Week 1 Methods: Growth

ECON 125: The Science of Population

Setup

Today, we analyze population growth and its components in this dataset:

- Population, births, deaths, net migration (in 1000s)
- One row per year since 1950 for every country
- From United Nations World Population Prospects

Start by setting up R, loading the dataset, and assigning it a name using <-

```
# Load tidyverse and clear the R environment
library(tidyverse)
rm(list=ls())

# Load dataset and call it country_year_df
country_year_df <- read_csv(url("https://github.com/tomvogl/econ125/raw

# Ask R to only report 2 significant digits
# This is not important and just makes the output easier to read
options(digits = 2)
```

Variables

You can see which variables are in the dataset by...

- Clicking on the data frame in the Environment pane in RStudio
- Using the `names()` function in your code

```
names(country_year_df)
```

```
## [1] "country"      "year"         "pop"          "births"       "dea
## [6] "netmigration"
```

Data structure

Let's look at the first 10 rows of the dataset

You can see that each country-year gets its own row

- In the 1950s, Burundi had 2-3 million people
- Each year, 120k were born, 55k died, and 15k more left than entered

```
head(country_year_df, 10)
```

```
## # A tibble: 10 x 6
##    country  year   pop births deaths netmigration
##    <chr>   <dbl> <dbl>  <dbl>  <dbl>        <dbl>
##  1 Burundi  1950 2255.   117.   52.7        -13.3
##  2 Burundi  1951 2306.   118.   54.4        -13.2
##  3 Burundi  1952 2356.   119.   55.4        -13.7
##  4 Burundi  1953 2405.   120.   56.1        -14.9
##  5 Burundi  1954 2455.   121.   56.7        -14.6
##  6 Burundi  1955 2505.   122.   57.3        -14.6
##  7 Burundi  1956 2555.   123.   57.7        -14.7
##  8 Burundi  1957 2606.   125.   58.1        -15.0
##  9 Burundi  1958 2657.   126.   58.6        -17.0
## 10 Burundi  1959 2710.   128.   59.0        -14.8
```

Country-level to world-level

Let's study the world population by summing across countries in each year

We'll use three tools from `tidyverse`: the pipe `|>`, `group_by()`, `summarise()`

```
world_year_df <-
  country_year_df |>
  group_by(year) |>
  summarise(world_pop = sum(pop),
            world_births = sum(births),
            world_deaths = sum(deaths),
            world_netmigration = sum(netmigration))
```

Line-by-line explanation:

1. Create a new data frame called `world_year_df`
2. Start with our existing data frame `country_year_df`, and then...
3. Group the data by `year`, and then...
4. Within each year, sum each variable across countries

New data frame

Here are the first 10 rows of the new data frame we created

- Net migration is always zero (slight nonzero values due to rounding errors)
- Aliens did not land on earth, and Elon did not start his Mars colony

```
head(world_year_df, 10)
```

```
## # A tibble: 10 x 5
##     year world_pop world_births world_deaths world_netmigration
##    <dbl>     <dbl>        <dbl>        <dbl>              <dbl>
##  1  1950  2493093.       91824.       48487.           -0.00100
##  2  1951  2536927.       92507.       48176.           -0.00700
##  3  1952  2584086.       97371.       47383.           -0.00100
##  4  1953  2634106.       97291.       47240.           -0.00200
##  5  1954  2685895.      100188.       46662.           -0.00300
##  6  1955  2740214.      101748.       46636.           -0.00200
##  7  1956  2795410       101759.       46479.            0.00100
##  8  1957  2852618.      106018.       46881.            0.00100
##  9  1958  2911250.      104644.       46518.           -0.00400
## 10  1959  2965950.      102000.       50725.           -0.00100
```

Summary statistics

R offers many ways to calculate summary statistics for a variable

- We will use `summarise()` from `tidyverse`, along with the functions `mean()`, `sd()`, `min()`, and `max()`
- The `summary()` function from base R is also useful, but its syntax is different from `tidyverse`, so we will skip it to avoid confusion

Here is an example for net migration, which again confirms that it is always zero:

```
world_year_df |> summarise(mean = mean(world_netmigration),
                            std_dev = sd(world_netmigration),
                            minimum = min(world_netmigration),
                            maximum = max(world_netmigration))
```

