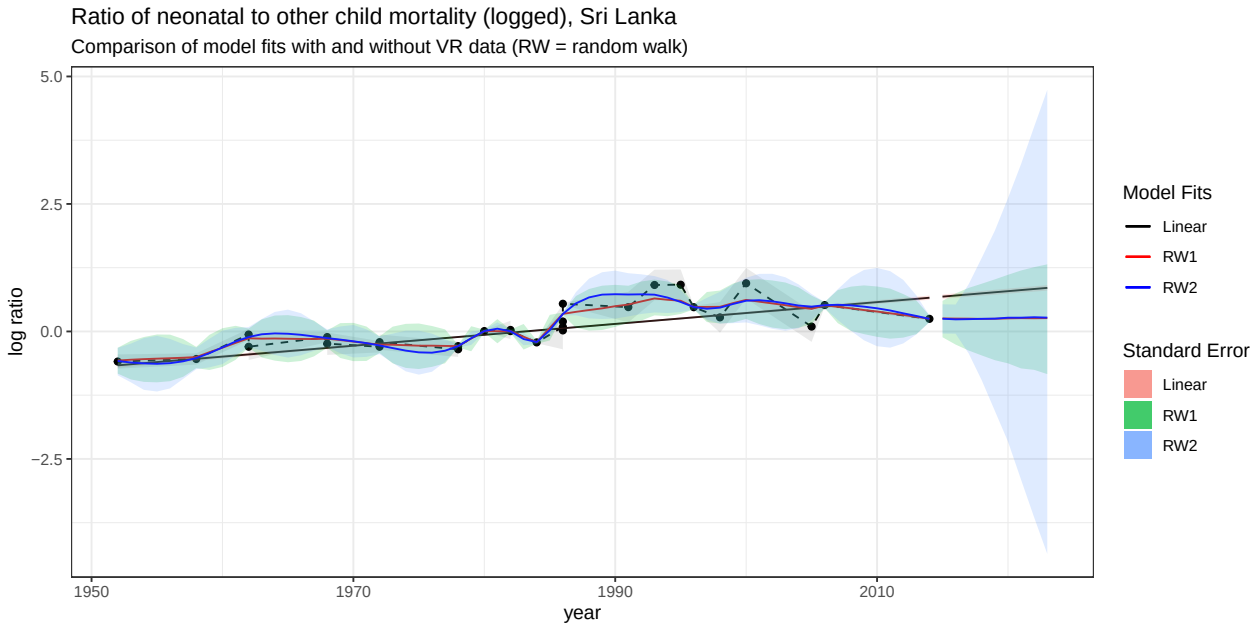
Question 4

Run the first order and second order random walk models, including projections out to 2023. Compare these estimates with the linear fit by plotting everything on the same graph.

Use the outcome of the linear model from Q1, the first- and second-order random walk model from Q2 and Q3.

```
ggplot(lka, aes(year, logit_ratio)) + geom_point() + geom_line(lty = 2) + geom_ribbon(aes(ymin = logit_
    se, ymax = logit_ratio + se), alpha = 0.1) + geom_line(data = res1, aes(year,
    .value, col = "Linear")) + geom_ribbon(data = res1, aes(y = .value, ymin = .lower,
    ymax = .upper, fill = "Linear"), alpha = 0.2) + geom_line(data = res_p, aes(year,
```

```
    .value, col = "Linear")) + geom_ribbon(data = res_p, aes(y = .value, ymin = .lower,
    ymax = .upper, fill = "Linear"), alpha = 0.2) + theme_bw() + geom_line(data = res_rw1,
    aes(year, .value, col = "RW1")) + geom_ribbon(data = res_rw1, aes(y = .value,
    ymin = .lower, ymax = .upper, fill = "RW1"), alpha = 0.2) + geom_line(data = res_prw1,
    aes(year, .value, col = "RW1")) + geom_ribbon(data = res_prw1, aes(y = .value,
    ymin = .lower, ymax = .upper, fill = "RW1"), alpha = 0.2) + geom_line(data = res_rw2,
    aes(year, .value, col = "RW2")) + geom_ribbon(data = res_rw2, aes(y = .value,
    ymin = .lower, ymax = .upper, fill = "RW2"), alpha = 0.2) + geom_line(data = res_prw2,
    aes(year, .value, col = "RW2")) + geom_ribbon(data = res_prw2, aes(y = .value,
    ymin = .lower, ymax = .upper, fill = "RW2"), alpha = 0.2) + labs(title = "Ratio of neonatal to other
    y = "log ratio", subtitle = "Comparison of model fits with and without VR data (RW = random walk)",
    col = "Median fits", fill = "Standard Error") + scale_color_manual(name = "Model Fits",
    values = c(Linear = "black", RW1 = "red", RW2 = "blue"))
```



