

gganimate Tasks

Team Blue

4/13/2022

We will need the following packages for our exercises.

```
library(tidyverse)
library(ggplot2)
library(gganimate)
library(gapminder)
library(gifs)
library(scales)
library(readr)
```

Exercise 1: Measles heat map using transition_time Great work!

Recall from earlier in the semester when we worked on the measles assignment. Our assignment from last time was to construct a grid plot of incidence of measles cases over time, and conduct a comparison of the incidence before and after when the vaccines were introduced.

For the sake of brevity and time, our intention today with this exercise is to use the `transition_time()` element of the `gganimate` package to enhance a `geom_point()` plot to show the incidence of measles cases over time.

Below you will find a quick-and-dirty `geom_point` plot of measles cases.

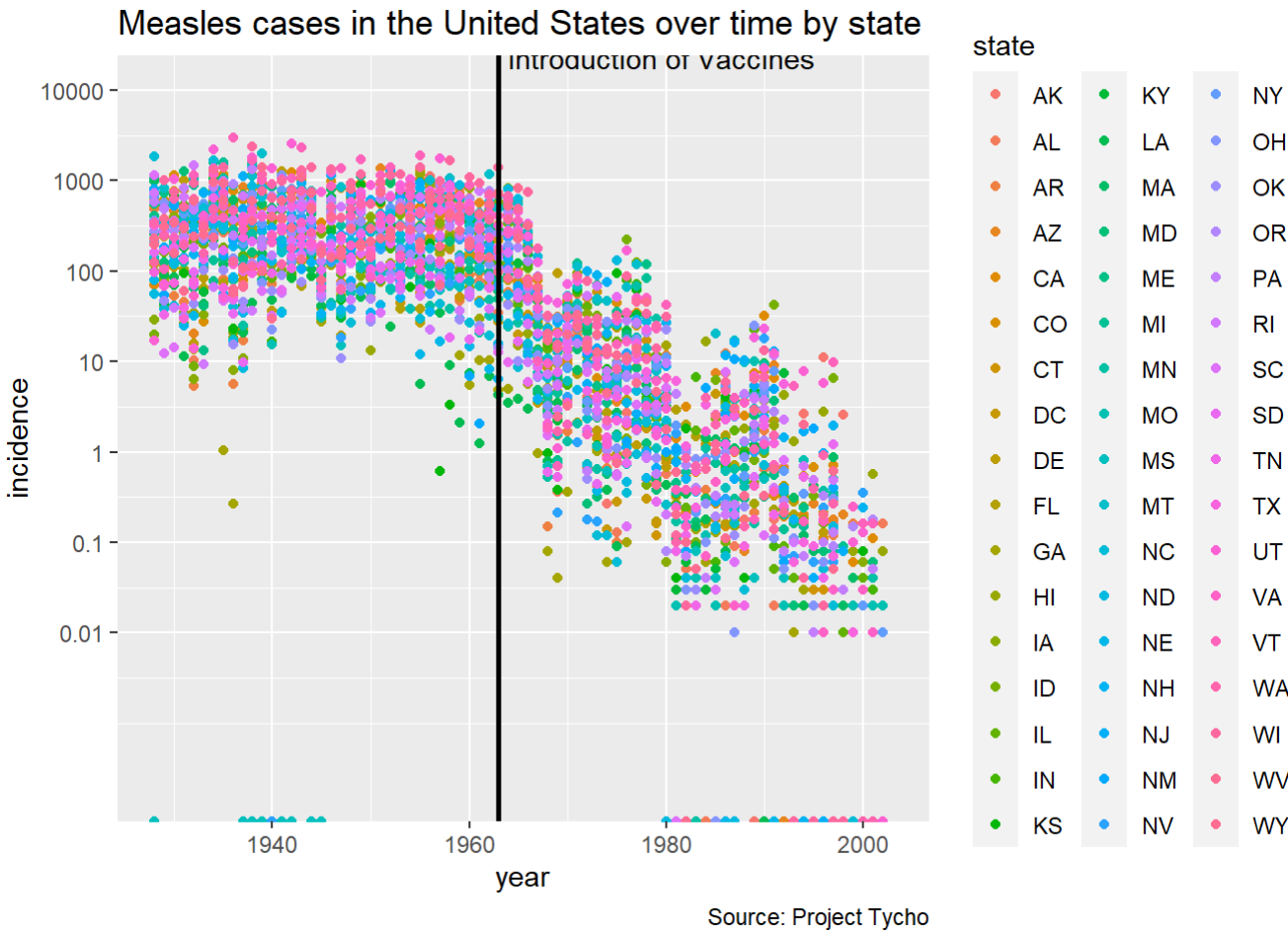
```
measles <- read_csv("measles_for_ggplot2.csv")

measles_plot <- ggplot(measles, aes(year, incidence)) +
  geom_point(aes(colour = state)) +
  labs(
    title = "Measles cases in the United States over time by state",
    caption = "Source: Project Tycho"
  ) +
  scale_y_log10(limits = c(2e-4, NA), breaks = 10^(-2:4), labels = prettyNum) +
  geom_vline(xintercept = 1963, size = 1) +
  annotate(
    geom = "text", label = "Introduction of Vaccines",
    x = 1950, y = 10000, hjust = -0.45, vjust = -1
  )

measles_plot
```

```
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 543 rows containing missing values (geom_point).
```



Your task is to build upon the above code.

