

# Week 1: Intro to demographic concepts

## SOC6708 ADA

Monica Alexander

### Read in the data

All these data come from the [UN World Population Prospects 2024](#).

Packages:

```
library(tidyverse)
library(here)
library(readxl)
library(janitor)
```

Population data: *d\_male is data <- means “assigned to” \$ means access a variable (can be either pre-existed or new) d\_male\$sexaccesssexvariableind<sub>male</sub> \* d<sub>male</sub>\$sex <-“male” when “male” put value on sex d\_female <- d\_female |> drop\_na(Year) drop if year is missing(N/A), and save it to original data d <- rbind(d\_male, d\_female) means combining two datasets together, these two datasets have same structures. similar to merge in stata. rm means removing, deleting clean\_names()is lower-cases,spaces replaced by\_, and removing special characters select is to keep rename: new names first \*mutate as.numeric number*

```
d_male <- read_xlsx(here("/Users/liangqishen/Desktop/WPP2024_POP_F01_2_POPULATION_SINGLE_AGE")
d_male$sex <- "Male"
d_male <- d_male |> drop_na(Year)
d_female <- read_xlsx(here("/Users/liangqishen/Desktop//WPP2024_POP_F01_3_POPULATION_SINGLE_AGE")
d_female$sex <- "Female"
d_female <- d_female |> drop_na(Year)

d <- rbind(d_male, d_female)
rm(d_male, d_female)

d <- d |>
```

```

clean_names() |>
select(region_subregion_country_or_area, iso3_alpha_code, year, x0:sex) |>
rename(region = region_subregion_country_or_area) |>
mutate(across(x0:x100, as.numeric))

head(d)

# A tibble: 6 x 105
  region iso3_alpha_code year     x0     x1     x2     x3     x4     x5     x6
  <chr>  <lgcl>        <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
1 World   NA            1950 41603. 36997. 34245. 32359. 30007. 28065. 27335.
2 World   NA            1951 42615. 38588. 35577. 33456. 31876. 29664. 27829.
3 World   NA            1952 43984. 39615. 37182. 34798. 32980. 31545. 29427.
4 World   NA            1953 45183. 41001. 38243. 36421. 34326. 32646. 31320.
5 World   NA            1954 45934. 42199. 39641. 37489. 35955. 33999. 32422.
6 World   NA            1955 47059. 42972. 40848. 38889. 37023. 35636. 33784.
# i 95 more variables: x7 <dbl>, x8 <dbl>, x9 <dbl>, x10 <dbl>, x11 <dbl>,
# x12 <dbl>, x13 <dbl>, x14 <dbl>, x15 <dbl>, x16 <dbl>, x17 <dbl>,
# x18 <dbl>, x19 <dbl>, x20 <dbl>, x21 <dbl>, x22 <dbl>, x23 <dbl>,
# x24 <dbl>, x25 <dbl>, x26 <dbl>, x27 <dbl>, x28 <dbl>, x29 <dbl>,
# x30 <dbl>, x31 <dbl>, x32 <dbl>, x33 <dbl>, x34 <dbl>, x35 <dbl>,
# x36 <dbl>, x37 <dbl>, x38 <dbl>, x39 <dbl>, x40 <dbl>, x41 <dbl>,
# x42 <dbl>, x43 <dbl>, x44 <dbl>, x45 <dbl>, x46 <dbl>, x47 <dbl>, ...

```

Mortality data:

```

d_male <- read_xlsx(here("/Users/liangqishen/Desktop/mortality_data/WPP2024_MORT_F01_2_DEATHS"))
d_male$sex <- "Male"
d_male <- d_male |> drop_na(Year)
d_female <- read_xlsx(here("/Users/liangqishen/Desktop/mortality_data/WPP2024_MORT_F01_3_DEATHS"))
d_female$sex <- "Female"
d_female <- d_female |> drop_na(Year)

dm <- rbind(d_male, d_female)
rm(d_male, d_female)

dm <- dm |>
clean_names() |>
select(region_subregion_country_or_area, iso3_alpha_code, year, x0:sex) |>
rename(region = region_subregion_country_or_area) |>
mutate(across(x0:x100, as.numeric))

```

```

head(dm)

# A tibble: 6 x 105
  region iso3_alpha_code year    x0     x1     x2     x3     x4     x5     x6     x7
  <chr> <lgl>        <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 World  NA            1950 6779. 1815. 967. 588. 380. 255. 183. 142.
2 World  NA            1951 6780. 1796. 968. 582. 382. 257. 181. 138.
3 World  NA            1952 6867. 1743. 943. 578. 375. 252. 176. 132.
4 World  NA            1953 6831. 1726. 933. 572. 379. 264. 191. 142.
5 World  NA            1954 6841. 1706. 928. 568. 375. 252. 180. 136.
6 World  NA            1955 6815. 1663. 917. 566. 373. 256. 181. 137.
# i 94 more variables: x8 <dbl>, x9 <dbl>, x10 <dbl>, x11 <dbl>, x12 <dbl>,
#   x13 <dbl>, x14 <dbl>, x15 <dbl>, x16 <dbl>, x17 <dbl>, x18 <dbl>,
#   x19 <dbl>, x20 <dbl>, x21 <dbl>, x22 <dbl>, x23 <dbl>, x24 <dbl>,
#   x25 <dbl>, x26 <dbl>, x27 <dbl>, x28 <dbl>, x29 <dbl>, x30 <dbl>,
#   x31 <dbl>, x32 <dbl>, x33 <dbl>, x34 <dbl>, x35 <dbl>, x36 <dbl>,
#   x37 <dbl>, x38 <dbl>, x39 <dbl>, x40 <dbl>, x41 <dbl>, x42 <dbl>,
#   x43 <dbl>, x44 <dbl>, x45 <dbl>, x46 <dbl>, x47 <dbl>, x48 <dbl>, ...

