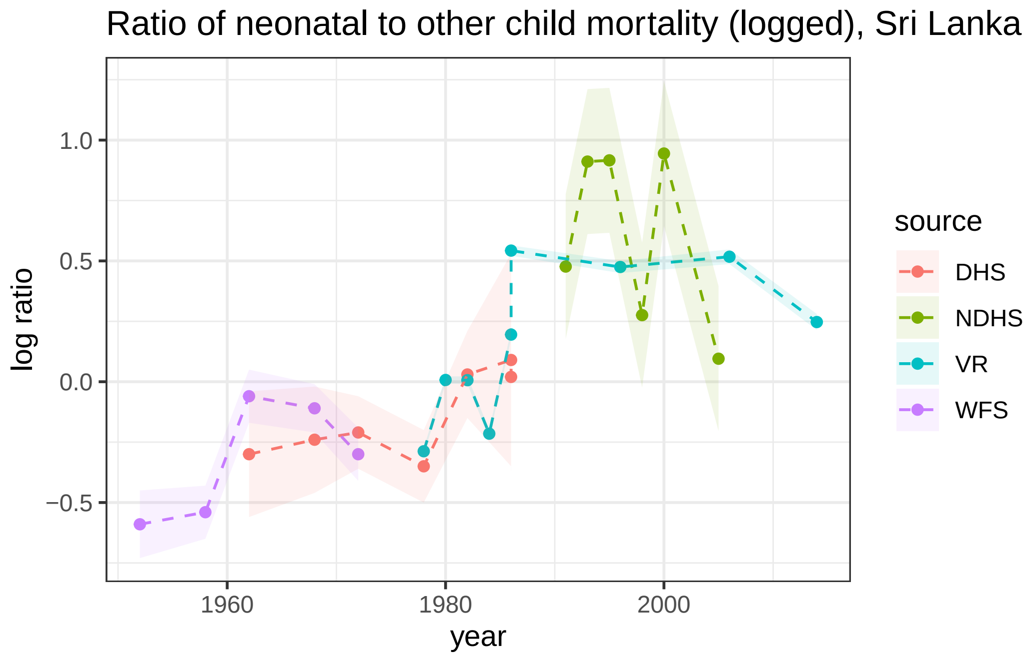
Week 10: Temporal data

24/03/24

Child mortality in Sri Lanka

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

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


lka <- read_csv(here("C:/D-drive/PhD/Year 1/2023 Winter/STA2201/Week11/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("lka_linear_me.stan"))
```

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

```
Chain 1:
Chain 1: Gradient evaluation took 2.7e-05 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.27 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: 0.047 seconds (Warm-up)
Chain 1:                0.044 seconds (Sampling)
Chain 1:                0.091 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 6e-06 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.06 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: 0.05 seconds (Warm-up)
Chain 2:                0.04 seconds (Sampling)
Chain 2:                0.09 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
Chain 3:
Chain 3: Gradient evaluation took 6e-06 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.06 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: 0.048 seconds (Warm-up)
Chain 3:                0.047 seconds (Sampling)
Chain 3:                0.095 seconds (Total)
Chain 3:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
Chain 4:
Chain 4: Gradient evaluation took 8e-06 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.08 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: 0.048 seconds (Warm-up)
Chain 4:                0.04 seconds (Sampling)
Chain 4:                0.088 seconds (Total)
Chain 4: 
```

Extract the results:

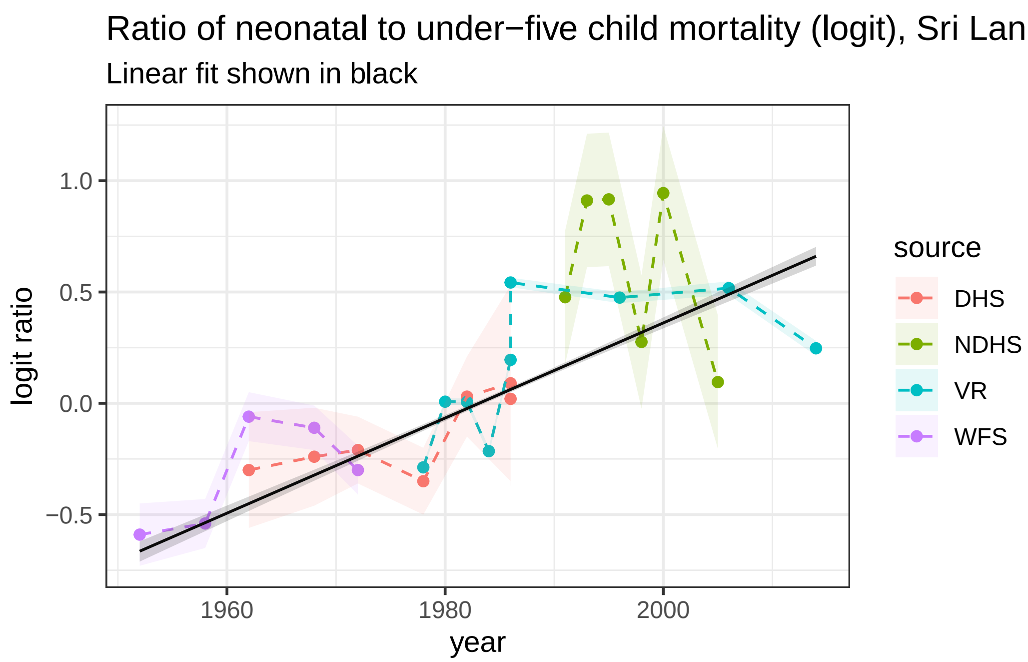
```
res <- mod %>%
  gather_draws(mu[t]) %>%
  median_qi() %>%
  mutate(year = years[t])
res
```

