Midterm I

September 30 2022

Section I: Gapminder Data

Health and income outcomes for 142 countries from 1952 to 2007 in increments of 5 years. The variables in the dataset are `country`, `continent`, `year`, `lifeExp`, `pop`, and `gdpPercap`. The descriptions for the variables are:

- `country` : name of the country, factor with 142 levels
- `continent`: name of the continent, factor with 5 levels
- `year` : ranges from 1952 to 2007 in increments of 5 years (12 distinct years)
- `lifeExp`: life expectancy at birth, in years
- `pop` : population
- `gdpPercap` : GDP per capita (US$, inflation-adjusted)

```
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, ~
```

Part 1: What do the following codes do?

Provide a thorough and intuitive (2-3 sentences) description of the output from each of the following R chunks. The chunks either produce a new data set or a new plot; if it's a new data set, give the dimensions in addition to your description. Write your descriptions in regular English, without using variable names.

a.

```
gapminder %>%
  filter(
    continent == "Africa",
    country == "Somalia",
     year %in% seq(1952, 2007, 5)) %>%
 slice_min(lifeExp, n=1)
```

b.

```
gapminder %>%
  filter(year == 2007) %>%
  group_by(continent) %>%
  summarise(totalPop = sum(pop)) %>%
  arrange(totalPop)
```

c.

```
gapminder %>%
  group_by(country, continent) %>%
  summarize(meanPop = mean(pop)/1000000, meanLifeExp = mean(lifeExp)) %>%
  filter(continent == "Europe") %>%
  ggplot(aes(x = meanPop, y = meanLifeExp, color = country, label = country)) +
  geom_point() +
  scale_color_discrete(guide = "none") +
  ggrepel::geom_text_repel()
```

d.

```
gapminder %>%
  filter(year == 2007) %>%
  mutate(rank = min_rank(desc(lifeExp))) %>%
  filter(rank < 50) %>%
  arrange(rank) %>%
  slice_min(rank, n = 10) %>%
  select(-continent, -pop, -year, -gdpPercap)
```

e.

```
gapminder %>%
  group_by(continent, year) %>%
  summarize(
      mean_le = mean(lifeExp),
      median_le = quantile(lifeExp, 0.50),
      min_le = min(lifeExp),
      max_le = max(lifeExp),
      se_le = sd(lifeExp)/sqrt(n()))%>%
 slice_max(mean_le, n=1) %>%
 ungroup()
```

Part 2: Miscellaneous

a. Use piping operation to combine the following set of codes into one chained line of code to produce the same output. Also, explain the output of the code.