```

## Crude Rates

Calculate the crude death rates for Kenya and Canada in 2023 *filter means to keep pivot\_longer* from x0 to x100, put these variable names as a new variable called age, and values of each x0 to x100 become population variable \*group\_by only happened within the group

```

# get total populations across all age and sex
total_pops <- d |>
  filter(region=="Kenya" | region=="Canada") |>
  pivot_longer(x0:x100, names_to = "age", values_to = "pop") |>
  mutate(age = as.numeric(str_remove(age, "x")))) |>
  group_by(region, year) |>
  summarize(total_pop = sum(pop))

# get total deaths

total_deaths <- dm |>
  filter(region=="Kenya" | region=="Canada") |>
  pivot_longer(x0:x100, names_to = "age", values_to = "deaths") |>
  mutate(age = as.numeric(str_remove(age, "x")))) |>
  group_by(region, year) |>

```

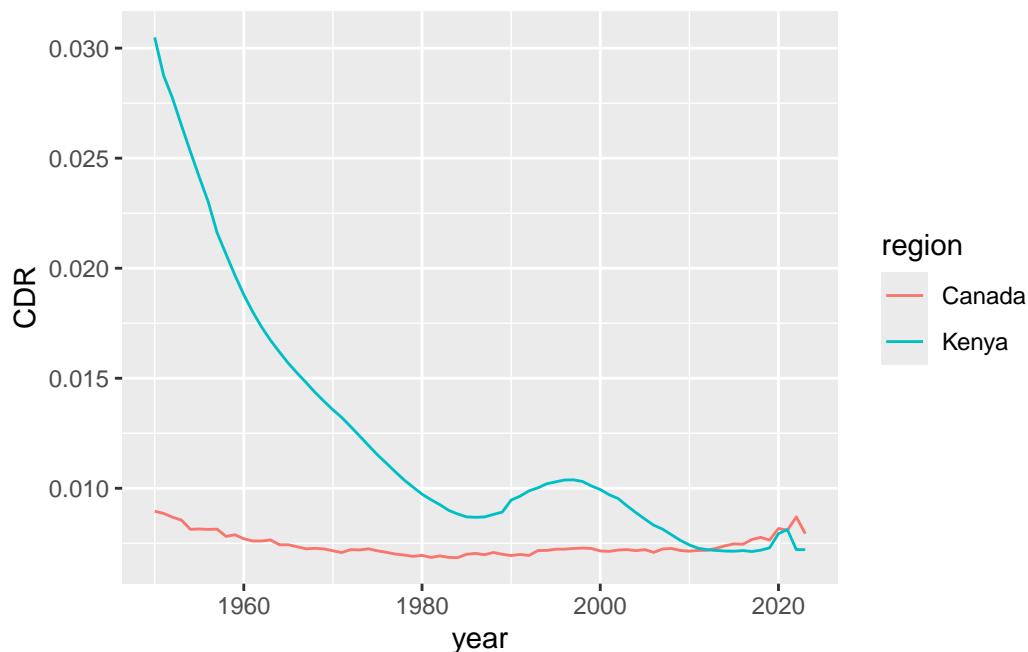
```

summarize(total_deaths = sum(deaths))

# join these

total_pops |>
  left_join(total_deaths) |>
  mutate(CDR = total_deaths/total_pop) |>
  ggplot(aes(year, CDR, color = region)) +
  geom_line()

```



```

total_pops |>
  left_join(total_deaths) |>
  mutate(CDR = total_deaths/total_pop) |>
  filter(year==2023)

```

```

# A tibble: 2 x 5
# Groups:   region [2]
  region year total_pop total_deaths      CDR
  <chr>   <dbl>     <dbl>        <dbl>    <dbl>
1 Canada   2023     39299.       312.  0.00793
2 Kenya    2023     55339.       399.  0.00721

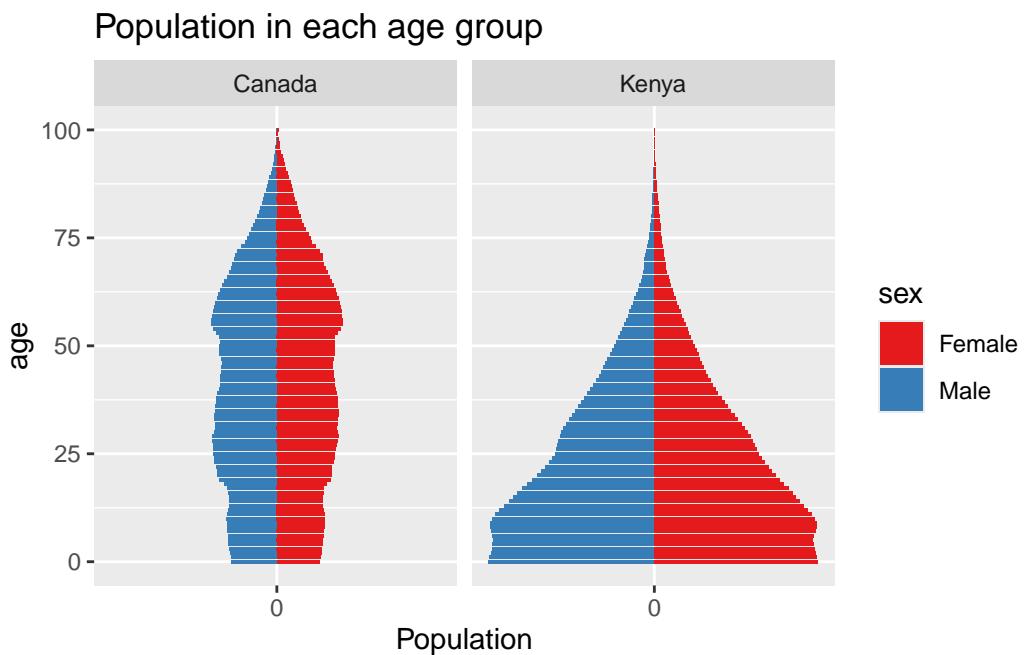
```