```
# A tibble: 63 x 9
       t .variable .value .lower .upper .width .point .interval  year
   <int> <chr>      <dbl>  <dbl>  <dbl>  <dbl> <chr>  <chr>     <int>
 1     1 mu        -0.665 -0.711 -0.620   0.95 median qi         1952
 2     2 mu        -0.643 -0.688 -0.600   0.95 median qi         1953
 3     3 mu        -0.622 -0.666 -0.580   0.95 median qi         1954
 4     4 mu        -0.600 -0.643 -0.560   0.95 median qi         1955
 5     5 mu        -0.579 -0.620 -0.540   0.95 median qi         1956
 6     6 mu        -0.558 -0.597 -0.520   0.95 median qi         1957
 7     7 mu        -0.536 -0.575 -0.500   0.95 median qi         1958
 8     8 mu        -0.515 -0.552 -0.480   0.95 median qi         1959
 9     9 mu        -0.494 -0.529 -0.459   0.95 median qi         1960
10    10 mu        -0.472 -0.506 -0.439   0.95 median qi         1961
# i 53 more rows
```

Plot the results:

```
ggplot(lka, aes(year, logit_ratio)) +
  geom_point(aes( color = source)) +
  geom_line(aes( color = source), lty = 2) +
```

```
  geom_ribbon(aes(ymin = logit_ratio - se,
                  ymax = logit_ratio + se,
                  fill =  source), alpha = 0.1) +
  theme_bw()+
  geom_line(data = res, aes(year, .value)) +
  geom_ribbon(data = res, aes(y = .value, ymin = .lower, ymax = .upper), alpha = 0.2)+
  theme_bw()+
  labs(title = "Ratio of neonatal to under-five child mortality (logit), Sri Lanka",
       y = "logit ratio", subtitle = "Linear fit shown in black")
```



Question 1

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

```
nobservations <- nrow(lka)
mid_year <- mean(years)
projection_years <- max(years):2022
nprojection <- length(projection_years)
```

```
stan_data_proj2022 <- list(
  y = lka$logit_ratio,
  year_i = match(lka$year, years),
  T = nyears,
  years = years,
  N = nobservations,
  mid_year = mid_year,
  se = lka$se,
  P = nprojection
  )


# Compile and fit the Stan model
mod_proj2022 <- stan(file = "C:/D-drive/PhD/Year 1/2023 Winter/STA2201/Week11/lka_linear_m
            data = stan_data_proj2022,
            iter = 4000,
            chains = 4)
```

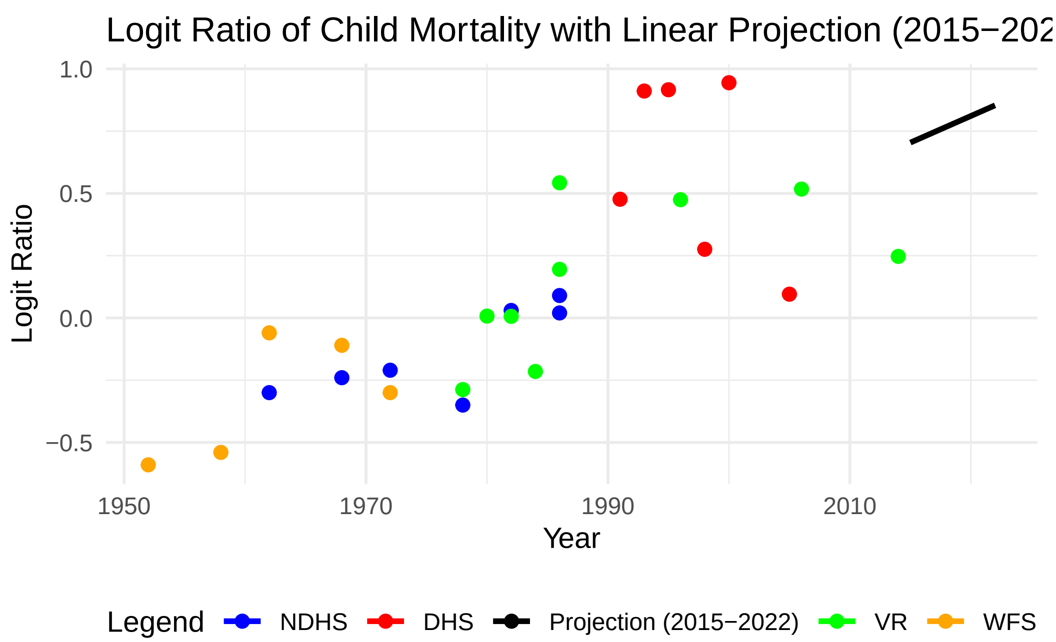
```
SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 3e-05 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.3 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration:    1 / 4000 [  0%]  (Warmup)
Chain 1: Iteration:  400 / 4000 [ 10%]  (Warmup)
Chain 1: Iteration:  800 / 4000 [ 20%]  (Warmup)
Chain 1: Iteration: 1200 / 4000 [ 30%]  (Warmup)
Chain 1: Iteration: 1600 / 4000 [ 40%]  (Warmup)
Chain 1: Iteration: 2000 / 4000 [ 50%]  (Warmup)
Chain 1: Iteration: 2001 / 4000 [ 50%]  (Sampling)
Chain 1: Iteration: 2400 / 4000 [ 60%]  (Sampling)
Chain 1: Iteration: 2800 / 4000 [ 70%]  (Sampling)
Chain 1: Iteration: 3200 / 4000 [ 80%]  (Sampling)
Chain 1: Iteration: 3600 / 4000 [ 90%]  (Sampling)
Chain 1: Iteration: 4000 / 4000 [100%]  (Sampling)
Chain 1:
Chain 1:  Elapsed Time: 0.157 seconds (Warm-up)
Chain 1:                0.106 seconds (Sampling)
Chain 1:                0.263 seconds (Total)
```

