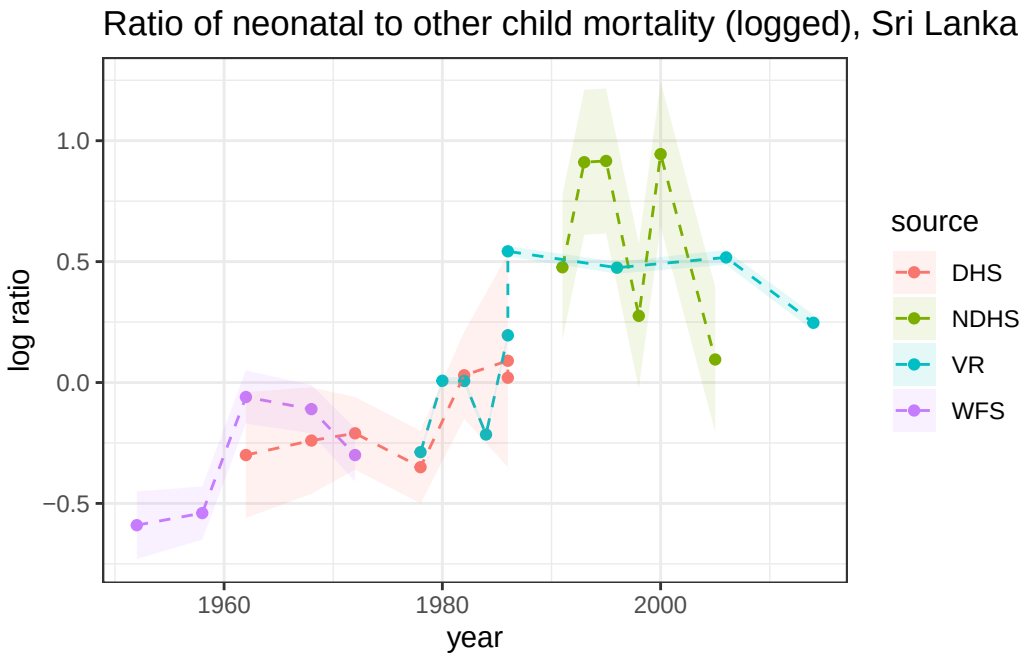
Week 10: Temporal data

26/03/23

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(here("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 = here("code/models/lka_linear_me.stan"))
```

```
Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
clang -mmacosx-version-min=10.13 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEI
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.1/Resources/library/StanHeac
In file included from /Library/Frameworks/R.framework/Versions/4.1/Resources/library/RcppEige
In file included from /Library/Frameworks/R.framework/Versions/4.1/Resources/library/RcppEige
```

```
/Library/Frameworks/R.framework/Versions/4.1/Resources/library/RcppEigen/include/Eigen/src/Co
namespace Eigen {
^
/Library/Frameworks/R.framework/Versions/4.1/Resources/library/RcppEigen/include/Eigen/src/Co
namespace Eigen {
              ^
              ;
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.1/Resources/library/StanHeac
In file included from /Library/Frameworks/R.framework/Versions/4.1/Resources/library/RcppEige
/Library/Frameworks/R.framework/Versions/4.1/Resources/library/RcppEigen/include/Eigen/Core:9
#include <complex>
         ^~~~~~~~
3 errors generated.
make: *** [foo.o] Error 1

SAMPLING FOR MODEL 'lka_linear_me' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 2.5e-05 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.25 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: 0.049601 seconds (Warm-up)
Chain 1:                0.048482 seconds (Sampling)
Chain 1:                0.098083 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'lka_linear_me' NOW (CHAIN 2).
Chain 2:
```

```
Chain 2: Gradient evaluation took 7e-06 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.07 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: 0.052183 seconds (Warm-up)
Chain 2:                0.038259 seconds (Sampling)
Chain 2:                0.090442 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'lka_linear_me' NOW (CHAIN 3).
Chain 3:
Chain 3: Gradient evaluation took 8e-06 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.08 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: 0.05046 seconds (Warm-up)
Chain 3:                0.044944 seconds (Sampling)
Chain 3:                0.095404 seconds (Total)
Chain 3:

SAMPLING FOR MODEL 'lka_linear_me' NOW (CHAIN 4).
Chain 4:
Chain 4: Gradient evaluation took 9e-06 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.09 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: 0.051747 seconds (Warm-up)
Chain 4:                0.050504 seconds (Sampling)
Chain 4:                0.102251 seconds (Total)
Chain 4:
```

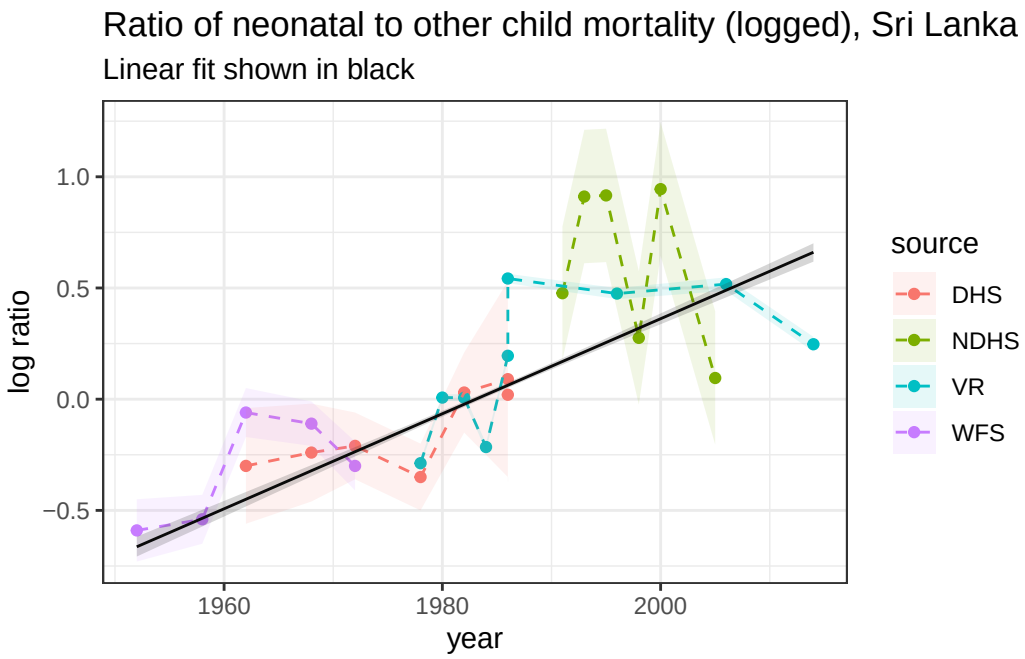
Extract the results:

```
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 other child mortality (logged), Sri Lanka",
     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.

```
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=9)

mod2 <- stan(data = stan_data,
```

```
              file = here("code/models/mod10_1.stan"))
```

```
Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
clang -mmacosx-version-min=10.13 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEI
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.1/Resources/library/StanHeac
In file included from /Library/Frameworks/R.framework/Versions/4.1/Resources/library/RcppEige
In file included from /Library/Frameworks/R.framework/Versions/4.1/Resources/library/RcppEige
/Library/Frameworks/R.framework/Versions/4.1/Resources/library/RcppEigen/include/Eigen/src/Co
namespace Eigen {
^
/Library/Frameworks/R.framework/Versions/4.1/Resources/library/RcppEigen/include/Eigen/src/Co
namespace Eigen {
               ^
               ;
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.1/Resources/library/StanHeac
In file included from /Library/Frameworks/R.framework/Versions/4.1/Resources/library/RcppEige
/Library/Frameworks/R.framework/Versions/4.1/Resources/library/RcppEigen/include/Eigen/Core:9
#include <complex>
         ^~~~~~~~~
3 errors generated.
make: *** [foo.o] Error 1

SAMPLING FOR MODEL 'mod10_1' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 3.4e-05 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.34 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: 0.055262 seconds (Warm-up)
Chain 1:                0.041736 seconds (Sampling)
Chain 1:                0.096998 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'mod10_1' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 1.1e-05 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.11 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: 0.050746 seconds (Warm-up)
Chain 2:                0.040914 seconds (Sampling)
Chain 2:                0.09166 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'mod10_1' NOW (CHAIN 3).
Chain 3:
Chain 3: Gradient evaluation took 8e-06 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.08 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: 0.055909 seconds (Warm-up)
Chain 3:                0.040589 seconds (Sampling)
Chain 3:                0.096498 seconds (Total)
Chain 3:

SAMPLING FOR MODEL 'mod10_1' NOW (CHAIN 4).
Chain 4:
Chain 4: Gradient evaluation took 9e-06 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.09 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: 0.05396 seconds (Warm-up)
Chain 4:                0.04281 seconds (Sampling)
Chain 4:                0.09677 seconds (Total)
Chain 4:
```

```
mod2
```

```
Inference for Stan model: mod10_1.
```

