Exploratory Data Analysis

John

Libraries and setup

```
library(here)
library(dplyr)
library(ggplot2)
library(ggforce)
source(here("R", "create_finaldata.R"))
```

Get the column variable names. Then call head() and tail() functions to take a quick look at the dataset.

```
names(finaldata)
```

```
 [1] "country_name" "ISO"          "region"       "year"         "gdp1000"
 [6] "OECD"         "OECD2023"     "popdens"      "urban"        "agedep"
[11] "male_edu"     "temp"         "rainfall1000" "matmor"       "infmor"
[16] "neomor"       "un5mor"       "earthquake"   "drought"      "totdeath"
[21] "armconf1"
```

```
head(finaldata, 10)
```

```
   country_name ISO        region year   gdp1000 OECD OECD2023  popdens
1   Afghanistan AFG Southern Asia 2000        NA    0        0 14.13654
2   Afghanistan AFG Southern Asia 2001        NA    0        0 14.23156
3   Afghanistan AFG Southern Asia 2002 0.1835328    0        0 14.32270
4   Afghanistan AFG Southern Asia 2003 0.2004626    0        0 14.40691
5   Afghanistan AFG Southern Asia 2004 0.2216576    0        0 15.21947
6   Afghanistan AFG Southern Asia 2005 0.2550551    0        0 15.33619
```

```
7   Afghanistan AFG Southern Asia 2006 0.2740005    0        0 15.43982
8   Afghanistan AFG Southern Asia 2007 0.3750781    0        0 15.65217
9   Afghanistan AFG Southern Asia 2008 0.3878492    0        0 15.74447
10  Afghanistan AFG Southern Asia 2009 0.4438452    0        0 15.83043
      urban   agedep male_edu     temp rainfall1000 matmor infmor neomor un5mor
1  16.25324 108.3466 2.762086 12.69959    0.2763704   1450   90.5   60.9  129.2
2  16.25661 108.9899 2.856936 12.85570    0.2793079   1390   87.9   59.7  125.2
3  16.42654 109.3472 2.954241 12.71081    0.3805710   1300   85.3   58.5  121.1
4  16.60701 109.4475 3.054121 12.16592    0.4288939   1240   82.7   57.2  116.9
5  16.71367 109.2868 3.156706 13.04643    0.3754336   1180   80.0   55.9  112.6
6  16.85096 107.9646 3.262133 12.23141    0.4415680   1140   77.3   54.6  108.4
7  16.98105 106.3262 3.370551 12.96153    0.4437097   1120   74.6   53.2  104.1
8  17.12259 108.3381 3.482112 12.47451    0.4092555   1090   71.9   51.7   99.9
9  17.26919 109.2404 3.596977 12.63527    0.3901204   1030   69.2   50.3   95.7
10 17.43508 106.8458 3.715306 12.61764    0.4808727    993   66.7   48.9   91.7
   earthquake drought totdeath armconf1
1           1       0     5065        1
2           0       1     5394        1
3           0       1     5553        1
4           0       1     1157        1
5           0       1      944        1
6           0       1      817        1
7           1       1     1711        1
8           0       0     4982        1
9           1       0     7020        1
10          0       1     5660        1
```

```
tail(finaldata, 10)
```

```
     country_name ISO             region year   gdp1000 OECD OECD2023  popdens
3711     Zimbabwe ZWE Sub-Saharan Africa 2010 0.9378403    0        0 25.51039
3712     Zimbabwe ZWE Sub-Saharan Africa 2011 1.0826158    0        0 25.53206
3713     Zimbabwe ZWE Sub-Saharan Africa 2012 1.2901940    0        0 25.55349
3714     Zimbabwe ZWE Sub-Saharan Africa 2013 1.4083678    0        0 25.53286
3715     Zimbabwe ZWE Sub-Saharan Africa 2014 1.4070343    0        0 26.52884
3716     Zimbabwe ZWE Sub-Saharan Africa 2015 1.4103292    0        0 26.54454
3717     Zimbabwe ZWE Sub-Saharan Africa 2016 1.4217878    0        0 26.53811
3718     Zimbabwe ZWE Sub-Saharan Africa 2017 1.1921070    0        0 26.49281
3719     Zimbabwe ZWE Sub-Saharan Africa 2018 2.2691770    0        0 26.47943
3720     Zimbabwe ZWE Sub-Saharan Africa 2019 1.4218686    0        0 26.46341
        urban   agedep male_edu     temp rainfall1000 matmor infmor neomor
3711 23.28851 85.56457 8.250225 21.53473    0.7290925    598   52.1   30.8
```

```
3712 23.43075 86.40049 8.358820 20.87452    0.8582386    557   50.8   30.1
3713 23.70160 86.71712 8.466529 20.98071    0.6259767    528   46.5   29.4
3714 24.04603 86.44543 8.573429 20.77221    0.6717220    509   44.8   28.7
3715 24.40427 85.87550 8.679591 20.87651    0.6777257    494   42.9   28.2
3716 24.75233 85.08337 8.785078 21.45470    0.4490721    480   42.1   27.8
3717 25.02842 84.11222 8.889947 21.39290    0.4939246    468   40.8   27.4
3718 25.29333 83.10129 8.994252 20.85962    0.9533149    458   39.9   27.0
3719 25.53759 82.12335 9.098048 20.86041    0.9535655     NA   38.8   26.6
3720 25.70572 81.20786 9.201384 20.86120    0.9538138     NA   38.1   26.2
     un5mor earthquake drought totdeath armconf1
3711   86.4          1       0        0        0
3712   80.8          0       0        0        0
3713   72.2          0       0        1        0
3714   66.3          1       0        1        0
3715   62.7          0       0        0        0
3716   61.3          0       0        0        0
3717   58.7          0       0        0        0
3718   57.0          1       0        0        0
3719   54.8          0       0        0        0
3720   54.2          0       0        4        0
```