```
Chain 1:

SAMPLING FOR MODEL 'anon_model' 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 / 4000 [  0%]  (Warmup)
Chain 2: Iteration:  400 / 4000 [ 10%]  (Warmup)
Chain 2: Iteration:  800 / 4000 [ 20%]  (Warmup)
Chain 2: Iteration: 1200 / 4000 [ 30%]  (Warmup)
Chain 2: Iteration: 1600 / 4000 [ 40%]  (Warmup)
Chain 2: Iteration: 2000 / 4000 [ 50%]  (Warmup)
Chain 2: Iteration: 2001 / 4000 [ 50%]  (Sampling)
Chain 2: Iteration: 2400 / 4000 [ 60%]  (Sampling)
Chain 2: Iteration: 2800 / 4000 [ 70%]  (Sampling)
Chain 2: Iteration: 3200 / 4000 [ 80%]  (Sampling)
Chain 2: Iteration: 3600 / 4000 [ 90%]  (Sampling)
Chain 2: Iteration: 4000 / 4000 [100%]  (Sampling)
Chain 2:
Chain 2:  Elapsed Time: 0.26 seconds (Warm-up)
Chain 2:                0.094 seconds (Sampling)
Chain 2:                0.354 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'anon_model' 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 / 4000 [  0%]  (Warmup)
Chain 3: Iteration:  400 / 4000 [ 10%]  (Warmup)
Chain 3: Iteration:  800 / 4000 [ 20%]  (Warmup)
Chain 3: Iteration: 1200 / 4000 [ 30%]  (Warmup)
Chain 3: Iteration: 1600 / 4000 [ 40%]  (Warmup)
Chain 3: Iteration: 2000 / 4000 [ 50%]  (Warmup)
Chain 3: Iteration: 2001 / 4000 [ 50%]  (Sampling)
Chain 3: Iteration: 2400 / 4000 [ 60%]  (Sampling)
Chain 3: Iteration: 2800 / 4000 [ 70%]  (Sampling)
```

```
Chain 3: Iteration: 3200 / 4000 [ 80%]  (Sampling)
Chain 3: Iteration: 3600 / 4000 [ 90%]  (Sampling)
Chain 3: Iteration: 4000 / 4000 [100%]  (Sampling)
Chain 3:
Chain 3:  Elapsed Time: 0.258 seconds (Warm-up)
Chain 3:                0.089 seconds (Sampling)
Chain 3:                0.347 seconds (Total)
Chain 3:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
Chain 4:
Chain 4: Gradient evaluation took 1.6e-05 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.16 seconds.
Chain 4: Adjust your expectations accordingly!
Chain 4:
Chain 4:
Chain 4: Iteration:    1 / 4000 [  0%]  (Warmup)
Chain 4: Iteration:  400 / 4000 [ 10%]  (Warmup)
Chain 4: Iteration:  800 / 4000 [ 20%]  (Warmup)
Chain 4: Iteration: 1200 / 4000 [ 30%]  (Warmup)
Chain 4: Iteration: 1600 / 4000 [ 40%]  (Warmup)
Chain 4: Iteration: 2000 / 4000 [ 50%]  (Warmup)
Chain 4: Iteration: 2001 / 4000 [ 50%]  (Sampling)
Chain 4: Iteration: 2400 / 4000 [ 60%]  (Sampling)
Chain 4: Iteration: 2800 / 4000 [ 70%]  (Sampling)
Chain 4: Iteration: 3200 / 4000 [ 80%]  (Sampling)
Chain 4: Iteration: 3600 / 4000 [ 90%]  (Sampling)
Chain 4: Iteration: 4000 / 4000 [100%]  (Sampling)
Chain 4:
Chain 4:  Elapsed Time: 0.275 seconds (Warm-up)
Chain 4:                0.091 seconds (Sampling)
Chain 4:                0.366 seconds (Total)
Chain 4:
```

