Data visualization intermdiates

Jae Yeon Kim

05 January, 2019

Motivation

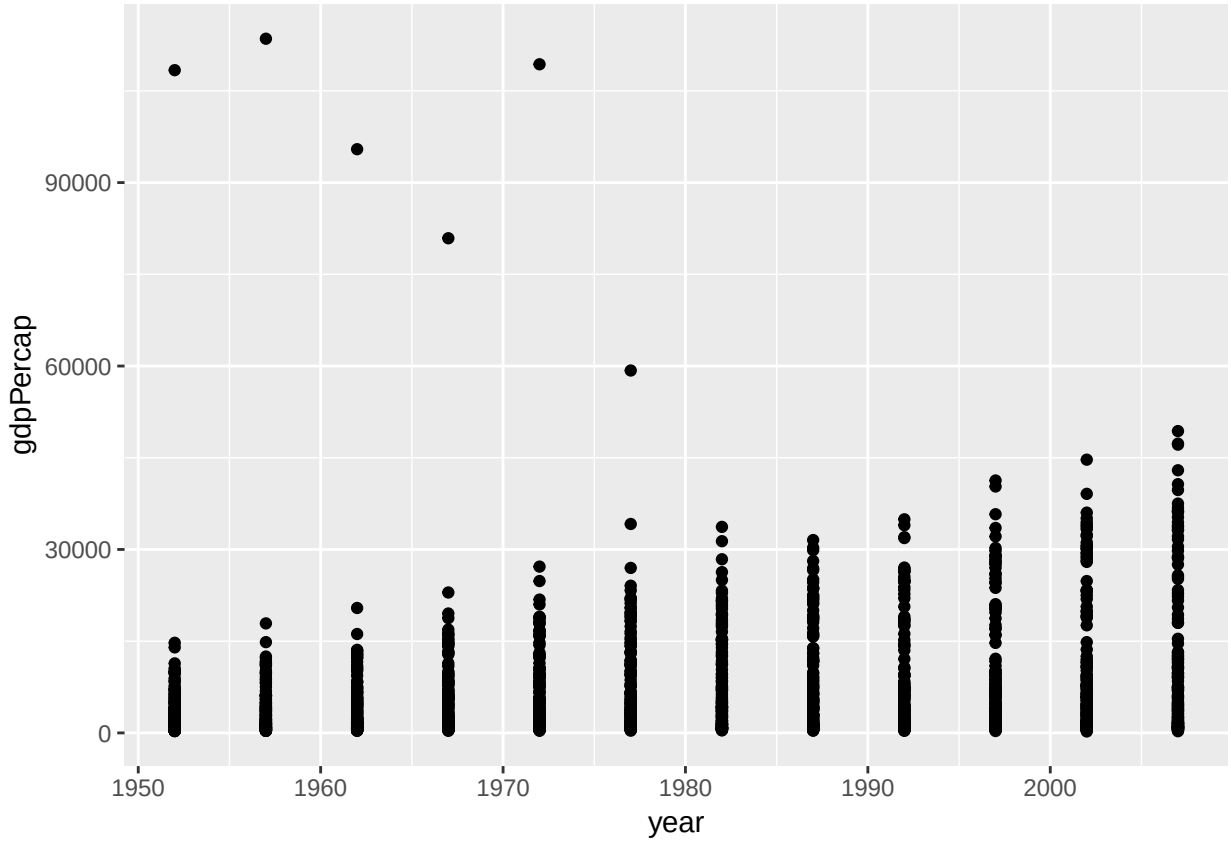
- The following material is adapted from Kieran Healy's wonderful book (2018) on data visualization.

