Homework: Data Visualization with ggplot2

Evans

2023-10-01

Instruction 1(a)

Set the following global chunk options in the setup chunk: eval = TRUE, fig.width = 8, fig.height = 3.5..

I placed this code at the beginning of R Markdown document, right after the YAML front matter to configure the behavior of code chunks throughout the document, this will evaluate it by default, and setting the default width and height o 8 and 3.5 respectively .

Instruction 1(b)

Load the tidyverse, skimr, and ggthemes packages, and explore the gapminder dataset using glimpse() and ?gapminder to learn about the variables. You may need to install ggthemes as well.

```
library("gapminder")
library("ggplot2")
library("dplyr")
library("skimr")
library("tidyverse")
library("ggthemes")
library("dplyr")
library("scales")
```

What are the quantitative variables in this data set? What are the categorical variables? Note that ?gapminder will not produce anything in the R Markdown output, but should open documentation in your browser for your viewing.

Provide the variable names, a description of each variable, and the type of each variable.

```
data(gapminder)
```

```
glimpse(gapminder)
```

```
## Rows: 1,704
## Columns: 6
## $ country   <fct> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanistan", ~
## $ continent <fct> Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, ~
## $ year      <int> 1952, 1957, 1962, 1967, 1972, 1977, 1982, 1987, 1992, 1997, ~
## $ lifeExp   <dbl> 28.801, 30.332, 31.997, 34.020, 36.088, 38.438, 39.854, 40.8~
## $ pop       <int> 8425333, 9240934, 10267083, 11537966, 13079460, 14880372, 12~
## $ gdpPercap <dbl> 779.4453, 820.8530, 853.1007, 836.1971, 739.9811, 786.1134, ~
```

```
quantitative_vars <- c("year", "pop", "gdpPercap", "lifeExp")
```

```
categorical_vars <- c("country", "continent")
```

Instruction 1(c)

Use skim() from the skimr package to further explore the data set and any missing data patterns. How many missing values are there for this data set?

```
cat("Quantitative Variables:\n")
```

```
## Quantitative Variables:
```

```
cat(paste("- ", quantitative_vars, collapse = "\n"))
```

```
## -  year
## -  pop
## -  gdpPercap
## -  lifeExp
```

```
data(gapminder)

skimmed_data <- skim(gapminder)
```

```
skimmed_data
```

Table 1: Data summary

Name	gapminder
Number of rows	1704
Number of columns	6
Column type frequency:	
factor	2
numeric	4
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
country	0	1	FALSE	142	Afg: 12, Alb: 12, Alg: 12, Ang: 12
continent	0	1	FALSE	5	Afr: 624, Asi: 396, Eur: 360, Ame: 300

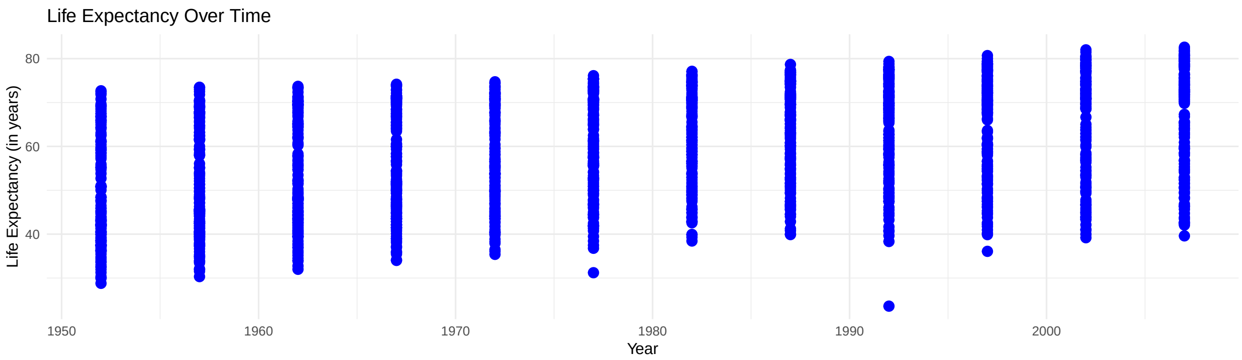
Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
year	0	1	1979.50	17.27	1952.00	1965.75	1979.50	1993.25	2007.0	
lifeExp	0	1	59.47	12.92	23.60	48.20	60.71	70.85	82.6	
pop	0	1	29601212.32	106157896.74	60011.00	2793664.00	7023595.50	19585221.75	1318683096.0	
gdpPercap	0	1	7215.33	9857.45	241.17	1202.06	3531.85	9325.46	113523.1	

Instruction 1(c)

Using ggplot2, create a scatterplot showing life expectancy across time, adding descriptive labels of the axes and overall plot. What trend do you notice?

```r
ggplot(gapminder, aes(x = year, y = lifeExp)) +
  geom_point(color = "blue", size = 3) +


  labs(
    title = "Life Expectancy Over Time",
    x = "Year",
    y = "Life Expectancy (in years)"
  ) +
  theme_minimal()
```



The scatterplot showing life expectancy across time reveals the following trends:

1. Overall Increase in Life Expectancy: Over the years, there is a noticeable upward trend in life expectancy. In general, life expectancy has increased across the world.

2. Variability: Although there is an overall increasing trend, there is also considerable variability in life expectancy within and between years. Some countries experience rapid improvements in life expectancy, while others progress at a slower pace.