```r
# Extract the projections
fit <- extract(mod_proj2022)
mu_projected <- apply(fit$mu_projected, 2, median)

# Prepare data for plotting
projection_data <- data.frame(
  year = c(years, projection_years),
  mu_projected = mu_projected
```

```
)

# Plot the observed data and projections for 2015-2022 with a legend for the projection
ggplot() +
  geom_point(data = lka, aes(x = year, y = logit_ratio, color = source), size = 2) +
  geom_line(data = filter(projection_data, year >= 2015 & year <= 2022),
            aes(x = year, y = mu_projected, color = "Projection"), size = 1) +
  scale_color_manual(values = c("Projection" = "black", "DHS" = "blue", "NDHS" = "red", "V
                     name = "Legend",
                     labels = c("NDHS", "DHS", "Projection (2015-2022)", "VR", "WFS")) +
  labs(title = "Logit Ratio of Child Mortality with Linear Projection (2015-2022)",
       x = "Year", y = "Logit Ratio") +
  theme_minimal() +
  theme(legend.position = "bottom")
```



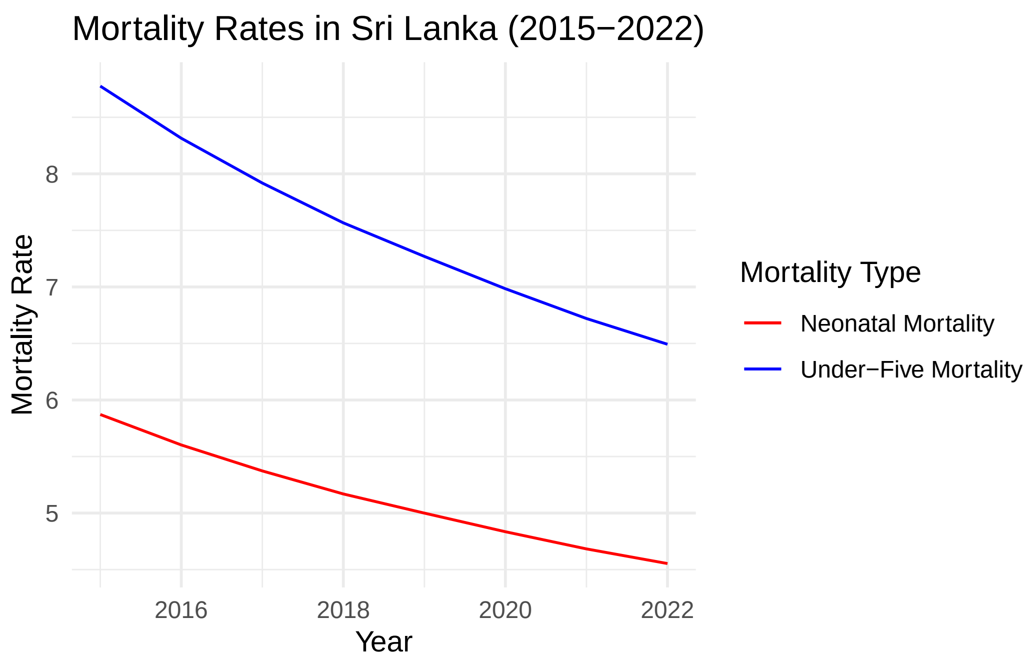
Question 2

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

```r
# Read the downloaded mortality rate data
under_five_mortality <- read_csv("C:/D-drive/PhD/Year 1/2023 Winter/STA2201/Week11/LKA-Und

under_five_mortality <- under_five_mortality |>
  select(Year, Estimate) |>
  mutate(MortalityRate = as.numeric(Estimate))
```

```r
proportion_projected <- 1 / (1 + exp(-mu_projected[72:65]))

# Calculate neonatal mortality using the under-five mortality rates
# This is a placeholder operation; adjust the calculation as per your actual requirements
neonatal_mortality <- under_five_mortality |>
  filter(Year >= 2015 & Year <= 2022) |>
  mutate(NeonatalMortality = MortalityRate * proportion_projected)
```

```r
# Combine data for plotting
combined_data <- data.frame(
  Year = neonatal_mortality$Year,
  UnderFiveMortality = neonatal_mortality$MortalityRate,
  NeonatalMortality = neonatal_mortality$NeonatalMortality
)

# Plot
ggplot(combined_data, aes(x = Year)) +
  geom_line(aes(y = UnderFiveMortality, color = "Under-Five Mortality")) +
  geom_line(aes(y = NeonatalMortality, color = "Neonatal Mortality")) +
  labs(title = "Mortality Rates in Sri Lanka (2015-2022)",
       y = "Mortality Rate", x = "Year") +
  scale_color_manual(values = c("Under-Five Mortality" = "blue", "Neonatal Mortality" = "r
  theme_minimal() +
  guides(color = guide_legend(title = "Mortality Type"))
```



Random walks

Question 3

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

