Exploring Factors (HW5 - STAT 545)

Phuong (Sam) Can

15/10/2019

Contents

`here` Package

We know that the `here` package is recommended over base R functions to set or get a working directory. Why? First, because functions in the `here` package uses relative paths instead of absolute paths like set_wd() or get_wd(). Moreover, `here` allows working with files that are not necessarily in the working directory. Second, by setting a relative path for a file (an R file, an image, or a dataset), `here` helps retrieving it from the root up even if the file is removed from the folder that contains the working directory. Third, `here` functions to specify a file path simply contain the name of two folders (the one that contains the working directory and the one before it) and the file name, which is much simpler than the base R functions. All of these functionalities in `here` help increase reproducibility in R projects.

Factor Management

I fell in love with gapminder and the wealth of useful information it contains. Let's look at the dataset one more time in this last assignment of STAT 545.

We will examine the levels in the factor `continent` and drop one of them: Oceania.

Before dropping the level and all related observations, we have 5 levels in the factor `continent`: Africa, Americas, Asia, Europe, Oceania. The dataset now has 1704 rows.

Let's now drop Oceania:

```
gap_drop <-
  gapminder %>%
  filter(!continent == "Oceania")
gap_drop <- gap_drop %>% droplevels()
attach(gap_drop)
```

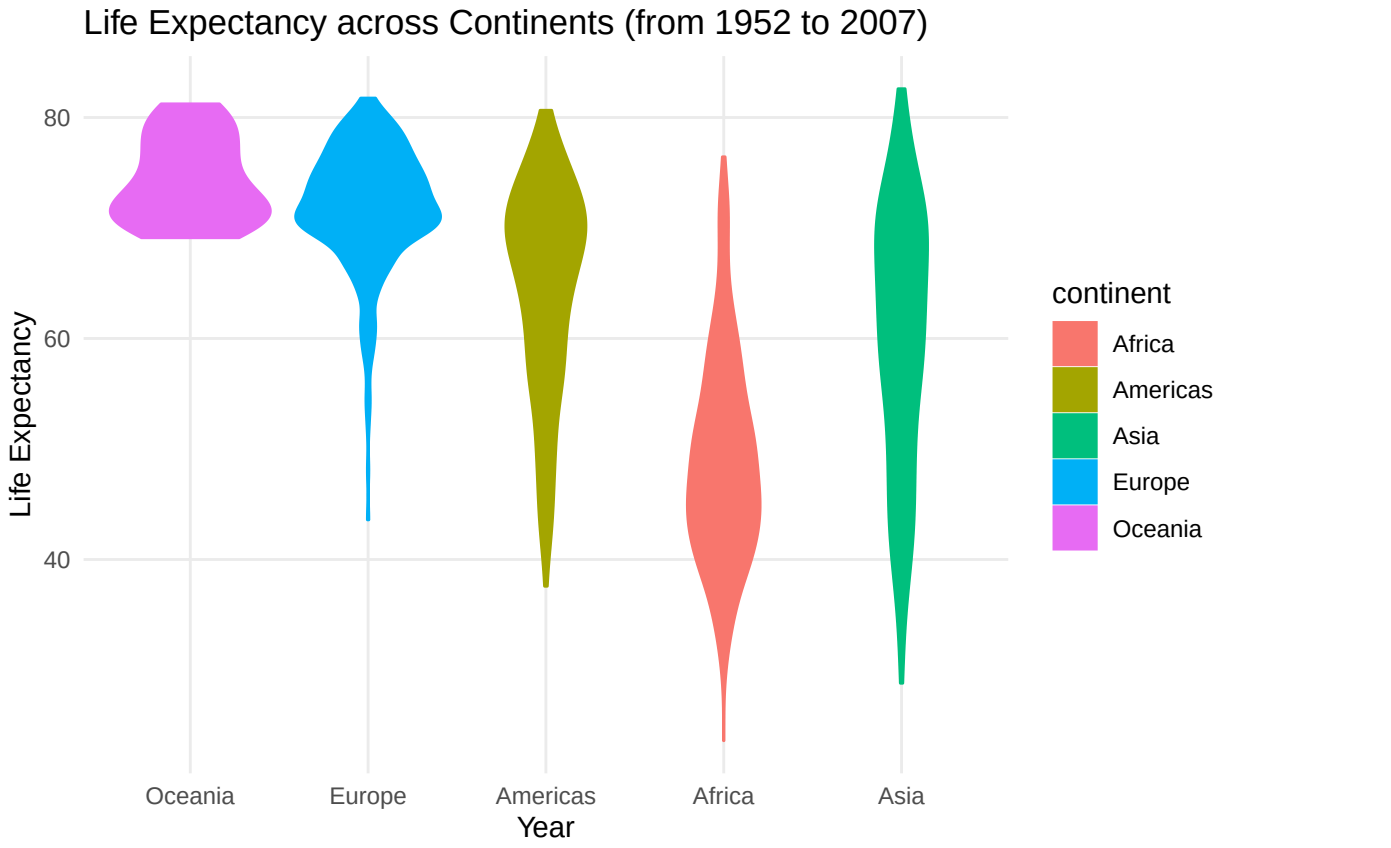
We should have 4 levels in the factor `continent` now as following: Africa, Americas, Asia, Europe. The dataset now only has 1680 rows, which means that the dropped level, Oceania, contains 24 observations.

