PS5: markdown, basic plots

Bruce Xu

2023-02-19

1 Load and check data (5pt) You first task is to do a very simple data check:

1. (1pt) For solving the problems, and answering the questions,create a new rmarkdown document with an appropriate title. See https://faculty.washington.edu/otoomet/info201-book/r-markdown.html#r-markdown-rstudio-creating.

2. (2pt) Load data. How many rows/columns do we have?

```
df <-read_delim("gapminder.csv.bz2")
```

```
## Rows: 13055 Columns: 25
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr  (6): iso3, name, iso2, region, sub-region, intermediate-region
## dbl (19): time, totalPopulation, fertilityRate, lifeExpectancy, childMortali...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
nrow(df)
```

```
## [1] 13055
```

```
ncol(df)
```

```
## [1] 25
```

3. (2pt) Print a small sample of data. Does it look OK?

```
head(df)
```

```
## # A tibble: 6 x 25
##   iso3  name  iso2  region sub-r~1 inter~2  time total~3 ferti~4 lifeE~5 child~6
##   <chr> <chr> <chr> <chr>  <chr>   <chr>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 ABW   Aruba AW    Ameri~ Latin ~ Caribb~  1960   54211    4.82    65.7      NA
## 2 ABW   Aruba AW    Ameri~ Latin ~ Caribb~  1961   55438    4.66    66.1      NA
## 3 ABW   Aruba AW    Ameri~ Latin ~ Caribb~  1962   56225    4.47    66.4      NA
## 4 ABW   Aruba AW    Ameri~ Latin ~ Caribb~  1963   56695    4.27    66.8      NA
## 5 ABW   Aruba AW    Ameri~ Latin ~ Caribb~  1964   57032    4.06    67.1      NA
## 6 ABW   Aruba AW    Ameri~ Latin ~ Caribb~  1965   57360    3.84    67.4      NA
## # ... with 14 more variables: youthFemaleLiteracy <dbl>,
## #   youthMaleLiteracy <dbl>, adultLiteracy <dbl>, GDP_PC <dbl>,
```

```
## #   accessElectricity <dbl>, agriculturalLand <dbl>, agricultureTractors <dbl>,
## #   cerealProduction <dbl>, fertilizerHa <dbl>, co2 <dbl>,
## #   greenhouseGases <dbl>, co2_PC <dbl>, pm2.5_35 <dbl>, battleDeaths <dbl>,
## #   and abbreviated variable names 1: `sub-region`, 2: `intermediate-region`,
## #   3: totalPopulation, 4: fertilityRate, 5: lifeExpectancy, ...
## # i Use `colnames()` to see all variable names
```

2 Descriptive statistics (15pt)

1. (3pt) How many countries are there in the dataset? Analyze all three: iso3, iso2 and name.

```
count(df, iso3, name, iso2) %>% summarise_all(funs(n_distinct))
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
## # A tibble: 1 x 4
##    iso3  name  iso2     n
##   <int> <int> <int> <int>
## 1   253   250   249     3
```

2. If you did this correctly, you saw that there are more names than iso-2 codes, and there are even more iso3-codes. What is going on? Can you find it out?

(a) (5pt) Find how many names are there for each iso-2 code. Are there any iso-2 codes that correspond to more than one name? What are these countries?

```
iso2_counts <- df %>%
  group_by(iso2) %>%
  filter(!is.na(iso2)) %>%
  summarise(num_names = n_distinct(name))
iso2_duplicates <- iso2_counts %>%
  filter(num_names > 1)
df %>%
  filter(iso2 %in% iso2_duplicates$iso2) %>%
  select(iso2, name) %>%
  distinct() %>%
  arrange(iso2)
```

```
## # A tibble: 0 x 2
## # ... with 2 variables: iso2 <chr>, name <chr>
## # i Use `colnames()` to see all variable names
```

(b) (5pt) Now repeat the same for name and iso3-code. Are there country names that have more than one iso3-code? What are these countries? Hint: two of these entitites are CHANISL and NLD CURACAO.

```
name_counts <- df %>%
  group_by(name) %>%
  summarise(num_iso3 = n_distinct(iso3))
name_duplicates <- name_counts %>%
  filter(num_iso3 > 1)
df %>%
  filter(name %in% name_duplicates$name) %>%
  select(name, iso3) %>%
  distinct() %>%
  arrange(name, iso3)
```

```
## # A tibble: 4 x 2
##   name  iso3
##   <chr> <chr>
## 1 <NA>  CHANISL
## 2 <NA>  GBM
## 3 <NA>  KOS
## 4 <NA>  NLD_CURACAO
```

3. (2pt) What is the minimum and maximum year in these data?

```
df %>%
  select(time) %>%
  filter(!is.na(time)) %>%
  summarise(min(time), max(time))
```

```
## # A tibble: 1 x 2
##   `min(time)` `max(time)`
##         <dbl>       <dbl>
## 1        1960        2019
```

3 CO2 emissions (30pt)

Next,let's analyze CO2 emissions.