Question 5

Rerun the RW2 model excluding the VR data. Briefly comment on the differences between the two data situations.

Need to filter out the VR data.

```
# Filter out the VR data
lka_nVR <- lka %>%
    filter(source != "VR")
obs_years <- lka_nVR$year
years <- min(obs_years):max(obs_years)
nyears <- length(years)
```

```
max(years)
```

```
## [1] 2005
```

After filtering out the VR data, the most recent year becomes 2005, therefore, we need to add in total 18 years into the projection.

```
stan_data_nvr <- list(N = length(obs_years), T = nyears, mid_year = mean(years),
    y = lka_nVR$logit_ratio, se = lka_nVR$se, years = years, year_i = obs_years -
        years[1] + 1, P = 18)
```

```
mod_rw_nvr <- stan(data = stan_data_nvr, file = "./lka_linear_me_rw2.stan", cores = parallel::detectCor
```

```
## Warning in readLines(file, warn = TRUE): incomplete final line found on 'E:
## \Code-archiv\R\STA2201_2023\Lab10\lka_linear_me_rw2.stan'
```

```
## Warning: There were 67 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup
## to find out why this is a problem and how to eliminate them.
```

```
## Warning: There were 4 chains where the estimated Bayesian Fraction of Missing Information was low. S
## https://mc-stan.org/misc/warnings.html#bfmi-low
```

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

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

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

```
mod_rw_nvr
```

```
## Inference for Stan model: anon_model.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##            mean se_mean    sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
## mu[1]     -0.61    0.00  0.13  -0.87  -0.70  -0.61  -0.52  -0.36  3762 1.00
## mu[2]     -0.60    0.00  0.12  -0.83  -0.67  -0.59  -0.51  -0.37  3463 1.00
## mu[3]     -0.58    0.00  0.12  -0.82  -0.65  -0.57  -0.50  -0.35  2928 1.00
## mu[4]     -0.55    0.00  0.12  -0.80  -0.62  -0.55  -0.47  -0.33  2027 1.00
## mu[5]     -0.52    0.00  0.11  -0.76  -0.59  -0.52  -0.45  -0.32  1317 1.00
## mu[6]     -0.49    0.00  0.10  -0.71  -0.55  -0.48  -0.42  -0.30  1108 1.00
## mu[7]     -0.44    0.00  0.09  -0.63  -0.50  -0.43  -0.37  -0.27  1231 1.00
## mu[8]     -0.37    0.00  0.09  -0.56  -0.43  -0.37  -0.31  -0.21  2263 1.00
## mu[9]     -0.31    0.00  0.09  -0.48  -0.36  -0.30  -0.25  -0.13  3193 1.00
## mu[10]    -0.24    0.00  0.09  -0.41  -0.30  -0.24  -0.18  -0.07  2659 1.00
## mu[11]    -0.18    0.00  0.09  -0.35  -0.24  -0.18  -0.12  -0.01  1193 1.00
## mu[12]    -0.14    0.00  0.10  -0.33  -0.21  -0.15  -0.08   0.05   966 1.00
## mu[13]    -0.12    0.00  0.10  -0.31  -0.19  -0.12  -0.05   0.10   979 1.00
## mu[14]    -0.11    0.00  0.11  -0.31  -0.18  -0.12  -0.04   0.11  1116 1.00
## mu[15]    -0.11    0.00  0.10  -0.30  -0.18  -0.12  -0.05   0.09  1284 1.00
## mu[16]    -0.13    0.00  0.09  -0.29  -0.18  -0.13  -0.07   0.06  1592 1.00
## mu[17]    -0.15    0.00  0.08  -0.29  -0.20  -0.15  -0.10   0.02  2114 1.00
## mu[18]    -0.17    0.00  0.08  -0.32  -0.22  -0.17  -0.12  -0.01  2891 1.00
## mu[19]    -0.20    0.00  0.08  -0.35  -0.25  -0.20  -0.15  -0.04  3271 1.00
## mu[20]    -0.23    0.00  0.08  -0.38  -0.28  -0.23  -0.18  -0.08  2922 1.00
## mu[21]    -0.26    0.00  0.08  -0.41  -0.31  -0.25  -0.20  -0.10  1853 1.00
## mu[22]    -0.28    0.00  0.09  -0.46  -0.33  -0.27  -0.21  -0.11  1460 1.00
## mu[23]    -0.29    0.00  0.10  -0.51  -0.36  -0.29  -0.22  -0.10  1354 1.00
## mu[24]    -0.30    0.00  0.12  -0.54  -0.37  -0.30  -0.22  -0.08  1220 1.00
```