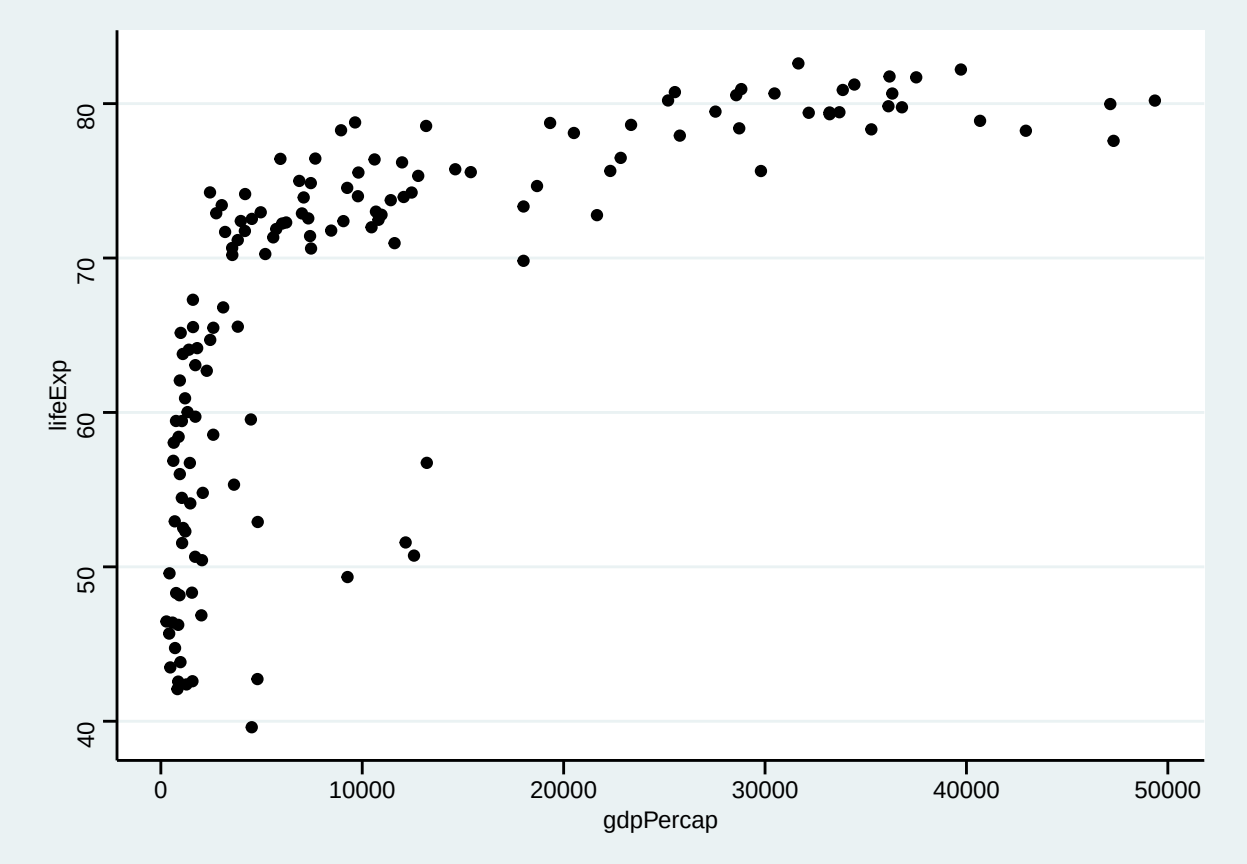
```
arrange(filter(gapminder, year == 1952), desc(gdpPercap))
```

b. Use piping operation to combine the following set of codes into one chained line of code to produce the same output. Also, explain the output of the code.

```
plot_data <- filter(gapminder, year == 1972 & continent %in% c("Africa", "Americas"))
ggplot(plot_data, aes(x = lifeExp, y = gdpPercap)) +
  geom_point()
```