4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

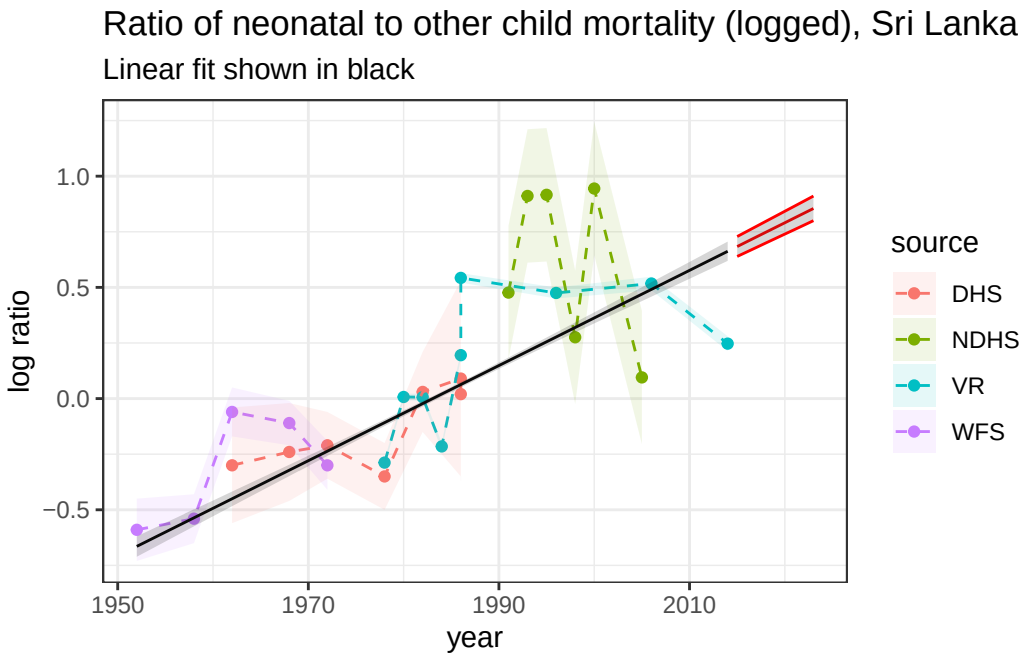
	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
alpha	0.00	0.00	0.01	-0.01	-0.01	0.00	0.00	0.01	858	1.01
beta	0.02	0.00	0.00	0.02	0.02	0.02	0.02	0.02	4001	1.00
mu[1]	-0.66	0.00	0.02	-0.71	-0.68	-0.66	-0.65	-0.62	2740	1.00
mu[2]	-0.64	0.00	0.02	-0.69	-0.66	-0.64	-0.63	-0.60	2692	1.00
mu[3]	-0.62	0.00	0.02	-0.67	-0.64	-0.62	-0.61	-0.58	2648	1.00
mu[4]	-0.60	0.00	0.02	-0.64	-0.61	-0.60	-0.59	-0.56	2601	1.00
mu[5]	-0.58	0.00	0.02	-0.62	-0.59	-0.58	-0.56	-0.54	2553	1.00
mu[6]	-0.56	0.00	0.02	-0.60	-0.57	-0.56	-0.54	-0.52	2502	1.00
mu[7]	-0.54	0.00	0.02	-0.57	-0.55	-0.54	-0.52	-0.50	2450	1.00
mu[8]	-0.51	0.00	0.02	-0.55	-0.53	-0.51	-0.50	-0.48	2395	1.00
mu[9]	-0.49	0.00	0.02	-0.53	-0.51	-0.49	-0.48	-0.46	2338	1.00
mu[10]	-0.47	0.00	0.02	-0.51	-0.48	-0.47	-0.46	-0.44	2278	1.00
mu[11]	-0.45	0.00	0.02	-0.48	-0.46	-0.45	-0.44	-0.42	2217	1.00
mu[12]	-0.43	0.00	0.02	-0.46	-0.44	-0.43	-0.42	-0.40	2153	1.00
mu[13]	-0.41	0.00	0.02	-0.44	-0.42	-0.41	-0.40	-0.38	2086	1.00
mu[14]	-0.39	0.00	0.01	-0.42	-0.40	-0.39	-0.38	-0.36	2018	1.00
mu[15]	-0.37	0.00	0.01	-0.39	-0.37	-0.36	-0.36	-0.34	1947	1.00
mu[16]	-0.34	0.00	0.01	-0.37	-0.35	-0.34	-0.33	-0.32	1873	1.00
mu[17]	-0.32	0.00	0.01	-0.35	-0.33	-0.32	-0.31	-0.30	1798	1.00
mu[18]	-0.30	0.00	0.01	-0.33	-0.31	-0.30	-0.29	-0.28	1721	1.00
mu[19]	-0.28	0.00	0.01	-0.30	-0.29	-0.28	-0.27	-0.26	1642	1.00
mu[20]	-0.26	0.00	0.01	-0.28	-0.27	-0.26	-0.25	-0.24	1562	1.00
mu[21]	-0.24	0.00	0.01	-0.26	-0.24	-0.24	-0.23	-0.22	1482	1.00
mu[22]	-0.22	0.00	0.01	-0.24	-0.22	-0.22	-0.21	-0.20	1401	1.00
mu[23]	-0.19	0.00	0.01	-0.21	-0.20	-0.19	-0.19	-0.18	1321	1.00
mu[24]	-0.17	0.00	0.01	-0.19	-0.18	-0.17	-0.17	-0.16	1243	1.00
mu[25]	-0.15	0.00	0.01	-0.17	-0.16	-0.15	-0.15	-0.13	1168	1.00
mu[26]	-0.13	0.00	0.01	-0.15	-0.13	-0.13	-0.12	-0.11	1097	1.00
mu[27]	-0.11	0.00	0.01	-0.12	-0.11	-0.11	-0.10	-0.09	1032	1.00
mu[28]	-0.09	0.00	0.01	-0.10	-0.09	-0.09	-0.08	-0.07	974	1.00
mu[29]	-0.07	0.00	0.01	-0.08	-0.07	-0.07	-0.06	-0.05	926	1.00
mu[30]	-0.04	0.00	0.01	-0.06	-0.05	-0.04	-0.04	-0.03	889	1.00
mu[31]	-0.02	0.00	0.01	-0.04	-0.03	-0.02	-0.02	-0.01	866	1.01
mu[32]	0.00	0.00	0.01	-0.01	-0.01	0.00	0.00	0.01	858	1.01
mu[33]	0.02	0.00	0.01	0.01	0.02	0.02	0.02	0.03	867	1.01
mu[34]	0.04	0.00	0.01	0.03	0.04	0.04	0.05	0.05	896	1.01
mu[35]	0.06	0.00	0.01	0.05	0.06	0.06	0.07	0.07	945	1.00
mu[36]	0.08	0.00	0.01	0.07	0.08	0.08	0.09	0.10	1015	1.00
mu[37]	0.11	0.00	0.01	0.09	0.10	0.11	0.11	0.12	1106	1.00