```
# Prepare the data for Stan
N <- nrow(lka)
y <- lka$logit_ratio
se <- lka$se
P <- 2022 - max(lka$year)

stan_data_rw <- list(N = N,
                     y = y,
                     se = se,
                     P = P)

# Fit the random walk model
fit <- stan(file = 'C:/D-drive/PhD/Year 1/2023 Winter/STA2201/Week11/lka_rw.stan',
            data = stan_data_rw,
```

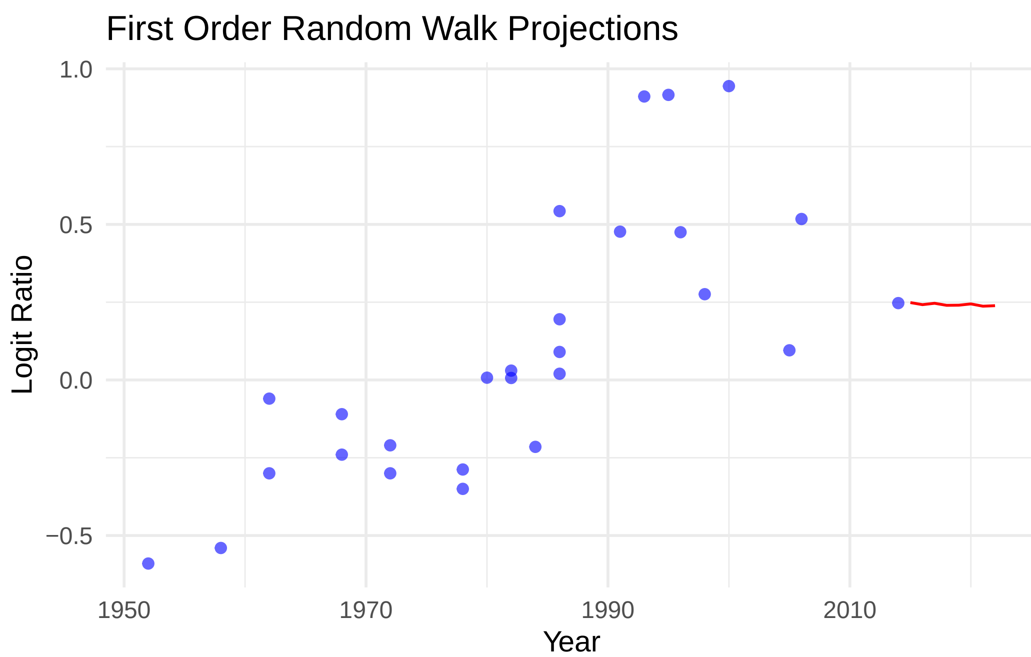
```r
              iter = 4000,
              chains = 4)
```

```
SAMPLING FOR MODEL 'anon_model' 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 / 4000 [  0%]  (Warmup)
Chain 1: Iteration:  400 / 4000 [ 10%]  (Warmup)
Chain 1: Iteration:  800 / 4000 [ 20%]  (Warmup)
Chain 1: Iteration: 1200 / 4000 [ 30%]  (Warmup)
Chain 1: Iteration: 1600 / 4000 [ 40%]  (Warmup)
Chain 1: Iteration: 2000 / 4000 [ 50%]  (Warmup)
Chain 1: Iteration: 2001 / 4000 [ 50%]  (Sampling)
Chain 1: Iteration: 2400 / 4000 [ 60%]  (Sampling)
Chain 1: Iteration: 2800 / 4000 [ 70%]  (Sampling)
Chain 1: Iteration: 3200 / 4000 [ 80%]  (Sampling)
Chain 1: Iteration: 3600 / 4000 [ 90%]  (Sampling)
Chain 1: Iteration: 4000 / 4000 [100%]  (Sampling)
Chain 1:
Chain 1:  Elapsed Time: 0.16 seconds (Warm-up)
Chain 1:                0.176 seconds (Sampling)
Chain 1:                0.336 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'anon_model' 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 / 4000 [  0%]  (Warmup)
Chain 2: Iteration:  400 / 4000 [ 10%]  (Warmup)
Chain 2: Iteration:  800 / 4000 [ 20%]  (Warmup)
Chain 2: Iteration: 1200 / 4000 [ 30%]  (Warmup)
Chain 2: Iteration: 1600 / 4000 [ 40%]  (Warmup)
Chain 2: Iteration: 2000 / 4000 [ 50%]  (Warmup)
```