ggplot2 intermediates

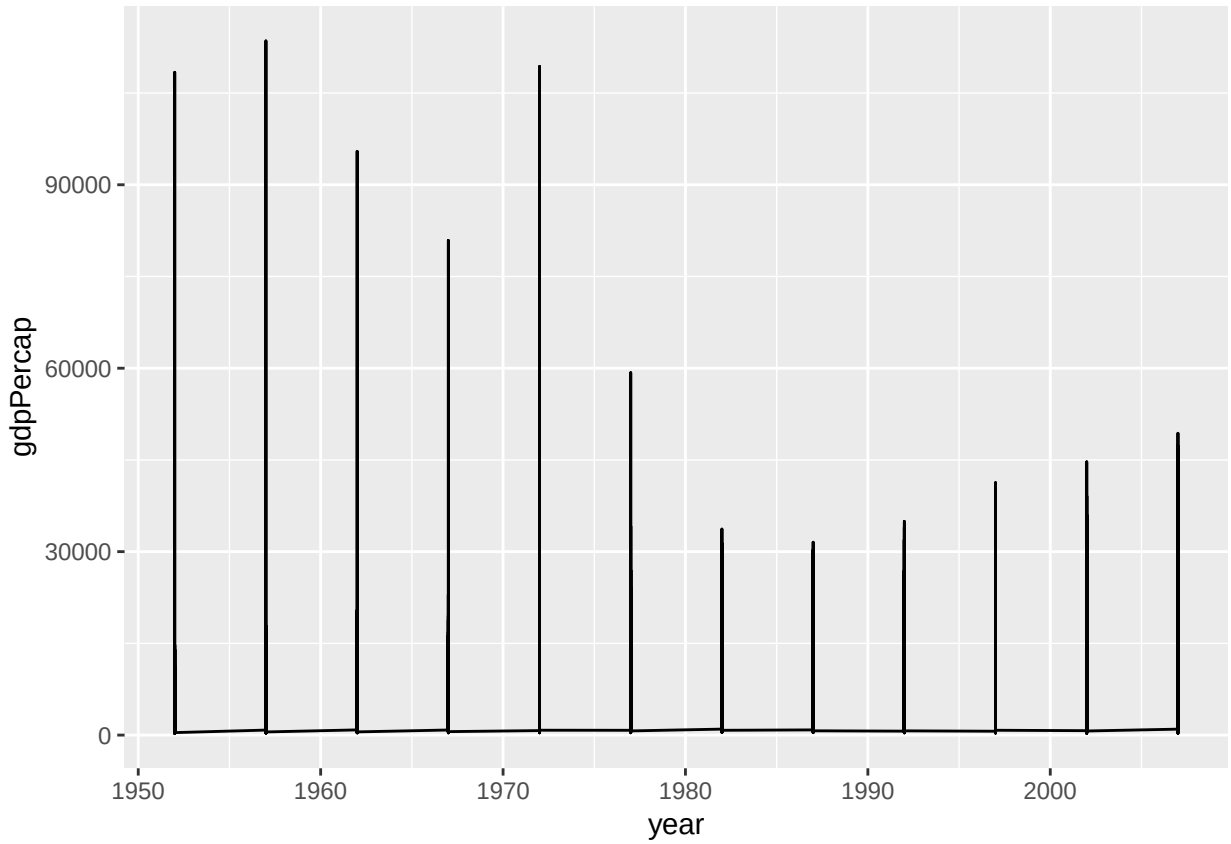
Grouping and facetting

- Can you guess what's wrong?

```
p <- ggplot(gapminder, aes(x = year, y = gdpPercap))

p + geom_point()
```




```
p + geom_line()
```