Now we would undrop the level Oceania by using `gap_life_diff` instead of `gap_drop` and reorder the levels of factor `continent` based on the spread of each continent's `lifeExp` throughout the years from 1952 to 2007.

```
detach(gap_drop)

gap_life_diff <-
  gapminder %>%
  group_by(continent) %>%
  mutate(life_diff = max(lifeExp) - min(lifeExp))

gap_life_diff %>%
  ggplot(aes(continent, lifeExp, color = continent, fill = continent)) +
  geom_violin(aes(fct_reorder(continent, life_diff))) +
  theme_minimal() +
  scale_x_discrete(drop = FALSE) +
  theme(panel.grid.minor = element_blank()) +
  labs(title="Life Expectancy across Continents (from 1952 to 2007)", x="Year", y="Life Expectancy")
```



File Input/Output

Now, assuming that we are ignorant and don't care about the statistics in Oceania, only the other four continents. We will save `gap_drop` as a separate file under a sub-folder of the folder that contains my current working directory and call it **sub_hw5**.

```
write_csv(gap_drop, here::here("sub_hw5", "gap_drop.csv"))
```

Then, let's import the file into R again.

```
read_csv(here::here("sub_hw5", "gap_drop.csv"))
```

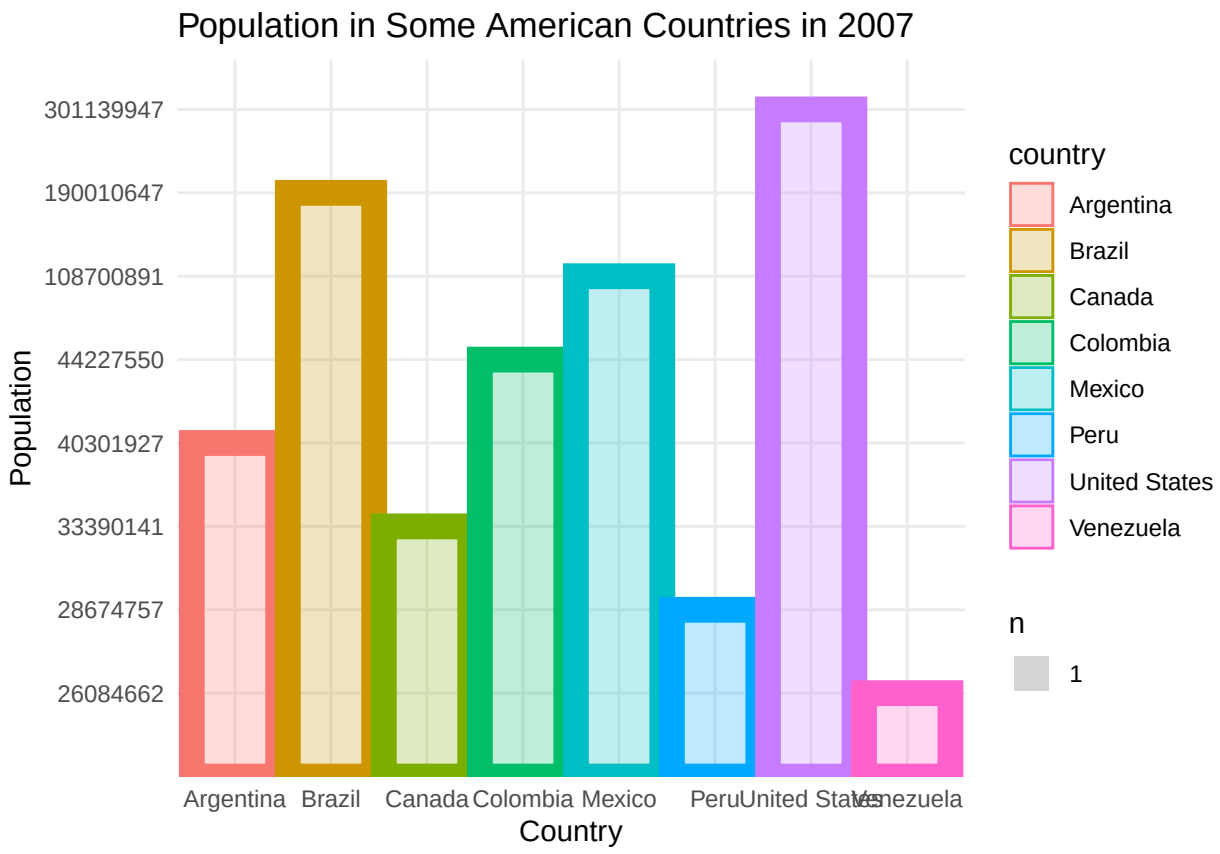
Notice that: specifying other folders in `here()` only works for sub-folders of the current working directory. A random file path wouldn't work. This means that if I planned to save the file in the current directory, I wouldn't have to specify any folder name but simply the name of the new file.

Now, let's examine and reorder countries within one of the four continents: Americas, based on the countries' population in 2007, via a dotplot. We will only work with countries with `pop` higher than double the median population of Americas in 2007.