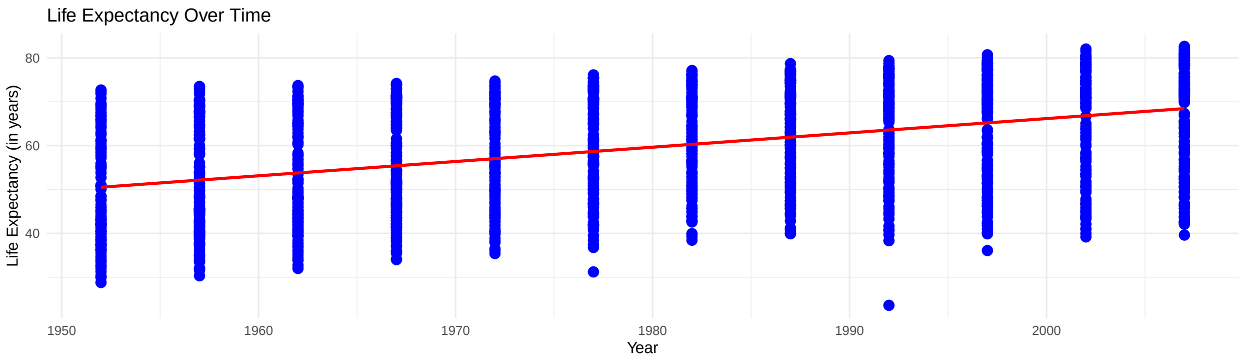
3. Periodic Patterns: In certain time periods, you may observe periodic patterns of peaks and troughs in life expectancy. These patterns can be influenced by various factors, including historical events, healthcare advancements, and economic conditions.

4. Outliers: There may be outliers or specific data points that deviate significantly from the general trend. These outliers could represent countries or regions with unique circumstances that led to deviations from the overall trend.

Instruction 1(d)

Recreate the plot of life expectancy across time, this time adding an additional smooth line of best fit through the data using geom_smooth(). Include the option se = FALSE in geom_smooth() to suppress the standard error bands around the smooth curves.

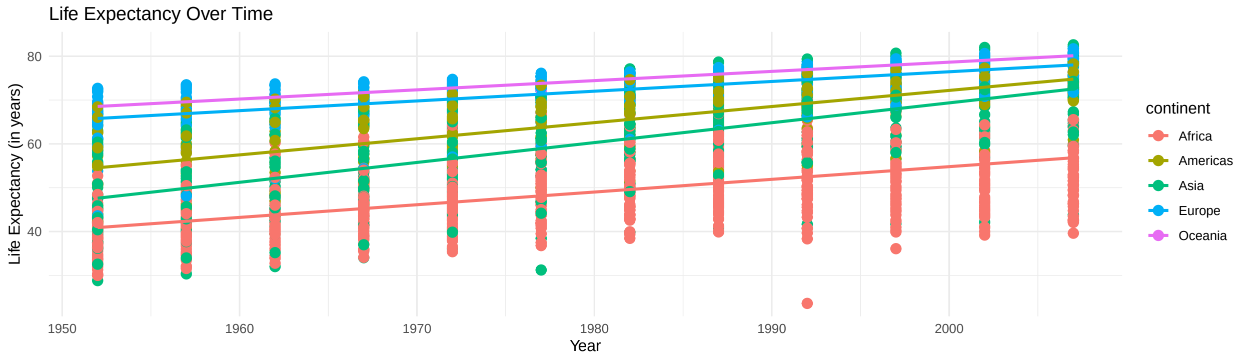
```r
ggplot(gapminder, aes(x = year, y = lifeExp)) +
  geom_point(color = "blue", size = 3) +
   geom_smooth(se = FALSE, method = "lm", color = "red", linetype = "solid") +



  labs(
    title = "Life Expectancy Over Time",
    x = "Year",
    y = "Life Expectancy (in years)"
  ) +
  theme_minimal()
```



Instruction 1(f)

Color the points based on which continent the points are representing, including smoothed lines through the points using geom_smooth() so that the lines are still colored by continent as well. Which continent / region has the highest life expectancy on average?

```
ggplot(gapminder, aes(x = year, y = lifeExp,color = continent)) +
  geom_point(size = 3) +
  geom_smooth(se = FALSE, method = "lm", linetype = "solid") +

  labs(
    title = "Life Expectancy Over Time",
    x = "Year",
    y = "Life Expectancy (in years)"
  ) +
  theme_minimal()
```



Which continent / region has the highest life expectancy on average?