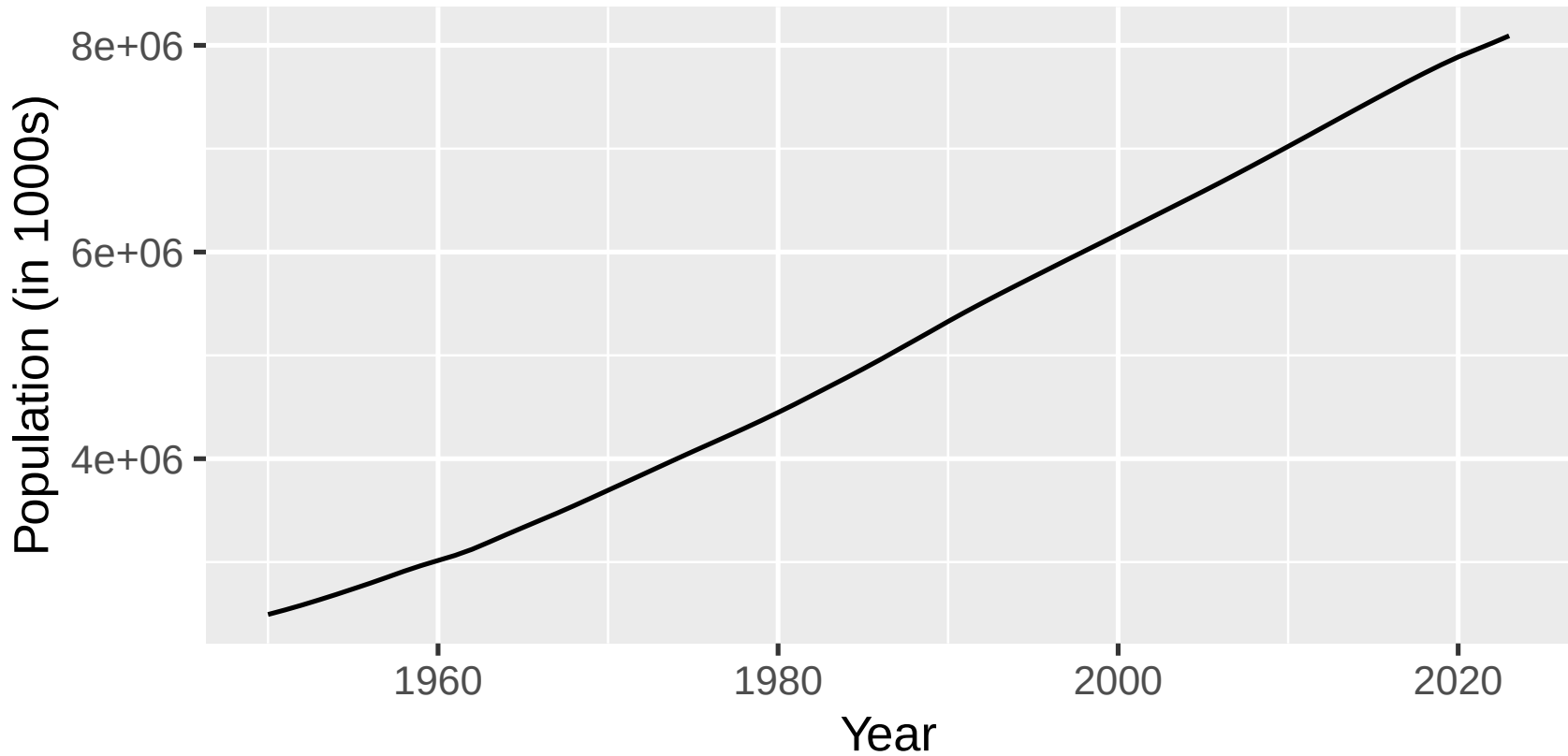
```
## # A tibble: 1 x 4
##         mean std_dev  minimum maximum
##        <dbl>   <dbl>    <dbl>   <dbl>
## 1 -0.000784 0.00241 -0.00700 0.00600
```