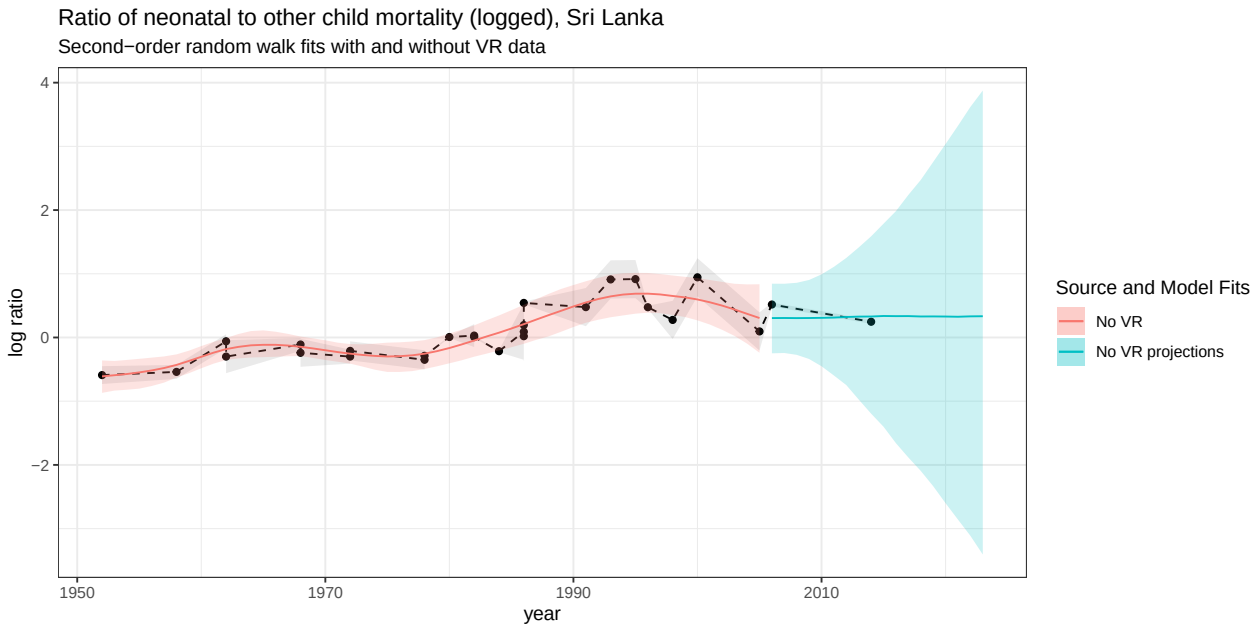
```
## mu[25]    -0.30    0.00  0.12  -0.54  -0.37  -0.29  -0.21  -0.08  1133 1.00
## mu[26]    -0.28    0.00  0.12  -0.53  -0.36  -0.28  -0.20  -0.06  1094 1.00
## mu[27]    -0.26    0.00  0.12  -0.50  -0.33  -0.25  -0.17  -0.04  1187 1.00
## mu[28]    -0.22    0.00  0.12  -0.45  -0.29  -0.21  -0.14   0.00  1769 1.00
## mu[29]    -0.17    0.00  0.12  -0.41  -0.25  -0.17  -0.08   0.06  2007 1.00
## mu[30]    -0.11    0.00  0.12  -0.36  -0.19  -0.11  -0.03   0.12  2116 1.00
## mu[31]    -0.05    0.00  0.13  -0.30  -0.14  -0.05   0.03   0.19  2229 1.00
## mu[32]     0.01    0.00  0.13  -0.25  -0.08   0.02   0.10   0.27  2273 1.00
## mu[33]     0.07    0.00  0.14  -0.20  -0.02   0.08   0.17   0.35  2241 1.00
## mu[34]     0.14    0.00  0.15  -0.15   0.04   0.14   0.24   0.43  2103 1.00
## mu[35]     0.21    0.00  0.16  -0.10   0.11   0.21   0.31   0.51  1921 1.00
## mu[36]     0.28    0.00  0.16  -0.04   0.17   0.28   0.38   0.60  1891 1.00
## mu[37]     0.35    0.00  0.17   0.02   0.24   0.35   0.46   0.68  1786 1.00
## mu[38]     0.42    0.00  0.17   0.09   0.31   0.42   0.53   0.76  1407 1.00
## mu[39]     0.49    0.01  0.17   0.16   0.38   0.49   0.60   0.82  1014 1.00
## mu[40]     0.55    0.01  0.17   0.23   0.44   0.55   0.66   0.88   726 1.01
## mu[41]     0.61    0.01  0.17   0.28   0.50   0.60   0.71   0.94   572 1.01
## mu[42]     0.65    0.01  0.16   0.32   0.54   0.65   0.76   0.98   482 1.01
## mu[43]     0.68    0.01  0.16   0.35   0.57   0.67   0.78   1.01   478 1.01
## mu[44]     0.69    0.01  0.16   0.37   0.58   0.69   0.79   1.01   502 1.01
## mu[45]     0.69    0.01  0.16   0.39   0.58   0.69   0.80   1.01   621 1.01
## mu[46]     0.68    0.01  0.16   0.38   0.57   0.68   0.78   1.00   806 1.01
## mu[47]     