```
Chain 2: Iteration: 2001 / 4000 [ 50%]  (Sampling)
Chain 2: Iteration: 2400 / 4000 [ 60%]  (Sampling)
Chain 2: Iteration: 2800 / 4000 [ 70%]  (Sampling)
Chain 2: Iteration: 3200 / 4000 [ 80%]  (Sampling)
Chain 2: Iteration: 3600 / 4000 [ 90%]  (Sampling)
Chain 2: Iteration: 4000 / 4000 [100%]  (Sampling)
Chain 2:
Chain 2:  Elapsed Time: 0.168 seconds (Warm-up)
Chain 2:                0.145 seconds (Sampling)
Chain 2:                0.313 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
Chain 3:
Chain 3: Gradient evaluation took 1e-05 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.1 seconds.
Chain 3: Adjust your expectations accordingly!
Chain 3:
Chain 3:
Chain 3: Iteration:    1 / 4000 [  0%]  (Warmup)
Chain 3: Iteration:  400 / 4000 [ 10%]  (Warmup)
Chain 3: Iteration:  800 / 4000 [ 20%]  (Warmup)
Chain 3: Iteration: 1200 / 4000 [ 30%]  (Warmup)
Chain 3: Iteration: 1600 / 4000 [ 40%]  (Warmup)
Chain 3: Iteration: 2000 / 4000 [ 50%]  (Warmup)
Chain 3: Iteration: 2001 / 4000 [ 50%]  (Sampling)
Chain 3: Iteration: 2400 / 4000 [ 60%]  (Sampling)
Chain 3: Iteration: 2800 / 4000 [ 70%]  (Sampling)
Chain 3: Iteration: 3200 / 4000 [ 80%]  (Sampling)
Chain 3: Iteration: 3600 / 4000 [ 90%]  (Sampling)
Chain 3: Iteration: 4000 / 4000 [100%]  (Sampling)
Chain 3:
Chain 3:  Elapsed Time: 0.169 seconds (Warm-up)
Chain 3:                0.158 seconds (Sampling)
Chain 3:                0.327 seconds (Total)
Chain 3:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
Chain 4:
Chain 4: Gradient evaluation took 8e-06 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.08 seconds.
Chain 4: Adjust your expectations accordingly!
Chain 4:
```

```
Chain 4:
Chain 4: Iteration:    1 / 4000 [  0%]  (Warmup)
Chain 4: Iteration:  400 / 4000 [ 10%]  (Warmup)
Chain 4: Iteration:  800 / 4000 [ 20%]  (Warmup)
Chain 4: Iteration: 1200 / 4000 [ 30%]  (Warmup)
Chain 4: Iteration: 1600 / 4000 [ 40%]  (Warmup)
Chain 4: Iteration: 2000 / 4000 [ 50%]  (Warmup)
Chain 4: Iteration: 2001 / 4000 [ 50%]  (Sampling)
Chain 4: Iteration: 2400 / 4000 [ 60%]  (Sampling)
Chain 4: Iteration: 2800 / 4000 [ 70%]  (Sampling)
Chain 4: Iteration: 3200 / 4000 [ 80%]  (Sampling)
Chain 4: Iteration: 3600 / 4000 [ 90%]  (Sampling)
Chain 4: Iteration: 4000 / 4000 [100%]  (Sampling)
Chain 4:
Chain 4:  Elapsed Time: 0.149 seconds (Warm-up)
Chain 4:                0.111 seconds (Sampling)
Chain 4:                0.26 seconds (Total)
Chain 4:
```

```
# Extract projections
projections <- extract(fit)$mu_projected

# Prepare data for plotting
years <- seq(max(lka$year)+1, 2022)
projection_means <- apply(projections, 2, mean)[(N+1):(N+P)]

plot_data <- data.frame(year = years,
                        mu_projected = projection_means)

# Plot the results
ggplot() +
  geom_point(data = lka, aes(x = year, y = logit_ratio), color = 'blue', alpha = 0.6) +
  geom_line(data = plot_data, aes(x = year, y = mu_projected), color = 'red') +
  labs(title = "First Order Random Walk Projections",
       x = "Year", y = "Logit Ratio") +
  theme_minimal()
```



Question 4

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

```
# Fit the second-order random walk model
fit_rw2 <- stan(file = 'C:/D-drive/PhD/Year 1/2023 Winter/STA2201/Week11/lka_rw_second.sta
                data = stan_data_rw,
                iter = 4000,
                chains = 4)
```

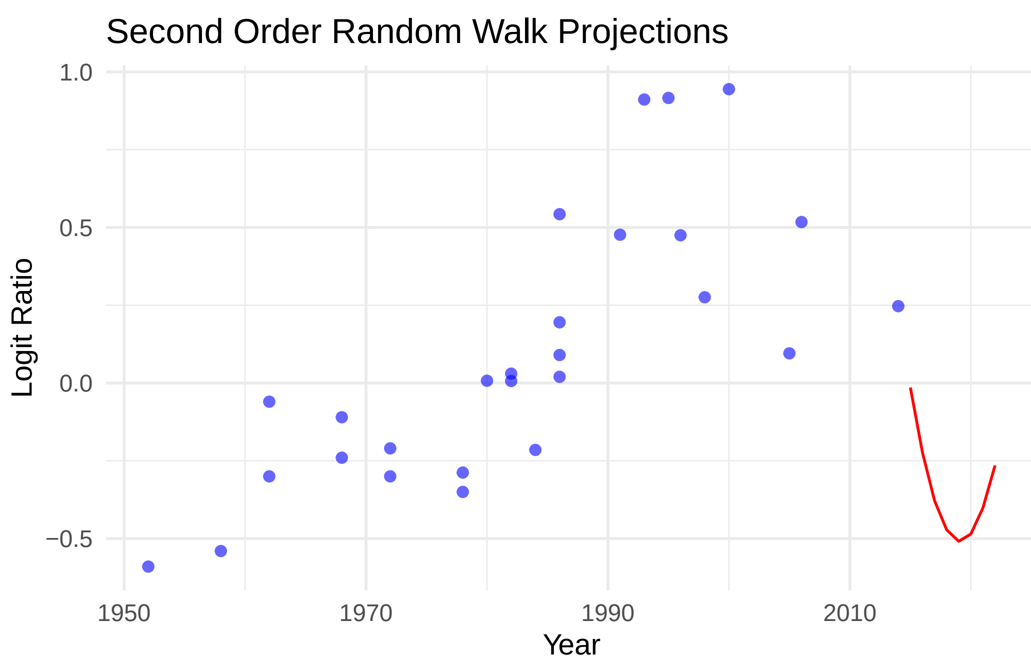
```
SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 6.5e-05 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.65 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration:    1 / 4000 [  0%]  (Warmup)
Chain 1: Iteration:  400 / 4000 [ 10%]  (Warmup)
Chain 1: Iteration:  800 / 4000 [ 20%]  (Warmup)
```