```
gap_drop07 <-
  gap_drop %>%
  filter(year == 2007) %>%
  filter(continent == "Americas") %>%
  filter(pop > median(pop)*2)
gap_drop07$pop <- format(gap_drop07$pop, scientific = FALSE)
gap_drop07$country <- droplevels(gap_drop07$country)
gap_drop07
```

```
## # A tibble: 8 x 6
##   country       continent  year lifeExp pop         gdpPercap
##   <fct>         <fct>     <int>   <dbl> <chr>           <dbl>
## 1 Argentina     Americas   2007    75.3 " 40301927"    12779.
## 2 Brazil        Americas   2007    72.4 190010647       9066.
## 3 Canada        Americas   2007    80.7 " 33390141"    36319.
## 4 Colombia      Americas   2007    72.9 " 44227550"     7007.
## 5 Mexico        Americas   2007    76.2 108700891      11978.
## 6 Peru          Americas   2007    71.4 " 28674757"     7409.
## 7 United States Americas   2007    78.2 301139947      42952.
## 8 Venezuela     Americas   2007    73.7 " 26084662"    11416.
```

```
gap_drop07 %>%
  ggplot(aes(country, pop, color = country, fill = country)) +
  geom_bar(stat = "sum", alpha = 0.25) +
  theme_minimal() +
  theme(panel.grid.minor = element_blank()) +
  labs(title="Population in Some American Countries in 2007", x="Country", y="Population")
```


Population in Some American Countries in 2007
301139947
190010647
108700891
44227550
40301927
33390141
28674757
26084662
Population
Argentina
Brazil
Canada
Colombia
Mexico
Peru
United States
Venezuela
Country
country
Argentina
Brazil
Canada
Colombia
Mexico
Peru
United States
Venezuela
n
1