- a. Convert the above code from a static plot to an animated plot, using the `transition_time()` element of the `gganimate` package.

```
measles_plot +
  transition_time(year)
```

```
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 4 rows containing missing values (geom_point).
## Removed 4 rows containing missing values (geom_point).

## Warning: Removed 5 rows containing missing values (geom_point).
## Removed 5 rows containing missing values (geom_point).
## Removed 5 rows containing missing values (geom_point).
## Removed 5 rows containing missing values (geom_point).

## Warning: Removed 4 rows containing missing values (geom_point).
## Removed 4 rows containing missing values (geom_point).
## Removed 4 rows containing missing values (geom_point).
## Removed 4 rows containing missing values (geom_point).
## Removed 4 rows containing missing values (geom_point).
## Removed 4 rows containing missing values (geom_point).
## Removed 4 rows containing missing values (geom_point).
## Removed 4 rows containing missing values (geom_point).

## Warning: Removed 3 rows containing missing values (geom_point).

## Warning: Removed 4 rows containing missing values (geom_point).

## Warning: Removed 3 rows containing missing values (geom_point).

## Warning: Removed 2 rows containing missing values (geom_point).
## Removed 2 rows containing missing values (geom_point).
## Removed 2 rows containing missing values (geom_point).

## Warning: Removed 3 rows containing missing values (geom_point).
## Removed 3 rows containing missing values (geom_point).

## Warning: Removed 2 rows containing missing values (geom_point).
## Removed 2 rows containing missing values (geom_point).

## Warning: Removed 3 rows containing missing values (geom_point).
## Removed 3 rows containing missing values (geom_point).

## Warning: Removed 2 rows containing missing values (geom_point).
## Removed 2 rows containing missing values (geom_point).
## Removed 2 rows containing missing values (geom_point).
## Removed 2 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_point).
## Removed 1 rows containing missing values (geom_point).

## Warning: Removed 2 rows containing missing values (geom_point).
## Removed 2 rows containing missing values (geom_point).
## Removed 2 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_point).
## Removed 1 rows containing missing values (geom_point).
## Removed 1 rows containing missing values (geom_point).

## Warning: Removed 2 rows containing missing values (geom_point).
## Removed 2 rows containing missing values (geom_point).
## Removed 2 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_point).
## Removed 1 rows containing missing values (geom_point).
## Removed 1 rows containing missing values (geom_point).
## Removed 1 rows containing missing values (geom_point).

## Warning: Removed 3 rows containing missing values (geom_point).
## Removed 3 rows containing missing values (geom_point).
## Removed 3 rows containing missing values (geom_point).
```

```
## Warning: Removed 8 rows containing missing values (geom_point).

## Warning: Removed 14 rows containing missing values (geom_point).

## Warning: Removed 27 rows containing missing values (geom_point).

## Warning: Removed 22 rows containing missing values (geom_point).
## Removed 22 rows containing missing values (geom_point).

## Warning: Removed 16 rows containing missing values (geom_point).

## Warning: Removed 19 rows containing missing values (geom_point).

## Warning: Removed 6 rows containing missing values (geom_point).

## Warning: Removed 15 rows containing missing values (geom_point).

## Warning: Removed 18 rows containing missing values (geom_point).

## Warning: Removed 19 rows containing missing values (geom_point).

## Warning: Removed 4 rows containing missing values (geom_point).

## Warning: Removed 5 rows containing missing values (geom_point).

## Warning: Removed 11 rows containing missing values (geom_point).

## Warning: Removed 15 rows containing missing values (geom_point).

## Warning: Removed 30 rows containing missing values (geom_point).

## Warning: Removed 24 rows containing missing values (geom_point).

## Warning: Removed 18 rows containing missing values (geom_point).

## Warning: Removed 23 rows containing missing values (geom_point).

## Warning: Removed 32 rows containing missing values (geom_point).

## Warning: Removed 16 rows containing missing values (geom_point).

## Warning: Removed 6 rows containing missing values (geom_point).

## Warning: Removed 28 rows containing missing values (geom_point).

## Warning: Removed 46 rows containing missing values (geom_point).

## Warning: Removed 31 rows containing missing values (geom_point).

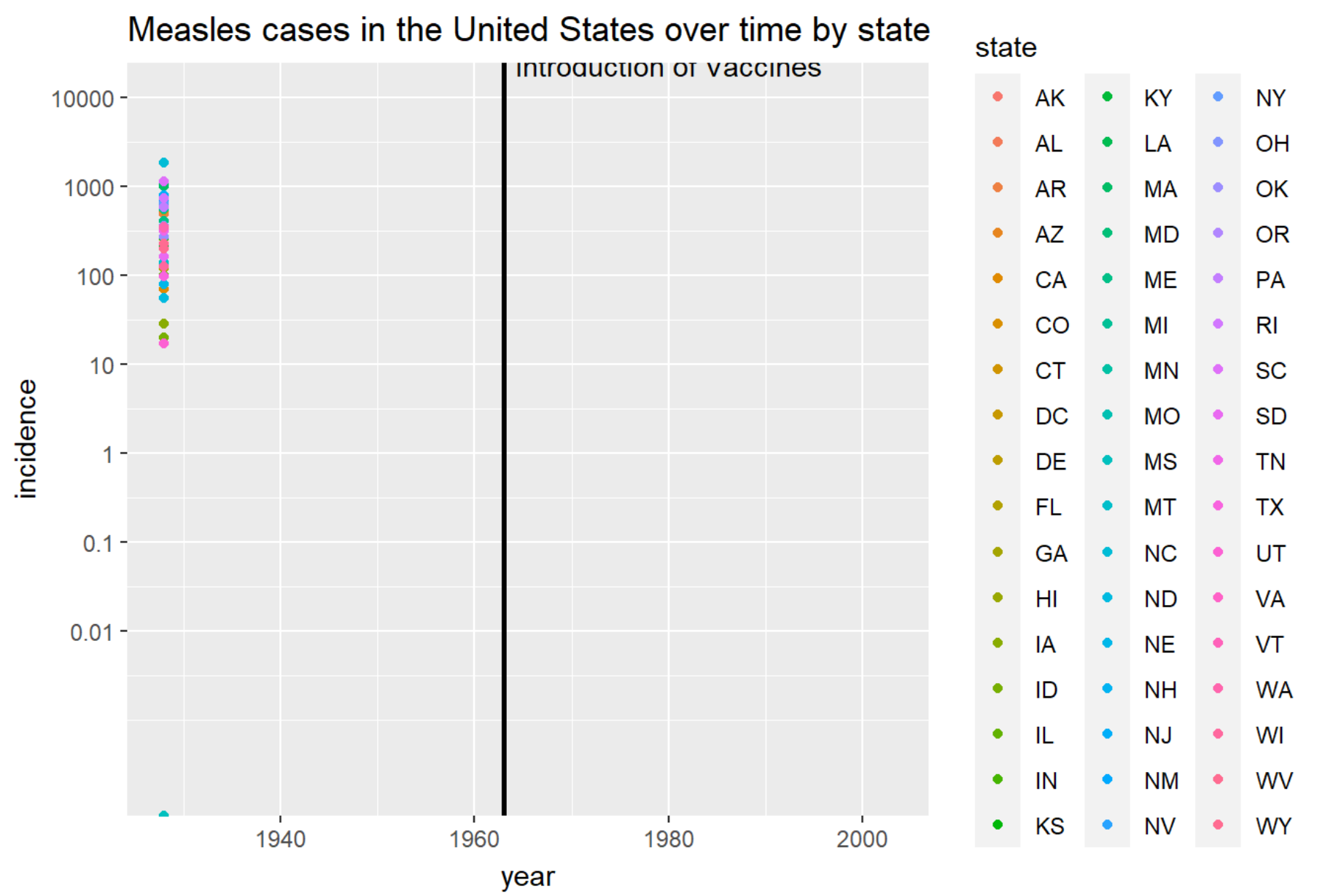
## Warning: Removed 30 rows containing missing values (geom_point).

## Warning: Removed 28 rows containing missing values (geom_point).

## Warning: Removed 43 rows containing missing values (geom_point).

## Warning: Removed 37 rows containing missing values (geom_point).

## Warning: Removed 51 rows containing missing values (geom_point).
```



Source: Project Tycho

b. Add a subtitle to the code with a frame-time element so we know which year each animation refers to, in order to make a good comparison.

The resulting plot should see the geom_points moving vertically as the measles incidence changes and horizontally as the year changes

Note: the code can take around 90 seconds to run, so be sure of your answer before you go ahead and run, otherwise you may risk running out of time! You may also wish to work on other exercises while the code runs to save time.

```
measles_plot +
  transition_time(as.integer(year)) +
  labs(subtitle = 'Year: {frame_time}')
```

```
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 4 rows containing missing values (geom_point).
## Removed 4 rows containing missing values (geom_point).

## Warning: Removed 5 rows containing missing values (geom_point).
## Removed 5 rows containing missing values (geom_point).
## Removed 5 rows containing missing values (geom_point).
## Removed 5 rows containing missing values (geom_point).

## Warning: Removed 4 rows containing missing values (geom_point).
## Removed 4 rows containing missing values (geom_point).
## Removed 4 rows containing missing values (geom_point).
## Removed 4 rows containing missing values (geom_point).
## Removed 4 rows containing missing values (geom_point).
## Removed 4 rows containing missing values (geom_point).
## Removed 4 rows containing missing values (geom_point).

## Warning: Removed 3 rows containing missing values (geom_point).

## Warning: Removed 4 rows containing missing values (geom_point).

## Warning: Removed 3 rows containing missing values (geom_point).

## Warning: Removed 2 rows containing missing values (geom_point).
## Removed 2 rows containing missing values (geom_point).
## Removed 2 rows containing missing values (geom_point).

## Warning: Removed 3 rows containing missing values (geom_point).
## Removed 3 rows containing missing values (geom_point).

## Warning: Removed 2 rows containing missing values (geom_point).
## Removed 2 rows containing missing values (geom_point).
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
## Removed 3 rows containing missing values (geom_point).

## Warning: Removed 2 rows containing missing values (geom_point).
## Removed 2 rows containing missing values (geom_point).
## Removed 2 rows containing missing values (geom_point).
## Removed 2 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_point).
## Removed 1 rows containing missing values (geom_point).

## Warning: Removed 2 rows containing missing values (geom_point).
## Removed 2 rows containing missing values (geom_point).
## Removed 2 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_point).
## Removed 1 rows containing missing values (geom_point).
## Removed 1 rows containing missing values (geom_point).
## Removed 1 rows containing missing values (geom_point).

## Warning: Removed 2 rows containing missing values (geom_point).
## Removed 2 rows containing missing values (geom_point).
## Removed 2 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_point).
## Removed 1 rows containing missing values (geom_point).
## Removed 1 rows containing missing values (geom_point).
## Removed 1 rows containing missing values (geom_point).
## Removed 1 rows containing missing values (geom_point).

## Warning: Removed 3 rows containing missing values (geom_point).
## Removed 3 rows containing missing values (geom_point).
## Removed 3 rows containing missing values (geom_point).

## Warning: Removed 8 rows containing missing values (geom_point).

## Warning: Removed 14 rows containing missing values (geom_point).

## Warning: Removed 27 rows containing missing values (geom_point).

## Warning: Removed 22 rows containing missing values (geom_point).
## Removed 22 rows containing missing values (geom_point).

## Warning: Removed 16 rows containing missing values (geom_point).

## Warning: Removed 19 rows containing missing values (geom_point).

## Warning: Removed 6 rows containing missing values (geom_point).

## Warning: Removed 15 rows containing missing values (geom_point).

## Warning: Removed 18 rows containing missing values (geom_point).

## Warning: Removed 19 rows containing missing values (geom_point).

## Warning: Removed 4 rows containing missing values (geom_point).

## Warning: Removed 5 rows containing missing values (geom_point).

## Warning: Removed 11 rows containing missing values (geom_point).

## Warning: Removed 15 rows containing missing values (geom_point).

## Warning: Removed 30 rows containing missing values (geom_point).

## Warning: Removed 24 rows containing missing values (geom_point).

## Warning: Removed 18 rows containing missing values (geom_point).

## Warning: Removed 23 rows containing missing values (geom_point).

## Warning: Removed 32 rows containing missing values (geom_point).

## Warning: Removed 16 rows containing missing values (geom_point).

## Warning: Removed 6 rows containing missing values (geom_point).

## Warning: Removed 28 rows containing missing values (geom_point).

## Warning: Removed 46 rows containing missing values (geom_point).

## Warning: Removed 31 rows containing missing values (geom_point).

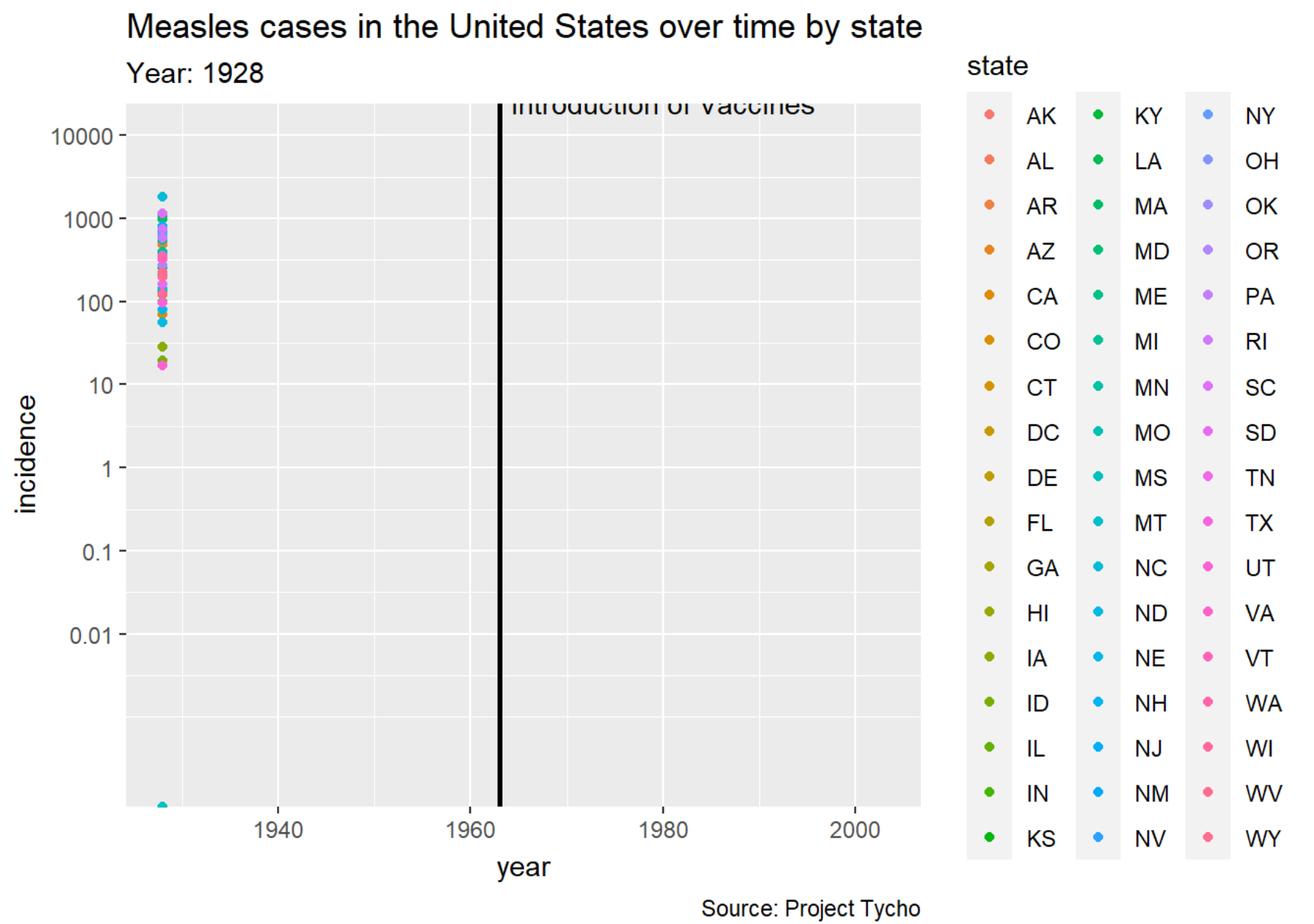
## Warning: Removed 30 rows containing missing values (geom_point).

## Warning: Removed 28 rows containing missing values (geom_point).

## Warning: Removed 43 rows containing missing values (geom_point).

## Warning: Removed 37 rows containing missing values (geom_point).

## Warning: Removed 51 rows containing missing values (geom_point).
```