```
Chain 1: Iteration: 1200 / 4000 [ 30%]  (Warmup)
Chain 1: Iteration: 1600 / 4000 [ 40%]  (Warmup)
Chain 1: Iteration: 2000 / 4000 [ 50%]  (Warmup)
Chain 1: Iteration: 2001 / 4000 [ 50%]  (Sampling)
Chain 1: Iteration: 2400 / 4000 [ 60%]  (Sampling)
Chain 1: Iteration: 2800 / 4000 [ 70%]  (Sampling)
Chain 1: Iteration: 3200 / 4000 [ 80%]  (Sampling)
Chain 1: Iteration: 3600 / 4000 [ 90%]  (Sampling)
Chain 1: Iteration: 4000 / 4000 [100%]  (Sampling)
Chain 1:
Chain 1:  Elapsed Time: 0.26 seconds (Warm-up)
Chain 1:                0.24 seconds (Sampling)
Chain 1:                0.5 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 9e-06 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.09 seconds.
Chain 2: Adjust your expectations accordingly!
Chain 2:
Chain 2:
Chain 2: Iteration:    1 / 4000 [  0%]  (Warmup)
Chain 2: Iteration:  400 / 4000 [ 10%]  (Warmup)
Chain 2: Iteration:  800 / 4000 [ 20%]  (Warmup)
Chain 2: Iteration: 1200 / 4000 [ 30%]  (Warmup)
Chain 2: Iteration: 1600 / 4000 [ 40%]  (Warmup)
Chain 2: Iteration: 2000 / 4000 [ 50%]  (Warmup)
Chain 2: Iteration: 2001 / 4000 [ 50%]  (Sampling)
Chain 2: Iteration: 2400 / 4000 [ 60%]  (Sampling)
Chain 2: Iteration: 2800 / 4000 [ 70%]  (Sampling)
Chain 2: Iteration: 3200 / 4000 [ 80%]  (Sampling)
Chain 2: Iteration: 3600 / 4000 [ 90%]  (Sampling)
Chain 2: Iteration: 4000 / 4000 [100%]  (Sampling)
Chain 2:
Chain 2:  Elapsed Time: 0.267 seconds (Warm-up)
Chain 2:                0.242 seconds (Sampling)
Chain 2:                0.509 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'anon_model' 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 / 4000 [  0%]  (Warmup)
Chain 3: Iteration:  400 / 4000 [ 10%]  (Warmup)
Chain 3: Iteration:  800 / 4000 [ 20%]  (Warmup)
Chain 3: Iteration: 1200 / 4000 [ 30%]  (Warmup)
Chain 3: Iteration: 1600 / 4000 [ 40%]  (Warmup)
Chain 3: Iteration: 2000 / 4000 [ 50%]  (Warmup)
Chain 3: Iteration: 2001 / 4000 [ 50%]  (Sampling)
Chain 3: Iteration: 2400 / 4000 [ 60%]  (Sampling)
Chain 3: Iteration: 2800 / 4000 [ 70%]  (Sampling)
Chain 3: Iteration: 3200 / 4000 [ 80%]  (Sampling)
Chain 3: Iteration: 3600 / 4000 [ 90%]  (Sampling)
Chain 3: Iteration: 4000 / 4000 [100%]  (Sampling)
Chain 3:
Chain 3:  Elapsed Time: 0.249 seconds (Warm-up)
Chain 3:                0.232 seconds (Sampling)
Chain 3:                0.481 seconds (Total)
Chain 3:

SAMPLING FOR MODEL 'anon_model' 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 / 4000 [  0%]  (Warmup)
Chain 4: Iteration:  400 / 4000 [ 10%]  (Warmup)
Chain 4: Iteration:  800 / 4000 [ 20%]  (Warmup)
Chain 4: Iteration: 1200 / 4000 [ 30%]  (Warmup)
Chain 4: Iteration: 1600 / 4000 [ 40%]  (Warmup)
Chain 4: Iteration: 2000 / 4000 [ 50%]  (Warmup)
Chain 4: Iteration: 2001 / 4000 [ 50%]  (Sampling)
Chain 4: Iteration: 2400 / 4000 [ 60%]  (Sampling)
Chain 4: Iteration: 2800 / 4000 [ 70%]  (Sampling)
Chain 4: Iteration: 3200 / 4000 [ 80%]  (Sampling)
Chain 4: Iteration: 3600 / 4000 [ 90%]  (Sampling)
Chain 4: Iteration: 4000 / 4000 [100%]  (Sampling)
Chain 4:
Chain 4:  Elapsed Time: 0.272 seconds (Warm-up)
```