Missing Data Analysis

Call summary to take a glance at the summary statistics of our dataset.

```
finaldata |>
  summary()
```

```
 country_name           ISO               region               year
 Length:3720        Length:3720        Length:3720        Min.   :2000
 Class :character   Class :character   Class :character   1st Qu.:2005
 Mode  :character   Mode  :character   Mode  :character   Median :2010
                                                          Mean   :2010
                                                          3rd Qu.:2014
                                                          Max.   :2019

    gdp1000              OECD          OECD2023         popdens
 Min.   :  0.1105   Min.   :0.000   Min.   :0.0000   Min.   : 0.00
 1st Qu.:  1.2383   1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:14.79
 Median :  4.0719   Median :0.000   Median :0.0000   Median :27.52
```

```
Mean   : 11.4917   Mean   :0.171   Mean   :0.1882   Mean   :30.57
3rd Qu.: 13.1531   3rd Qu.:0.000   3rd Qu.:0.0000   3rd Qu.:40.72
Max.   :123.6787   Max.   :1.000   Max.   :1.0000   Max.   :99.86
NA's   :62                                          NA's   :20
    urban             agedep          male_edu          temp
Min.   : 0.1025   Min.   : 16.17   Min.   : 1.067   Min.   :-2.405
1st Qu.:17.2872   1st Qu.: 47.94   1st Qu.: 5.904   1st Qu.:12.928
Median :30.2535   Median : 55.51   Median : 8.368   Median :21.958
Mean   :30.6948   Mean   : 61.94   Mean   : 8.258   Mean   :19.625
3rd Qu.:41.6558   3rd Qu.: 77.11   3rd Qu.:10.849   3rd Qu.:25.869
Max.   :93.4135   Max.   :111.48   Max.   :14.441   Max.   :29.676
NA's   :20                         NA's   :20       NA's   :20
 rainfall1000         matmor          infmor          neomor
Min.   :0.01993   Min.   :   2.0   Min.   :  1.60   Min.   : 0.80
1st Qu.:0.59146   1st Qu.:  17.0   1st Qu.:  7.60   1st Qu.: 4.90
Median :1.01288   Median :  66.0   Median : 18.90   Median :12.10
Mean   :1.20216   Mean   : 210.6   Mean   : 28.90   Mean   :16.18
3rd Qu.:1.68706   3rd Qu.: 299.8   3rd Qu.: 44.52   3rd Qu.:25.32
Max.   :4.71081   Max.   :2480.0   Max.   :138.10   Max.   :60.90
NA's   :20        NA's   :426      NA's   :20       NA's   :20
    un5mor          earthquake        drought          totdeath
Min.   :  2.00   Min.   :0.00000   Min.   :0.00000   Min.   :    0.0
1st Qu.:  9.00   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:    0.0
Median : 22.20   Median :0.00000   Median :0.00000   Median :    0.0
Mean   : 40.50   Mean   :0.08737   Mean   :0.08333   Mean   :  361.1
3rd Qu.: 61.33   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:    2.0
Max.   :224.90   Max.   :1.00000   Max.   :1.00000   Max.   :78644.0
NA's   :20
   armconf1
Min.   :0.0000
1st Qu.:0.0000
Median :0.0000
Mean   :0.1892
3rd Qu.:0.0000
Max.   :1.0000
```