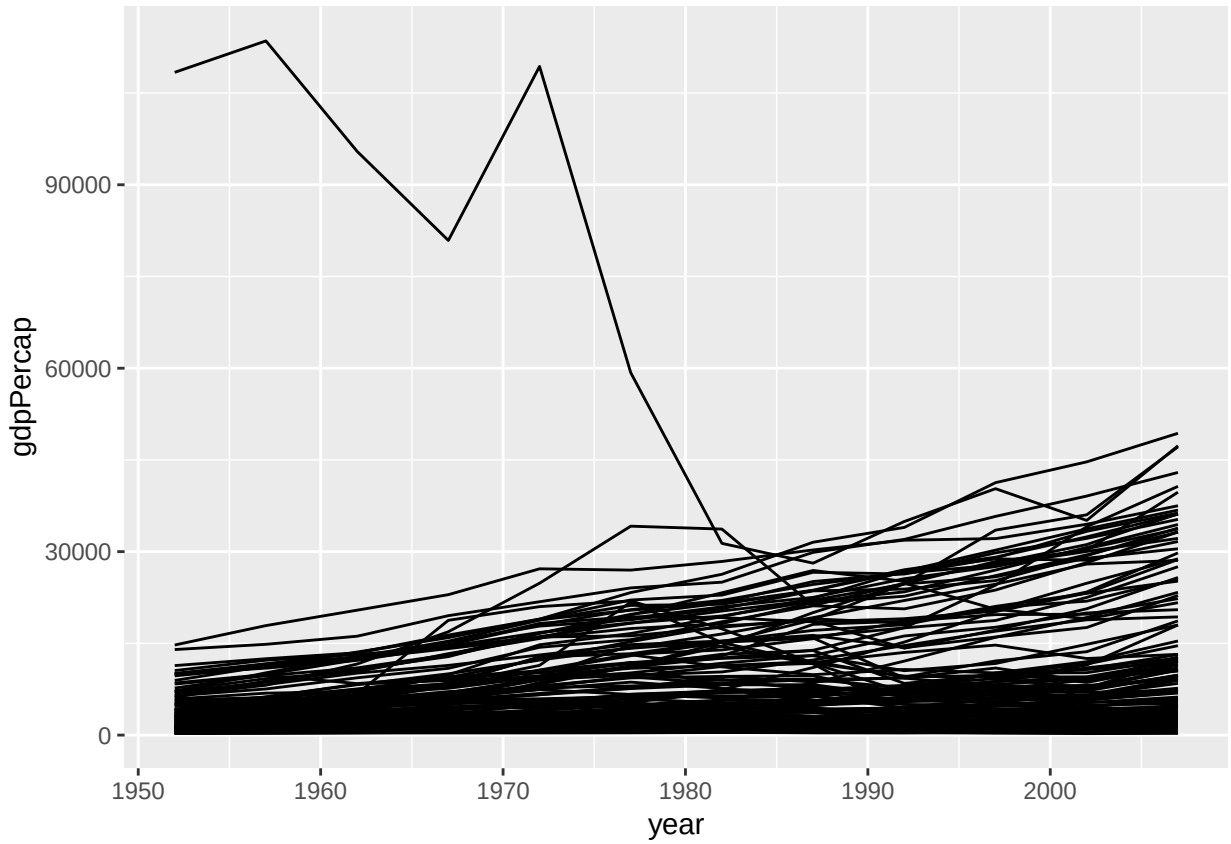



```
gapminder
```

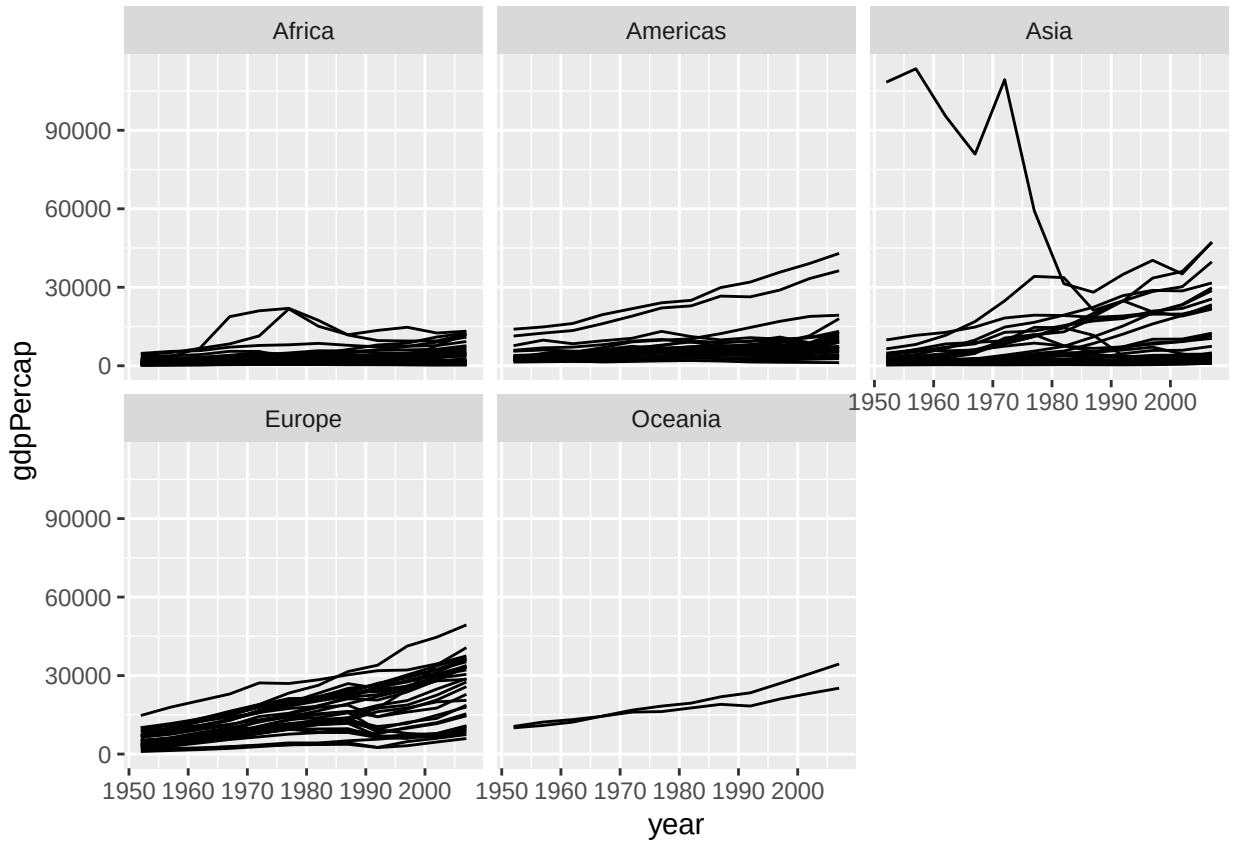
```
## # A tibble: 1,704 x 6
##    country     continent  year lifeExp      pop gdpPercap
##    <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Afghanistan Asia       1952    28.8  8425333      779.
##  2 Afghanistan Asia       1957    30.3  9240934      821.
##  3 Afghanistan Asia       1962    32.0 10267083      853.
##  4 Afghanistan Asia       1967    34.0 11537966      836.
##  5 Afghanistan Asia       1972    36.1 13079460      740.
##  6 Afghanistan Asia       1977    38.4 14880372      786.
##  7 Afghanistan Asia       1982    39.9 12881816      978.
##  8 Afghanistan Asia       1987    40.8 13867957      852.
##  9 Afghanistan Asia       1992    41.7 16317921      649.
## 10 Afghanistan Asia       1997    41.8 22227415      635.
## # ... with 1,694 more rows
```