## Population Pyramids

Population age structures of Kenya and Canada in 2019:

```
d_long <- d  |>
  pivot_longer(x0:x100, names_to = "age", values_to = "pop") |>
  mutate(age = as.numeric(str_remove(age, "x")))

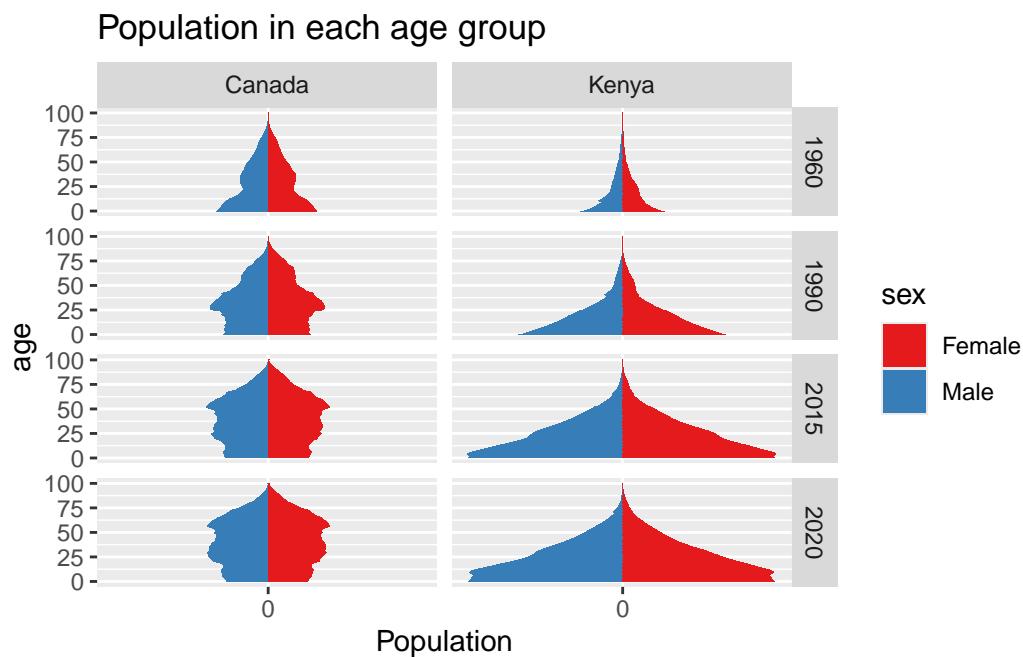
d_long|>
  filter(region=="Kenya" | region=="Canada", year == 2019) |>
  mutate(population=ifelse(sex=="Male", -pop, pop)) |>
  ggplot(aes(x = age, y = population, fill = sex)) +
  facet_wrap(~region) +
  geom_bar(stat="identity")+
  ggtitle("Population in each age group")+
  ylab("Population")+
  coord_flip() +
  scale_y_continuous(breaks = seq(-4000, 4000, 1000),
                     labels = c(seq(4000, 0, -1000), seq(1000, 4000, 1000))) +
  scale_fill_brewer(palette = "Set1")
```



```
ggsave(here("plots", "CAN_KEN_pyramid.pdf"))
```

Change in age structures over time:

```
d_long |>
  filter(region=="Kenya" | region=="Canada", year %in% c(1960, 1990, 2015, 2020)) |>
  mutate(population = ifelse(sex == "Male", -pop, pop)) %>%
  ggplot(aes(x = age, y = population, fill = sex)) +
  facet_grid(year ~ region) +
  geom_bar(stat = "identity") +
  ggtitle("Population in each age group") +
  ylab("Population") +
  coord_flip() +
  scale_y_continuous(breaks = seq(-4000, 4000, 1000),
                     labels = c(seq(4000, 0, -1000), seq(1000, 4000, 1000))) +
  scale_fill_brewer(palette = "Set1")
```



```
ggsave(here("plots", "CAN_KEN_pyramid_time.pdf"))
```