1. (2pt) How many missing co2 emissions are there for each year? Analyze both missing CO2 and co2_PC. Which years have most missing data?

```
missing_counts <- df %>%
  group_by(time) %>%
  summarise(missing_co2 = sum(is.na(co2)),
            missing_co2_pc = sum(is.na(co2_PC)))

most_missing <- missing_counts %>%
  arrange(desc(missing_co2)) %>%
  slice(1)

print(missing_counts)
```

```
## # A tibble: 61 x 3
##     time missing_co2 missing_co2_pc
##    <dbl>       <int>          <int>
##  1  1960          60             60
##  2  1961          60             60
##  3  1962          58             58
##  4  1963          57             57
##  5  1964          51             51
##  6  1965          51             51
##  7  1966          51             51
##  8  1967          51             51
##  9  1968          51             51
## 10  1969          51             51
## # ... with 51 more rows
## # i Use `print(n = ...)` to see more rows
```

```
print(most_missing)
```

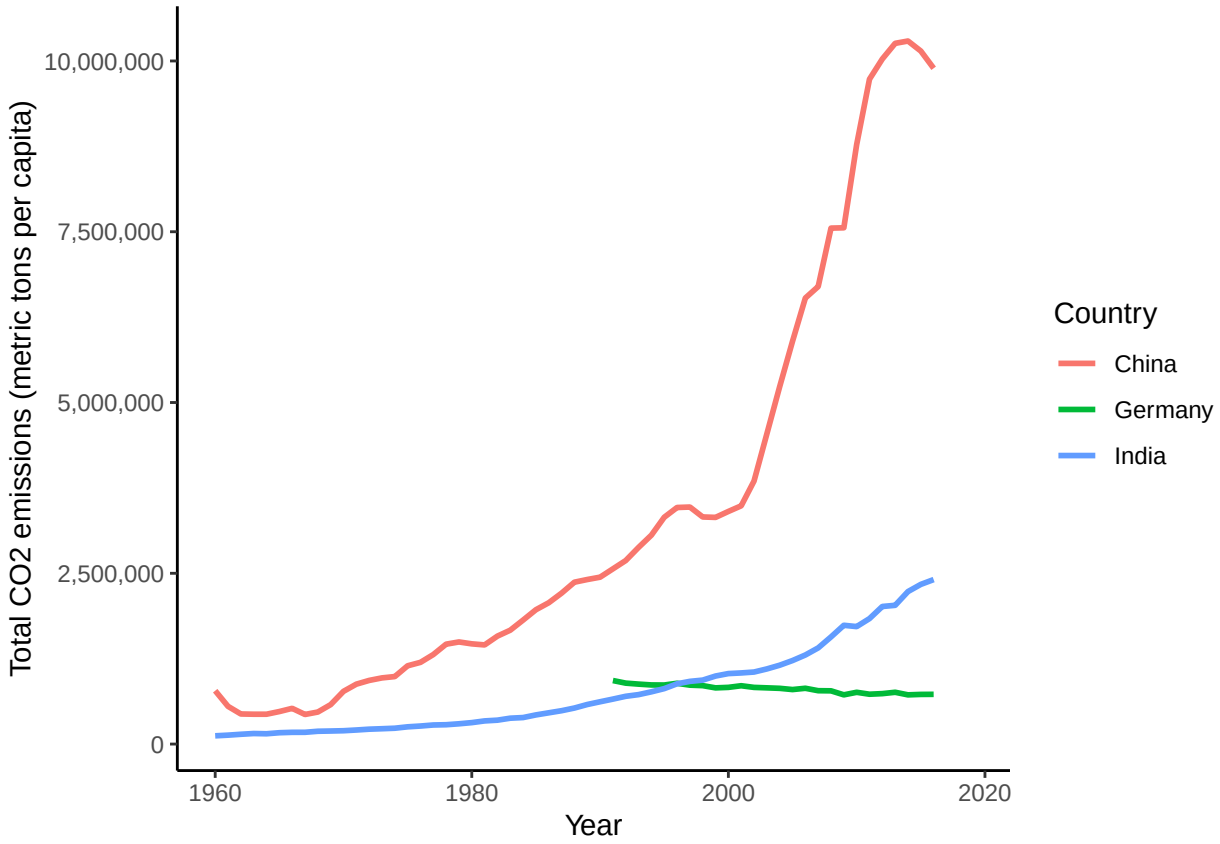
```
## # A tibble: 1 x 3
##    time missing_co2 missing_co2_pc
##   <dbl>       <int>          <int>
## 1  2017         217            217
```