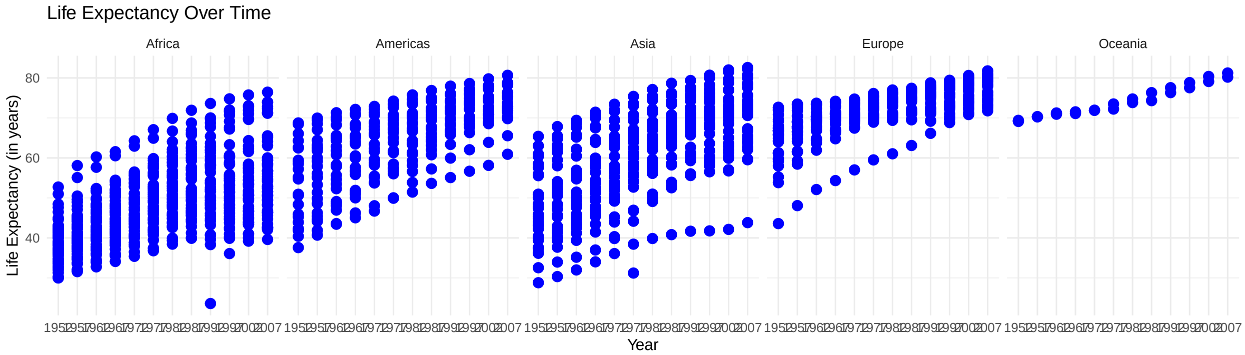
```
  # I will calculate the mean life expectancy by continent
mean_life_expectancy <- gapminder %>%
  group_by(continent) %>%
  summarise(mean_life_expectancy = mean(lifeExp))

# Then will find the continent with the highest mean life expectancy
continent_with_highest_expectancy <- mean_life_expectancy %>%
  filter(mean_life_expectancy == max(mean_life_expectancy)) %>%
  pull(continent)

# below is the result
cat("Continent with the highest average life expectancy:", continent_with_highest_expectancy)
```

```
## Continent with the highest average life expectancy: 5
```

Instruction 1(g)

Extend the plot from the previous part by faceting by the continent associated with each point so that each continent has its own column.

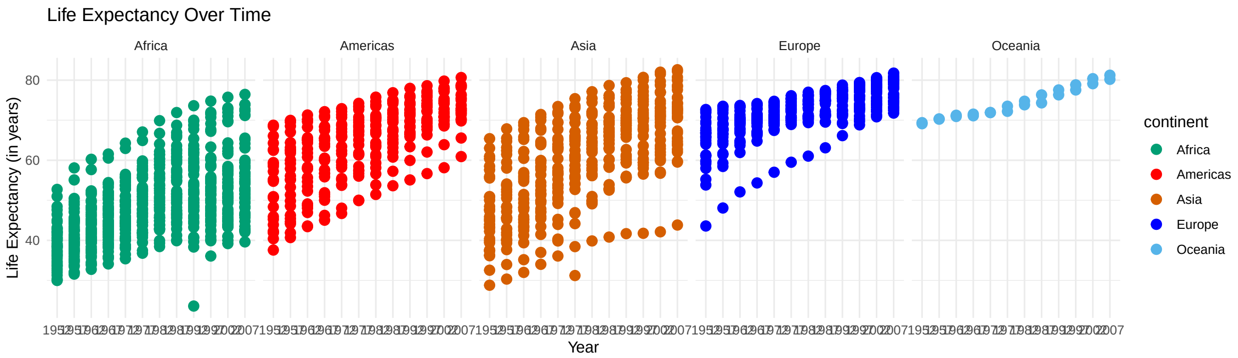
```
ggplot(gapminder, aes(x = str_wrap(as.character(year)), y = lifeExp)) +
  geom_point(color = "blue", size = 3) +


  labs(
    title = "Life Expectancy Over Time",
    x = "Year",
    y = "Life Expectancy (in years)"
  ) +
  facet_wrap(~continent,nrow=1) +
  theme_minimal()
```



Instruction 1(h)

Modify the colors for each continent to be color-blind friendly using this palette, and separately customize the overall theme of the plot by specifying a complete theme of your choice.

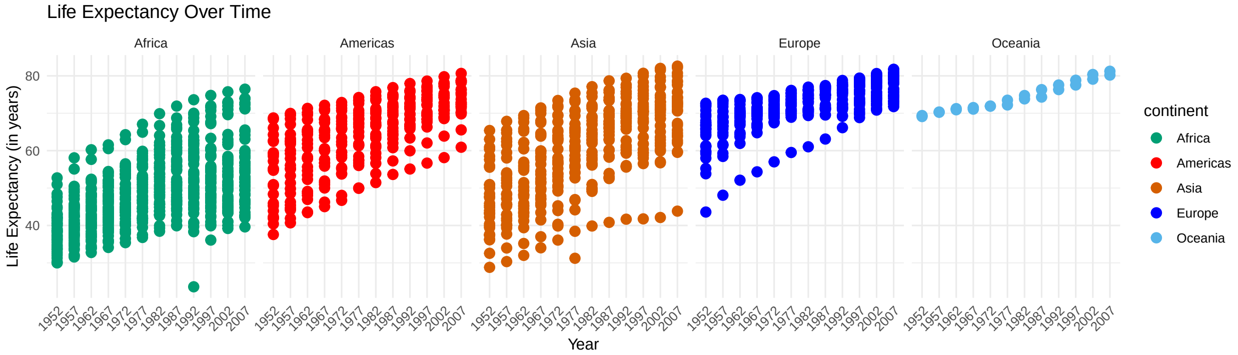
```
ggplot(gapminder, aes(x = str_wrap(as.character(year)), y = lifeExp, color = continent)) +
  geom_point( size = 3) +
  scale_fill_manual(values = continent_colors) +

  labs(
    title = "Life Expectancy Over Time",
    x = "Year",
    y = "Life Expectancy (in years)"
  ) +
  facet_wrap(~continent,nrow=1) +
   scale_color_manual(values = c("Asia" = "#d55e00", "Europe" = "#0000FF", "Africa" = "#009E73", "Ameri
  theme_minimal()
```



Instruction 1(i)