Count and proportion of NAs for each variable

```
var <- c("country_name", "ISO", "region", "year", "gdp1000", "OECD", "OECD2023",
              "popdens", "urban", "agedep", "male_edu", "temp", "rainfall1000", "matmor",
```

```
              "infmor", "neomor", "un5mor", "earthquake", "drought", "totdeath", "armconf1")

# Calculate the number and proportion of missing values for each variable
na_table <- data.frame(
  Variable = var,
  NA_Count = sapply(var, function(var) sum(is.na(finaldata[[var]]))),
  Proportion_NA = sapply(var, function(var) sum(is.na(finaldata[[var]])) / nrow(finaldata))
)

print(na_table)
```

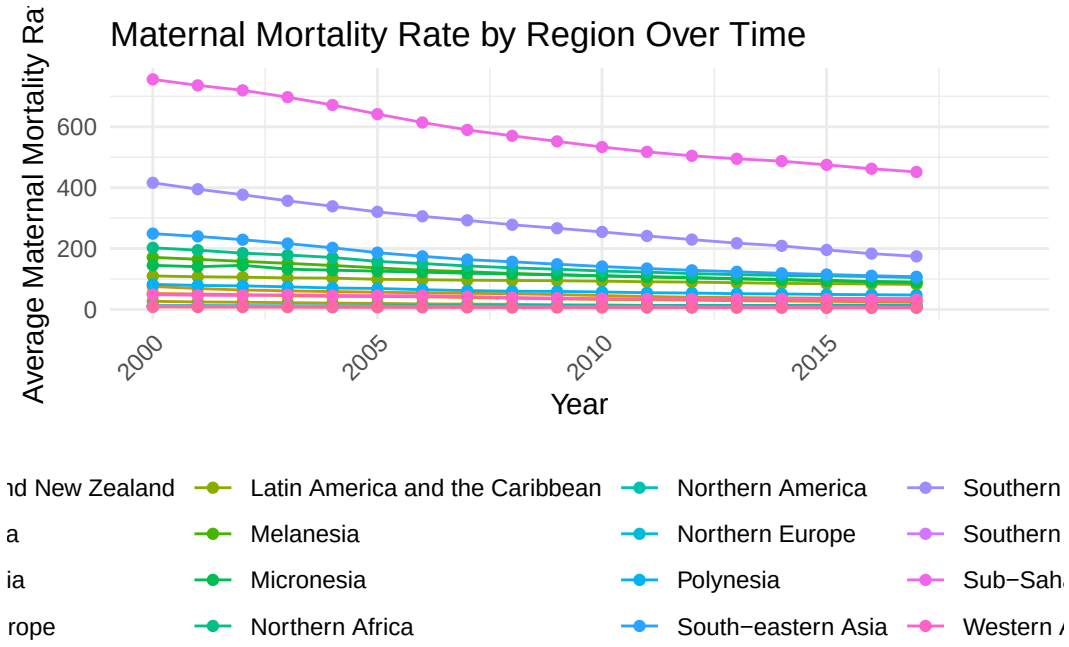
```
                Variable NA_Count Proportion_NA
country_name country_name        0   0.000000000
ISO                 ISO          0   0.000000000
region           region          0   0.000000000
year               year          0   0.000000000
gdp1000         gdp1000         62   0.016666667
OECD               OECD          0   0.000000000
OECD2023       OECD2023          0   0.000000000
popdens         popdens         20   0.005376344
urban             urban         20   0.005376344
agedep           agedep          0   0.000000000
male_edu       male_edu         20   0.005376344
temp               temp         20   0.005376344
rainfall1000 rainfall1000       20   0.005376344
matmor           matmor        426   0.114516129
infmor           infmor         20   0.005376344
neomor           neomor         20   0.005376344
un5mor           un5mor         20   0.005376344
earthquake   earthquake          0   0.000000000
drought         drought          0   0.000000000
totdeath       totdeath          0   0.000000000
armconf1       armconf1          0   0.000000000
```

Not sure which missing mechanism each variable is –MCAR, MAR, MNAR. I think it is best to ask some domain experts.