2. (5pt) Make a plot of total CO2 emissions over time for the U.S, China, and India. Add a few more countries of your choice. Explain what do you see.

```
co2_data <- df %>%
  select(name, time, co2) %>%
  filter(name %in% c("United States", "China", "India", "Russia", "Germany"))

ggplot(co2_data, aes(x = time, y = co2, color = name)) +
  geom_line(size = 1) +
  scale_y_continuous(labels = scales::comma) +
  labs(x = "Year", y = "Total CO2 emissions (metric tons per capita)", color = "Country") +
  theme_classic()
```

```
## Warning: Removed 40 row(s) containing missing values (geom_path).
```



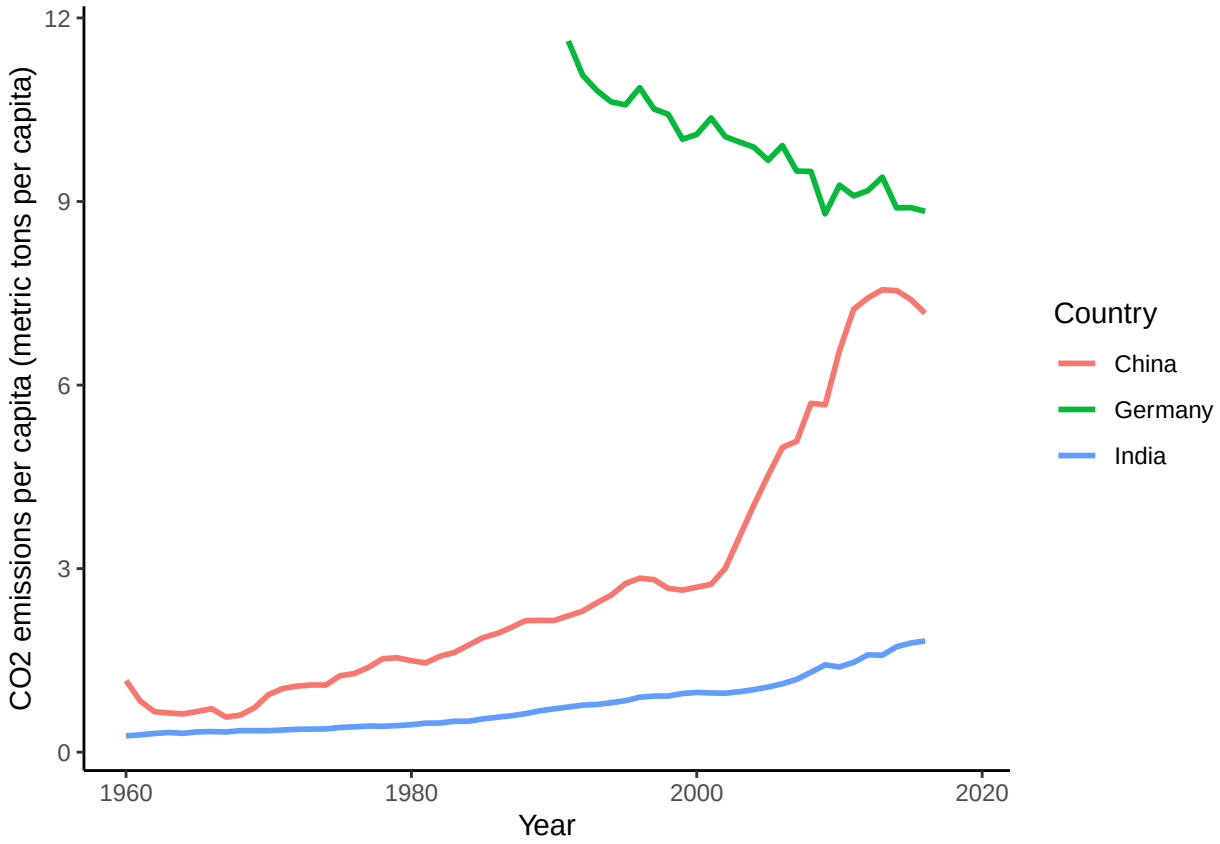
this plot shows the different trends in total CO2 emissions over time for these selected countries, and highlights the varying levels of CO2 emissions between them