mu[38]	0.13	0.00	0.01	0.11	0.12	0.13	0.13	0.14	1217	1.00
mu[39]	0.15	0.00	0.01	0.13	0.14	0.15	0.15	0.16	1348	1.00
mu[40]	0.17	0.00	0.01	0.15	0.16	0.17	0.18	0.18	1496	1.00
mu[41]	0.19	0.00	0.01	0.18	0.19	0.19	0.20	0.21	1654	1.00
mu[42]	0.21	0.00	0.01	0.20	0.21	0.21	0.22	0.23	1817	1.00
mu[43]	0.23	0.00	0.01	0.22	0.23	0.23	0.24	0.25	1985	1.00
mu[44]	0.26	0.00	0.01	0.24	0.25	0.26	0.26	0.27	2157	1.00
mu[45]	0.28	0.00	0.01	0.26	0.27	0.28	0.28	0.30	2329	1.00
mu[46]	0.30	0.00	0.01	0.28	0.29	0.30	0.31	0.32	2498	1.00
mu[47]	0.32	0.00	0.01	0.30	0.31	0.32	0.33	0.34	2666	1.00
mu[48]	0.34	0.00	0.01	0.32	0.33	0.34	0.35	0.37	2863	1.00
mu[49]	0.36	0.00	0.01	0.34	0.35	0.36	0.37	0.39	3056	1.00
mu[50]	0.38	0.00	0.01	0.36	0.38	0.38	0.39	0.41	3242	1.00
mu[51]	0.41	0.00	0.01	0.38	0.40	0.41	0.41	0.43	3421	1.00
mu[52]	0.43	0.00	0.01	0.40	0.42	0.43	0.44	0.46	3591	1.00
mu[53]	0.45	0.00	0.02	0.42	0.44	0.45	0.46	0.48	3763	1.00
mu[54]	0.47	0.00	0.02	0.44	0.46	0.47	0.48	0.50	3872	1.00
mu[55]	0.49	0.00	0.02	0.46	0.48	0.49	0.50	0.52	3907	1.00
mu[56]	0.51	0.00	0.02	0.48	0.50	0.51	0.52	0.55	3939	1.00
mu[57]	0.53	0.00	0.02	0.50	0.52	0.53	0.55	0.57	3966	1.00
mu[58]	0.56	0.00	0.02	0.52	0.54	0.56	0.57	0.59	3991	1.00
mu[59]	0.58	0.00	0.02	0.54	0.56	0.58	0.59	0.61	4012	1.00
mu[60]	0.60	0.00	0.02	0.56	0.58	0.60	0.61	0.64	4030	1.00
mu[61]	0.62	0.00	0.02	0.58	0.61	0.62	0.63	0.66	4047	1.00
mu[62]	0.64	0.00	0.02	0.60	0.63	0.64	0.65	0.68	4061	1.00
mu[63]	0.66	0.00	0.02	0.62	0.65	0.66	0.68	0.71	4074	1.00
mu_p[1]	0.68	0.00	0.02	0.64	0.67	0.68	0.70	0.73	4085	1.00
mu_p[2]	0.70	0.00	0.02	0.66	0.69	0.71	0.72	0.75	4094	1.00
mu_p[3]	0.73	0.00	0.02	0.68	0.71	0.73	0.74	0.77	4103	1.00
mu_p[4]	0.75	0.00	0.02	0.70	0.73	0.75	0.76	0.80	4110	1.00
mu_p[5]	0.77	0.00	0.02	0.72	0.75	0.77	0.79	0.82	4116	1.00
mu_p[6]	0.79	0.00	0.03	0.74	0.77	0.79	0.81	0.84	4122	1.00
mu_p[7]	0.81	0.00	0.03	0.76	0.79	0.81	0.83	0.87	4127	1.00
mu_p[8]	0.83	0.00	0.03	0.78	0.82	0.83	0.85	0.89	4131	1.00
mu_p[9]	0.85	0.00	0.03	0.80	0.84	0.85	0.87	0.91	4135	1.00
lp__	-618.28	0.03	1.03	-621.03	-618.67	-617.98	-617.55	-617.28	1184	1.00

Samples were drawn using NUTS(diag_e) at Sun Mar 26 16:19:35 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).

```
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)
```

```
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, c
  theme_bw()+
  labs(title = "Ratio of neonatal to other child mortality (logged), Sri Lanka",
       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.

```
mod3 <- stan(data = stan_data,
             file = here("code/models/mod10_2.stan"))
```

```
Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
clang -mmacosx-version-min=10.13 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEI
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.1/Resources/library/StanHea
In file included from /Library/Frameworks/R.framework/Versions/4.1/Resources/library/RcppEig
In file included from /Library/Frameworks/R.framework/Versions/4.1/Resources/library/RcppEig
/Library/Frameworks/R.framework/Versions/4.1/Resources/library/RcppEigen/include/Eigen/src/Co
namespace Eigen {
^
/Library/Frameworks/R.framework/Versions/4.1/Resources/library/RcppEigen/include/Eigen/src/Co
namespace Eigen {
```

```
               ^
               ;
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.1/Resources/library/StanHea
In file included from /Library/Frameworks/R.framework/Versions/4.1/Resources/library/RcppEig
/Library/Frameworks/R.framework/Versions/4.1/Resources/library/RcppEigen/include/Eigen/Core:
#include <complex>
         ^~~~~~~~
3 errors generated.
make: *** [foo.o] Error 1

SAMPLING FOR MODEL 'mod10_2' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 3.7e-05 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.37 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: 0.277264 seconds (Warm-up)
Chain 1:                0.205434 seconds (Sampling)
Chain 1:                0.482698 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'mod10_2' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 1e-05 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.1 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: 0.260948 seconds (Warm-up)
Chain 2:                0.189551 seconds (Sampling)
Chain 2:                0.450499 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'mod10_2' NOW (CHAIN 3).
Chain 3:
Chain 3: Gradient evaluation took 1.1e-05 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.11 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: 0.24851 seconds (Warm-up)
Chain 3:                0.173516 seconds (Sampling)
Chain 3:                0.422026 seconds (Total)
Chain 3:
```

```
SAMPLING FOR MODEL 'mod10_2' NOW (CHAIN 4).
Chain 4:
Chain 4: Gradient evaluation took 1.2e-05 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.12 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: 0.245155 seconds (Warm-up)
Chain 4:                0.2209 seconds (Sampling)
Chain 4:                0.466055 seconds (Total)
Chain 4:
```