c. Comment on your plot. What are its advantages and disadvantages? What can be done to improve the plot? You are not expected to implement these improvements into the R code, please write them down in bullet point format. Advantages:

- It is easy to spot the general trend if one does not want to look too closely at the plot and analyze the different data points.
- The line signifying the introduction of vaccines makes it easier to explain why the incidence of measles decreased.

Disadvantages: - While you can see the incidence decreasing generally, you cannot identify the trend of each state. - The animation is too fast to focus on one state, and the colors representing each state are very similar to one another, making it difficult to differentiate them. - A line plot would make it easier to see the trends of each state.

Exercise 2: Singapore population pyramid

We use the following the dataset for this exercise.

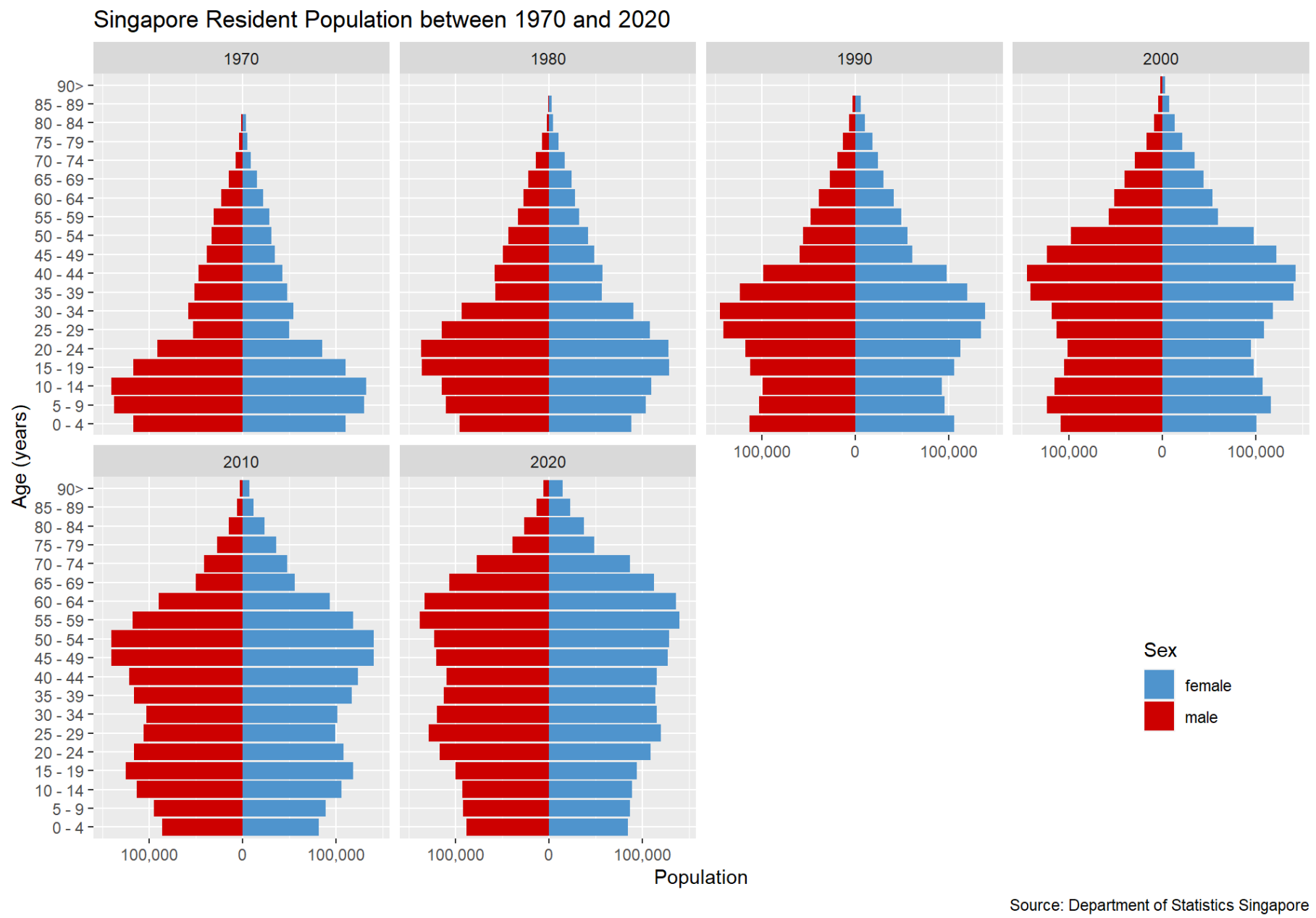
```
data <- read_csv("data.csv")
data <- data |>
  mutate(
    age = factor(
      age,
      levels = c(
        "0 - 4",
        "5 - 9",
        "10 - 14",
        "15 - 19",
        "20 - 24",
        "25 - 29",
        "30 - 34",
        "35 - 39",
        "40 - 44",
        "45 - 49",
        "50 - 54",
        "55 - 59",
        "60 - 64",
        "65 - 69",
        "70 - 74",
        "75 - 79",
        "80 - 84",
        "85 - 89",
        "90+",
      )
    )
  )
```

A few weeks back, we worked on creating a faceted bar graph visualizing the Singapore resident population between 1970 and 2020. The code for that graph is here.

```
p <- ggplot(data, aes(pop2, age, fill = sex)) +
  geom_col() +
  facet_wrap(~year, ncol = 4) +
  labs(
    title = "Singapore Resident Population between 1970 and 2020",
    x = "Population",
    y = "Age (years)",
    caption = "Source: Department of Statistics Singapore"
  ) +
  theme(legend.position = c(0.9, 0.2)) +
  scale_fill_manual(name = "Sex", values = c("steelblue3", "red3"), labels = c("female", "male")) +
  scale_x_continuous(labels = ~ comma(abs(.)))

p
```

Warning: Removed 8 rows containing missing values (position_stack).