3. (5pt) Now let's analyze the CO2 emissions per capita (co2_PC). Make a similar plot of the same countries. What does this figure suggest?

```
co2_pc_data <- df %>%
  select(name, time, co2_PC) %>%
  filter(name %in% c("United States", "China", "India", "Russia", "Germany"))

ggplot(co2_pc_data, aes(x = time, y = co2_PC, color = name)) +
  geom_line(size = 1) +
  scale_y_continuous(labels = scales::comma) +
  labs(x = "Year", y = "CO2 emissions per capita (metric tons per capita)", color = "Country") +
  theme_classic()
```

```
## Warning: Removed 40 row(s) containing missing values (geom_path).
```



this plot suggests that while the total CO2 emissions may be dominated by countries with larger populations and economies, the CO2 emissions per capita can highlight the differences in carbon intensity and efficiency of different countries. ## 4. (6pt) Compute average CO2 emissions per capita across the continents (assume region is the same as continent). Comment what do you see. Note: just compute averages over countries and ignore the fact that countries are of different size. Hint: Americas 2016 should be 4.80.

```
co2_pc_continent <- df %>%
  select(region, co2_PC) %>%
  group_by(region) %>%
  summarize(avg_co2_pc = mean(co2_PC, na.rm = TRUE))

co2_pc_continent <- co2_pc_continent[order(co2_pc_continent$region),]

co2_pc_continent
```

```
## # A tibble: 6 x 2
##   region   avg_co2_pc
##   <chr>         <dbl>
## 1 Africa        0.930
## 2 Americas      6.46
## 3 Asia          6.21
## 4 Europe        7.95
## 5 Oceania       4.39
## 6 <NA>         16.2
```

5. (7pt) Make a barplot where you show the previous results–average CO2 emissions per capita across continents in 1960 and 2016. Hint: it should look something along these lines:0 2 4 6 Africa Americas Asia Europe Oceania Continent Average CO2 per capita Year 1960 2016 6. Which countries are the three largest, and three smallest CO2 emitters (in terms of CO2 per capita) in 2019 for each continent? (Assume region is continent).

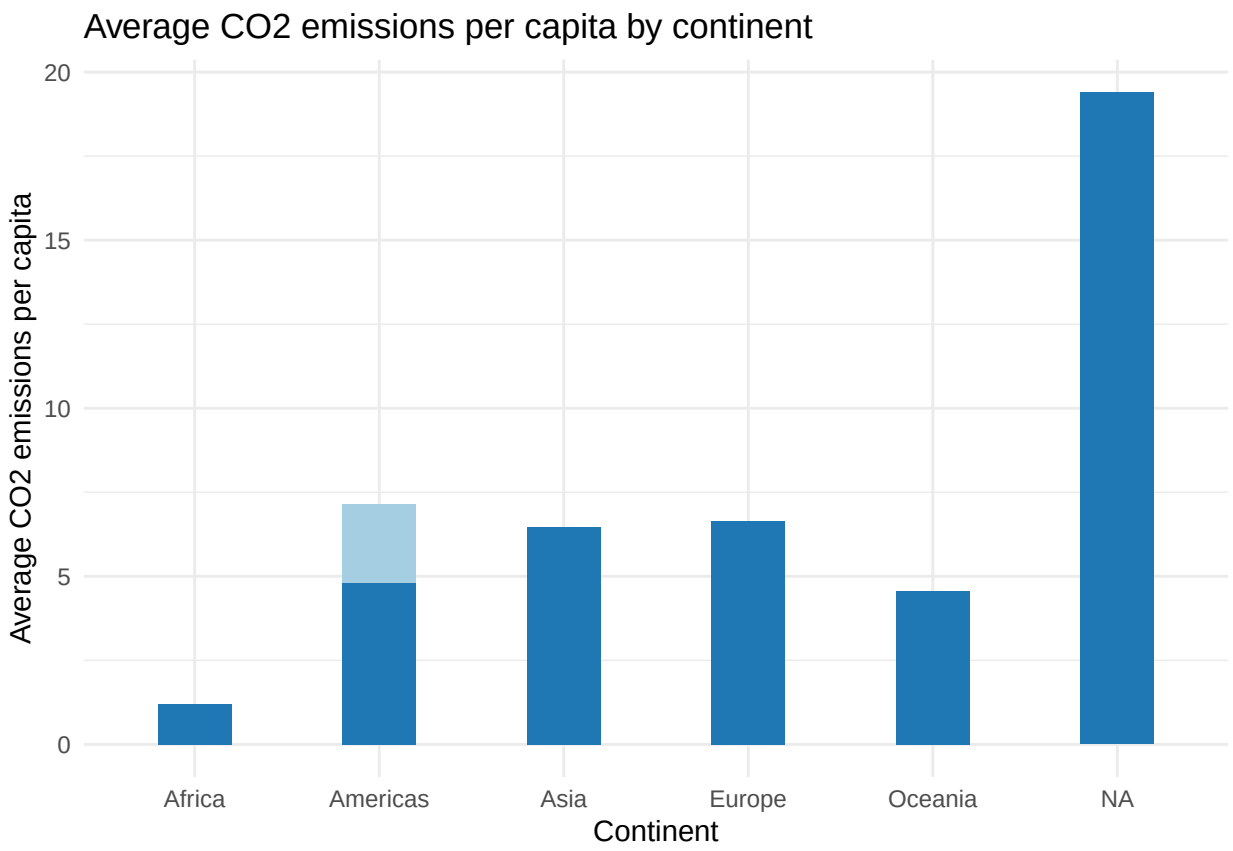
```
co2_pc_by_year <- df %>%
  select(region, time, co2_PC) %>%
  group_by(region, time) %>%
  summarize(avg_co2_pc = mean(co2_PC, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'region'. You can override using the
## `.groups` argument.
```

```
co2_pc_by_year_wide <- co2_pc_by_year %>%
  pivot_wider(names_from = time, values_from = avg_co2_pc)

ggplot(co2_pc_by_year_wide, aes(x = region)) +
  geom_bar(aes(y = `1960`), stat = "identity", fill = "#a6cee3", width = 0.4) +
  geom_bar(aes(y = `2016`), stat = "identity", fill = "#1f78b4", width = 0.4) +
  labs(x = "Continent", y = "Average CO2 emissions per capita", fill = "Year") +
  ggtitle("Average CO2 emissions per capita by continent") +
  theme_minimal()
```