```
mod3
```

```
Inference for Stan model: mod10_2.
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 -0.56 -0.48 -0.31  4242 1.00
mu[2]   -0.56    0.00  0.20  -0.95  -0.69 -0.56 -0.44 -0.17  3052 1.00
mu[3]   -0.55    0.00  0.22  -1.01  -0.69 -0.55 -0.40 -0.12  2864 1.00
mu[4]   -0.54    0.00  0.23  -1.00  -0.69 -0.53 -0.39 -0.10  2828 1.00
mu[5]   -0.53    0.00  0.22  -0.96  -0.67 -0.52 -0.39 -0.10  2892 1.00
mu[6]   -0.52    0.00  0.18  -0.88  -0.64 -0.52 -0.40 -0.16  3268 1.00
mu[7]   -0.51    0.00  0.10  -0.71  -0.58 -0.50 -0.44 -0.31  4235 1.00
mu[8]   -0.41    0.00  0.17  -0.76  -0.53 -0.41 -0.31 -0.07  3261 1.00
mu[9]   -0.32    0.00  0.19  -0.70  -0.45 -0.32 -0.20  0.05  3172 1.00
mu[10]  -0.23    0.00  0.17  -0.56  -0.34 -0.23 -0.12  0.12  3496 1.00
```

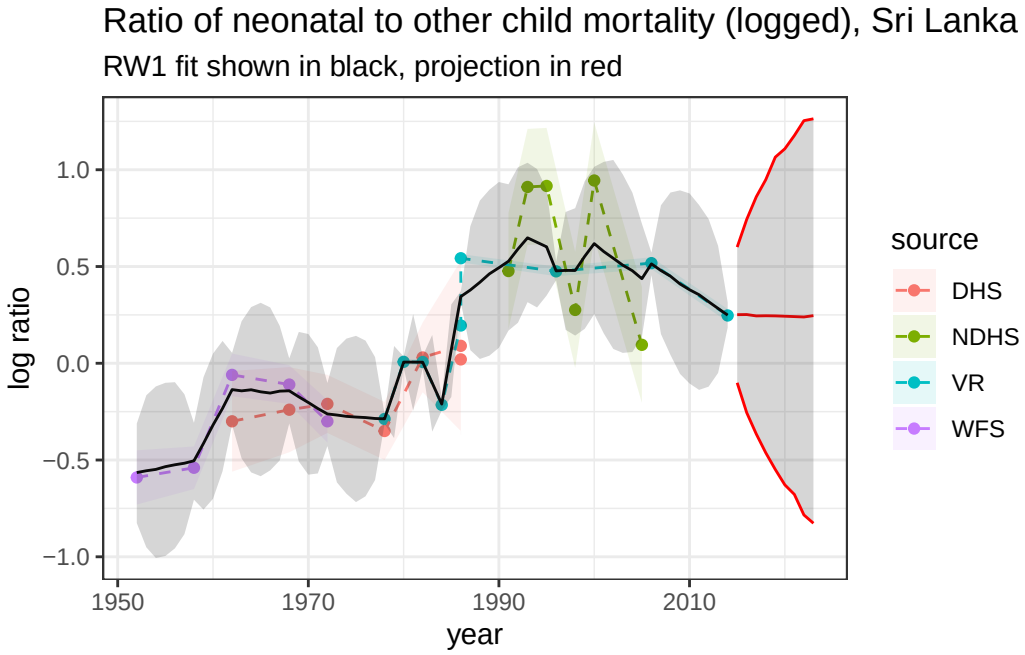
mu[11]	-0.13	0.00	0.09	-0.32	-0.20	-0.14	-0.07	0.05	3801	1.00
mu[12]	-0.14	0.00	0.18	-0.49	-0.25	-0.14	-0.02	0.22	3638	1.00
mu[13]	-0.14	0.00	0.21	-0.56	-0.28	-0.14	0.00	0.29	3228	1.00
mu[14]	-0.14	0.00	0.23	-0.58	-0.29	-0.15	0.01	0.31	3210	1.00
mu[15]	-0.15	0.00	0.22	-0.55	-0.29	-0.15	-0.01	0.29	3636	1.00
mu[16]	-0.15	0.00	0.18	-0.50	-0.27	-0.14	-0.03	0.19	3833	1.00
mu[17]	-0.14	0.00	0.09	-0.31	-0.20	-0.14	-0.09	0.02	4721	1.00
mu[18]	-0.17	0.00	0.17	-0.51	-0.28	-0.17	-0.06	0.16	3949	1.00
mu[19]	-0.21	0.00	0.19	-0.58	-0.32	-0.20	-0.09	0.15	3992	1.00
mu[20]	-0.24	0.00	0.17	-0.57	-0.35	-0.23	-0.13	0.08	3808	1.00
mu[21]	-0.26	0.00	0.08	-0.43	-0.32	-0.26	-0.21	-0.10	3922	1.00
mu[22]	-0.27	0.00	0.17	-0.62	-0.38	-0.27	-0.15	0.08	3857	1.00
mu[23]	-0.27	0.00	0.21	-0.67	-0.41	-0.27	-0.14	0.13	3252	1.00
mu[24]	-0.28	0.00	0.22	-0.72	-0.42	-0.28	-0.14	0.14	3315	1.00
mu[25]	-0.28	0.00	0.20	-0.69	-0.41	-0.28	-0.15	0.12	3552	1.00
mu[26]	-0.29	0.00	0.16	-0.60	-0.38	-0.29	-0.19	0.03	4059	1.00
mu[27]	-0.29	0.00	0.01	-0.31	-0.30	-0.29	-0.28	-0.26	6129	1.00
mu[28]	-0.14	0.00	0.12	-0.39	-0.22	-0.14	-0.06	0.09	7368	1.00
mu[29]	0.01	0.00	0.01	-0.02	0.00	0.01	0.02	0.03	6519	1.00
mu[30]	0.01	0.00	0.13	-0.24	-0.07	0.01	0.09	0.25	6499	1.00
mu[31]	0.01	0.00	0.02	-0.03	-0.01	0.01	0.02	0.04	5511	1.00
mu[32]	-0.10	0.00	0.12	-0.35	-0.18	-0.10	-0.02	0.15	6647	1.00
mu[33]	-0.21	0.00	0.02	-0.24	-0.22	-0.21	-0.20	-0.18	5733	1.00
mu[34]	0.07	0.00	0.12	-0.17	-0.01	0.07	0.14	0.30	7651	1.00
mu[35]	0.34	0.00	0.01	0.32	0.33	0.34	0.35	0.37	5743	1.00
mu[36]	0.38	0.00	0.16	0.07	0.28	0.38	0.49	0.71	4758	1.00
mu[37]	0.42	0.00	0.21	0.02	0.29	0.42	0.55	0.84	3417	1.00
mu[38]	0.46	0.00	0.22	0.04	0.32	0.46	0.60	0.90	3203	1.00
mu[39]	0.50	0.00	0.21	0.08	0.36	0.49	0.64	0.94	2847	1.00
mu[40]	0.54	0.00	0.19	0.17	0.41	0.53	0.66	0.92	3071	1.00
mu[41]	0.60	0.00	0.20	0.21	0.46	0.59	0.73	1.01	2593	1.00
mu[42]	0.65	0.00	0.18	0.32	0.53	0.65	0.77	1.04	2261	1.00
mu[43]	0.63	0.00	0.18	0.29	0.51	0.62	0.74	1.00	2903	1.00
mu[44]	0.61	0.00	0.14	0.35	0.51	0.60	0.70	0.90	2815	1.00
mu[45]	0.48	0.00	0.02	0.43	0.46	0.48	0.49	0.53	6509	1.00
mu[46]	0.48	0.00	0.15	0.17	0.38	0.48	0.58	0.78	5182	1.00
mu[47]	0.48	0.00	0.17	0.14	0.37	0.48	0.58	0.80	4426	1.00
mu[48]	0.55	0.00	0.20	0.18	0.42	0.55	0.68	0.94	3563	1.00
mu[49]	0.62	0.00	0.19	0.25	0.49	0.62	0.74	1.02	2922	1.00
mu[50]	0.58	0.00	0.23	0.15	0.43	0.58	0.73	1.04	2654	1.00
mu[51]	0.55	0.00	0.24	0.07	0.39	0.54	0.70	1.05	2685	1.00
mu[52]	0.51	0.00	0.23	0.05	0.36	0.51	0.65	0.98	2621	1.00
mu[53]	0.47	0.00	0.21	0.06	0.34	0.48	0.61	0.88	2946	1.00

```
mu[54]   0.44    0.00  0.15   0.14   0.34  0.44  0.53  0.72  3200 1.00
mu[55]   0.51    0.00  0.03   0.46   0.49  0.51  0.53  0.57  5505 1.00
mu[56]   0.48    0.00  0.17   0.15   0.38  0.48  0.59  0.82  3971 1.00
mu[57]   0.45    0.00  0.22   0.04   0.31  0.45  0.58  0.88  2850 1.00
mu[58]   0.41    0.00  0.24  -0.05   0.26  0.41  0.57  0.89  2844 1.00
mu[59]   0.38    0.00  0.24  -0.10   0.23  0.38  0.53  0.88  2742 1.00
mu[60]   0.35    0.00  0.24  -0.14   0.21  0.35  0.50  0.82  2900 1.00
mu[61]   0.31    0.00  0.22  -0.12   0.18  0.32  0.45  0.75  2993 1.00
mu[62]   0.28    0.00  0.16  -0.05   0.18  0.28  0.39  0.61  3504 1.00
mu[63]   0.25    0.00  0.03   0.18   0.23  0.25  0.27  0.31  5093 1.00
sigma    0.17    0.00  0.04   0.11   0.14  0.17  0.19  0.26   597 1.01
mu_p[1]  0.25    0.00  0.18  -0.10   0.13  0.25  0.37  0.60  3607 1.00
mu_p[2]  0.25    0.00  0.25  -0.26   0.09  0.25  0.41  0.74  3825 1.00
mu_p[3]  0.24    0.00  0.31  -0.37   0.05  0.24  0.44  0.86  3823 1.00
mu_p[4]  0.24    0.01  0.35  -0.46   0.02  0.25  0.47  0.95  3689 1.00
mu_p[5]  0.25    0.01  0.39  -0.55   0.01  0.24  0.49  1.07  3612 1.00
mu_p[6]  0.25    0.01  0.43  -0.63  -0.02  0.24  0.52  1.11  3830 1.00
mu_p[7]  0.24    0.01  0.47  -0.68  -0.05  0.24  0.54  1.18  3922 1.00
mu_p[8]  0.24    0.01  0.50  -0.78  -0.06  0.24  0.55  1.25  3915 1.00
mu_p[9]  0.24    0.01  0.53  -0.83  -0.09  0.25  0.57  1.26  4009 1.00
lp__    -7.51    0.54 11.91 -32.31 -15.27 -6.89  1.14 13.81   491 1.01

Samples were drawn using NUTS(diag_e) at Sun Mar 26 16:20:01 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).
```

```
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)
```

```
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, c
theme_bw()+
labs(title = "Ratio of neonatal to other child mortality (logged), Sri Lanka",
     y = "log ratio", subtitle = "RW1 fit shown in black, projection in red")
```



Question 3

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