```
Chain 4:                0.241 seconds (Sampling)
Chain 4:                0.513 seconds (Total)
Chain 4:
```

```r
# Extract projections for the second-order model
projections_rw2 <- extract(fit_rw2)$mu_projected

# Prepare data for plotting - ensure projection_years covers the correct range
projection_years <- seq(max(lka$year)+1, 2022)
projection_means_rw2 <- apply(projections_rw2, 2, mean)[(N+1):(N+P)]

plot_data_rw2 <- data.frame(year = projection_years,
                            mu_projected = projection_means_rw2)

# Plot the results with the second-order random walk model
ggplot() +
  geom_point(data = lka, aes(x = year, y = logit_ratio), color = 'blue', alpha = 0.6) +
  geom_line(data = plot_data_rw2, aes(x = year, y = mu_projected), color = 'red') +
  labs(title = "Second Order Random Walk Projections",
       x = "Year", y = "Logit Ratio") +
  theme_minimal()
```

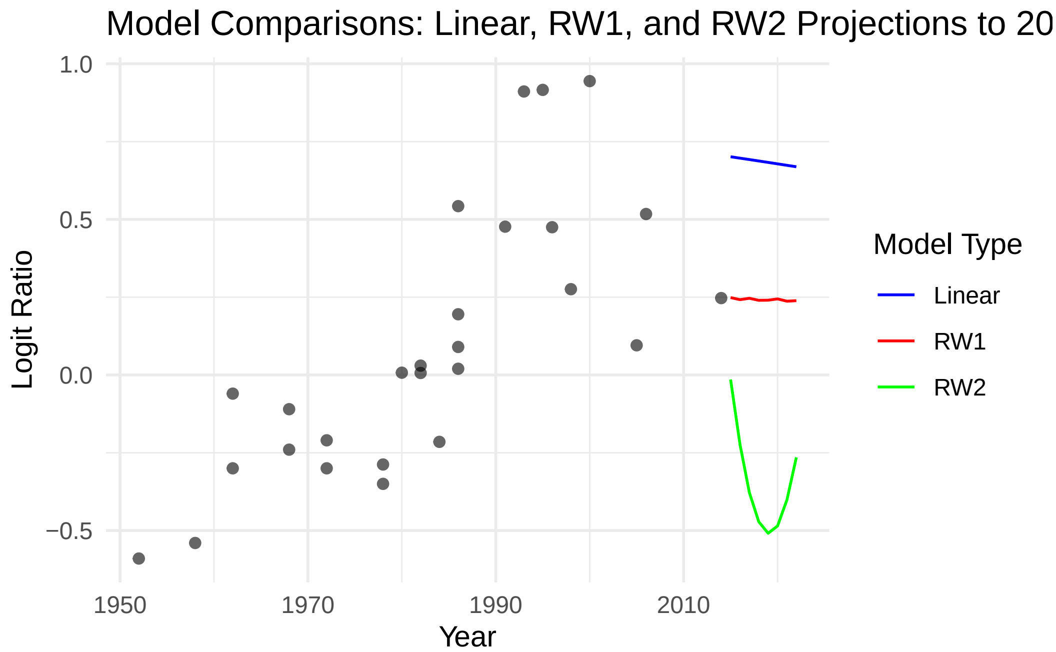


Question 5

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

```
# Assuming projection_means, projection_means_rw2, and projection_linear_means are already
# And projection_years contains the years from the end of your dataset to 2022
projection_linear_means <- combined_data$NeonatalMortality/combined_data$UnderFiveMortalit

# Preparing for plotting
plot_data <- data.frame(
  year = rep(projection_years, 3),
  mu_projected = c(projection_linear_means, projection_means, projection_means_rw2),
  model = factor(rep(c("Linear", "RW1", "RW2"), each = length(projection_years)))
)
```

```
ggplot() +
  geom_point(data = lka, aes(x = year, y = logit_ratio), color = 'black', alpha = 0.6) +
  geom_line(data = plot_data, aes(x = year, y = mu_projected, color = model)) +
  scale_color_manual(values = c("Linear" = "blue", "RW1" = "red", "RW2" = "green")) +
  labs(title = "Model Comparisons: Linear, RW1, and RW2 Projections to 2022",
       x = "Year", y = "Logit Ratio", color = "Model Type") +
  theme_minimal()
```



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 first-order random walk (RW1) model, with its modest variation, appears to strike a balance between the stable yet potentially oversimplistic linear trend and the highly volatile second-order random walk (RW2). For projecting child mortality rates, RW1's flexibility without overfitting suggests it may be the most prudent choice.