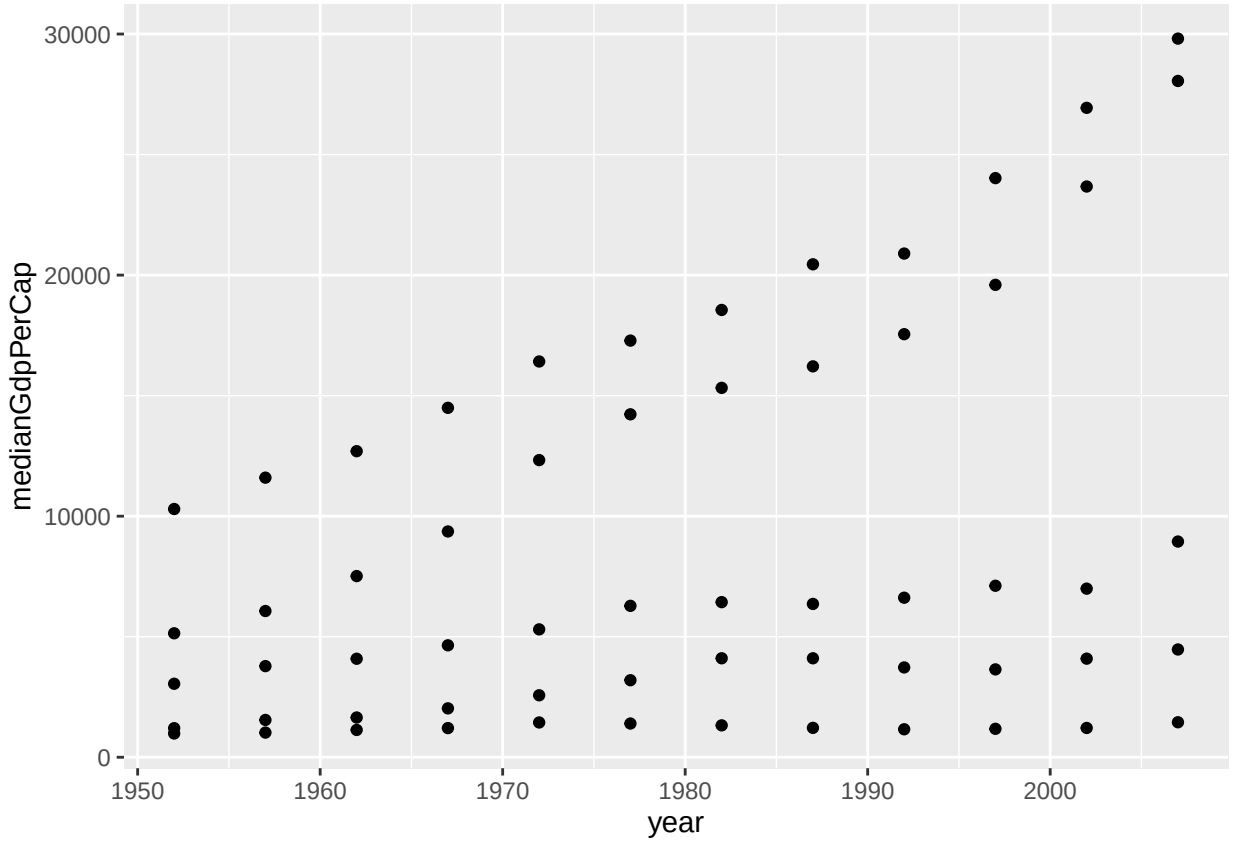
Visualization Design

To appreciate how much I have learned from STAT 545, I will revamp a graph that I made from the very beginning of the course and change it based on what I learned about ggplots.