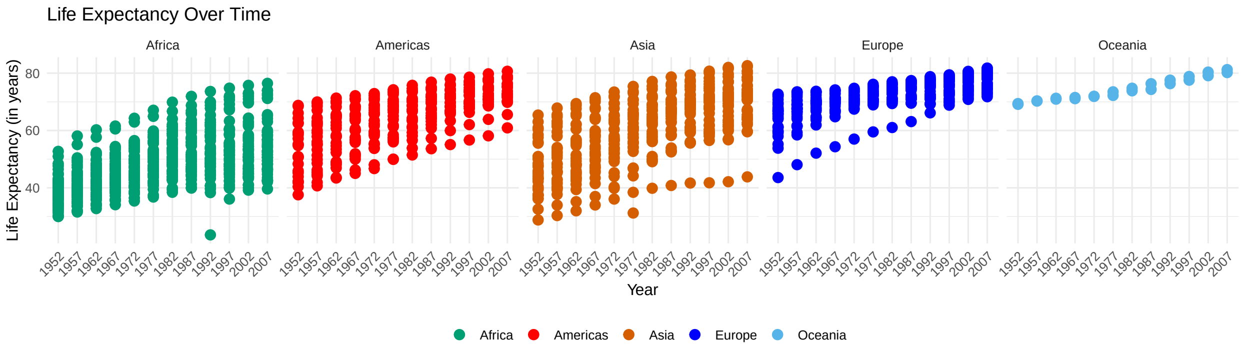
Rotate the labels on the x-axis 45 degrees by adding a theme() layer with the appropriate option. Hint: see this section of the reading to review how to customize aspects of the axes.

```
continent_colors <- c("Asia" = "#d55e00", "Europe" = "#0000FF", "Africa" = "#009E73", "Americas" = "#FF(


ggplot(gapminder, aes(x = str_wrap(as.character(year)), y = lifeExp, color = continent)) +
  geom_point(size = 3) +
  labs(
    title = "Life Expectancy Over Time",
    x = "Year",
    y = "Life Expectancy (in years)"
  ) +
  facet_wrap(~continent, nrow = 1) +
  scale_color_manual(values = continent_colors) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels by 45 degrees
```



# Instruction 1(j) ## Specify an argument in the theme() function to suppress the legend.

```
continent_colors <- c("Asia" = "#d55e00", "Europe" = "#0000FF", "Africa" = "#009E73", "Americas" = "#FF


ggplot(gapminder, aes(x = str_wrap(as.character(year)), y = lifeExp, color = continent)) +
  geom_point(size = 3) +
  labs(
    title = "Life Expectancy Over Time",
    x = "Year",
    y = "Life Expectancy (in years)"
  ) +
  facet_wrap(~continent, nrow = 1) +
  scale_color_manual(values = continent_colors) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),legend.position = "bottom") +
  theme(legend.position = "bottom", legend.title = element_blank())
```

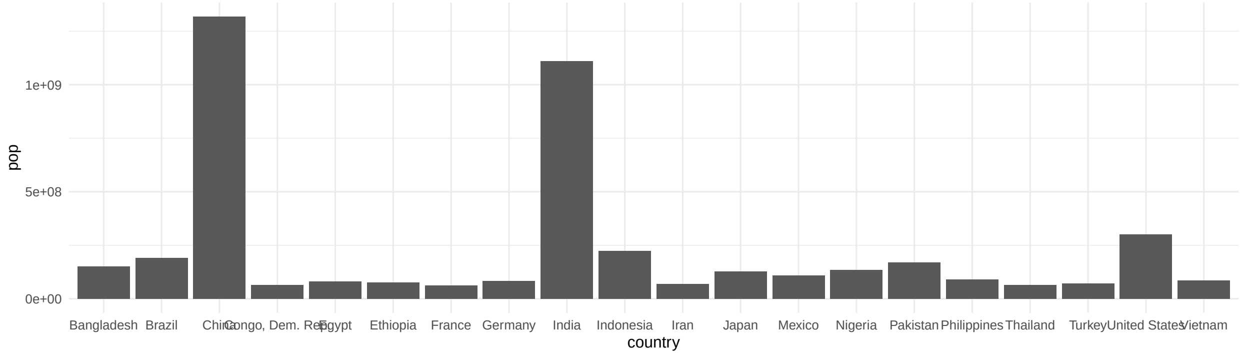


question 2

```
gapminder2007 <- gapminder %>% dplyr::filter(year == 2007) %>% slice_max(pop, n = 20)
```

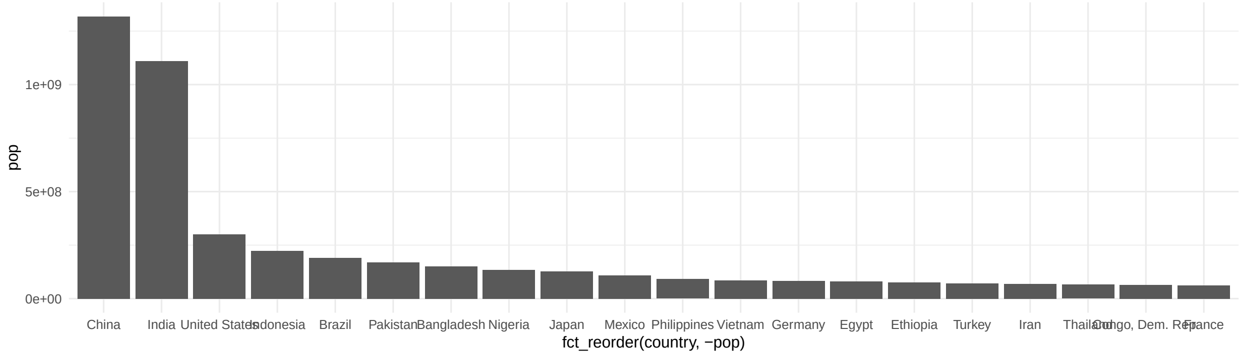
2 (a) First, create a bar plot displaying the population of each country using the gapminder2007 dataset using geom_col().

```
gapminder_2007 <- gapminder %>%
  filter(year == 2007) %>%
  arrange(desc(pop))

gapminder2007 <- head(gapminder_2007, 20)


  ggplot(gapminder2007, aes(y = pop, x = country)) +
 geom_col() +
  theme_minimal() +
  theme()
```