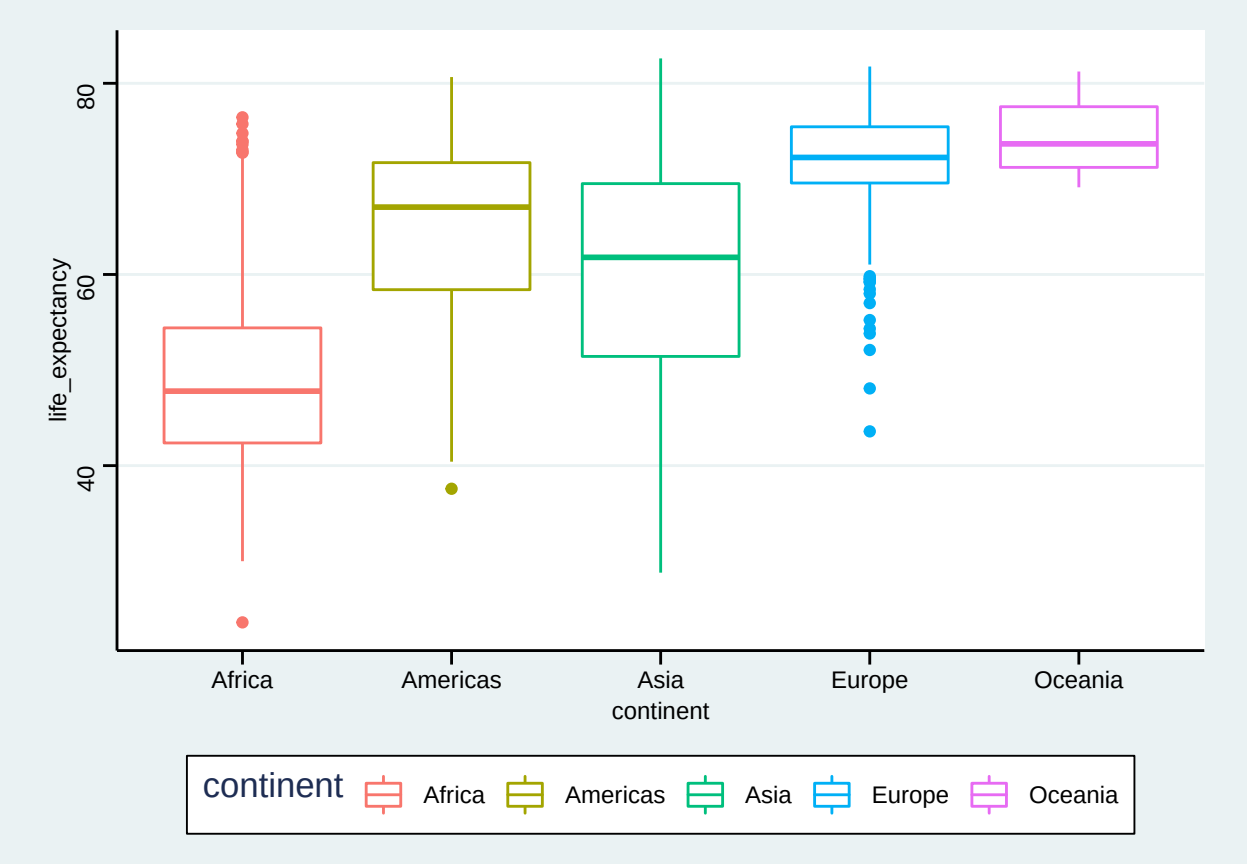
c. How would you improve the aesthetics of the plot given below? Write your code modifications.

```
gapminder %>%
  filter(year == 2007) %>%
  ggplot(aes(x = gdpPercap, y = lifeExp), size = pop, color = continent) +
  geom_point()
```



d. Please take a look at the `# code 1` and the resulting plot.

```
# code 1
gapminder %>%
  group_by(continent, year) %>%
  summarize(life_expectancy = lifeExp) %>%   # line 1
  ggplot(aes(x = continent, y = life_expectancy, color = continent)) +
  geom_boxplot()
```



Will the plot look the same or different if we switch to `# code 2`? Explain.

```
# code 2
gapminder %>%
  group_by(continent, year) %>%
  summarize(life_expectancy = mean(lifeExp)) %>%
  ggplot(aes(x = continent, y = life_expectancy, color = continent)) +
  geom_boxplot()
```

Section II: R objects

Part 3. Consider the following objects to answer the questions below.

```
x <- 1:4
y <- c(TRUE, as.factor(c(2,"3")), 4)
z <- list(z1 = x, z2 = y, z3 = c("cellar","door"))
```

(a) Consider the objects x, y and z. Which are atomic vectors and which are lists?

(b) What does the following command evaluate to? Briefly explain your answer.

```
typeof(x)  # line 1
typeof(y) # line 2
```

(c) What does the following command evaluate to? Briefly explain your answer.

```
z[["z3"]][1]
```

(d) What does the following command evaluate to? Briefly explain your answer.

```
z[3][[1]][2]
```

(e) What does the following command evaluate to? Briefly explain your answer.

```
x + y
```