Time series of the world population

Now let's plot the world population over time

We'll use the `tidyverse` tool for drawing graphs: `ggplot()`

```
ggplot(world_year_df, aes(x = year, y = world_pop)) +
  geom_line() +
  scale_y_continuous("Population (in 1000s)") +
  scale_x_continuous("Year")
```



Levels to growth rates

It's hard to see details in the time series of population

Let's calculate the growth rate and add it to `world_year_df`

```
world_year_df <-
  world_year_df |>
  arrange(year) |>
  mutate(world_popgrowth =
           100*(lead(world_pop) - world_pop)/world_pop)
```

Some comments:

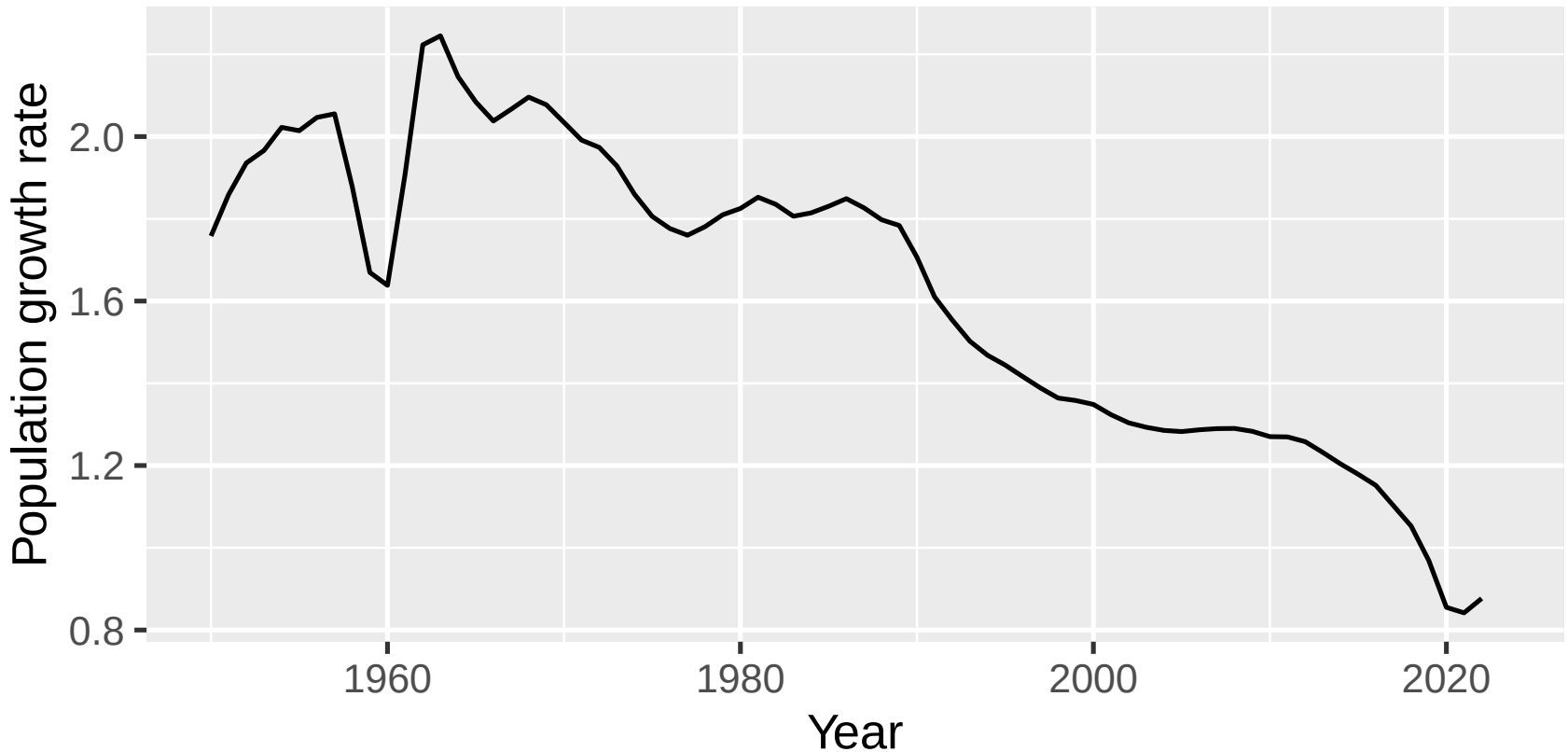
- `arrange()` sorted the data by year
- `mutate()` generated a new variable `world_popgrowth`
- `lead()` looked forward to the next value of `world_pop`
- The line breaks were not necessary; they just made my slides prettier

Plotting the population growth rate over time

When we plot the growth rate, the results are more interesting!