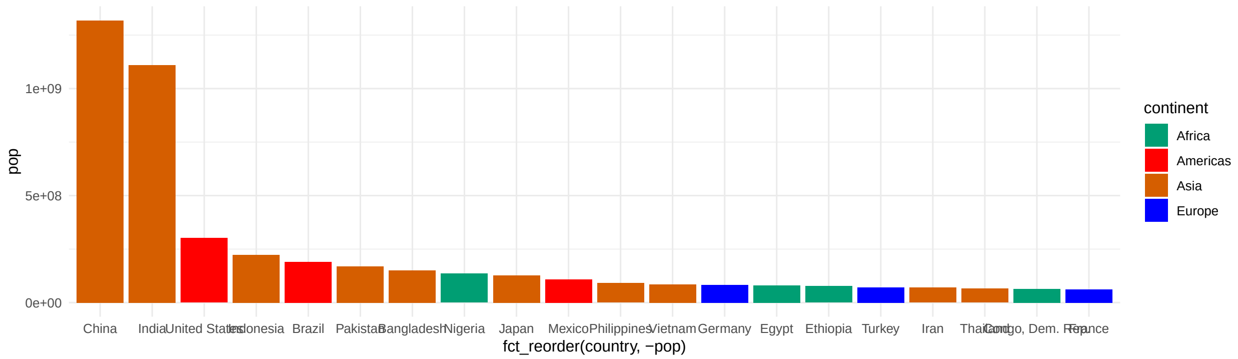
2 (b) In a new code chunk, modify the plot in the previous part so that the bars are sorted based on height using the fct__reorder(country, pop) for the x aesthetic.

```
ggplot(gapminder2007, aes(y = fct_reorder(country, -pop), x = pop)) +
  geom_col( ) +
   #scale_fill_manual(values = continent_colors) +



  theme_minimal() +
  theme() +
  coord_flip()
```



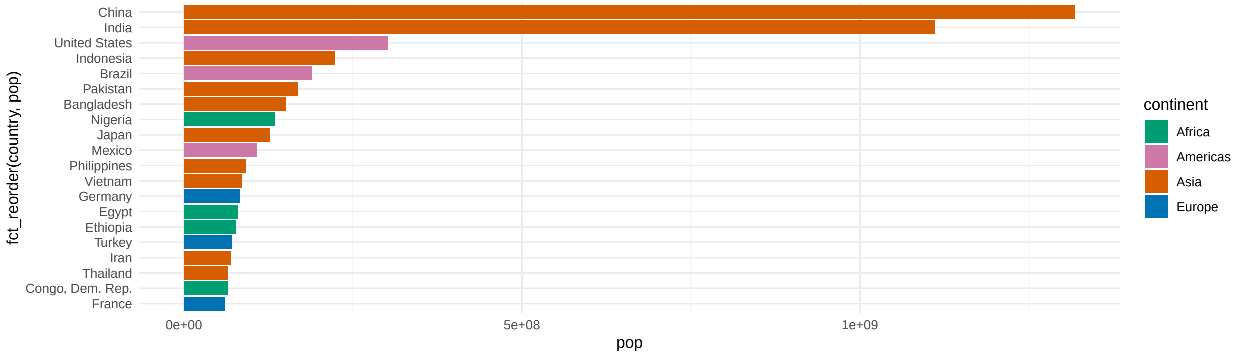
(c) In another new code chunk, modify the plot so that the color inside of the bars displays which continent each bar represents as well using the fill aesthetic, and change the outline of all bars in the plot to be black by manually setting the color aesthetic.

```
ggplot(gapminder2007, aes(y = fct_reorder(country, -pop), x = pop, fill = continent)) +
  geom_col() +
   scale_fill_manual(values = continent_colors) +

  theme_minimal() +
  theme() +
  coord_flip()
```



2(d) Use the coord_flip() function to make the barchart a horizontal bar chart rather than a vertical one to fix the issue of the country names overlapping.