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



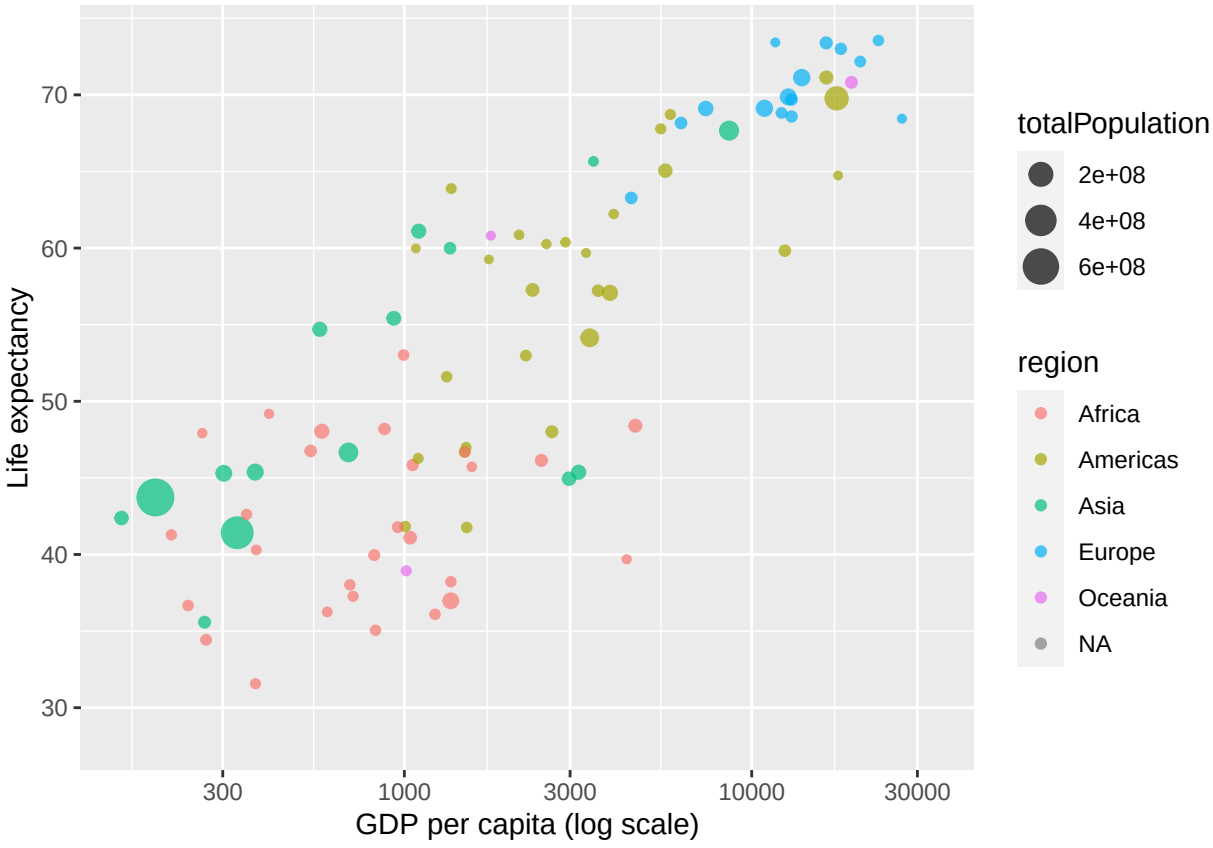
4 GDP per capita (50pt) Let's look at GDP per capita (GDP_PC).