```
mod4 <- stan(data = stan_data,
             file = here("code/models/mod10_3.stan"))
```

```
Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
clang -mmacosx-version-min=10.13 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEI
```

```
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.1/Resources/library/StanHea
In file included from /Library/Frameworks/R.framework/Versions/4.1/Resources/library/RcppEig
In file included from /Library/Frameworks/R.framework/Versions/4.1/Resources/library/RcppEig
/Library/Frameworks/R.framework/Versions/4.1/Resources/library/RcppEigen/include/Eigen/src/C
namespace Eigen {
^
/Library/Frameworks/R.framework/Versions/4.1/Resources/library/RcppEigen/include/Eigen/src/C
namespace Eigen {
              ^
              ;
In file included from <built-in>:1:
In file included from /Library/Frameworks/R.framework/Versions/4.1/Resources/library/StanHea
In file included from /Library/Frameworks/R.framework/Versions/4.1/Resources/library/RcppEig
/Library/Frameworks/R.framework/Versions/4.1/Resources/library/RcppEigen/include/Eigen/Core:
#include <complex>
         ^~~~~~~~~
3 errors generated.
make: *** [foo.o] Error 1

SAMPLING FOR MODEL 'mod10_3' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 5.1e-05 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.51 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: 0.725007 seconds (Warm-up)
Chain 1:                0.672799 seconds (Sampling)
Chain 1:                1.39781 seconds (Total)
```

```
Chain 1:

SAMPLING FOR MODEL 'mod10_3' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 1.5e-05 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.15 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: 0.70834 seconds (Warm-up)
Chain 2:                0.686556 seconds (Sampling)
Chain 2:                1.3949 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'mod10_3' NOW (CHAIN 3).
Chain 3:
Chain 3: Gradient evaluation took 1.2e-05 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.12 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: 0.669281 seconds (Warm-up)
Chain 3:                0.70173 seconds (Sampling)
Chain 3:                1.37101 seconds (Total)
Chain 3:

SAMPLING FOR MODEL 'mod10_3' NOW (CHAIN 4).
Chain 4:
Chain 4: Gradient evaluation took 1.2e-05 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.12 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: 0.713976 seconds (Warm-up)
Chain 4:                0.626357 seconds (Sampling)
Chain 4:                1.34033 seconds (Total)
Chain 4:
```

```
mod4
```

```
Inference for Stan model: mod10_3.
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  0.14  -0.85 -0.66 -0.57 -0.48 -0.31  6422 1.00
```

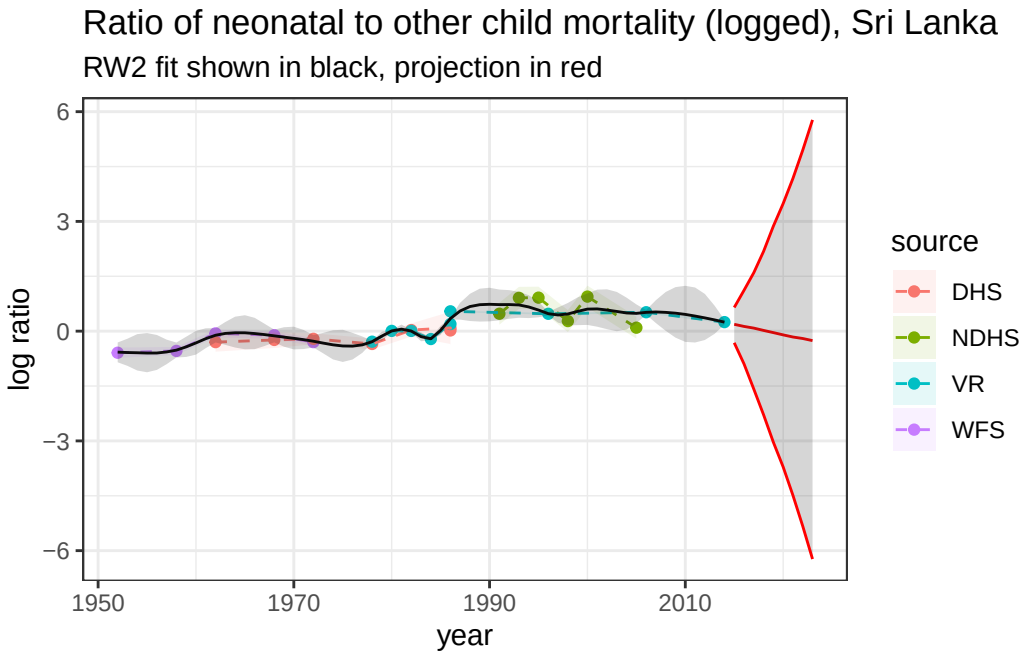
mu[2]	-0.58	0.00	0.19	-0.95	-0.70	-0.58	-0.46	-0.21	2250	1.00
mu[3]	-0.59	0.01	0.26	-1.08	-0.76	-0.59	-0.43	-0.08	1840	1.00
mu[4]	-0.60	0.01	0.27	-1.12	-0.77	-0.60	-0.43	-0.05	1771	1.00
mu[5]	-0.59	0.01	0.24	-1.06	-0.75	-0.60	-0.44	-0.10	1806	1.00
mu[6]	-0.57	0.00	0.17	-0.90	-0.68	-0.57	-0.46	-0.23	2250	1.00
mu[7]	-0.52	0.00	0.11	-0.72	-0.59	-0.52	-0.45	-0.31	3428	1.00
mu[8]	-0.43	0.00	0.14	-0.70	-0.52	-0.43	-0.34	-0.15	3024	1.00
mu[9]	-0.32	0.00	0.16	-0.64	-0.42	-0.32	-0.21	0.01	2714	1.00
mu[10]	-0.20	0.00	0.14	-0.46	-0.30	-0.21	-0.12	0.08	2830	1.00
mu[11]	-0.11	0.00	0.10	-0.31	-0.18	-0.11	-0.05	0.08	3629	1.00
mu[12]	-0.06	0.00	0.16	-0.37	-0.17	-0.06	0.04	0.25	2410	1.00
mu[13]	-0.04	0.00	0.22	-0.45	-0.18	-0.04	0.10	0.40	2001	1.00
mu[14]	-0.04	0.01	0.24	-0.50	-0.20	-0.04	0.10	0.44	1853	1.00
mu[15]	-0.07	0.00	0.22	-0.49	-0.20	-0.07	0.07	0.37	1918	1.00
mu[16]	-0.09	0.00	0.16	-0.41	-0.19	-0.09	0.01	0.20	2328	1.00
mu[17]	-0.13	0.00	0.09	-0.31	-0.19	-0.13	-0.07	0.04	4091	1.00
mu[18]	-0.16	0.00	0.13	-0.42	-0.25	-0.16	-0.08	0.11	3380	1.00
mu[19]	-0.20	0.00	0.16	-0.50	-0.30	-0.20	-0.09	0.12	2800	1.00
mu[20]	-0.23	0.00	0.13	-0.49	-0.32	-0.23	-0.15	0.03	3559	1.00
mu[21]	-0.28	0.00	0.09	-0.45	-0.34	-0.28	-0.22	-0.11	3759	1.00
mu[22]	-0.33	0.00	0.14	-0.61	-0.42	-0.33	-0.23	-0.05	2578	1.00
mu[23]	-0.37	0.00	0.20	-0.77	-0.50	-0.37	-0.25	0.03	2367	1.00
mu[24]	-0.41	0.00	0.21	-0.84	-0.54	-0.40	-0.27	0.04	2339	1.00
mu[25]	-0.41	0.00	0.18	-0.78	-0.52	-0.41	-0.30	-0.05	2584	1.00
mu[26]	-0.37	0.00	0.11	-0.59	-0.45	-0.37	-0.31	-0.15	3201	1.00
mu[27]	-0.29	0.00	0.01	-0.31	-0.30	-0.29	-0.28	-0.26	7480	1.00
mu[28]	-0.14	0.00	0.06	-0.26	-0.17	-0.13	-0.10	-0.01	4927	1.00
mu[29]	0.01	0.00	0.01	-0.02	0.00	0.01	0.02	0.04	8361	1.00
mu[30]	0.06	0.00	0.06	-0.07	0.02	0.05	0.09	0.18	7024	1.00
mu[31]	0.00	0.00	0.02	-0.03	-0.01	0.00	0.01	0.04	6898	1.00
mu[32]	-0.15	0.00	0.06	-0.28	-0.19	-0.15	-0.11	-0.03	6357	1.00
mu[33]	-0.21	0.00	0.02	-0.24	-0.22	-0.21	-0.20	-0.18	5612	1.00
mu[34]	0.02	0.00	0.06	-0.10	-0.02	0.02	0.06	0.15	5798	1.00
mu[35]	0.34	0.00	0.01	0.32	0.33	0.34	0.35	0.37	7974	1.00
mu[36]	0.55	0.00	0.11	0.32	0.48	0.55	0.63	0.78	2421	1.00
mu[37]	0.67	0.00	0.19	0.28	0.55	0.67	0.80	1.02	2074	1.00
mu[38]	0.72	0.01	0.23	0.26	0.57	0.72	0.88	1.16	1850	1.00
mu[39]	0.73	0.01	0.24	0.27	0.58	0.73	0.89	1.19	1942	1.00
mu[40]	0.73	0.00	0.21	0.30	0.59	0.73	0.87	1.14	2142	1.00
mu[41]	0.73	0.00	0.20	0.34	0.60	0.73	0.86	1.12	2519	1.00
mu[42]	0.72	0.00	0.18	0.37	0.60	0.72	0.84	1.08	2629	1.00
mu[43]	0.67	0.00	0.16	0.36	0.56	0.66	0.77	0.98	2536	1.00
mu[44]	0.58	0.00	0.11	0.38	0.52	0.58	0.65	0.81	2567	1.00