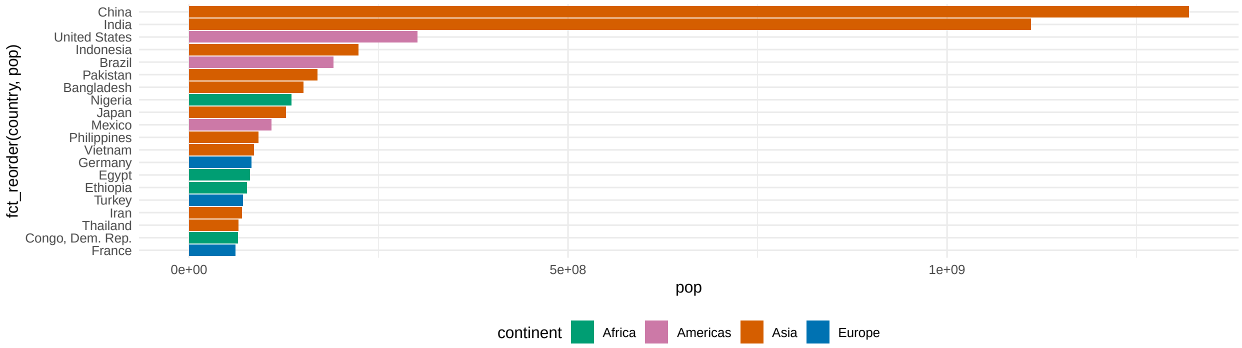
```
continent_colors <- c("Asia" = "#d55e00", "Europe" = "#0072B2", "Africa" = "#009E73", "Americas" = "#CC

ggplot(gapminder2007, aes(x = fct_reorder(country, pop), y = pop, fill = continent)) +
  geom_col() +
  scale_fill_manual(values = continent_colors) +



  theme_minimal() +
  theme() +
  coord_flip() +
  theme()
```



2(d) Move the legend to below the plot (to the "bottom").

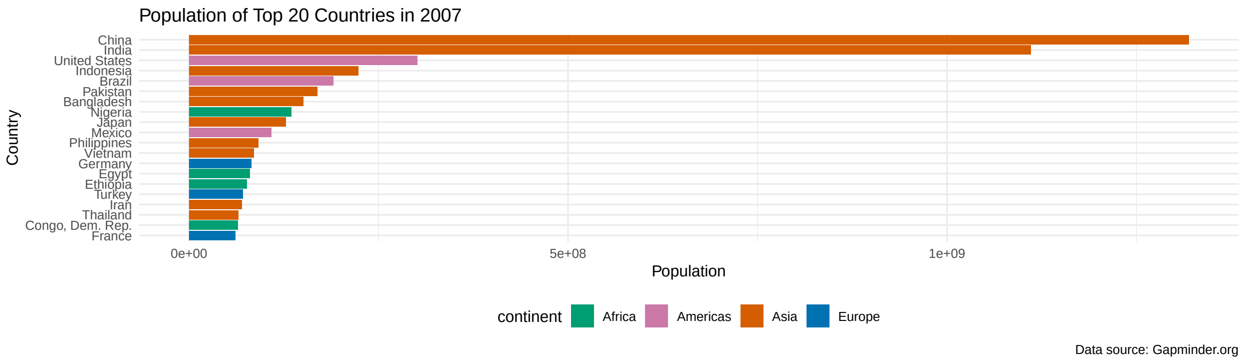
```
continent_colors <- c("Asia" = "#d55e00", "Europe" = "#0072B2", "Africa" = "#009E73", "Americas" = "#CC
```

```r
ggplot(gapminder2007, aes(x = fct_reorder(country, pop), y = pop, fill = continent)) +
  geom_col() +
  scale_fill_manual(values = continent_colors) +


  theme_minimal() +
  theme(axis.text.y = element_text(angle = 0, hjust = 1)) +
  coord_flip() +
  theme(legend.position = "bottom")
```



2(f) Add descriptive labels for the axes, title, and a caption below the plot.

```r
ggplot(gapminder2007, aes(x = fct_reorder(country, pop), y = pop, fill = continent)) +
  geom_col() +
  scale_fill_manual(values = continent_colors) +

  labs(
    title = "Population of Top 20 Countries in 2007",
    y = "Population",
    x = "Country",
    caption = "Data source: Gapminder.org"
  ) +
  theme_minimal() +
  theme(axis.text.y = element_text(angle = 0, hjust = 1)) +
  coord_flip() +
  theme(legend.position = "bottom")
```



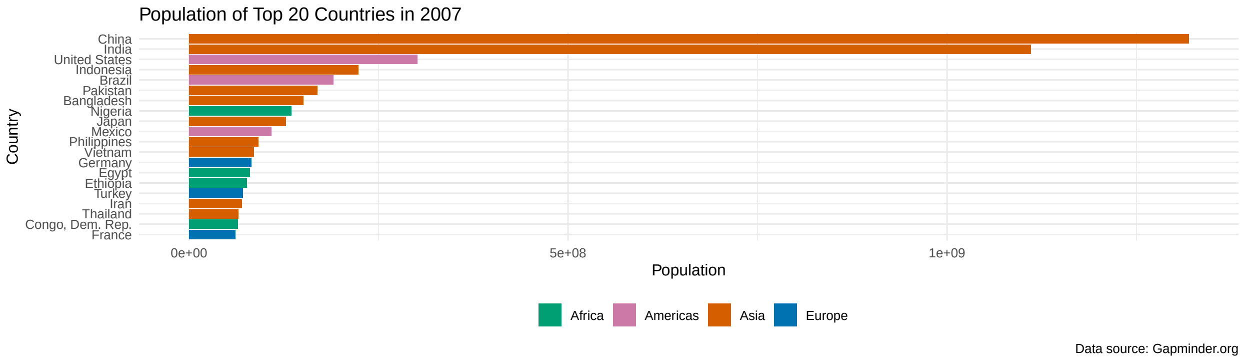
Data source: Gapminder.org

```
  theme(legend.position = "bottom", legend.title = element_blank())
```

```
## List of 2
##  $ legend.title   : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ legend.position: chr "bottom"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi FALSE
##  - attr(*, "validate")= logi TRUE
```