- World population growth steadily declining for decades
- Large spike down in 1960, due to the Great Famine in China

```
ggplot(world_year_df, aes(x = year, y = world_popgrowth)) +
  geom_line()+
  scale_y_continuous("Population growth rate") +
  scale_x_continuous("Year")
```



Decomposing population growth

Recall the demographic balancing equation:

$$P_1 = P_0 + (B - D) + (I - E)$$

We can rearrange as follows:

$$\frac{P_1 - P_0}{P_0} = \frac{B}{P_0} - \frac{D}{P_0} + \frac{I - E}{P_0}$$

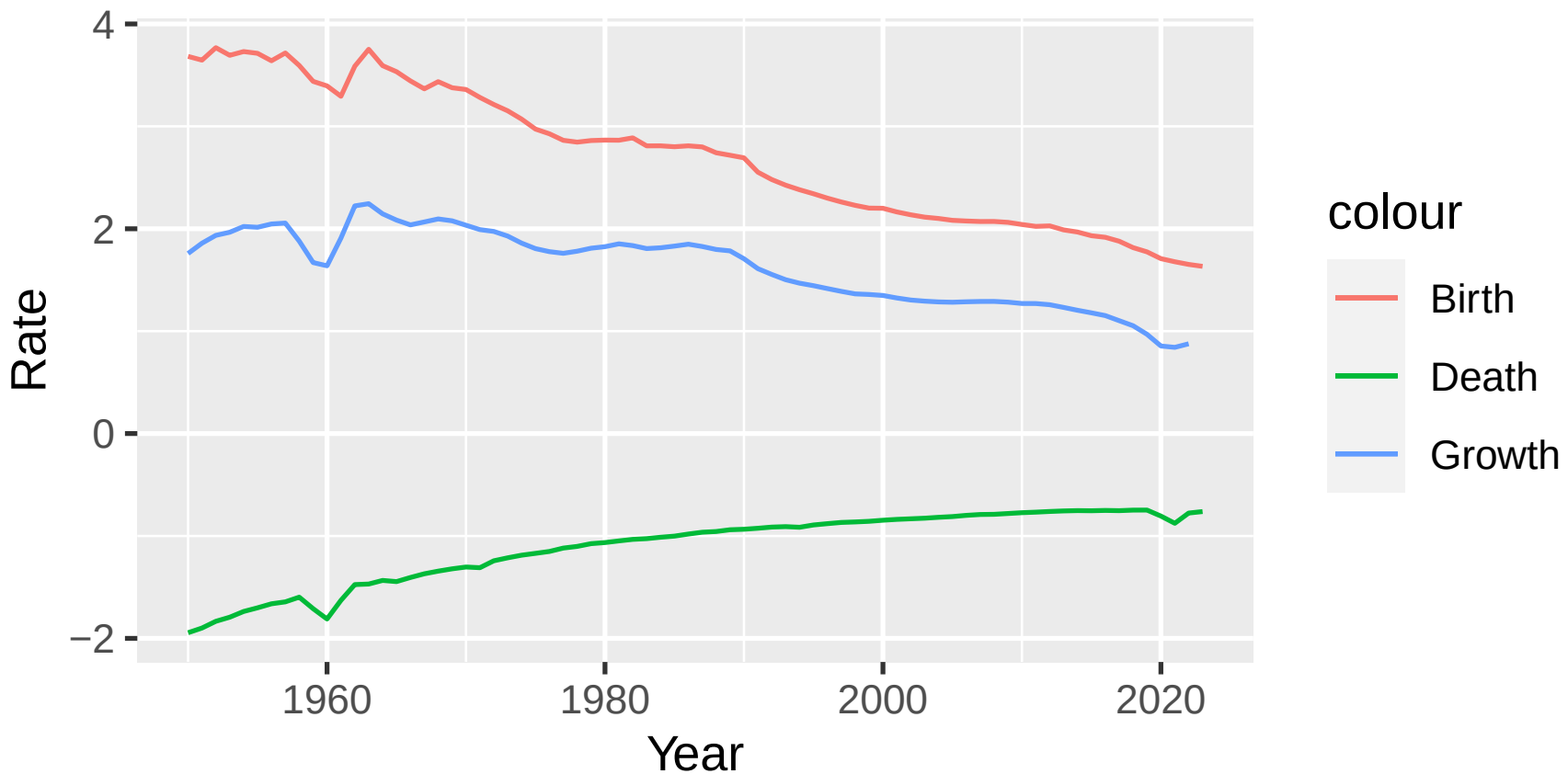
The population growth rate equals the birth rate minus the death rate plus the net migration rate. Since global net migration is 0, focus on births and deaths:

```
world_year_df <-
  world_year_df |>
  arrange(year) |>
  mutate(world_birthrate = 100*world_births/world_pop,
         world_deathrate = -100*world_deaths/world_pop)
```