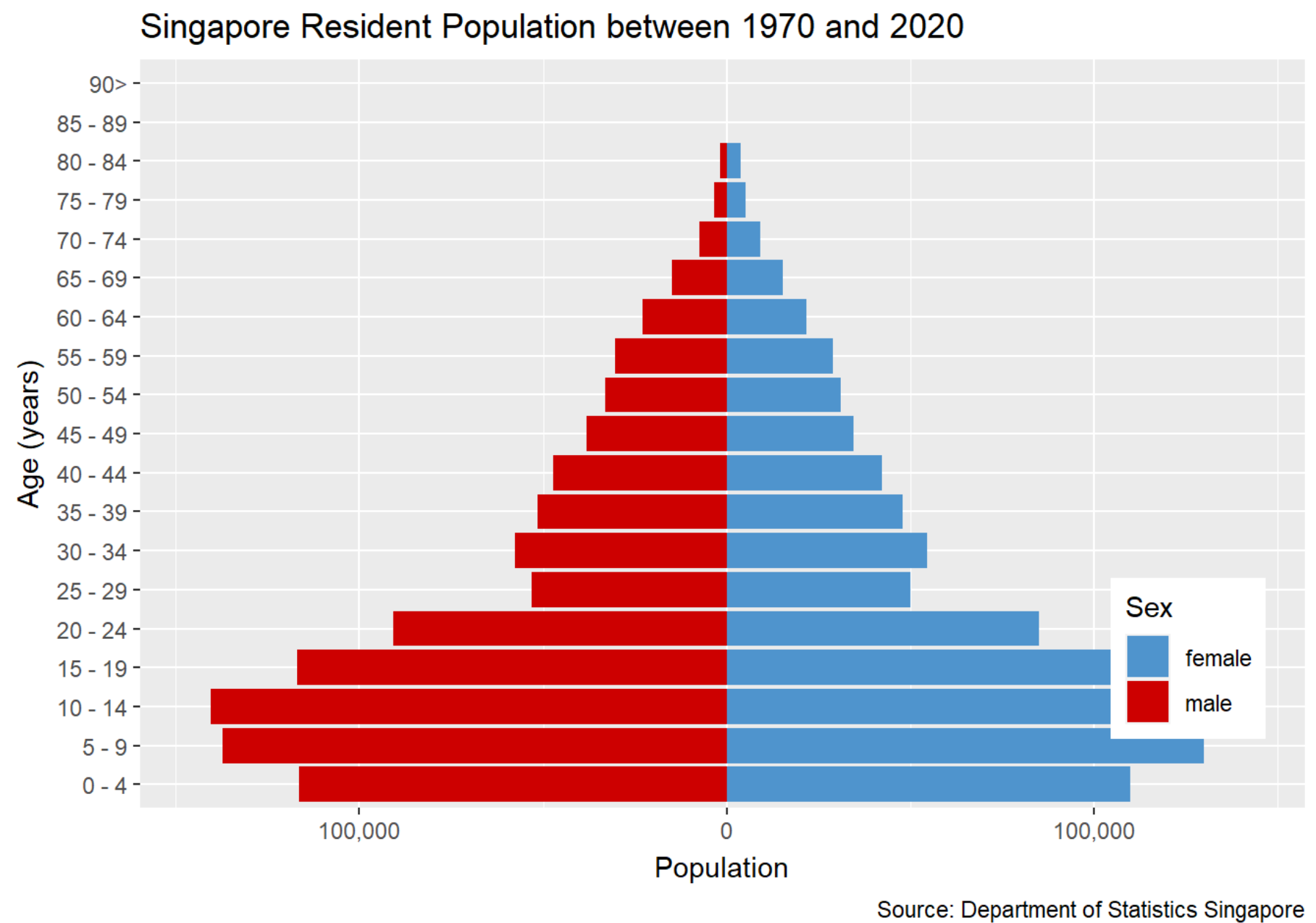


Do the following questions using `gganimate` commands.

a. Show transitions between each year, without intermediate states.

```
ggplot(data, aes(pop2, age, fill = sex)) +
  geom_col() +
  labs(
    title = 'Singapore Resident Population between 1970 and 2020',
    x = "Population",
    y = "Age (years)",
    caption = "Source: Department of Statistics Singapore"
  ) +
  theme(legend.position = c(0.9, 0.2)) +
  scale_fill_manual(name = "Sex", values = c("steelblue3", "red3"), labels = c("female", "male")) +
  scale_x_continuous(labels = ~ comma(abs(.))) +
  transition_manual(year)
```

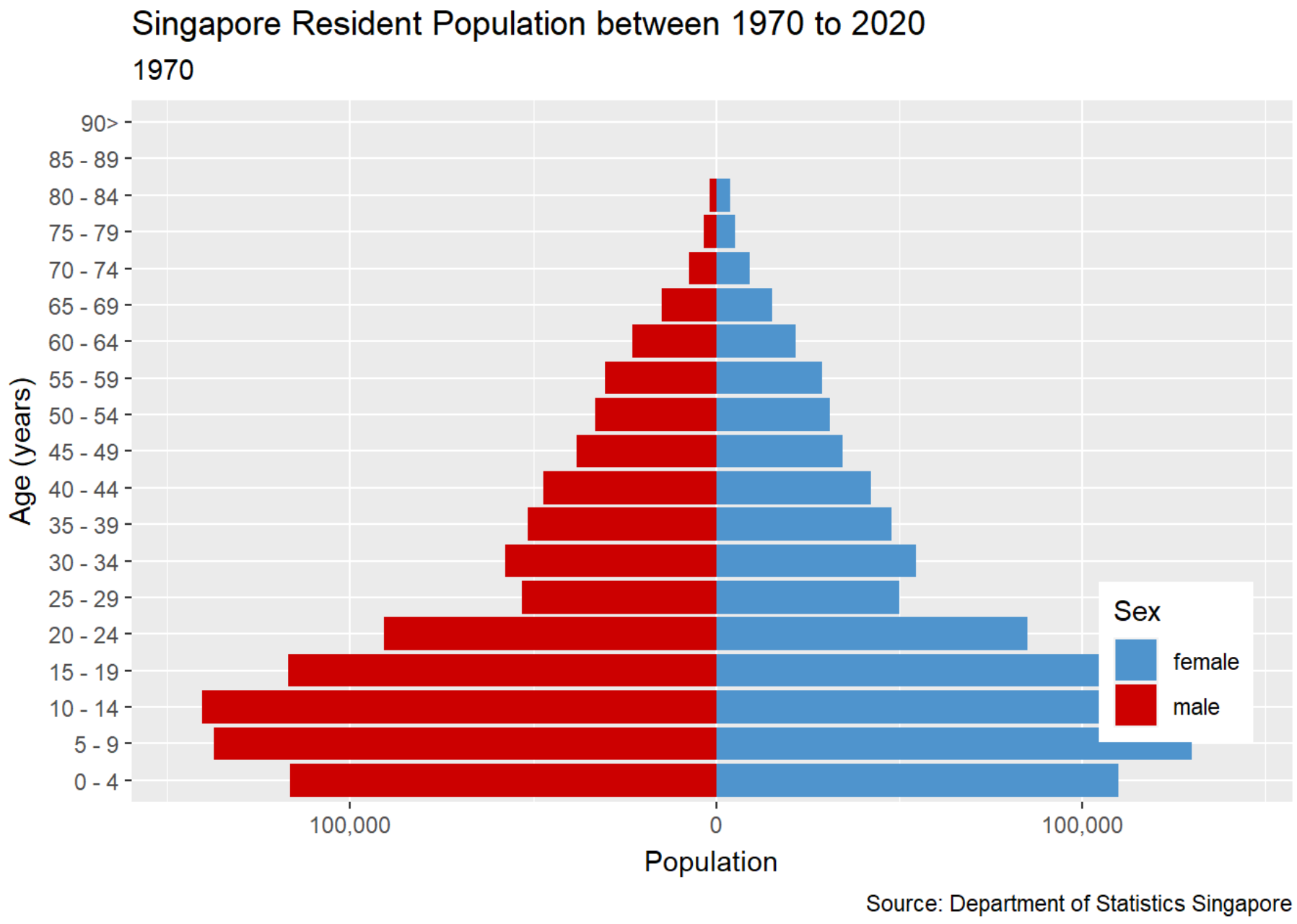
Warning: Removed 8 rows containing missing values (position_stack).