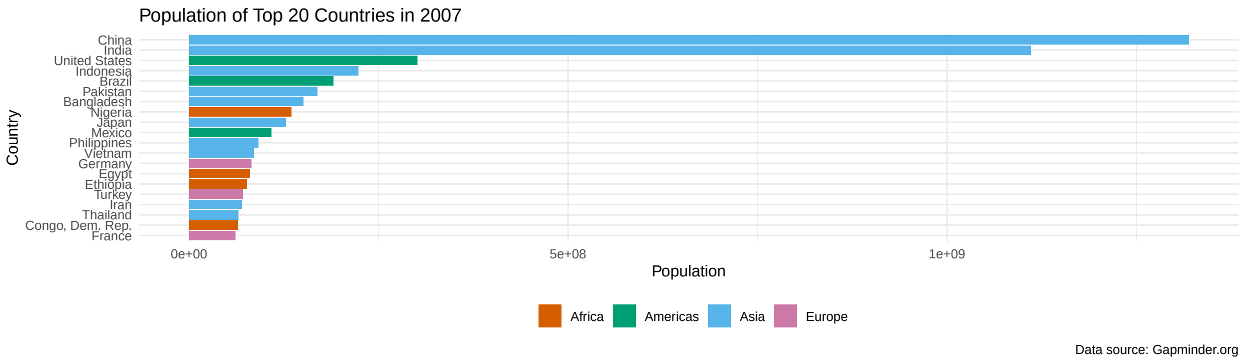
2(g) Add the option legend.title = element_blank() to the theme() function to remove the legend title.

```
ggplot(gapminder2007, aes(x = fct_reorder(country, pop), y = pop, fill = continent)) +
  geom_col() +
  scale_fill_manual(values = continent_colors) +

  labs(
    title = "Population of Top 20 Countries in 2007",
    y = "Population",
    x = "Country",
    caption = "Data source: Gapminder.org"
  ) +
  theme_minimal() +
  theme(axis.text.y = element_text(angle = 0, hjust = 1)) +
  coord_flip() +

  theme(legend.position = "bottom", legend.title = element_blank())
```



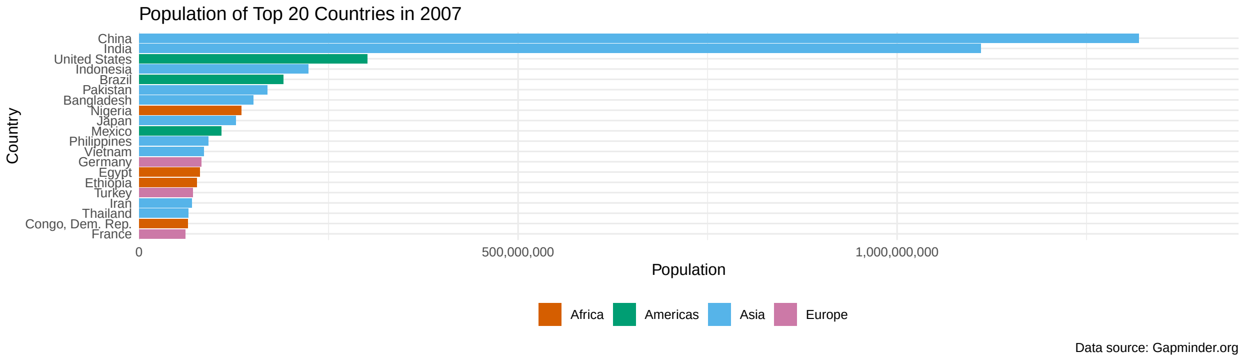
2(h) Use color-blind friendly colors by adding a scale_fill_manual() layer using the code below:

```
color_palette <- c("#D55E00", "#009E73", "#56B4E9", "#CC79A7")


ggplot(gapminder2007, aes(x = fct_reorder(country, pop), y = pop, fill = continent)) +
  geom_col() +
  scale_fill_manual(values = color_palette) +

  labs(
    title = "Population of Top 20 Countries in 2007",
    y = "Population",
    x = "Country",
    caption = "Data source: Gapminder.org"
  ) +
  theme_minimal() +
  theme(axis.text.y = element_text(angle = 0, hjust = 1)) +
  coord_flip() +
  theme(legend.position = "bottom", legend.title = element_blank())
```