Time series of population growth rate, birth rate, and death rate

To plot all three series, we modify the `ggplot()` syntax as follows:

```
ggplot(world_year_df, aes(x = year)) +
  geom_line(aes(y = world_popgrowth, color = "Growth")) +
  geom_line(aes(y = world_deathrate, color = "Death")) +
  geom_line(aes(y = world_birthrate, color = "Birth")) +
  scale_y_continuous("Rate") +
  scale_x_continuous("Year")
```



Country case study

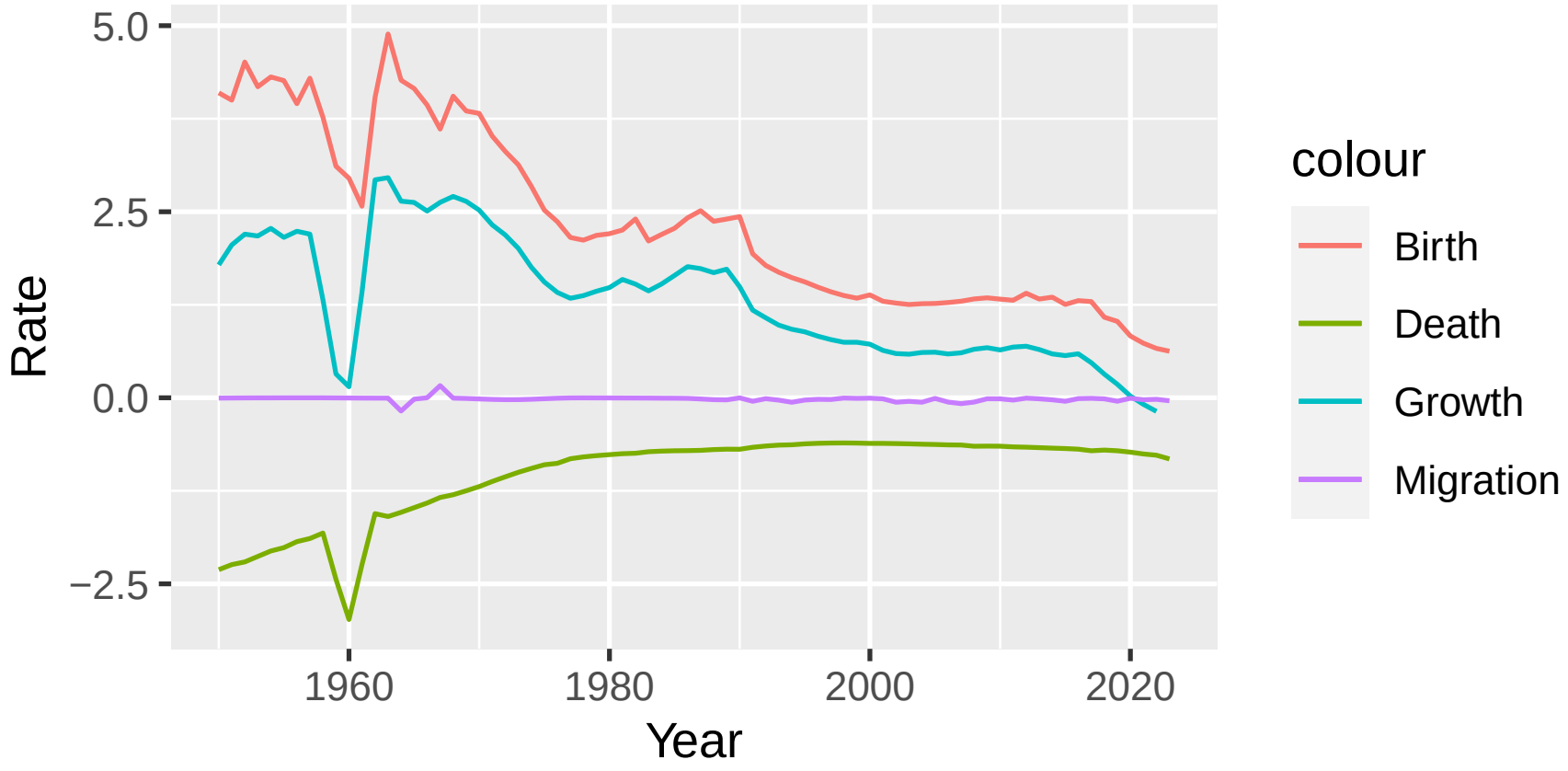
I claimed that Great Chinese Famine accounted for the blip in 1960. Let's check.