1. (8pt) Make a scatterplot of GDP per capita versus life expectancy by country, using data for 1960. Make the point size dependent on the country size, and color those according to the continent. Feel free to adjust the plot in other ways to make it better. Comment what do you see there.

```
data1960 <- df %>%
  filter(time == 1960) %>%
  select(name, region, GDP_PC, lifeExpectancy, totalPopulation)

ggplot(data1960, aes(x = GDP_PC, y = lifeExpectancy, size = totalPopulation, color = region)) +
  geom_point(alpha = 0.7) +
  scale_x_log10() +
  labs(x = "GDP per capita (log scale)", y = "Life expectancy")
```

```
## Warning: Removed 128 rows containing missing values (geom_point).
```



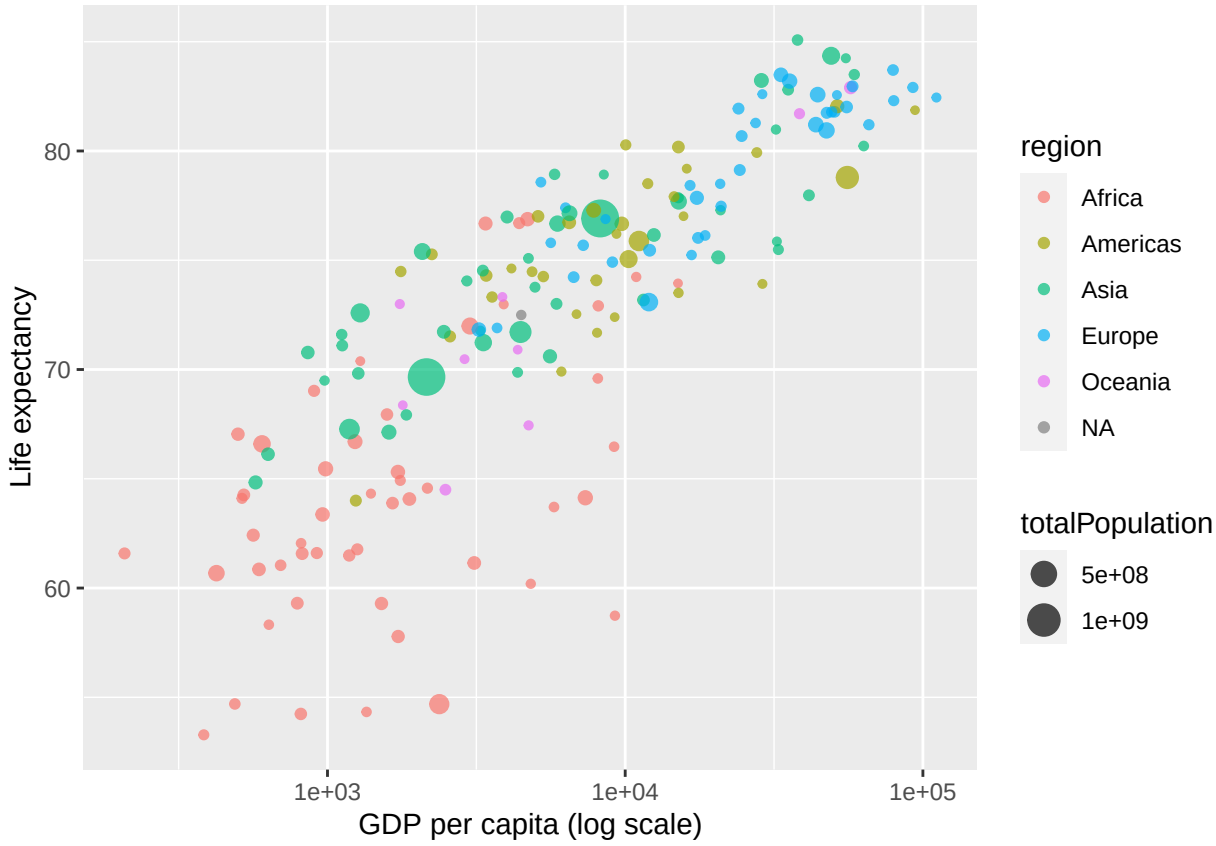
2. (4pt) Make a similar plot, but this time use 2019 data only.

```
data2019 <- df %>%
  filter(time == 2019) %>%
  select(name, region, GDP_PC, lifeExpectancy, totalPopulation)

ggplot(data2019, aes(x = GDP_PC, y = lifeExpectancy, size = totalPopulation, color = region)) +
```

```
  geom_point(alpha = 0.7) +
  scale_x_log10() +
  labs(x = "GDP per capita (log scale)", y = "Life expectancy")
```

```
## Warning: Removed 38 rows containing missing values (geom_point).
```