b. Show transitions between each year, with intermediate states. Add subtitle with year of each state shown.

```
ggplot(data, aes(pop2, age, fill = sex)) +
  geom_col() +
  labs(
    title = 'Singapore Resident Population between 1970 to 2020',
    x = "Population",
    y = "Age (years)",
    caption = "Source: Department of Statistics Singapore",
    subtitle = "{frame_time}"
  ) +
  theme(legend.position = c(0.9, 0.2)) +
  scale_fill_manual(name = "Sex", values = c("steelblue3", "red3"), labels = c("female", "male")) +
  scale_x_continuous(labels = ~ comma(abs(.))) +
  transition_time(as.integer(year))
```

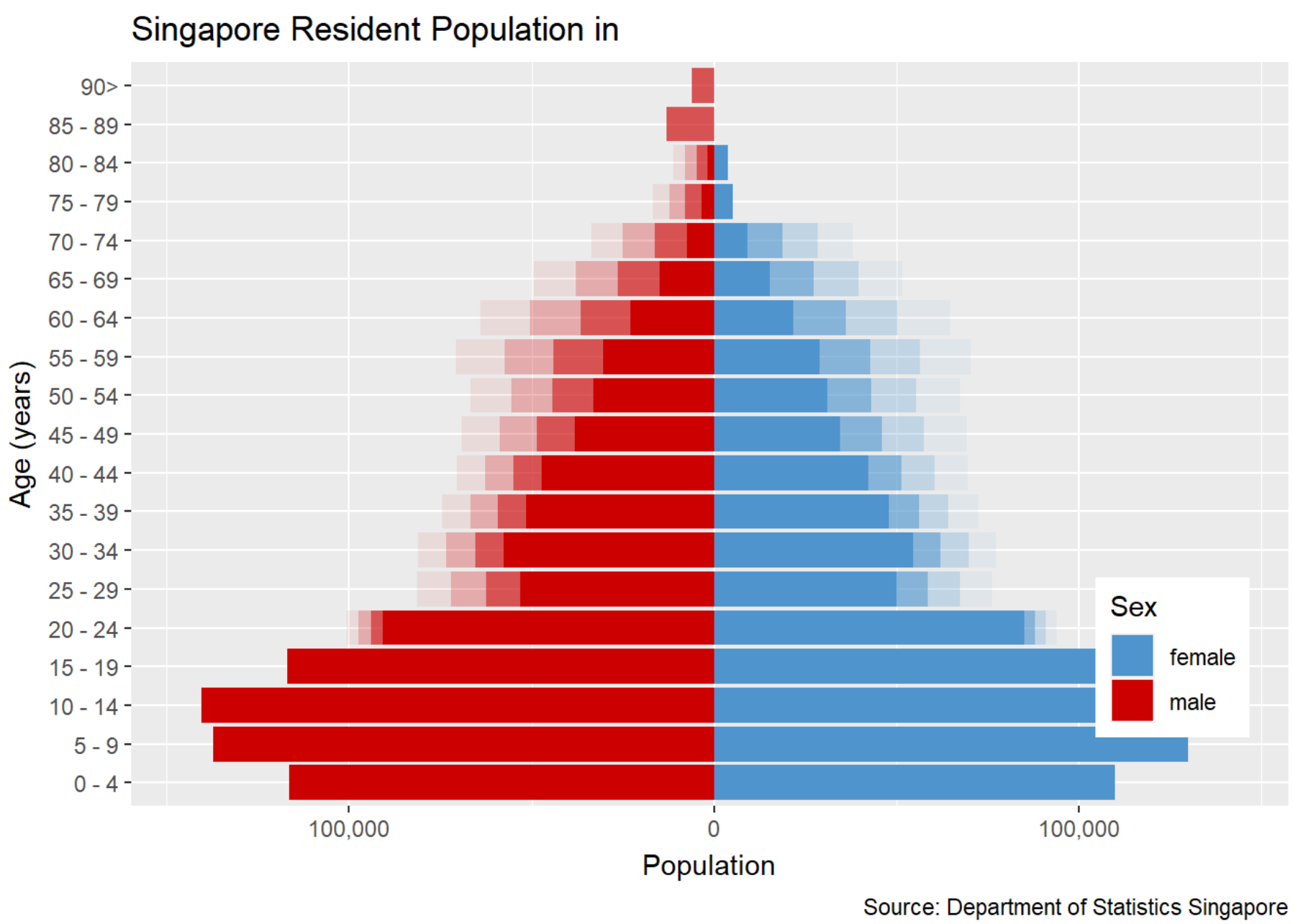
Warning: Removed 8 rows containing missing values (position_stack).



c. Furthermore, add a shadow such that the points have a wake of points with the data from the last 5% of frames.

```
ggplot(data, aes(pop2, age, fill = sex)) +
  geom_col() +
  labs(
    title = 'Singapore Resident Population in',
    x = "Population",
    y = "Age (years)",
    caption = "Source: Department of Statistics Singapore"
  ) +
  theme(legend.position = c(0.9, 0.2)) +
  scale_fill_manual(name = "Sex", values = c("steelblue3", "red3"), labels = c("female", "male")) +
  scale_x_continuous(labels = ~ comma(abs(.))) +
  transition_states(year, transition_length = 4, state_length = 3) +
  shadow_wake(wake_length = 0.05)
```

Warning: Removed 8 rows containing missing values (position_stack).



d. How do the animations add more value than the static, faceted plot?

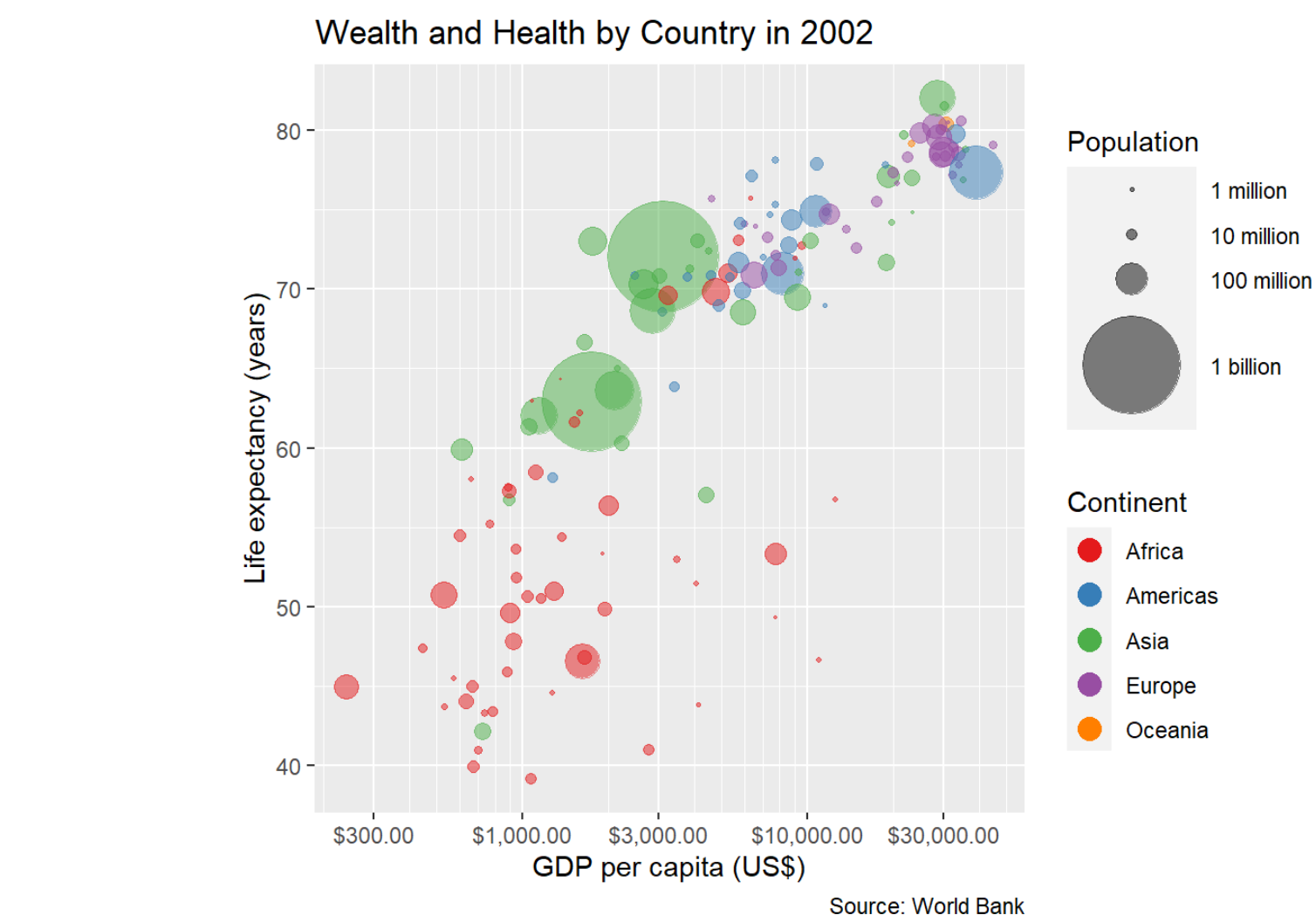
The animations make it easy to compare the population and change in distribution of all age groups at the same time.