To focus on China, we use `filter()`:

```
china_df <-
  country_year_df |>
  filter(country=="China") |>
  arrange(year) |>
  mutate(popgrowth = 100*(lead(pop) - pop)/pop,
         birthrate = 100*births/pop,
         deathrate = -100*deaths/pop,
         migrate = 100*netmigration/pop)
```

Population growth and its components in China

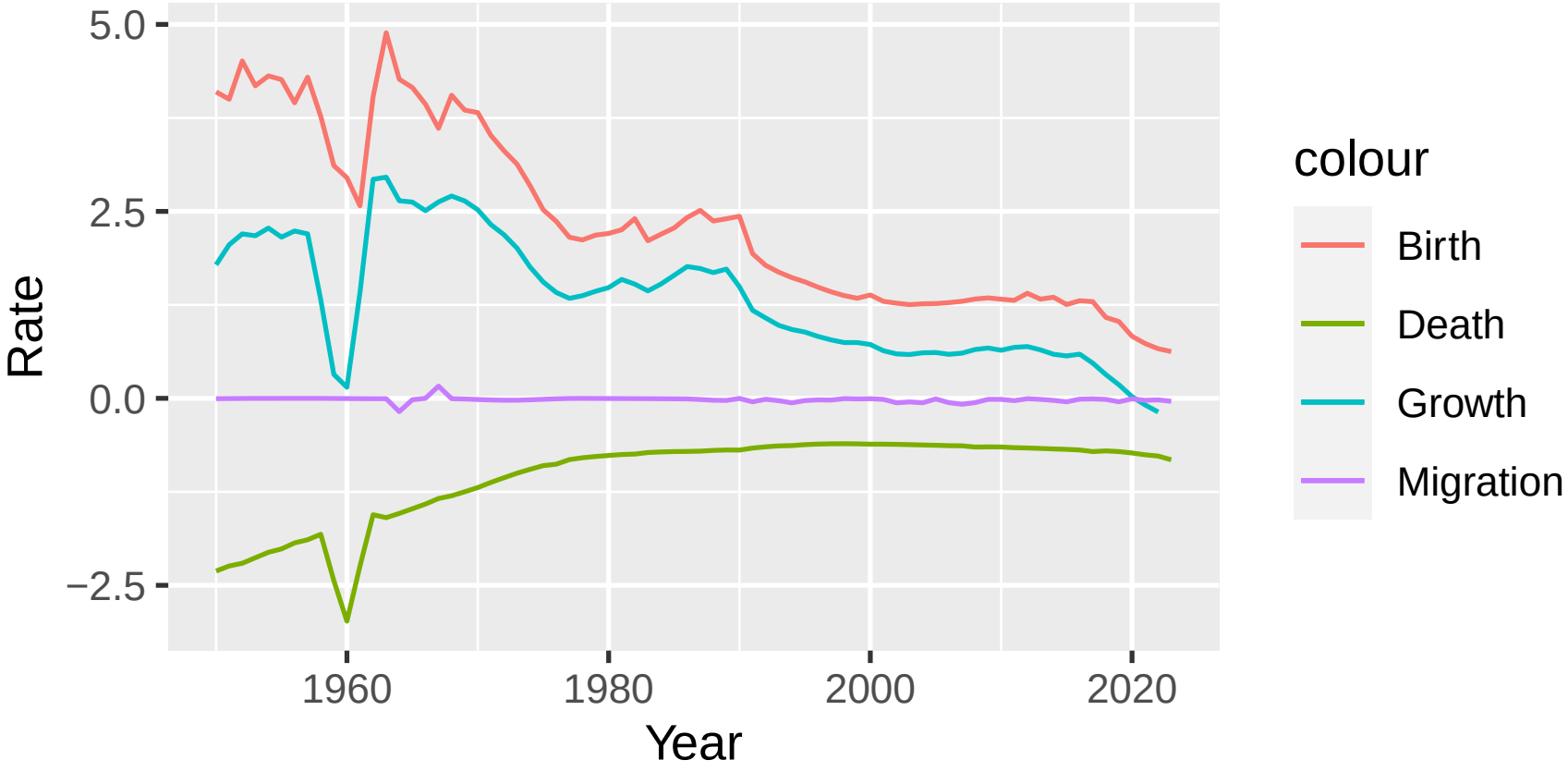
```
ggplot(china_df, aes(x = year)) +
  geom_line(aes(y = popgrowth, color = "Growth")) +
  geom_line(aes(y = deathrate, color = "Death")) +
  geom_line(aes(y = birthrate, color = "Birth")) +
  geom_line(aes(y = migrate, color = "Migration")) +
  scale_y_continuous("Rate") +
  scale_x_continuous("Year")
```