3. (6pt) Compare these two plots and comment what do you see. How has world developed through the last 60 years?

these two plots show that the world has developed significantly over the last 60 years, with many countries experiencing significant economic growth. However, this growth has come at a cost to the environment, and it will be important for countries to find ways to continue growing their economies while also reducing their carbon footprints and addressing the problem of climate change. ## 4. (6pt) Compute the average life expectancy for each continent in 1960 and 2019. Do the results fit with what do you see on the figures? Note: here as average I mean just average over countries, ignore the fact that countries are of different size.

```
avg_life_exp <- aggregate(lifeExpectancy ~ region + time, data = df, mean)

avg_life_exp_1960 <- subset(avg_life_exp, time == 1960)
avg_life_exp_2019 <- subset(avg_life_exp, time == 2019)

print(avg_life_exp_1960)
```

```
##     region time lifeExpectancy
## 1   Africa 1960       41.46600
## 2 Americas 1960       58.64651
## 3     Asia 1960       51.64931
```

```
## 4   Europe 1960       68.28254
## 5  Oceania 1960       56.39613
```

```
print(avg_life_exp_2019)
```

```
##       region time lifeExpectancy
## 296   Africa 2019       64.11014
## 297 Americas 2019       75.83206
## 298     Asia 2019       74.61739
## 299   Europe 2019       79.35714
## 300  Oceania 2019       73.52827
```

5. (8pt) Compute the average LE growth from 1960-2019 across the continents. Show the results in the order of growth. Explain what do you see. Hint: these data (data in long form) is not the simplest to compute growth. But you may want to check out the lag() function. And do not forget to group data by continent when using lag(), otherwise your results will be messed up! See https://faculty.washington.edu/otoomet/info201-book/dplyr.html#dplyr-helpers-compute.

```
df %>%
  filter(time == 2019 | time == 1960) %>%
  group_by(region, name) %>%
  arrange(time) %>%
  summarize(lifeExpectancy = last(lifeExpectancy)) %>%
  group_by(region) %>%
  mutate(growth = (lifeExpectancy / lag(lifeExpectancy) - 1) * 100) %>%
  filter(!is.na(growth)) %>%
  summarize(avg_growth = mean(growth)) %>%
  arrange(avg_growth)
```