Visualizing matmor, infmor, neomor, un5mor,

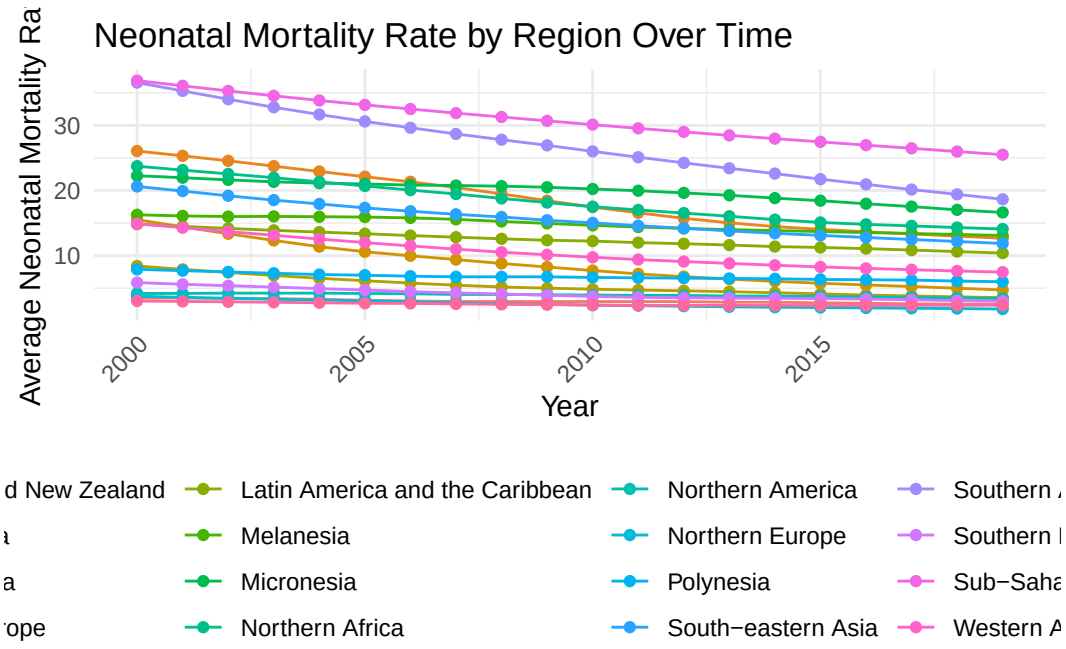
```r
grouped_data <- finaldata %>%
  group_by(region, year) %>%
  summarise(matmor = mean(matmor, na.rm = TRUE))  # Summarize matmor by region and year

ggplot(grouped_data, aes(x = year, y = matmor, group = region, color = region)) +
  geom_line() +
  geom_point() +
  labs(title = "Maternal Mortality Rate by Region Over Time",
       x = "Year",
       y = "Average Maternal Mortality Rate") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "bottom")  # Move the legend to the bottom
```