Exercise 3: Change of GDP and life expectancy of countries with time

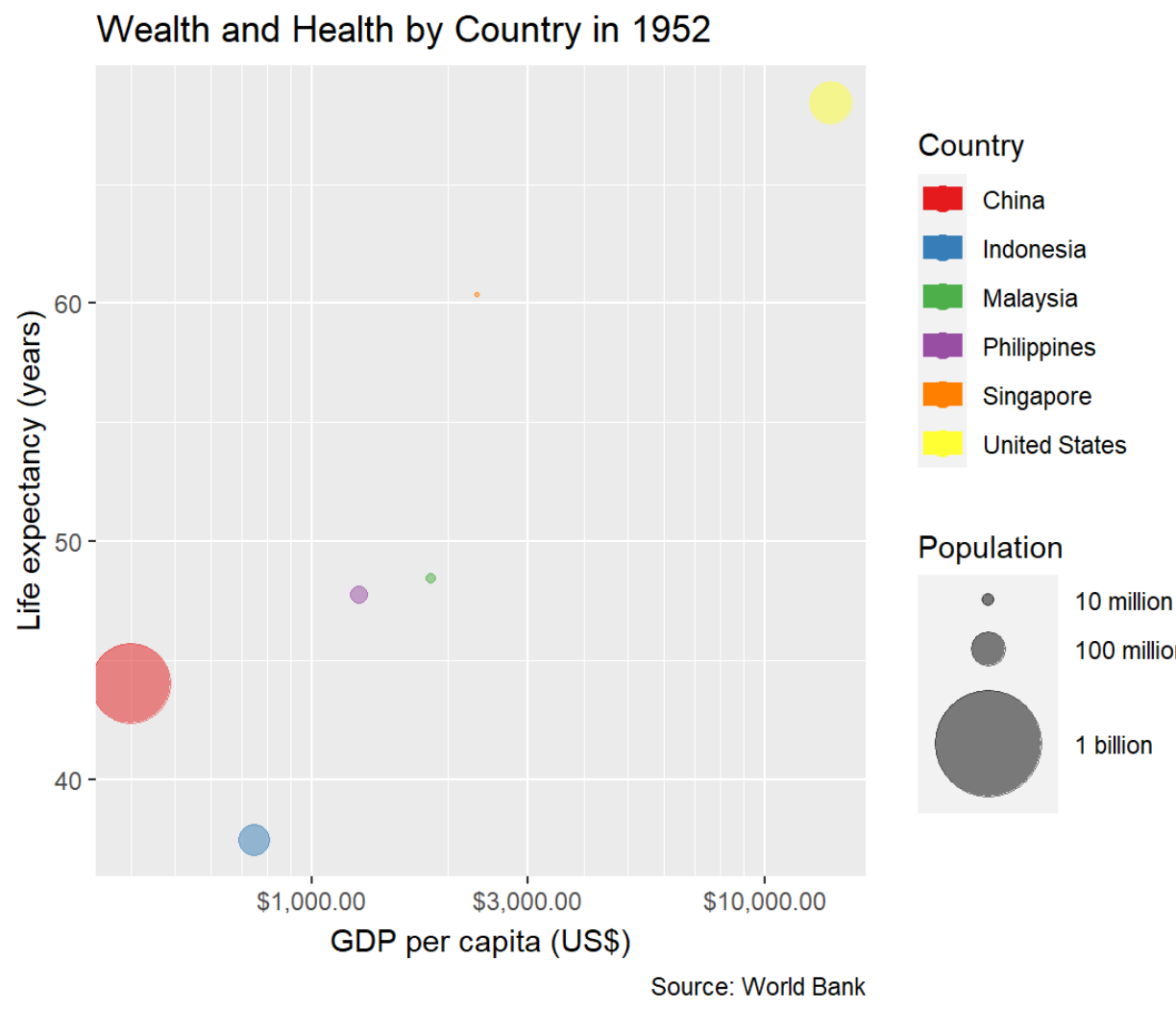
In a previous class, we made a similar plot to the one below using the `gapminder` dataset. The code for this plot is adopted from the Preston Curve assignment.

```
data(gapminder)
gapminder_2002 <- filter(gapminder, year == 2002)

ggplot(gapminder_2002, aes(
  gdpPercap,
  lifeExp,
  size = pop,
  color = continent
)) +
  geom_point(alpha = 0.5) +
  labs(
    title = "Wealth and Health by Country in 2002",
    x = "GDP per capita (US$)",
    y = "Life expectancy (years)",
    caption = "Source: World Bank"
  ) +
  scale_x_log10(labels = dollar, minor_breaks = 2:9 %>% 10^(2:6)) +
  scale_size_area(
    name = "Population",
    breaks = 10^(6:9),
    labels = c("1 million", "10 million", "100 million", "1 billion"),
    max_size = 20
  ) +
  scale_colour_brewer(name = "Continent", palette = "Set1") +
  guides(colour = guide_legend(override.aes = list(alpha = 1, size = 4)))
```

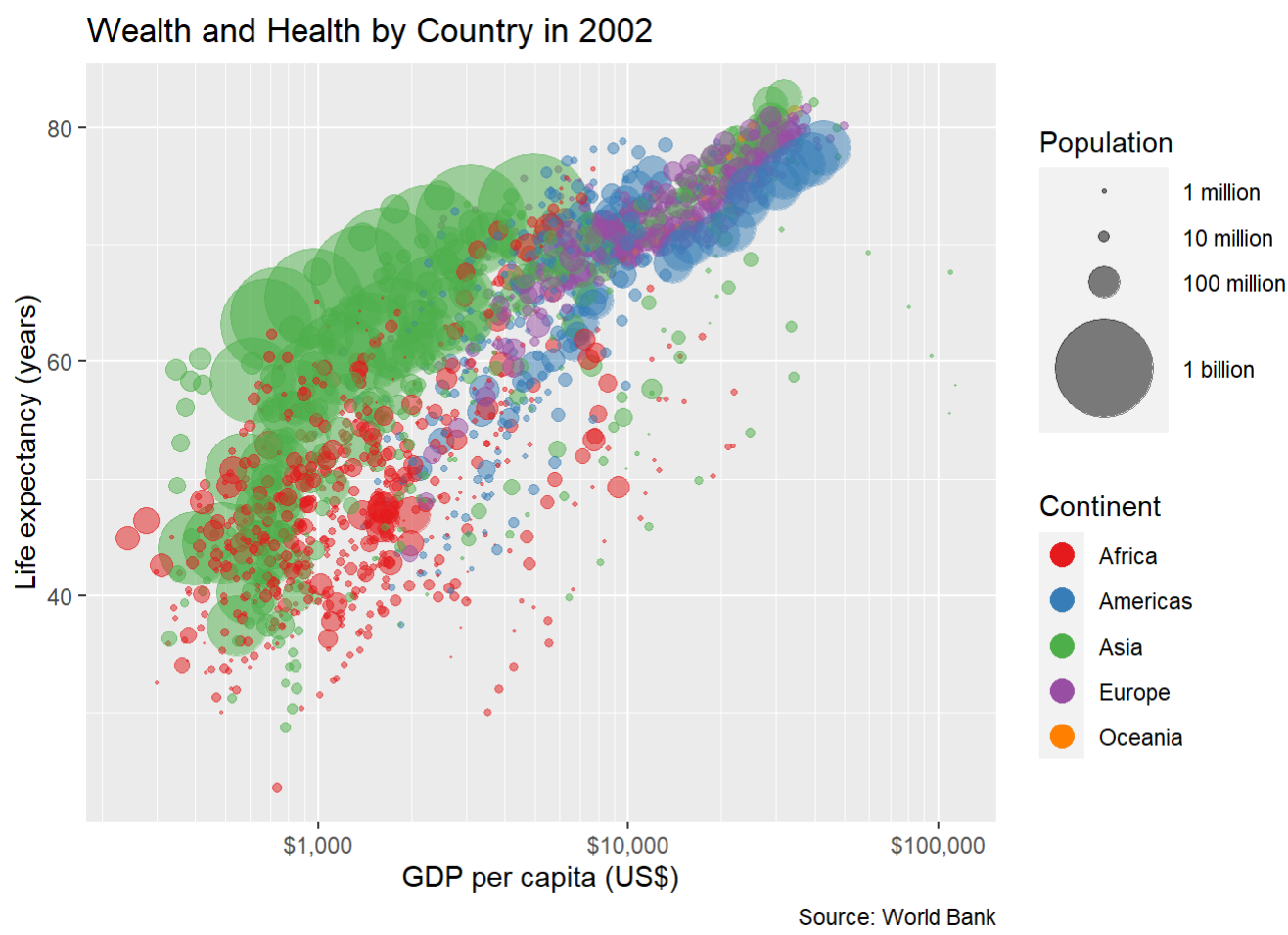


However, the above plot only shows the country statistics for a particular year. In this exercise, we are interested to understand how all the country statistics change with time. The plot below provides a good example.