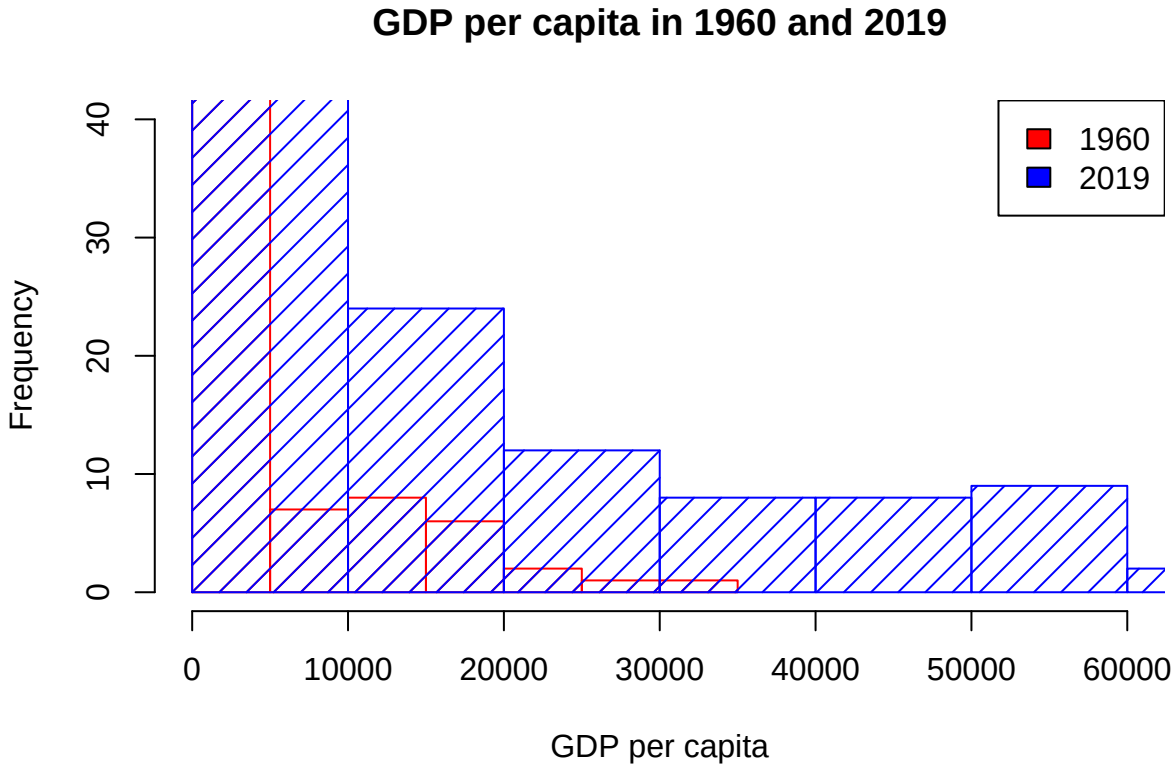
```
## `summarise()` has grouped output by 'region'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 5 x 2
##   region   avg_growth
##   <chr>         <dbl>
## 1 Europe      -0.107
## 2 Oceania      0.0762
## 3 Americas     0.270
## 4 Africa       0.346
## 5 Asia         0.490
```

6. (6pt) Show the histogram of GDP per capita for years of 1960 and 2019. Try to put both histograms on the same graph, see how well you can do it!

```
gdp_1960 <- df[df$time == 1960, ]
gdp_2019 <- df[df$time == 2019, ]
hist(gdp_1960$GDP_PC, col="red", main="GDP per capita in 1960 and 2019", xlab="GDP per capita", xlim=c(0
hist(gdp_2019$GDP_PC, col="blue", add=TRUE, density=10)

legend("topright", c("1960", "2019"), fill=c("red", "blue"))
```



7. (6pt) What was the ranking of US in terms of life expectancy in 1960 and in 2019? (When counting from top.) Hint: check out the function rank()! Hint2: 17 for 1960.

```
df %>%
  filter(!is.na(lifeExpectancy)) %>%
  filter(time == 1960) %>%
  select(name, lifeExpectancy) %>%
  arrange(desc(lifeExpectancy)) %>%
  mutate(rank = rank(desc(lifeExpectancy))) %>%
  filter(name == "United States of America")
```

```
## # A tibble: 1 x 3
##   name                     lifeExpectancy  rank
##   <chr>                             <dbl> <dbl>
## 1 United States of America           69.8    18
```

```
df %>%
  filter(!is.na(lifeExpectancy)) %>%
  filter(time == 2019) %>%
  select(name, lifeExpectancy) %>%
  arrange(desc(lifeExpectancy)) %>%
  mutate(rank = rank(desc(lifeExpectancy))) %>%
  filter(name == "United States of America")
```

```
## # A tibble: 1 x 3
##   name                     lifeExpectancy  rank
##   <chr>                             <dbl> <dbl>
## 1 United States of America           78.8    47
```

8. (6pt) If you did this correctly, then you noticed that US ranking has been falling quite a bit. But we also have more countries in 2019–what about the relative rank divided by the corresponding number of countries that have LE data in the corresponding year? Hint: 0.0904 for 1960.

```
le_2019 <- subset(df, time == 2019)
us_2019 <- subset(le_2019, name == "United States of America")
rank_2019 <- rank(-us_2019$lifeExpectancy)
rel_rank_2019 <- rank_2019 / sum(!is.na(le_2019$lifeExpectancy))
rel_rank_2019
```

```
## [1] 0.005050505
```

Finally tell us how many hours did you spend on this PS

About 5 hours