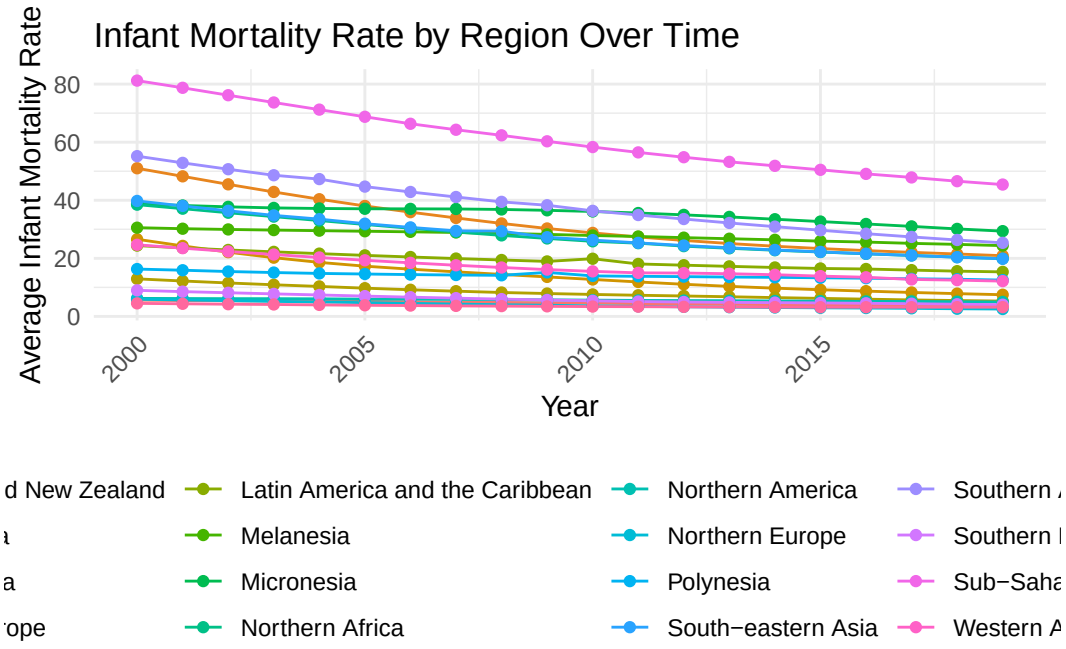
```r
# Group by region and year, and summarize the neomor
grouped_data <- finaldata %>%
  group_by(region, year) %>%
  summarise(neomor = mean(neomor, na.rm = TRUE))  # Summarize neomor by region and year

# Create the plot
ggplot(grouped_data, aes(x = year, y = neomor, group = region, color = region)) +
  geom_line() +
  geom_point() +
  labs(title = "Neonatal Mortality Rate by Region Over Time",
       x = "Year",
       y = "Average Neonatal Mortality Rate") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = 'bottom')  # Tilting x-axis labels for readability
```



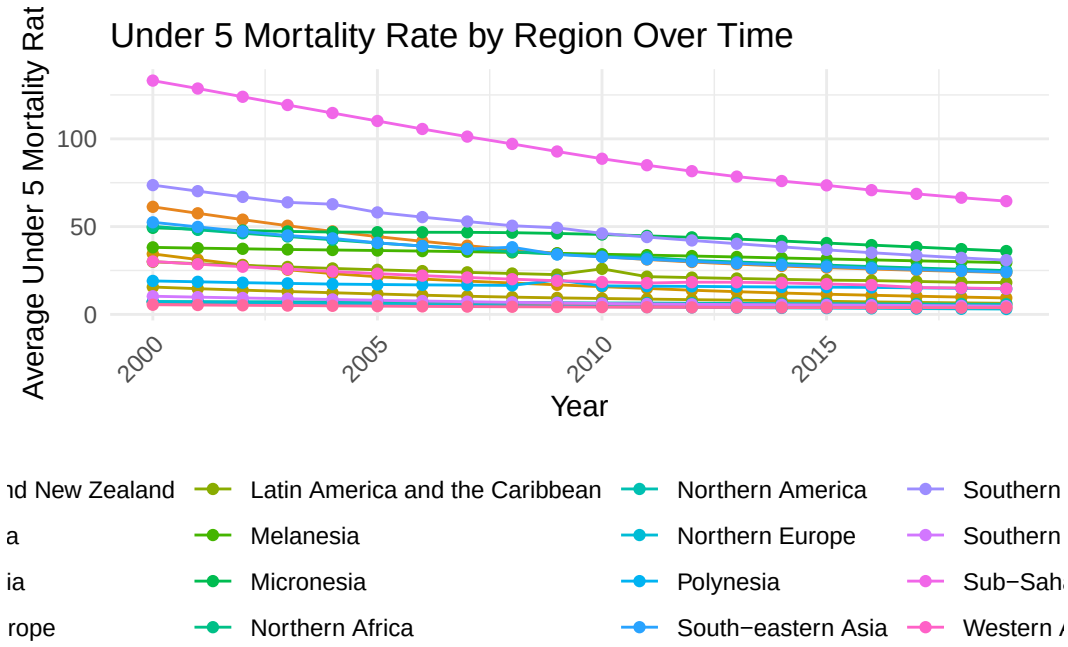
```r
# Group by region and year, and summarize the infmor
grouped_data <- finaldata %>%
  group_by(region, year) %>%
  summarise(infmor = mean(infmor, na.rm = TRUE))  # Summarize infmor by region and year

# Create the plot
ggplot(grouped_data, aes(x = year, y = infmor, group = region, color = region)) +
  geom_line() +
  geom_point() +
  labs(title = "Infant Mortality Rate by Region Over Time",
       x = "Year",
       y = "Average Infant Mortality Rate") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = 'bottom')  # Tilting x-axis labels for readability
```



```r
# Group by region and year, and summarize the un5mor
grouped_data <- finaldata %>%
  group_by(region, year) %>%
  summarise(un5mor = mean(un5mor, na.rm = TRUE))  # Summarize un5mor by region and year

# Create the plot
ggplot(grouped_data, aes(x = year, y = un5mor, group = region, color = region)) +
  geom_line() +
  geom_point() +
  labs(title = "Under 5 Mortality Rate by Region Over Time",
       x = "Year",
       y = "Average Under 5 Mortality Rate") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = 'bottom')  # Tilting x-axis labels for readability
```