```
mu[45]   0.48    0.00  0.02   0.43  0.46  0.48  0.49  0.52  7741 1.00
mu[46]   0.44    0.00  0.11   0.23  0.37  0.44  0.51  0.65  3208 1.00
mu[47]   0.47    0.00  0.15   0.16  0.36  0.47  0.57  0.77  2783 1.00
mu[48]   0.54    0.00  0.19   0.16  0.41  0.54  0.67  0.92  2498 1.00
mu[49]   0.60    0.00  0.22   0.17  0.45  0.60  0.75  1.03  2118 1.00
mu[50]   0.61    0.01  0.26   0.12  0.44  0.61  0.79  1.13  1989 1.00
mu[51]   0.58    0.01  0.28   0.05  0.40  0.58  0.77  1.14  1870 1.00
mu[52]   0.53    0.01  0.27   0.00  0.36  0.54  0.71  1.07  1823 1.00
mu[53]   0.50    0.01  0.22   0.04  0.36  0.50  0.64  0.92  1762 1.00
mu[54]   0.48    0.00  0.13   0.22  0.40  0.49  0.56  0.73  1744 1.00
mu[55]   0.51    0.00  0.03   0.45  0.49  0.51  0.54  0.58  9383 1.00
mu[56]   0.53    0.00  0.15   0.22  0.43  0.53  0.62  0.85  1511 1.00
mu[57]   0.52    0.01  0.27   0.01  0.35  0.51  0.69  1.07  1356 1.00
mu[58]   0.50    0.01  0.35  -0.18  0.27  0.49  0.71  1.20  1317 1.00
mu[59]   0.46    0.01  0.38  -0.29  0.21  0.46  0.69  1.24  1343 1.00
mu[60]   0.42    0.01  0.38  -0.31  0.17  0.41  0.65  1.19  1375 1.00
mu[61]   0.37    0.01  0.31  -0.24  0.17  0.36  0.56  1.02  1469 1.00
mu[62]   0.31    0.00  0.19  -0.07  0.19  0.31  0.42  0.69  1715 1.00
mu[63]   0.25    0.00  0.03   0.18  0.22  0.25  0.27  0.31  7677 1.00
sigma    0.14    0.00  0.03   0.09  0.11  0.13  0.15  0.22   439 1.01
mu_p[1]  0.19    0.01  0.24  -0.32  0.04  0.19  0.34  0.65  1857 1.00
mu_p[2]  0.13    0.01  0.50  -0.90 -0.16  0.13  0.43  1.12  1827 1.00
mu_p[3]  0.07    0.02  0.78  -1.58 -0.39  0.08  0.54  1.60  1868 1.00
mu_p[4]  0.01    0.02  1.09  -2.29 -0.63  0.02  0.68  2.18  1918 1.00
mu_p[5] -0.04    0.03  1.43  -3.04 -0.88 -0.04  0.82  2.87  2001 1.00
mu_p[6] -0.09    0.04  1.78  -3.71 -1.14 -0.10  0.98  3.49  2090 1.00
mu_p[7] -0.14    0.05  2.15  -4.49 -1.44 -0.16  1.17  4.18  2166 1.00
mu_p[8] -0.19    0.05  2.55  -5.33 -1.74 -0.20  1.39  4.95  2219 1.00
mu_p[9] -0.23    0.06  2.96  -6.23 -2.04 -0.26  1.62  5.78  2285 1.00
lp__     3.76    0.64 12.65 -24.14 -3.62  4.44 12.35 26.20   397 1.01

Samples were drawn using NUTS(diag_e) at Sun Mar 26 16:20:32 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).
```

```r
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)
```

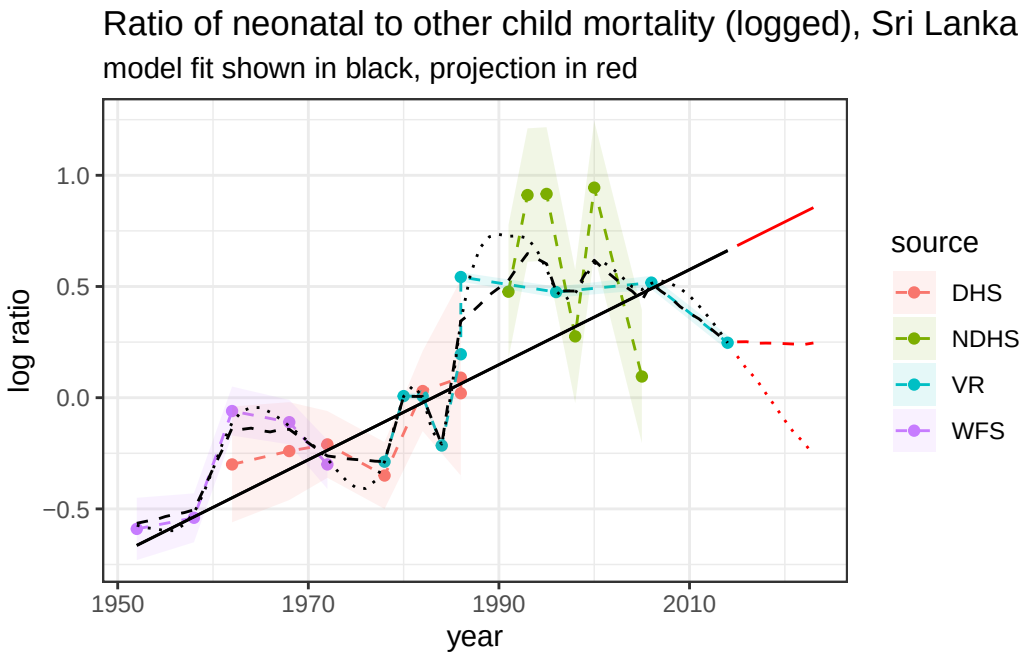
```
ggplot(lka, aes(year, logit_ratio)) +
  geom_point(aes( color = source)) +
  geom_line(aes( color = source), lty = 2) +
  geom_ribbon(aes(ymin = logit_ratio - se,
                  ymax = logit_ratio + se,
                  fill =  source), alpha = 0.1) +
  theme_bw()+
  geom_line(data = res4, aes(year, .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, c
  theme_bw()+
  labs(title = "Ratio of neonatal to other child mortality (logged), Sri Lanka",
       y = "log ratio", subtitle = "RW2 fit shown in black, projection in red")
```



Question 4

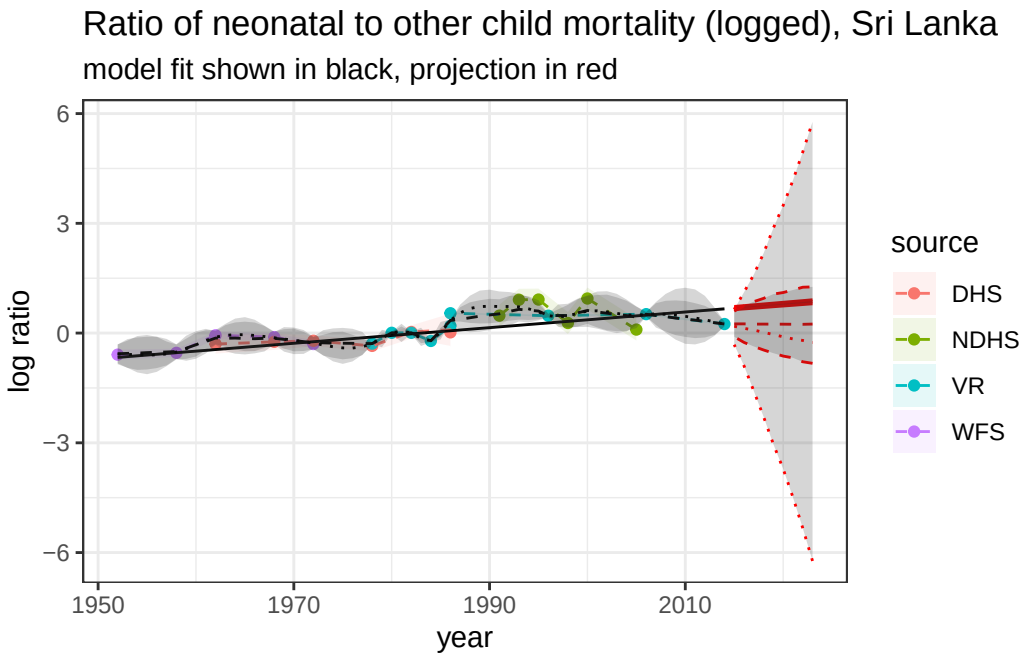
Run the first order and second order random walk models, including projections out to 2023. Compare these estimates with the linear fit by plotting everything on the same graph.