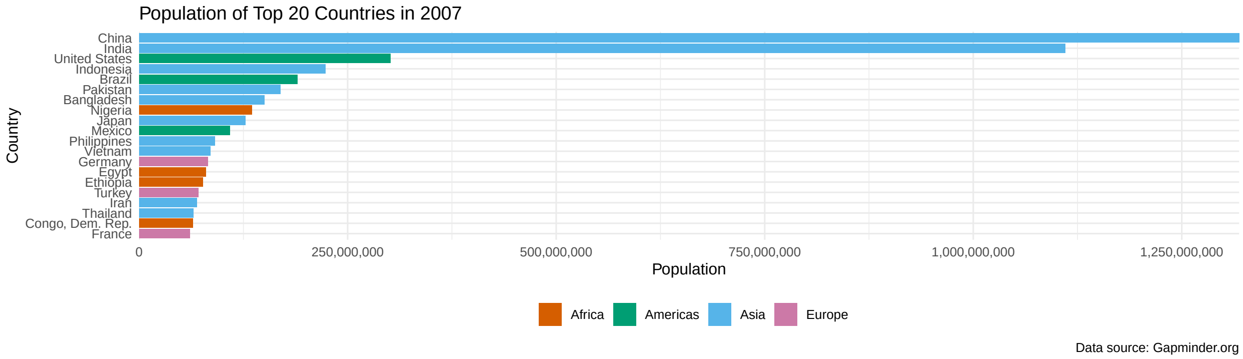
2(i) Display commas in the population numbers rather than scientific notation by adding a scale_y_continuous() layer using the code below. Note that the scales package will need to be installed to do this.

```
color_palette <- c("#D55E00", "#009E73", "#56B4E9", "#CC79A7")


ggplot(gapminder2007, aes(x = fct_reorder(country, pop), y = pop, fill = continent)) +
  geom_col() +
  scale_fill_manual(values = color_palette) +
   scale_y_continuous(labels =scales::label_comma(),
                         expand =expansion(mult =c(0,0.1)))+
  labs(
    title = "Population of Top 20 Countries in 2007",
    y = "Population",
    x = "Country",
    caption = "Data source: Gapminder.org"
  ) +
  theme_minimal() +
  theme(axis.text.y = element_text(angle = 0, hjust = 1)) +
  coord_flip() +
  theme(legend.position = "bottom", legend.title = element_blank())
```



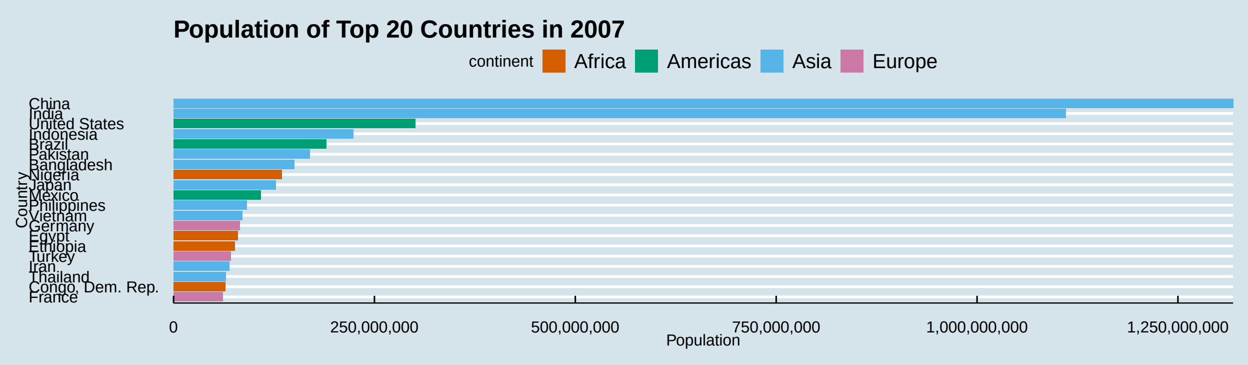
2(j) Remove the excess space in between the bars and the axis by specifying the expand argument inside of the scale_y_continuous() layer using expand = expansion(mult = c(0, .1)).

```
color_palette <- c("#D55E00", "#009E73", "#56B4E9", "#CC79A7")


ggplot(gapminder2007, aes(x = fct_reorder(country, pop), y = pop, fill = continent)) +
  geom_col() +
  scale_fill_manual(values = color_palette) +
   #scale_y_continuous(labels =scales::label_comma(0,0),expand = expansion(mult = c(0, 0)))+
   scale_y_continuous(labels =scales::comma,
                          expand =expansion(mult =c(0,0) ))+
  labs(
    title = "Population of Top 20 Countries in 2007",
    y = "Population",
    x = "Country",
    caption = "Data source: Gapminder.org"
  ) +
  theme_minimal() +
  theme(axis.text.y = element_text(angle = 0, hjust = 1)) +
  coord_flip() +
  theme(legend.position = "bottom", legend.title = element_blank())
```



2(k) Modify the previous plot by specifying a theme from the ggthemes package: https://yutannihilation.github.io/allYourFigureAreBelongToUs/ggthemes/. Make sure to add the custom theme layer before the final theme() call so that the positioning of the legend is kept at the bottom.

```r
ggplot(gapminder2007, aes(x = fct_reorder(country, pop), y = pop, fill = continent)) +
  geom_col() +
  scale_fill_manual(values = color_palette) +
  labs(
    title = "Population of Top 20 Countries in 2007",
     y = "Population",
    x = "Country",
  ) +
  theme_minimal() +
  theme(axis.text.y = element_text(angle = 0, hjust = 1)) +
  coord_flip() +
  theme(legend.position = "bottom", legend.title = element_blank()) +
  scale_y_continuous(
    labels = scales::label_comma(),
    expand = expansion(mult = c(0, 0))
  ) +
  theme_economist()
```


Population of Top 20 Countries in 2007
continent
Africa
Americas
Asia
Europe
Country
China
India
United States
Indonesia
Brazil
Pakistan
Bangladesh
Nigeria
Japan
Mexico
Philippines
Vietnam
Germany
Egypt
Ethiopia
Turkey
Iran
Thailand
Congo, Dem. Rep.
France
Population