## Age-specific rates

Create and plot age-specific mortality rates (across both sexes)

```
pops <- d_long

dm_long <- dm  |>
  pivot_longer(x0:x100, names_to = "age", values_to = "deaths") |>
  mutate(age = as.numeric(str_remove(age, "x")))

# join these two tibbles and calculate rates

asmr <- d_long |>
  left_join(dm_long) |>
  mutate(mx = deaths/pop)

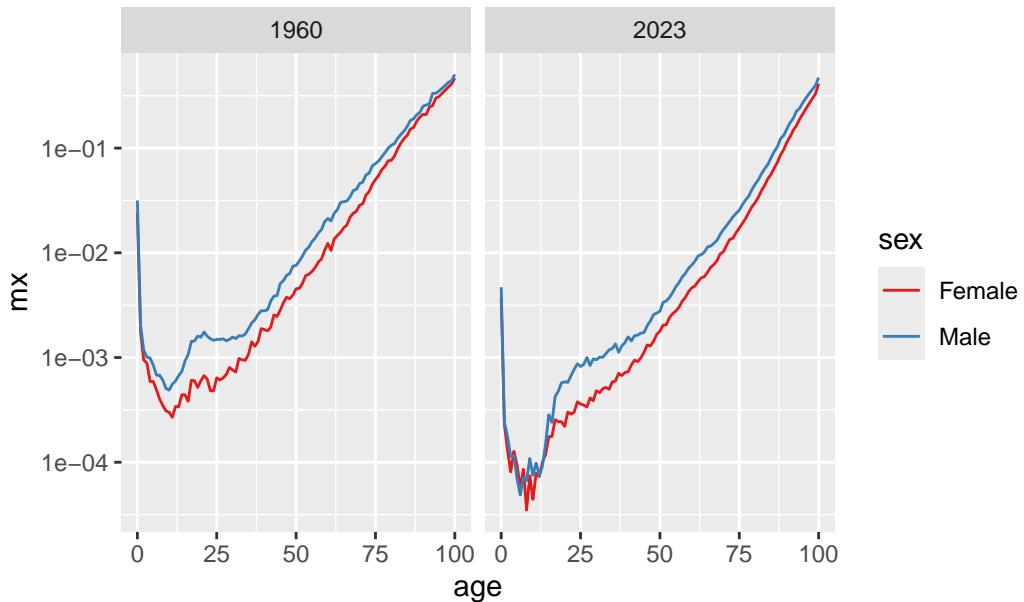
head(asmr)
```

```
# A tibble: 6 x 8
  region iso3_alpha_code year sex     age   pop deaths      mx
  <chr>    <lgcl>        <dbl> <chr> <dbl> <dbl> <dbl>    <dbl>
1 World     NA            1950 Male     0 41603. 6779. 0.163
2 World     NA            1950 Male     1 36997. 1815. 0.0491
3 World     NA            1950 Male     2 34245. 967.  0.0282
4 World     NA            1950 Male     3 32359. 588.  0.0182
5 World     NA            1950 Male     4 30007. 380.  0.0127
6 World     NA            1950 Male     5 28065. 255.  0.00909
```

Mortality rates for Canada

```
asmr |>
  filter(region == "Canada", year %in% c(1960, 2023)) |>
  ggplot(aes(age, mx, color = sex)) +
  geom_line()+
  scale_y_log10()+
  facet_wrap(~year)+
  labs(title ="Age- and sex-specific mortality rates for Canada")+
  scale_color_brewer(palette = "Set1")
```

## Age- and sex-specific mortality rates for Canada



```
ggsave(here("plots", "CAN_mortality.pdf"), width = 6, height = 4)
```

## Age-standardized rates

What would Kenya mortality look like in 2023 with Canada's age structure?

```
kenya_2023 <- asmr |>
  filter(year==2023, region=="Kenya") |>
  rename(kpop = pop, kdeath = deaths, kmx = mx) |>
  select(-region)

canada_2023 <- asmr |>
  ungroup() |>
  filter(year==2023, region=="Canada") |>
  rename(cpop = pop, cdeath = deaths, cmx = mx) |>
  select(-region, -age, -year, -sex, -iso3_alpha_code)

kc_2023 <- bind_cols(kenya_2023, canada_2023)

# now calculate age-standardized rates using canada's population
```

```
kc_2023 |>
  mutate(std_deaths_kenya = cpop*kmx,
        std_deaths_canada = cpop*cmx) |>
  summarise(std_rate_kenya = sum(std_deaths_kenya)/sum(cpop),
            std_rate_canada = sum(std_deaths_canada)/sum(cpop))
```

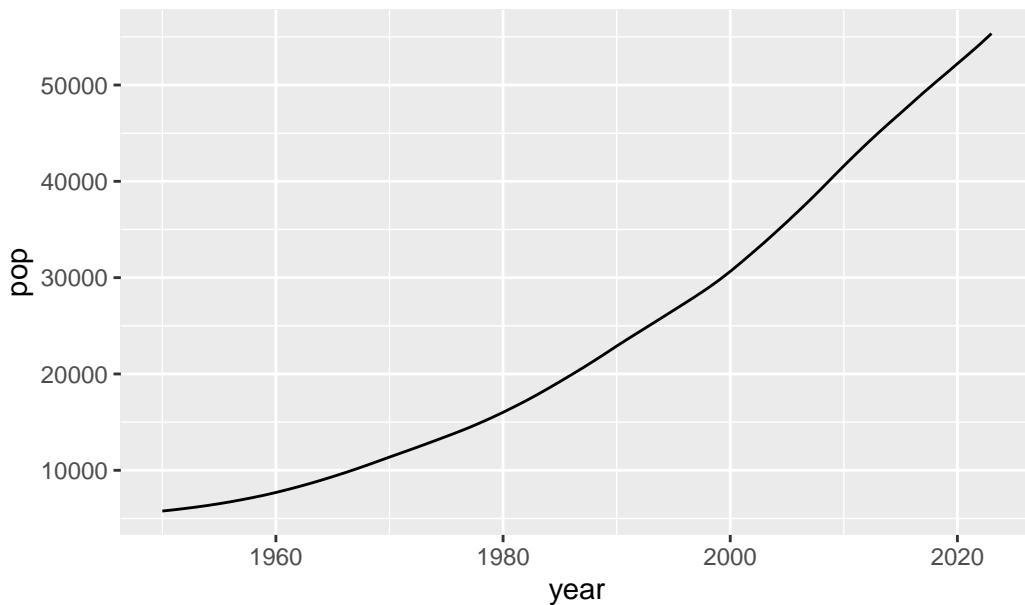
```
# A tibble: 1 x 2
  std_rate_kenya std_rate_canada
  <dbl>          <dbl>
1 0.0223         0.00793
```