0.66    0.00  0.16   0.35   0.54   0.65   0.76   0.97  1099 1.00
## mu[48]     0.63    0.00  0.16   0.32   0.52   0.63   0.74   0.95  1652 1.00
## mu[49]     0.60    0.00  0.17   0.27   0.49   0.60   0.71   0.93  2089 1.00
## mu[50]     0.55    0.00  0.18   0.20   0.44   0.55   0.67   0.90  2317 1.00
## mu[51]     0.50    0.00  0.19   0.12   0.37   0.50   0.63   0.87  2318 1.00
## mu[52]     0.44    0.00  0.21   0.02   0.30   0.44   0.58   0.84  2086 1.00
## mu[53]     0.37    0.01  0.24  -0.10   0.22   0.37   0.53   0.83  1261 1.00
## mu[54]     0.30    0.01  0.28  -0.24   0.12   0.31   0.49   0.84   868 1.00
## sigma      0.04    0.00  0.02   0.01   0.02   0.03   0.04   0.07   100 1.03
## mu_p[1]    0.30    0.01  0.28  -0.25   0.12   0.30   0.50   0.84   920 1.00
## mu_p[2]    0.30    0.01  0.28  -0.24   0.12   0.31   0.49   0.84   900 1.00
## mu_p[3]    0.30    0.01  0.29  -0.27   0.11   0.30   0.50   0.86   951 1.00
## mu_p[4]    0.30    0.01  0.32  -0.34   0.10   0.31   0.52   0.90  1153 1.00
## mu_p[5]    0.30    0.01  0.36  -0.45   0.07   0.31   0.54   0.99  1491 1.00
## mu_p[6]    0.30    0.01  0.42  -0.60   0.05   0.31   0.57   1.11  1873 1.00
## mu_p[7]    0.30    0.01  0.50  -0.75   0.01   0.32   0.61   1.25  2409 1.00
## mu_p[8]    0.30    0.01  0.58  -0.97  -0.03   0.33   0.66   1.41  2819 1.00
## mu_p[9]    0.30    0.01  0.68  -1.20  -0.08   0.33   0.71   1.59  3139 1.00
## mu_p[10]   0.30    0.01  0.78  -1.40  -0.13   0.34   0.76   1.79  3365 1.00
## mu_p[11]   0.30    0.01  0.89  -1.66  -0.19   0.33   0.82   1.98  3678 1.00
## mu_p[12]   0.30    0.02  1.00  -1.88  -0.25   0.34   0.87   2.24  4077 1.00
## mu_p[13]   0.29    0.02  1.12  -2.09  -0.32   0.33   0.93   2.47  4362 1.00
## mu_p[14]   0.29    0.02  1.25  -2.33  -0.38   0.33   1.01   2.75  