In general, all four mortality rates seems to be going down across all regions by year.

Combine matmor, infmor, neomor, and un5mor to create a new dataset by country.

```
totalmort <- finaldata %>%
  group_by(country_name) %>%
  summarise(
    total_matmor = sum(matmor, na.rm = TRUE),
    total_infmor = sum(infmor, na.rm = TRUE),
    total_neomor = sum(neomor, na.rm = TRUE),
    total_un5mor = sum(un5mor, na.rm = TRUE)
  ) %>%
  mutate(totalmort = total_matmor + total_infmor + total_neomor + total_un5mor)
```

```
# Find the country with the maximum totalmort
max_country <- totalmort %>%
  filter(totalmort == max(totalmort, na.rm = TRUE)) %>%
  select(country_name, totalmort)

# Find the country with the minimum totalmort
min_country <- totalmort %>%
  filter(totalmort == min(totalmort, na.rm = TRUE)) %>%
  select(country_name, totalmort)

# View the results
print(max_country, max_country$totalmort)
```

```
# A tibble: 1 x 2
  country_name totalmort
  <chr>            <dbl>
1 Sierra Leone    34614.
```

```
print(min_country, min_country$totalmort)
```

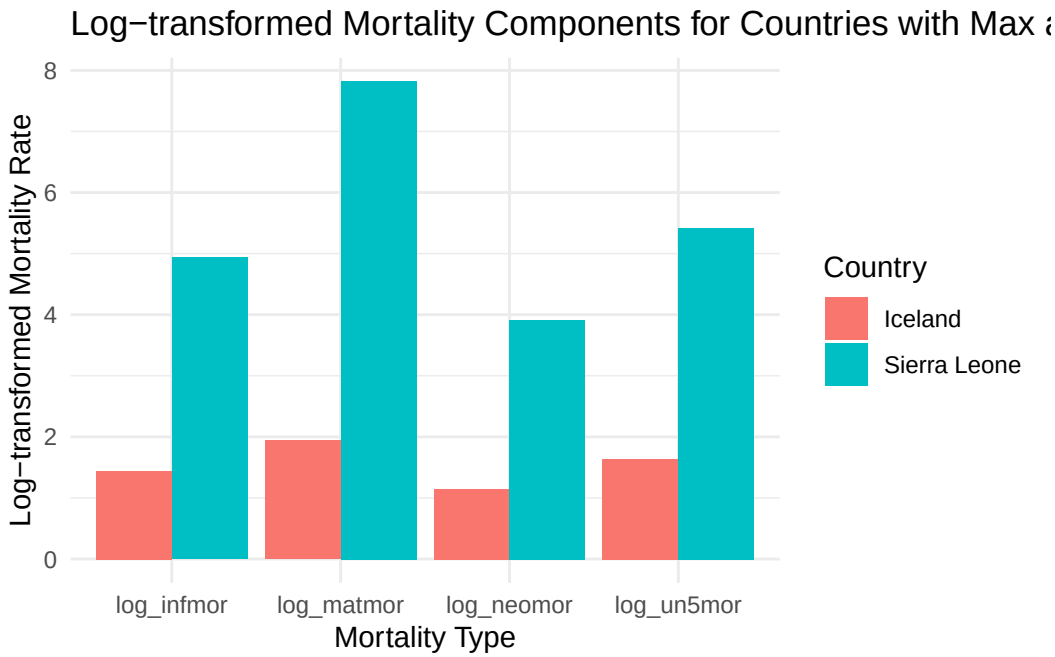
```
# A tibble: 1 x 2
  country_name totalmort
  <chr>            <dbl>
1 Iceland            221
```