## Population Growth

Plot Kenya total population over time

```
d_long |>
  filter(region=="Kenya") |>
  group_by(year) |>
  summarize(pop = sum(pop)) |>
  ggplot(aes(year, pop)) +
  geom_line() +
  ggtitle("Kenya population over time")
```

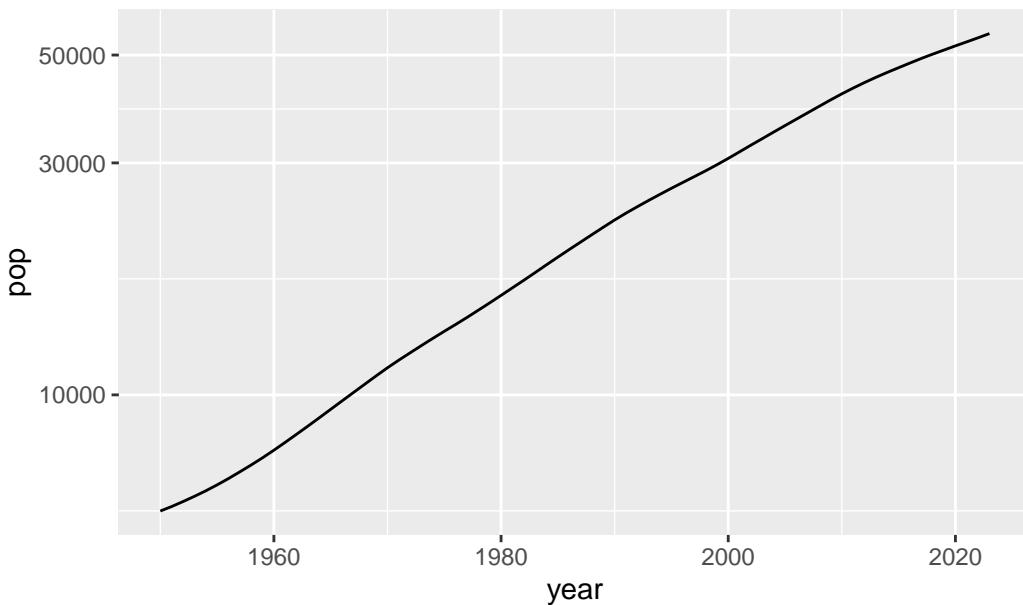
## Kenya population over time



What does the log look like?

```
d_long |>
  filter(region=="Kenya") |>
  group_by(year) |>
  summarize(pop = sum(pop)) |>
  ggplot(aes(year, pop)) +
  geom_line() +
  ggtitle("Kenya population over time")+
  scale_y_log10()
```

## Kenya population over time



```
ggsave(here("plots", "KEN_pop.pdf"))
```

Pretty straight! Let's calculate the growth rate from 1950 to 2015. This is just the slope of the logged graph. It's about 3% per year.

```
d_long |>
  filter(region=="Kenya") |>
  group_by(year) |>
  summarize(pop = sum(pop)) |>
  mutate(log_pop = log(pop))|>
  summarise(growth_rate = (log_pop[year==2023] - log_pop[year==1950])/(2023-1950))
```

```
# A tibble: 1 x 1
  growth_rate
  <dbl>
1      0.0310
```