a. Make a plot of the gapminder data similar to the original Preston Curve above. However, this time, include data for all the years available (not just for the year 2002 as above). You can use the above code as a template for this plot.

```
ggplot(gapminder, aes(
  gdpPercap,
  lifeExp,
  size = pop,
  color = continent
)) +
  geom_point(alpha = 0.5) +
  labs(
    title = "Wealth and Health by Country in 2002",
    x = "GDP per capita (US$)",
    y = "Life expectancy (years)",
    caption = "Source: World Bank"
  ) +
  scale_x_log10(labels = dollar, minor_breaks = 2:9 %>% 10^(2:6)) +
  scale_size_area(
    name = "Population",
    breaks = 10^(6:9),
    labels = c("1 million", "10 million", "100 million", "1 billion"),
    max_size = 20
  ) +
  scale_colour_brewer(name = "Continent", palette = "Set1") +
  guides(colour = guide_legend(override.aes = list(alpha = 1, size = 4)))
```

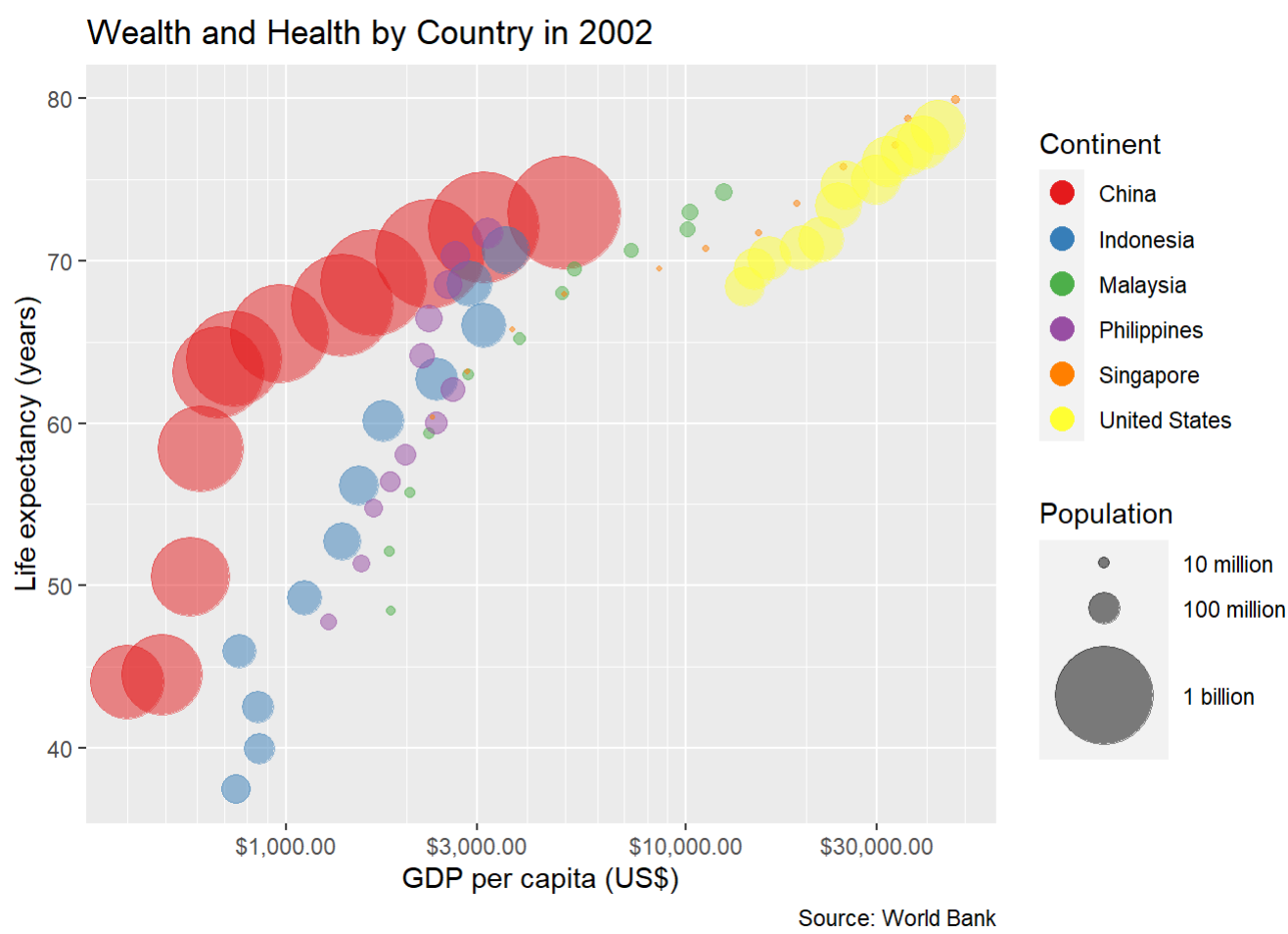


b. There are too many countries on the plot to make meaningful observations. The plot might be more useful if we pick just a few countries. Let us choose the following countries:

- China
- Indonesia
- Malaysia
- Philippines
- Singapore
- United States

Remake the above plot with only these countries. Instead of coloring by continent, color by country.

```
ggplot(filter(gapminder, country %in% c("China", "Indonesia", "Malaysia", "Philippines", "Singapore", "United States")), aes(
  gdpPercap,
  lifeExp,
  size = pop,
  color = country
)) +
  geom_point(alpha = 0.5) +
  labs(
    title = "Wealth and Health by Country in 2002",
    x = "GDP per capita (US$)",
    y = "Life expectancy (years)",
    caption = "Source: World Bank"
  ) +
  scale_x_log10(labels = dollar, minor_breaks = 2:9 %>% 10^(2:6)) +
  scale_size_area(
    name = "Population",
    breaks = 10^(6:9),
    labels = c("1 million", "10 million", "100 million", "1 billion"),
    max_size = 20
  ) +
  scale_colour_brewer(name = "Continent", palette = "Set1") +
  guides(colour = guide_legend(override.aes = list(alpha = 1, size = 4)))
```



All answers to the question below are compiled into one plot.

c. Now, show the points move and change shape as years progress. The objective is to make the points move and change smoothly from their initial to final position. Use one of the `transition_*()` functions to make the points animated.

For this answer, we wanted to see the use of `transition_time()` which makes the points move smoothly

d. Now we want to show the trend of the countries with time.

However, combined with the followup questions, `transition_reveal()` takes care of this.

i. Your plot should show the points move with time. It may also be useful to keep track of the path of the point movements. Add another geom to the plot to show the paths of the points.

Hint: Which geom would we use in this case to show a trend?

ii. In the same code, add a corresponding transition to your graph. The objective is to make the points move and change smoothly from their initial to final position, while also capturing the trend with time. Refer back to the tutorial. Which `transition_*()` is suitable here? Add this `transition_*()` function after you add the geom for the line.

Hint: Which transition is suitable to show gradual changes in time?

This provides a great example of how multiple transitions can be combined in the same plot.