The two countries with max and min total mortality rates are Sierra Leone with 34614.5 deaths and Iceland with 221 deaths.

```r
# Filter the data for the max and min countries and log-transform the mortality rates
combined_data <- finaldata %>%
  filter(country_name %in% c(max_country$country_name, min_country$country_name)) %>%
  select(country_name, matmor, infmor, neomor, un5mor) %>%
  mutate(
    log_matmor = log(matmor + 1),  # log transformation, adding 1 to avoid log(0)
    log_infmor = log(infmor + 1),
    log_neomor = log(neomor + 1),
    log_un5mor = log(un5mor + 1)
  ) %>%
  select(country_name, log_matmor, log_infmor, log_neomor, log_un5mor) %>%
  gather(key = "mortality_type", value = "log_mortality_rate", log_matmor, log_infmor, log_ne

# Visualize the log-transformed data
ggplot(combined_data, aes(x = mortality_type, y = log_mortality_rate, fill = country_name)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Log-transformed Mortality Components for Countries with Max and Min Total Mor
       x = "Mortality Type",
       y = "Log-transformed Mortality Rate",
       fill = "Country") +
  theme_minimal()
```

```
Warning: Removed 4 rows containing missing values (`geom_bar()`).
```



Taking a look at the log transformed mortality rates for Sierra Leone and Iceland. Intepretation: If the y-axis value is 2, the actual mortality rate is $\exp(2) - 1 \approx 6.39$ per N deaths where N is the measurement metric from the dataset.

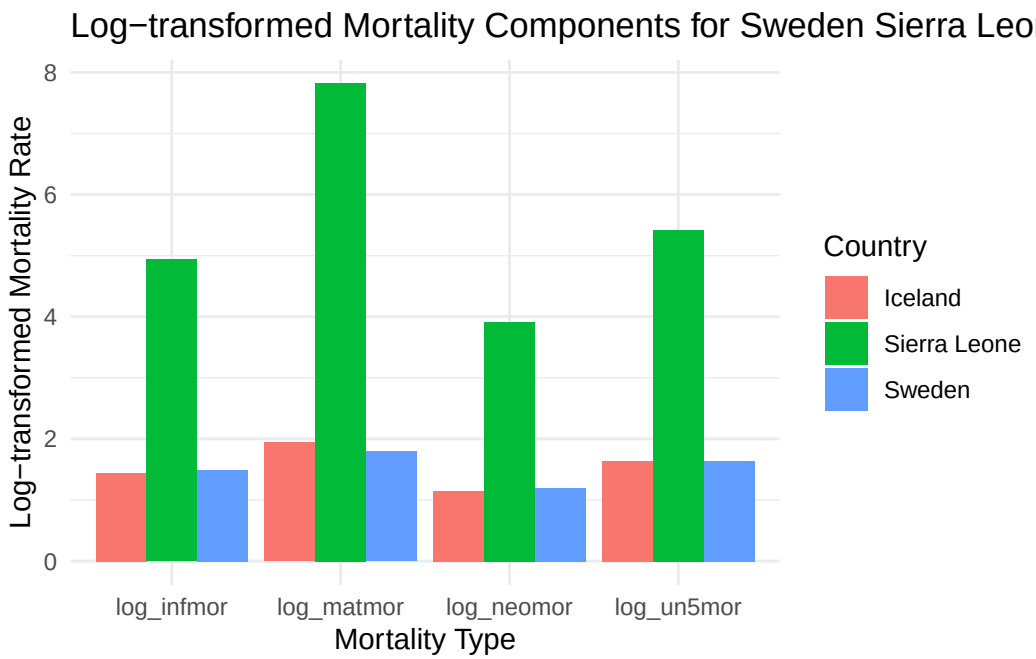
Now I will generate a random country and see if this country has similar rates to Sierra Leone and Iceland.

```
# Select a random country from the dataset
set.seed(123)  # Set seed for reproducibility
random_country <- finaldata %>%
  select(country_name) %>%
  distinct() %>%
  sample_n(1)  # Randomly select one country

# Specify Sierra Leone and Iceland
selected_countries <- c("Sierra Leone", "Iceland", random_country$country_name)

# Filter the data for the selected countries (random country, Sierra Leone, and Iceland)
combined_country_data <- finaldata %>%
  filter(country_name %in% selected_countries) %>%
  select(country_name, matmor, infmor, neomor, un5mor) %>%
  mutate(
    log_matmor = log(matmor + 1),  # log transformation, adding 1 to avoid log(0)
    log_infmor = log(infmor + 1),
    log_neomor = log(neomor + 1),
    log_un5mor = log(un5mor + 1)
  ) %>%
  select(country_name, log_matmor, log_infmor, log_neomor, log_un5mor) %>%
  gather(key = "mortality_type", value = "log_mortality_rate", log_matmor, log_infmor, log_ne

# Visualize the log-transformed data for the selected countries
ggplot(combined_country_data, aes(x = mortality_type, y = log_mortality_rate, fill = country_
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = paste("Log-transformed Mortality Components for",
                     random_country$country_name, "Sierra Leone, and Iceland"),
       x = "Mortality Type",
       y = "Log-transformed Mortality Rate",
       fill = "Country") +
  theme_minimal()
```