What about Canada? About half that.

```

d_long |>
  filter(region=="Canada") |>
  group_by(year) |>
  summarize(pop = sum(pop)) |>
  mutate(log_pop = log(pop))|>
  summarise(growth_rate = (log_pop[year==2023] - log_pop[year==1950])/(2023-1950))

# A tibble: 1 x 1
  growth_rate
  <dbl>
1      0.0144

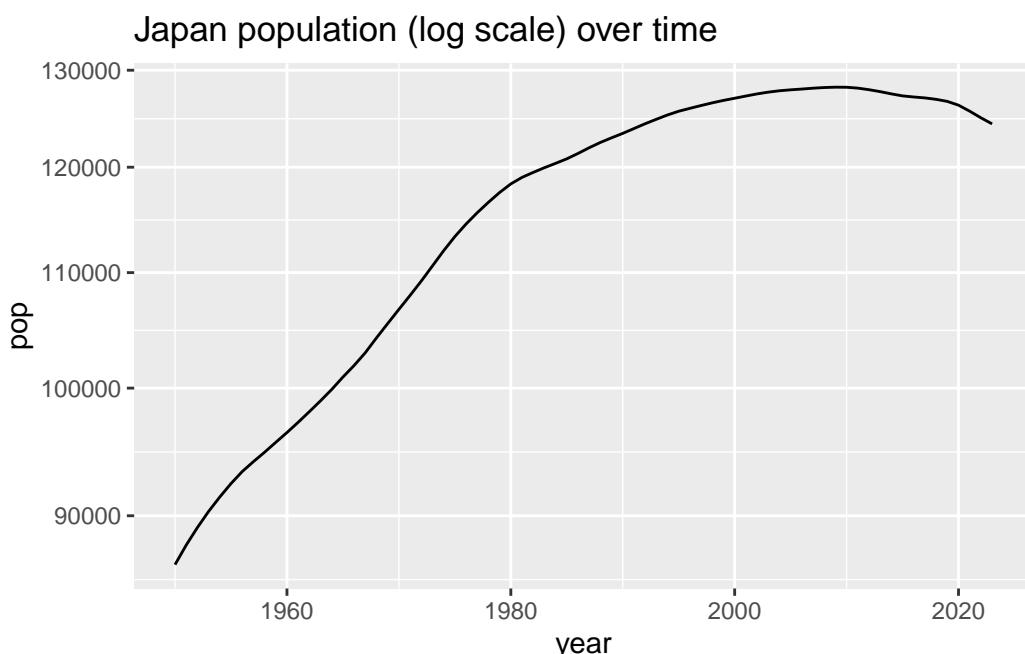
```

Some countries have stagnated:

```

d_long |>
  filter(region=="Japan") |>
  group_by(year) |>
  summarize(pop = sum(pop)) |>
  ggplot(aes(year, pop)) + geom_line()+
  scale_y_log10()+
  ggtitle("Japan population (log scale) over time")

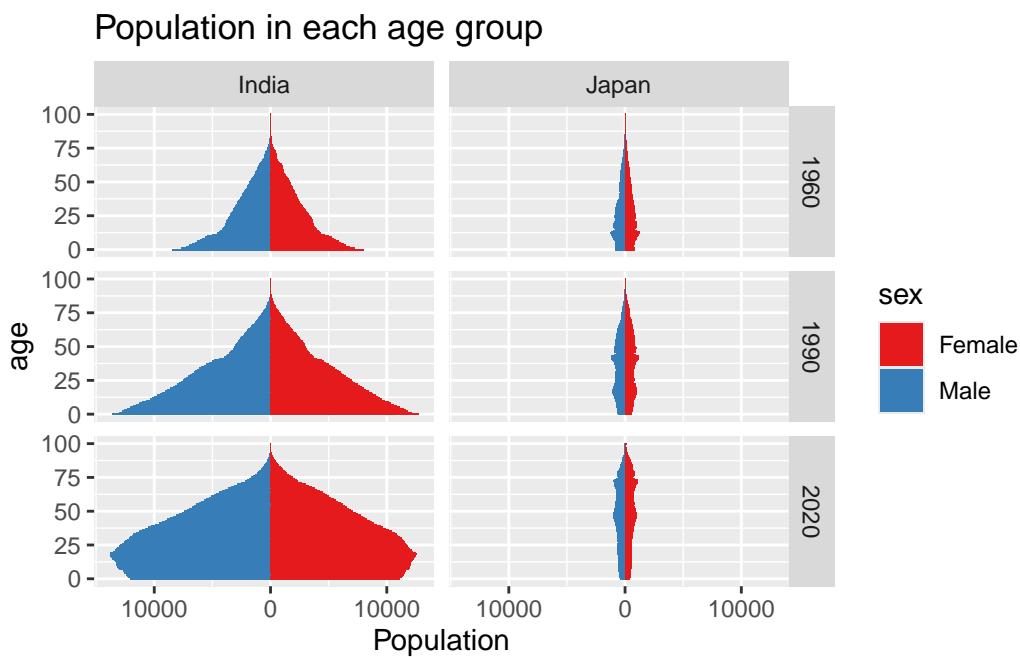
```



## Questions

- Pick two countries. Plot their population pyramids in 1960, 1990, and 2020. I select India and Japan.

```
d_long |>
  filter(region=="India" | region=="Japan", year%in% c(1960, 1990, 2020)) |>
  mutate(population=ifelse(sex=="Male", -pop, pop)) %>%
  ggplot(aes(x = age, y = population, fill = sex)) +
  facet_grid(year~region) +
  geom_bar(stat="identity") +
  ggtitle("Population in each age group") +
  ylab("Population") +
  coord_flip() +
  scale_y_continuous(breaks = seq(-40000, 40000, 10000),
                     labels = c(seq(40000, 0, -10000), seq(10000, 40000, 10000))) +
  scale_fill_brewer(palette = "Set1")
```