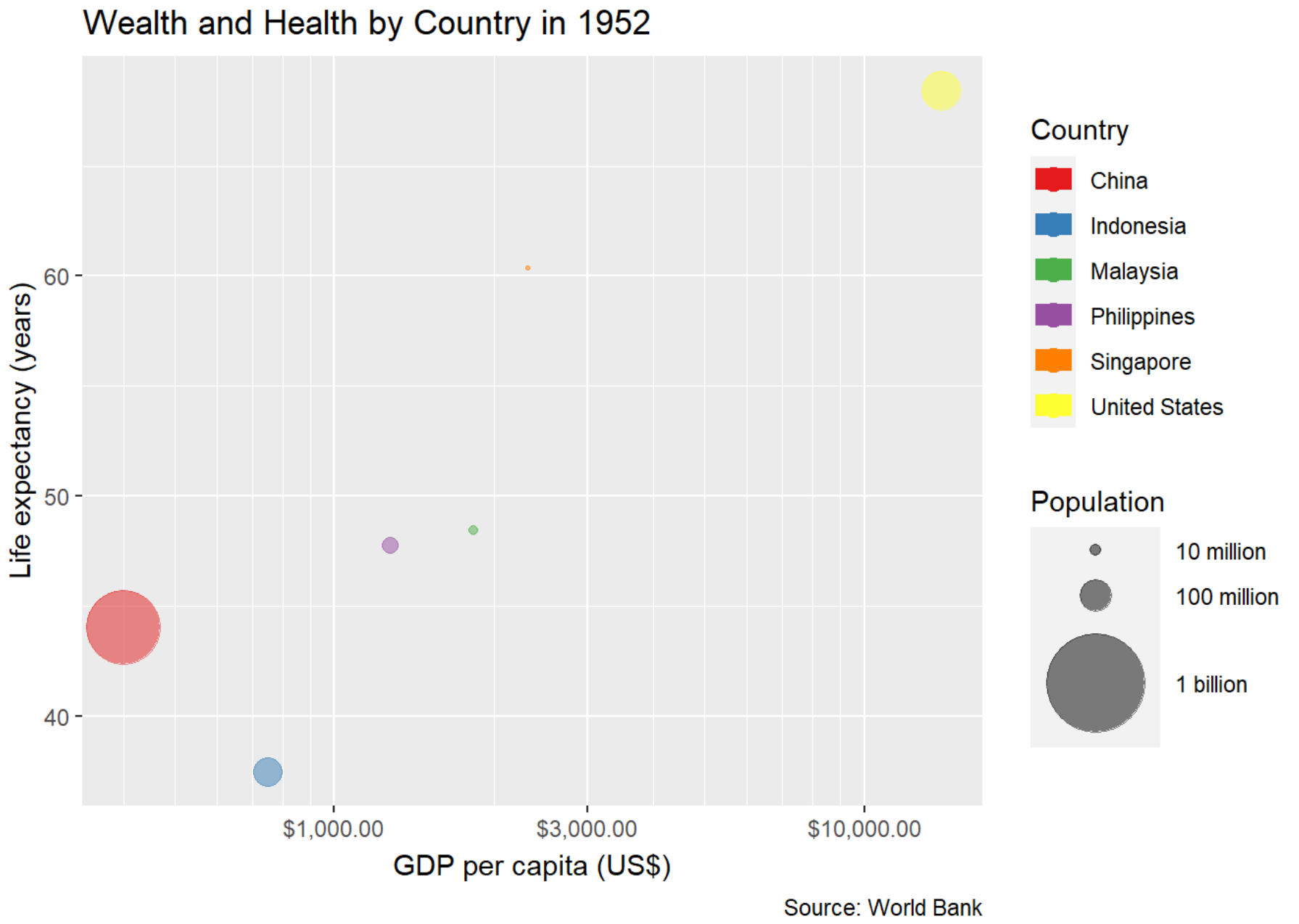
e. At the moment, our axis are static. This makes it difficult to observe the initial changes of the country pathways. It will be more useful to be able to make the axis dynamic and show all the points given at the current moment, slowly growing to include the full pathways. Use one of the `view_*()` functions to achieve this effect in our plot.

f. Our plot currently does not show the year for which the points are at a location. We want to use one of the label variables to include the year in our title. The tutorial introduced two labelling variables that can be used for this purpose. However, label variables differ for which `transition_*()` function we are using. Use the `gganimate` cheatsheet (https://ugoproto.github.io/ugo_r_doc/pdf/gganimate.pdf) to identify

the correct label variable to use for `transition_reveal()`, our latest `transition_*`() function, and incorporate it into the plot title to show the current year.

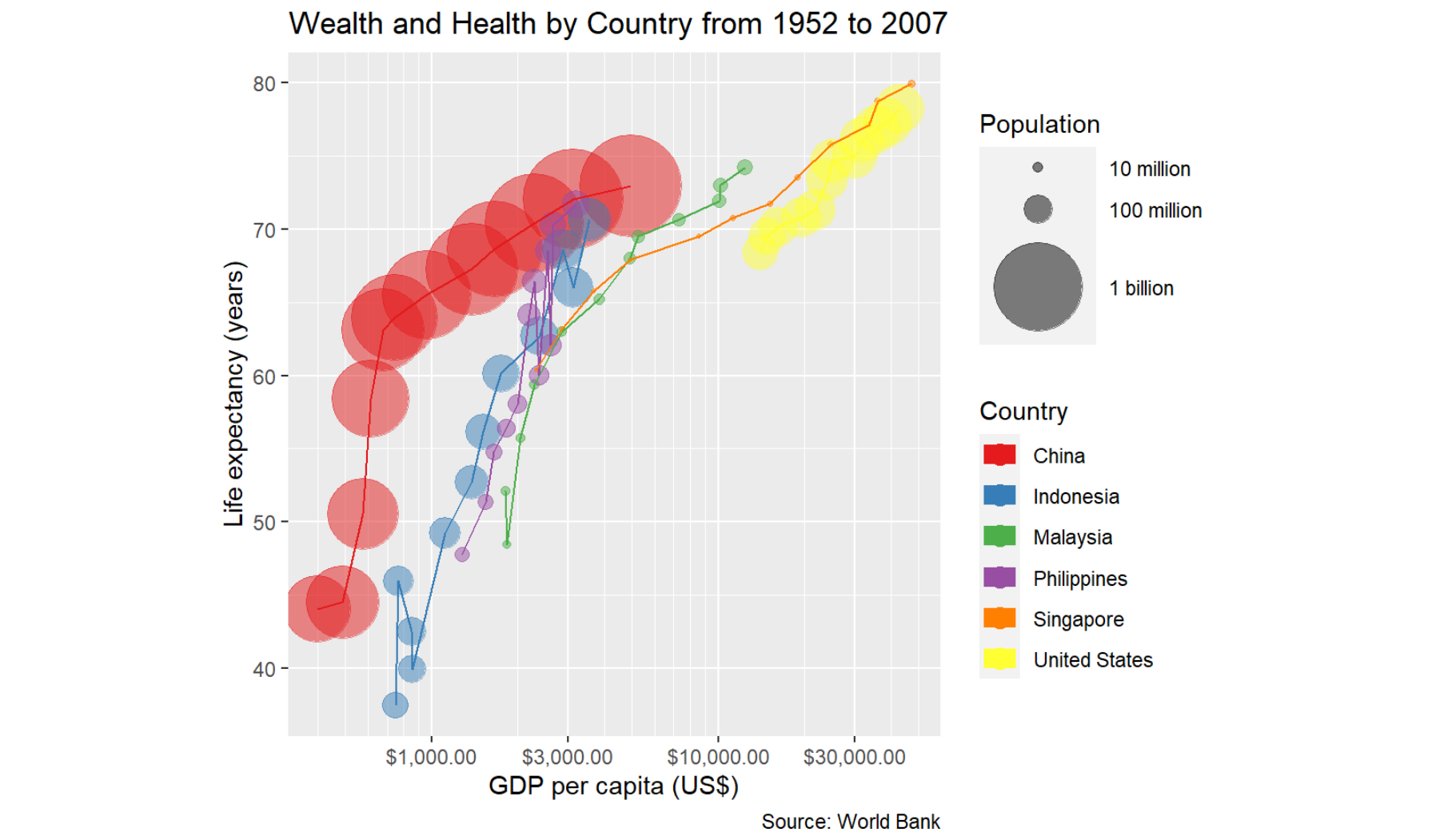
g. Please display your final plot here with adjusted figure dimensions and any other changes you think are suitable.

```
ggplot(filter(gapminder, country %in% c("China", "Indonesia", "Malaysia", "Philippines", "Singapore", "United States")), aes(
  gdpPercap,
  lifeExp,
  size = pop,
  color = country
)) +
  geom_point(alpha = 0.5) +
  labs(
    title = "Wealth and Health by Country in {frame_along}",
    x = "GDP per capita (US$)",
    y = "Life expectancy (years)",
    caption = "Source: World Bank"
  ) +
  scale_x_log10(labels = dollar, minor_breaks = 2:9 %>% 10^(2:6)) +
  scale_size_area(
    name = "Population",
    breaks = 10^(6:9),
    labels = c("1 million", "10 million", "100 million", "1 billion"),
    max_size = 20
  ) +
  scale_colour_brewer(name = "Country", palette = "Set1") +
  guides(colour = guide_legend(override.aes = list(alpha = 1, size = 4))) +
  geom_line(size = 0.5, aes(group = country)) +
  transition_reveal(year) +
  view_follow()
```



The chunk options could be adjusted here to make the plot look more closely in size to the rest of the document.

h. Shown below is a static plot for the tasks above. Compare the static plot with your final, animated plot. Which one is better, and why?



They have different advantages and disadvantages to them. For the animated plot, the viewer can identify which year the country had certain data points. Furthermore, the animated plot is less cluttered as there are less points. However, with the line plot, one can see where the different countries intersect or have similar values. They can also better analyze the plot as it is static, whereas the moving plot restarts every few seconds, and the viewer must follow a specific country to determine its trend.

I assume you are referring to the static plot when you say line plot. You are entitled to opion, but it may be more accurate to say the static plot is a bit cluttered and difficult to see the trends and gradual changes in trends. If you were to present it to an audience, they would not be able to observe much out of it. However the dynamic plot is easier to understand as it shows how the trends, and point sizes change with time. These aspects are a bit difficult to observe in the static.

It is however true that the animated plot keeps repeating.