Chinese demographic history

Some interesting facts:

1. Death rates rose **and** birth rates fell during Great Leap Forward (1958-62)
2. Most decline in the birth rate happened **before** the One Child Policy (1979)
3. After end of One Child Policy (2016), fertility fell more
4. Large reductions in death rate before 1980, some slippage recently



Setting up country comparisons

Let's compare the five countries with the largest populations in 1950

- The symbol | means **or**
- `group_by()` and `ungroup()` make sure we take leads in the same country

```
top5_df <-
  country_year_df |>
  filter(country == "China" |
         country == "India" |
         country == "United States of America" |
         country == "Russian Federation" |
         country == "Japan") |>
  group_by(country) |>
  arrange(year) |>
  mutate(popgrowth = 100*(lead(pop) - pop)/pop,
         birthrate = 100*births/pop,
         deathrate = -100*deaths/pop,
         migrate = 100*netmigration/pop) |>
  ungroup()
```

Shortening country names

Some country names are very long!

Use `unique()` to list the unique values of the variable `country` in the data frame `top5_df`

```
unique(top5_df$country)
```

```
## [1] "China"                          "Japan"
## [3] "India"                          "Russian Federation"
## [5] "United States of America"
```

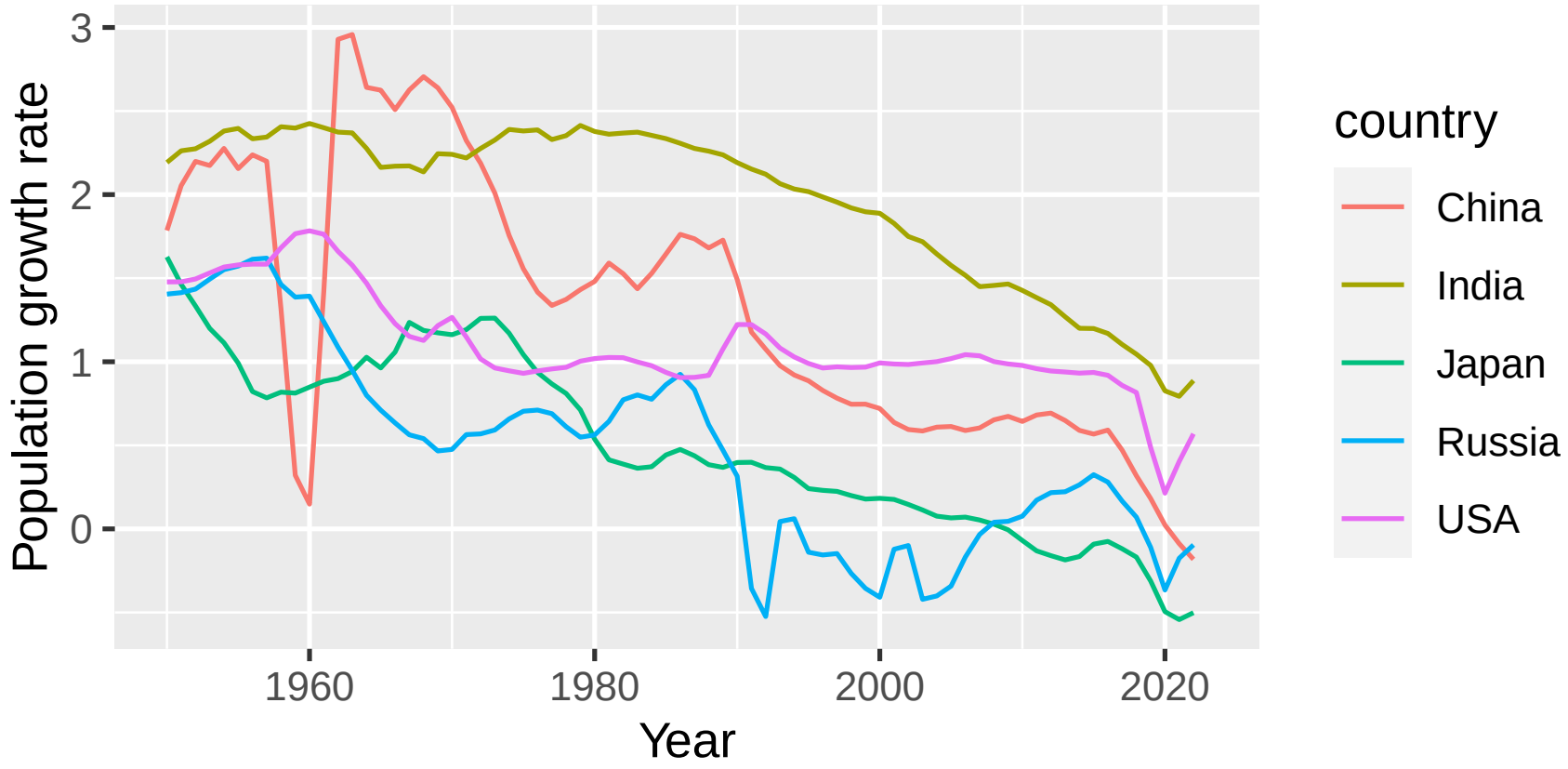
Let's shorten the longest ones using `mutate()` and `case_when()`

```
top5_df <-
  top5_df |>
  mutate(
  country = case_when(country=="United States of America" ~ "USA",
                      country=="Russian Federation" ~ "Russia",
                      .default = country)
  )
```

Population growth over time by country

Plot all five countries in one panel using the `color` option:

```
ggplot(top5_df, aes(x = year, y = popgrowth, color = country)) +
  geom_line()+
  scale_y_continuous("Population growth rate") +
  scale_x_continuous("Year")
```



Population growth and its components over time by country

Plot one panel per country using `facet_wrap()`:

```
ggplot(top5_df, aes(x = year)) +
  geom_line(aes(y = popgrowth, color = "Growth")) +
  geom_line(aes(y = deathrate, color = "Death")) +
  geom_line(aes(y = birthrate, color = "Birth")) +
  geom_line(aes(y = migrate, color = "Migration")) +
  scale_y_continuous("Rate") +
  scale_x_continuous("Year") +
  facet_wrap(~country)
```