4388 1.00
## mu_p[15]   0.29    0.02  1.38  -2.60  -0.45   0.33   1.10   3.04  4400 1.00
## mu_p[16]   0.29    0.02  1.51  -2.86  -0.53   0.33   1.18   3.32  4399 1.00
## mu_p[17]   0.29    0.02  1.65  -3.11  -0.60   0.33   1.25   3.62  4387 1.00
## mu_p[18]   0.29    0.03  1.79  -3.41  -0.67   0.33   1.34   3.88  4370 1.00
## lp__     140.64    2.26 20.93 102.13 126.04 140.49 154.59 183.40    86 1.04
##
## Samples were drawn using NUTS(diag_e) at Mon Mar 27 18:16:42 2023.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
```

```
## convergence, Rhat=1).
```

```
res_rw_nvr <- mod_rw_nvr |>
    gather_draws(mu[t]) |>
    median_qi() |>
    mutate(year = years[t])

res_prw_nvr <- mod_rw_nvr |>
    gather_draws(mu_p[p]) |>
    median_qi() |>
    mutate(year = years[nyears] + p)
```

```
ggplot(lka, aes(year, logit_ratio)) + geom_point() + geom_line(lty = 2) + geom_ribbon(aes(ymin = logit_r
    se, ymax = logit_ratio + se), alpha = 0.1) + geom_line(data = res_rw_nvr, aes(year,
    .value, col = "No VR")) + geom_ribbon(data = res_rw_nvr, aes(y = .value, ymin = .lower,
    ymax = .upper, fill = "No VR"), alpha = 0.2) + geom_line(data = res_prw_nvr,
    aes(year, .value, col = "No VR projections")) + geom_ribbon(data = res_prw_nvr,
    aes(y = .value, ymin = .lower, ymax = .upper, fill = "No VR projections"), alpha = 0.2) +
    theme_bw() + labs(title = "Ratio of neonatal to other child mortality (logged), Sri Lanka",
    y = "log ratio", subtitle = "Second-order random walk fits with and without VR data",
    col = "Source and Model Fits", fill = "Source and Model Fits")
```



Question 6

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

The linear model is the best here because it captures the upward trend of the temporal change of child mortality the best.