```
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_line(data = res2, aes(year, .value), linetype = 'solid') +
  geom_line(data = res2_p, aes(year, .value), col = 'red', linetype = 'solid') +
    geom_line(data = res3, aes(year, .value), linetype = 'dashed') +
  geom_line(data = res3_p, aes(year, .value), col = 'red', linetype = 'dashed') +
    geom_line(data = res4, aes(year, .value), linetype = 'dotted') +
  geom_line(data = res4_p, aes(year, .value), col = 'red', linetype = 'dotted') +
  theme_bw()+
  labs(title = "Ratio of neonatal to other child mortality (logged), Sri Lanka",
       y = "log ratio", subtitle = "model fit shown in black, projection in red")
```



solid lines: linear fit, dashed lines: RW1, dotted lines: RW2. I see that the linear model does not fit the data well, and its projection seems unplausible. RW1 and RW2 model are better fit than the linear model. RW1 projects a level estimate into 2023, while RW2 projection captures the declining trend at the end of 2010. Also note that RW2 projection has a much wider prediction interval as shown in the plot below.

```
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, c
    geom_line(data = res3, aes(year, .value), linetype = 'dashed') +
  geom_ribbon(data = res3, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.2 , li
    geom_line(data = res3_p, aes(year, .value), col = 'red' , linetype = 'dashed') +
  geom_ribbon(data = res3_p, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.2, c
    geom_line(data = res4, aes(year, .value), linetype = 'dotted') +
  geom_ribbon(data = res4, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.2,line
    geom_line(data = res4_p, aes(year, .value), col = 'red', linetype = 'dotted') +
  geom_ribbon(data = res4_p, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.2, c
  theme_bw()+
  labs(title = "Ratio of neonatal to other child mortality (logged), Sri Lanka",
       y = "log ratio", subtitle = "model fit shown in black, projection in red")
```



Question 5

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

Note that the country_name variable is missing for VR data. Only the VR data captures the ratio in 2014, data from other sources are all before 2006. If we exclude the VR data, our new projection might not capture the latest trend.
VR is the most reliable data source, other sources might have measurement error which could affect the prediction.
Note that VR data is not available in all year, there are some gap between years, we see that removing the VR data, the prediction interval becomes narrower and the fit is smoother.

```
lka_rVR <- lka[lka$source!='VR',]
observed_years <- lka_rVR$year
years <- min(observed_years):max(observed_years)
nyears <- length(years)
```

```
stan_data <- list(y = lka_rVR$logit_ratio, year_i = observed_years - years[1]+1,
                  T = nyears, years = years, N = length(observed_years),
                  mid_year = mean(years), se = lka_rVR$se, P=9)
```

```r
mod5 <- stan(data = stan_data,
             file = here("code/models/mod10_3.stan"))
```

```
SAMPLING FOR MODEL 'mod10_3' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 1.5e-05 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.15 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: 1.93081 seconds (Warm-up)
Chain 1:                3.72437 seconds (Sampling)
Chain 1:                5.65518 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'mod10_3' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 1.3e-05 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.13 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: 2.15017 seconds (Warm-up)
Chain 2:                3.77391 seconds (Sampling)
Chain 2:                5.92408 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'mod10_3' NOW (CHAIN 3).
Chain 3:
Chain 3: Gradient evaluation took 1.1e-05 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.11 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: 2.12975 seconds (Warm-up)
Chain 3:                3.20344 seconds (Sampling)
Chain 3:                5.33319 seconds (Total)
Chain 3:

SAMPLING FOR MODEL 'mod10_3' NOW (CHAIN 4).
Chain 4:
Chain 4: Gradient evaluation took 1.1e-05 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.11 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: 2.01803 seconds (Warm-up)
Chain 4:                1.40477 seconds (Sampling)
Chain 4:                3.4228 seconds (Total)
Chain 4:
```

```
mod5
```

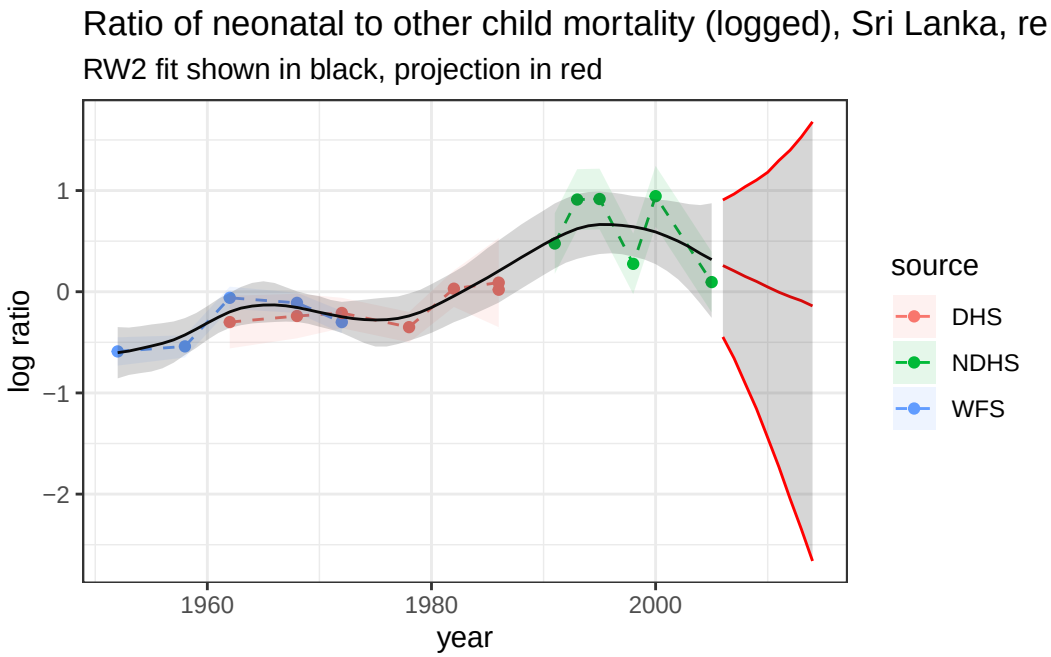
```
Inference for Stan model: mod10_3.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