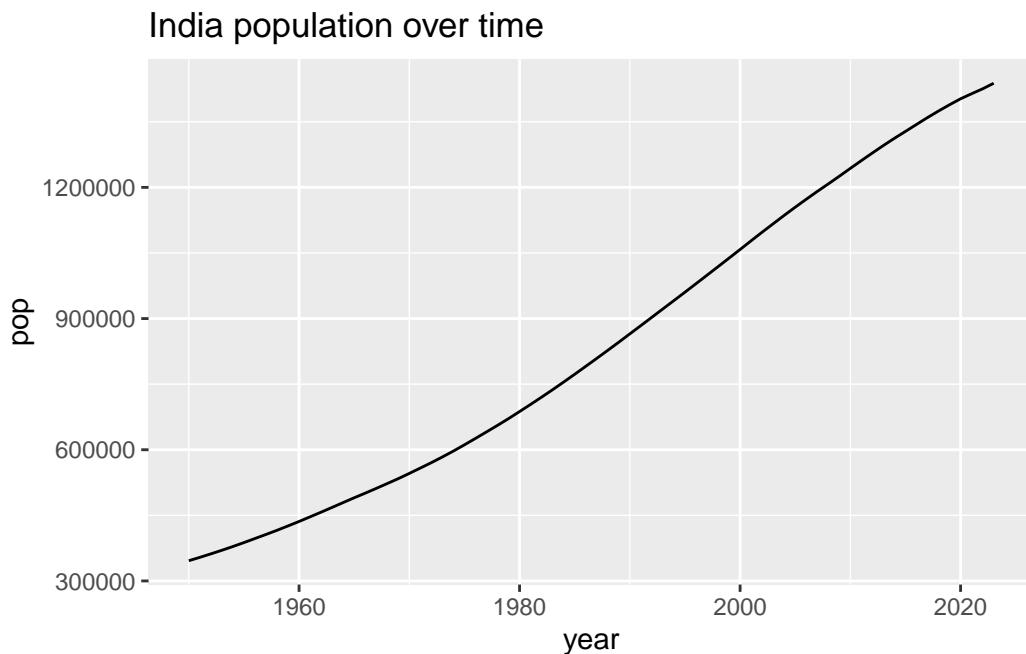


```
ggsave(here("plots", "INA_JAP_pyramid_time.pdf"))
```

- Based on the shape and change in population pyramids in 1, do you think the population growth rate in each of your chosen countries is positive or negative in recent years?  
India:positive Japan:negative

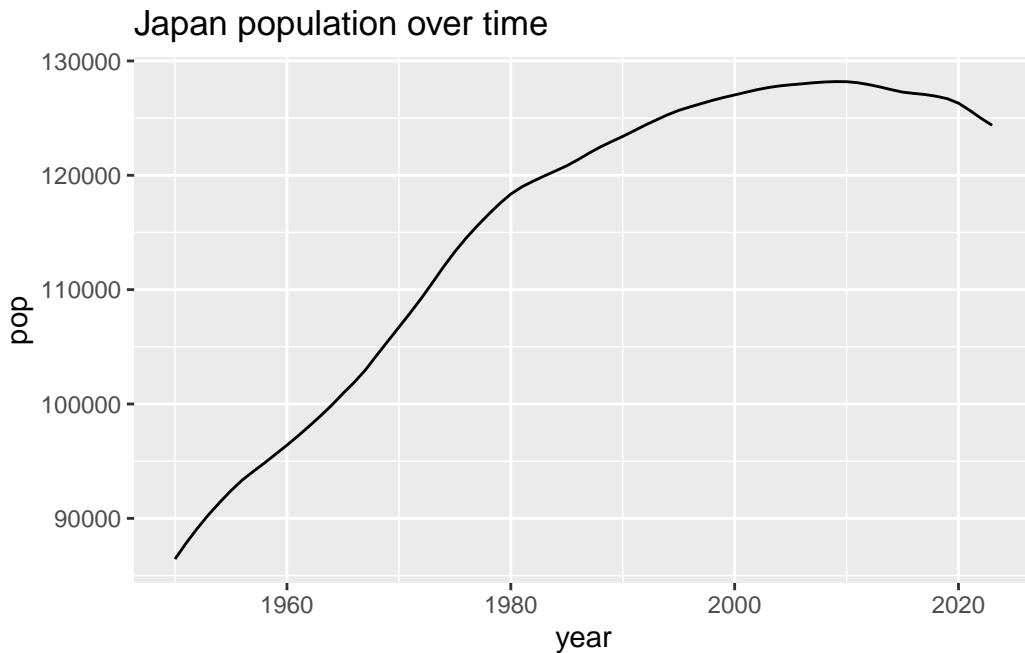
3. Confirm your guesses in 2 by plotting population over time in both countries. plot population growth curve and then compare them. India:

```
d_long |>
  filter(region=="India") |>
  group_by(year) |>
  summarize(pop = sum(pop)) |>
  ggplot(aes(year, pop)) +
  geom_line() +
  ggttitle("India population over time")
```



Japan:

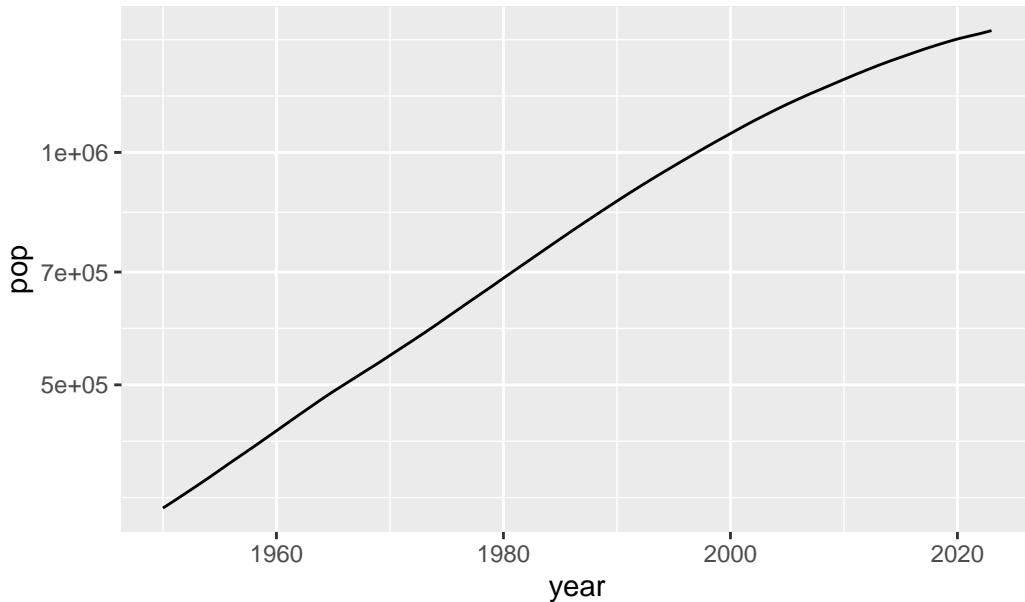
```
d_long |>
  filter(region=="Japan") |>
  group_by(year) |>
  summarize(pop = sum(pop)) |>
  ggplot(aes(year, pop)) +
  geom_line() +
  ggttitle("Japan population over time")
```



we can also look at log plot India

```
d_long |>
  filter(region=="India") |>
  group_by(year) |>
  summarize(pop = sum(pop)) |>
  ggplot(aes(year, pop)) +
  geom_line() +
  ggtitle("India population over time")+
  scale_y_log10()
```

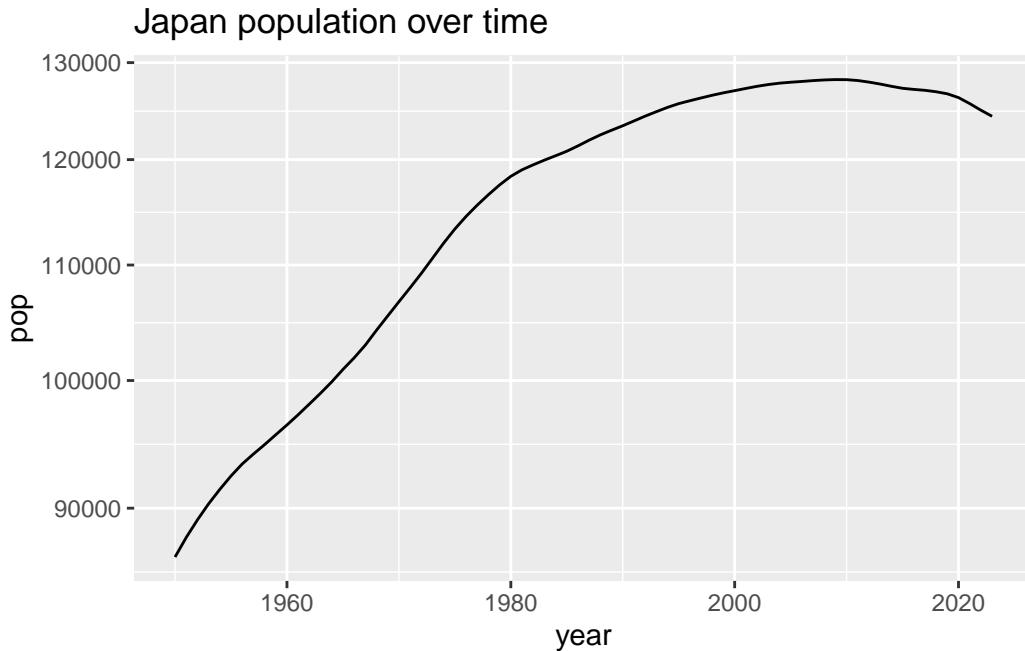
## India population over time



```
ggsave(here("plots", "IND_pop.pdf"))
```

Japan

```
d_long |>
  filter(region=="Japan") |>
  group_by(year) |>
  summarize(pop = sum(pop)) |>
  ggplot(aes(year, pop)) +
  geom_line() +
  ggtitle("Japan population over time")+
  scale_y_log10()
```



```
ggsave(here("plots", "JAP_pop.pdf"))
```

4. Taking the US population in 2000 as the standard population, calculate the standardized mortality rates for US in 2023 and Australia in 2023. How much higher was the death rate in US compared to Australia?

I calculate age-standardized mortality rates. age-specific mortality rates asmr have been obtained from above analyses. Answer: Death rate in US is 1.2 times higher compared to Australia

```
australia_2000 <- asmr |>
  filter(year==2000, region=="Australia") |>
  rename(apop = pop, adeath = deaths, amx = mx) |>
  select(-region)

usa_2000 <- asmr |>
  ungroup() |>
  filter(year==2000, region=="United States of America") |>
  rename(upop = pop, udeath = deaths, umx = mx) |>
  select(-region, -age, -year, -sex, -iso3_alpha_code)

au_2000 <- bind_cols(australia_2000, usa_2000)
```

```
# now calculate age-standardized rates using usa's population

au_2000 |>
  mutate(std_deaths_australia = upop*amx,
         std_deaths_usa = upop*umx) |>
  summarise(std_rate_australia = sum(std_deaths_australia)/sum(upop),
             std_rate_usa = sum(std_deaths_usa)/sum(upop))
```

```
# A tibble: 1 x 2
  std_rate_australia std_rate_usa
  <dbl>           <dbl>
1     0.00711        0.00861
```

Thank you for everything! I learn a lot from your class.