In assignment 1, I summarized the median GDP per capita within each continent within each year and plot the results to show the change of the median GDP per capita over time. The codes below are pasted from the assignment:

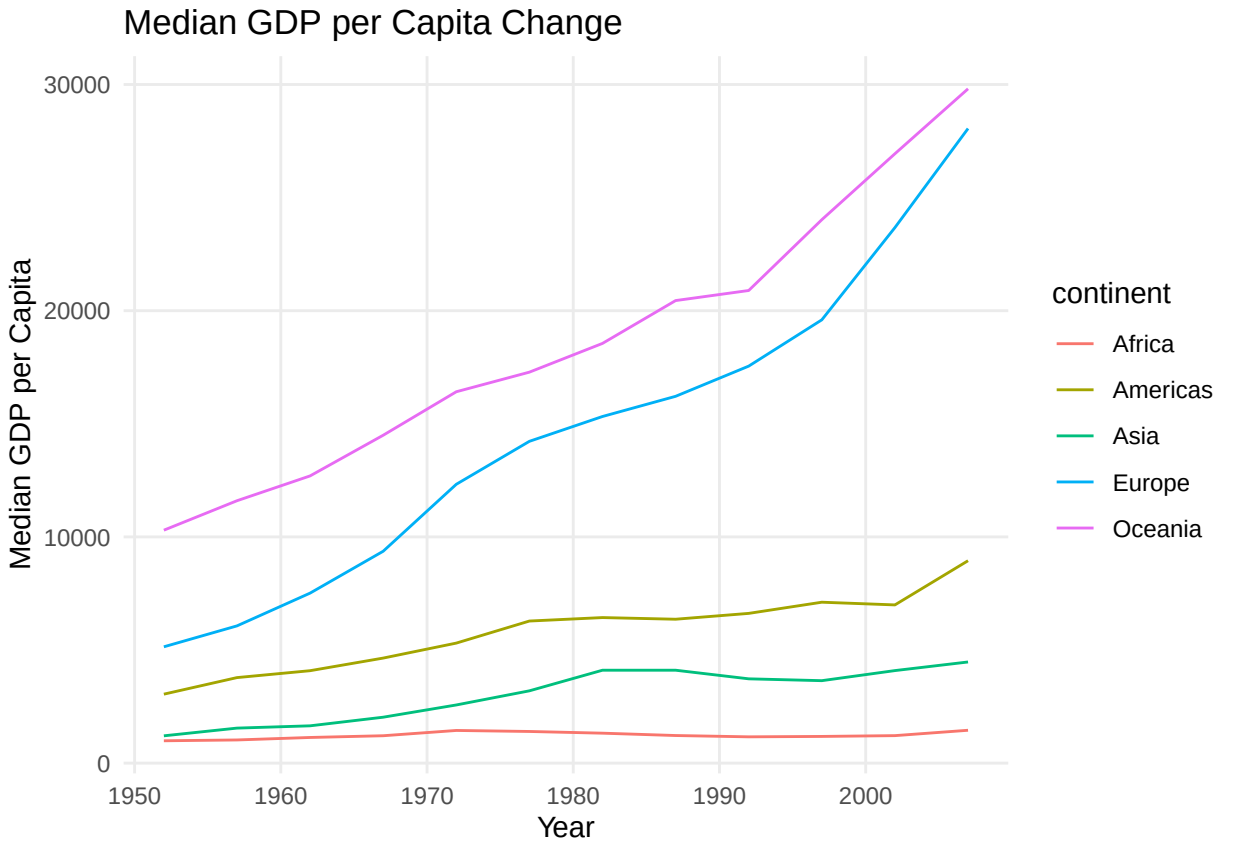
```
change_medianGdpPerCap <- gapminder %>% group_by(year, continent) %>% summarize(medianGdpPerCap = media

(plot1 <- ggplot(change_medianGdpPerCap, aes(x = year, y = medianGdpPerCap)) + geom_point())
```



I will declutter this plot by changing it into a line graph, relabeling the axes, adding colors corresponding to different continents, adopting the minimal theme, and eliminate unnecessary grid elements in the background.