          mean se_mean    sd  2.5%    25%    50%    75%  97.5% n_eff Rhat
mu[1]    -0.60    0.00  0.13 -0.86  -0.69  -0.60  -0.52  -0.35  3503 1.00
mu[2]    -0.58    0.00  0.12 -0.82  -0.66  -0.59  -0.50  -0.35  2702 1.00
mu[3]    -0.56    0.00  0.12 -0.80  -0.64  -0.56  -0.49  -0.34  1846 1.00
mu[4]    -0.54    0.00  0.12 -0.79  -0.62  -0.54  -0.46  -0.32  1184 1.00
mu[5]    -0.51    0.00  0.11 -0.76  -0.58  -0.51  -0.44  -0.31   818 1.00
mu[6]    -0.48    0.00  0.10 -0.70  -0.54  -0.48  -0.41  -0.29   621 1.01
mu[7]    -0.43    0.00  0.09 -0.63  -0.49  -0.43  -0.37  -0.26   763 1.00
mu[8]    -0.37    0.00  0.09 -0.56  -0.43  -0.37  -0.31  -0.21  2106 1.00
mu[9]    -0.31    0.00  0.09 -0.48  -0.36  -0.31  -0.25  -0.14  3148 1.00
mu[10]   -0.25    0.00  0.09 -0.41  -0.30  -0.25  -0.19  -0.07  1188 1.01
mu[11]   -0.19    0.00  0.09 -0.36  -0.25  -0.20  -0.14  -0.01   455 1.01
mu[12]   -0.16    0.01  0.10 -0.33  -0.22  -0.16  -0.10   0.05   371 1.02
mu[13]   -0.13    0.01  0.10 -0.31  -0.21  -0.14  -0.07   0.09   377 1.02
mu[14]   -0.12    0.01  0.10 -0.31  -0.20  -0.13  -0.06   0.10   420 1.01
mu[15]   -0.13    0.00  0.10 -0.30  -0.19  -0.13  -0.06   0.09   511 1.01
```

mu[16]	-0.14	0.00	0.09	-0.30	-0.19	-0.14	-0.08	0.04	664	1.01
mu[17]	-0.15	0.00	0.08	-0.30	-0.20	-0.16	-0.10	0.00	966	1.01
mu[18]	-0.17	0.00	0.08	-0.32	-0.23	-0.18	-0.12	-0.03	1996	1.00
mu[19]	-0.20	0.00	0.08	-0.35	-0.25	-0.20	-0.15	-0.04	3079	1.00
mu[20]	-0.23	0.00	0.08	-0.37	-0.27	-0.23	-0.18	-0.08	2614	1.00
mu[21]	-0.25	0.00	0.08	-0.41	-0.30	-0.25	-0.20	-0.10	1181	1.00
mu[22]	-0.27	0.00	0.09	-0.46	-0.33	-0.26	-0.21	-0.09	788	1.00
mu[23]	-0.28	0.00	0.11	-0.51	-0.34	-0.28	-0.21	-0.08	742	1.00
mu[24]	-0.28	0.00	0.12	-0.54	-0.35	-0.28	-0.21	-0.07	719	1.00
mu[25]	-0.28	0.00	0.12	-0.54	-0.36	-0.28	-0.20	-0.06	642	1.01
mu[26]	-0.27	0.00	0.12	-0.52	-0.34	-0.26	-0.19	-0.05	587	1.01
mu[27]	-0.24	0.00	0.12	-0.48	-0.32	-0.24	-0.17	-0.02	609	1.01
mu[28]	-0.20	0.00	0.12	-0.45	-0.28	-0.20	-0.12	0.02	744	1.01
mu[29]	-0.16	0.00	0.12	-0.41	-0.23	-0.16	-0.07	0.07	1014	1.01
mu[30]	-0.10	0.00	0.12	-0.36	-0.18	-0.10	-0.02	0.13	1417	1.01
mu[31]	-0.05	0.00	0.13	-0.30	-0.13	-0.04	0.04	0.19	1725	1.01
mu[32]	0.01	0.00	0.13	-0.26	-0.07	0.02	0.10	0.26	1808	1.01
mu[33]	0.07	0.00	0.14	-0.21	-0.02	0.08	0.17	0.34	1797	1.01
mu[34]	0.14	0.00	0.15	-0.16	0.04	0.14	0.23	0.43	1703	1.00
mu[35]	0.20	0.00	0.15	-0.11	0.10	0.21	0.30	0.51	1641	1.00
mu[36]	0.27	0.00	0.16	-0.04	0.17	0.27	0.38	0.58	1548	1.00
mu[37]	0.34	0.00	0.16	0.03	0.23	0.34	0.45	0.66	1403	1.00
mu[38]	0.41	0.00	0.16	0.10	0.30	0.40	0.52	0.73	1191	1.00
mu[39]	0.47	0.01	0.16	0.17	0.36	0.47	0.58	0.80	966	1.00
mu[40]	0.53	0.01	0.16	0.23	0.42	0.53	0.63	0.87	748	1.00
mu[41]	0.59	0.01	0.16	0.29	0.48	0.58	0.69	0.92	577	1.01
mu[42]	0.63	0.01	0.16	0.33	0.52	0.62	0.73	0.97	485	1.01
mu[43]	0.66	0.01	0.16	0.35	0.54	0.65	0.76	0.99	486	1.01
mu[44]	0.67	0.01	0.16	0.37	0.56	0.66	0.77	0.99	542	1.01
mu[45]	0.67	0.01	0.16	0.38	0.56	0.66	0.78	0.98	675	1.00
mu[46]	0.66	0.01	0.15	0.37	0.55	0.66	0.76	0.97	895	1.00
mu[47]	0.64	0.00	0.15	0.35	0.54	0.64	0.74	0.95	1222	1.00
mu[48]	0.62	0.00	0.16	0.32	0.51	0.62	0.72	0.93	1712	1.00
mu[49]	0.59	0.00	0.16	0.27	0.48	0.59	0.70	0.91	1969	1.00
mu[50]	0.55	0.00	0.18	0.21	0.43	0.55	0.67	0.90	2127	1.00
mu[51]	0.50	0.00	0.19	0.12	0.37	0.50	0.63	0.89	1753	1.00
mu[52]	0.44	0.01	0.22	0.01	0.30	0.44	0.58	0.87	1098	1.00
mu[53]	0.38	0.01	0.25	-0.12	0.21	0.38	0.55	0.86	756	1.00
mu[54]	0.31	0.01	0.29	-0.26	0.12	0.32	0.51	0.87	615	1.01
sigma	0.03	0.00	0.02	0.01	0.02	0.03	0.04	0.08	108	1.04
mu_p[1]	0.25	0.01	0.34	-0.45	0.03	0.26	0.49	0.91	579	1.01
mu_p[2]	0.19	0.02	0.41	-0.66	-0.07	0.21	0.47	0.97	595	1.01
mu_p[3]	0.13	0.02	0.49	-0.91	-0.17	0.15	0.47	1.04	635	1.01

```
mu_p[4]   0.07    0.02  0.57 -1.16  -0.28   0.10   0.46   1.10   691 1.00
mu_p[5]   0.01    0.02  0.66 -1.44  -0.39   0.05   0.46   1.18   759 1.00
mu_p[6]  -0.05    0.03  0.76 -1.73  -0.50   0.00   0.46   1.30   816 1.00
mu_p[7]  -0.11    0.03  0.86 -2.05  -0.61  -0.05   0.46   1.40   865 1.00
mu_p[8]  -0.17    0.03  0.97 -2.35  -0.73  -0.09   0.47   1.53   906 1.00
mu_p[9]  -0.23    0.04  1.08 -2.66  -0.85  -0.14   0.48   1.68   953 1.00
lp__    145.73    2.59 24.88 96.71 129.70 145.55 161.70 197.58    93 1.05

Samples were drawn using NUTS(diag_e) at Sun Mar 26 16:20:54 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).
```

```
res5 <- mod5 %>%
  gather_draws(mu[t]) %>%
  median_qi() %>%
  mutate(year = years[t])

res5_p <- mod5 %>%
  gather_draws(mu_p[p]) %>%
  median_qi() %>%
  mutate(year = years[nyears]+p)
```

```
ggplot(lka_rVR, 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 = res5, aes(year, .value)) +
  geom_ribbon(data = res5, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.2)+
    geom_line(data = res5_p, aes(year, .value), col = 'red') +
  geom_ribbon(data = res5_p, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.2, c
  theme_bw()+
  labs(title = "Ratio of neonatal to other child mortality (logged), Sri Lanka, remove VR
       y = "log ratio", subtitle = "RW2 fit shown in black, projection in red")
```



Question 6

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

I think the RW2 model is most appropriate. Although a linear model is usually a good starting point, but we see in this case the trend in child mortality (logged) rate is not linear. The RW1 model is better than the linear model as it captures the change across time, but it assumes current value only depends on the previous value. While a RW2 includes the previous two values of the time series, allowing it to capture more complex trends in the data.