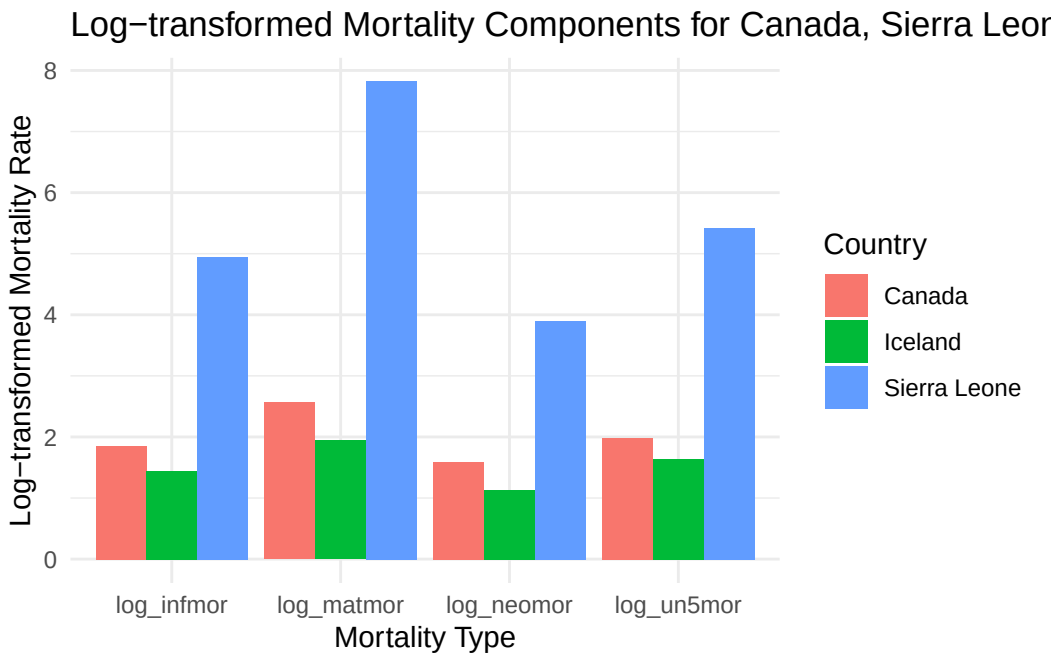
Now look at Canada with Sierra Leone and Iceland.

```
# Specify Sierra Leone, Iceland, and Canada
selected_countries <- c("Sierra Leone", "Iceland", "Canada")

# Filter the data for the selected countries (Canada, Sierra Leone, and Iceland)
combined_country_data <- finaldata %>%
  filter(country_name %in% selected_countries) %>%
  select(country_name, matmor, infmor, neomor, un5mor) %>%
  mutate(
    log_matmor = log(matmor + 1),  # log transformation, adding 1 to avoid log(0)
    log_infmor = log(infmor + 1),
    log_neomor = log(neomor + 1),
    log_un5mor = log(un5mor + 1)
  ) %>%
  select(country_name, log_matmor, log_infmor, log_neomor, log_un5mor) %>%
  gather(key = "mortality_type", value = "log_mortality_rate", log_matmor, log_infmor, log_ne

# Visualize the log-transformed data for the selected countries
ggplot(combined_country_data, aes(x = mortality_type, y = log_mortality_rate, fill = country_
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Log-transformed Mortality Components for Canada, Sierra Leone, and Iceland",
       x = "Mortality Type",
```

```
      y = "Log-transformed Mortality Rate",
      fill = "Country") +
 theme_minimal()
```



Summary

We looked at the column variable names, missing values for each variable and their proportions, visialized total mortality rate for the country with max/min mortality rates, and compared it to a random country and Canada.