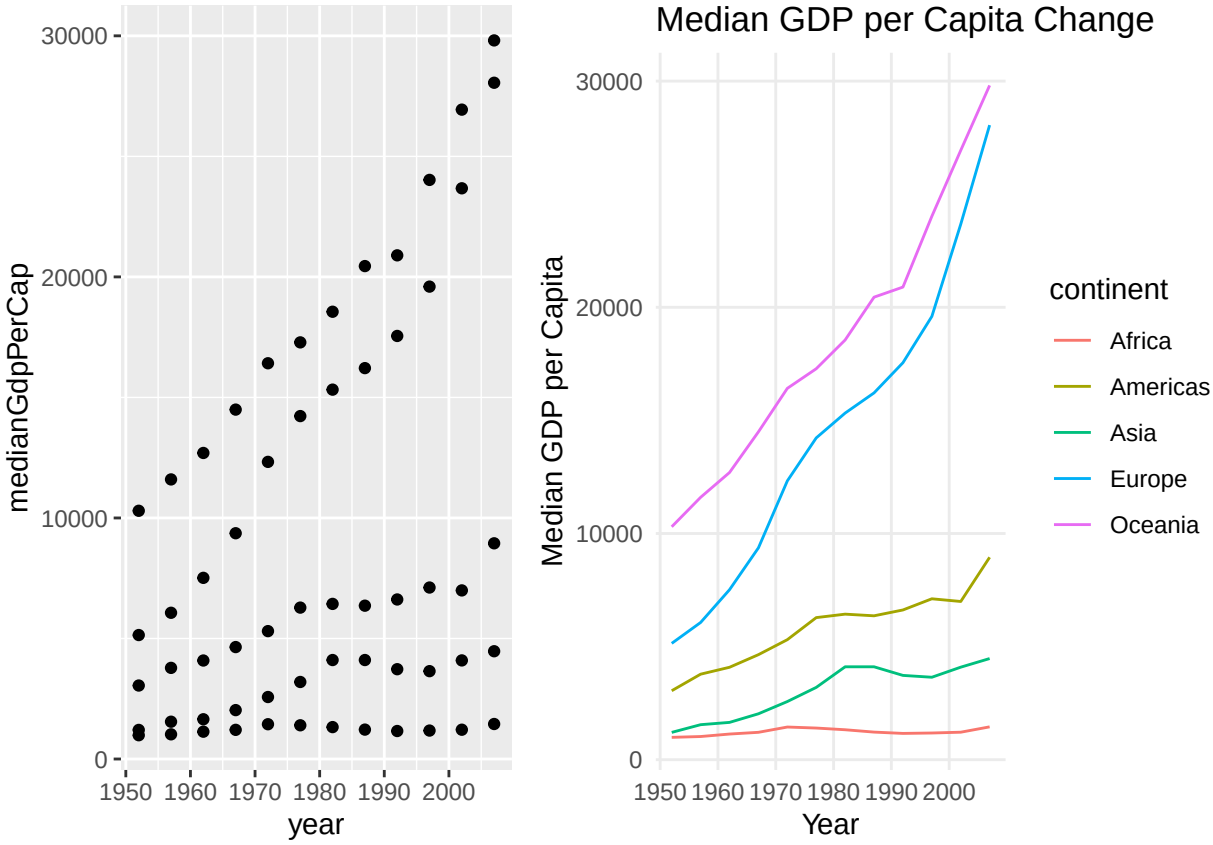
```
(plot2 <- ggplot(change_medianGdpPerCap, aes(x = year, y = medianGdpPerCap, color = continent)) +
  geom_line() +
  labs(title="Median GDP per Capita Change", x="Year", y="Median GDP per Capita") +
  theme_minimal() +
  theme(panel.grid.minor = element_blank()))
```



This graph hopefully looks better! A line graph works better than a scatterplot in this case, the axes are clearer, a white, minimal background and colored lines hopefully look more visually attractive and make more intuitive sense for readers.

Placing these plots side-by-side:

```
(plot_revamped <- gridExtra::grid.arrange(plot1, plot2, ncol = 2, widths = c(9, 12)))
```




```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
```

Writing Figures to File

Let's save the plot I just made using `ggsave()` into a `.png` file with specified width, height and resolution.

```
ggsave("median_gdp_change_revamped.png", plot = plot_revamped, width = 15, height = 10, dpi = 320)
```

You can find this plot file in my `hw05_gapminder` folder on Github.

The End