- Use grouping and facetting to clarify

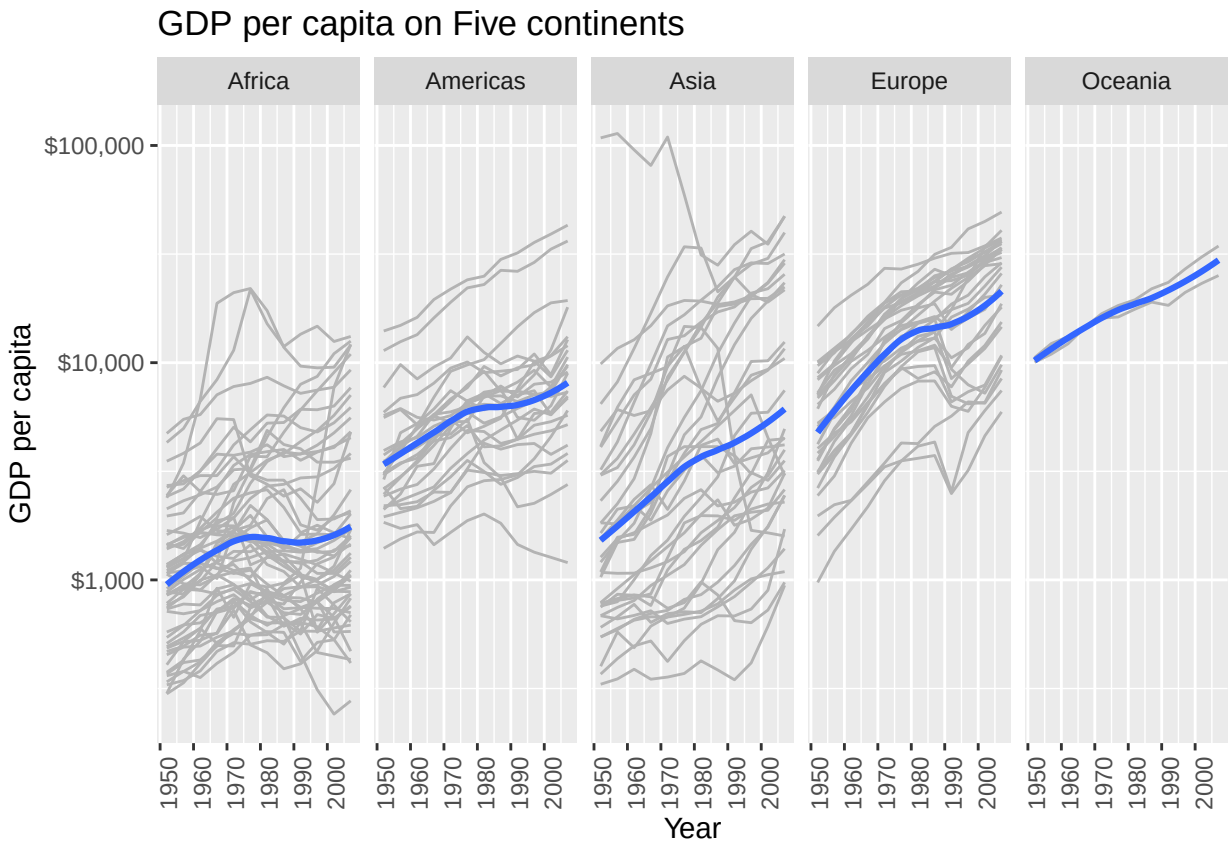
```
p <- ggplot(gapminder, aes(x = year, y = gdpPercap))

p + geom_line(aes(group = country)) # group by
```




```
p + geom_line(aes(group = country)) + facet_wrap(~continent) # facetting
```

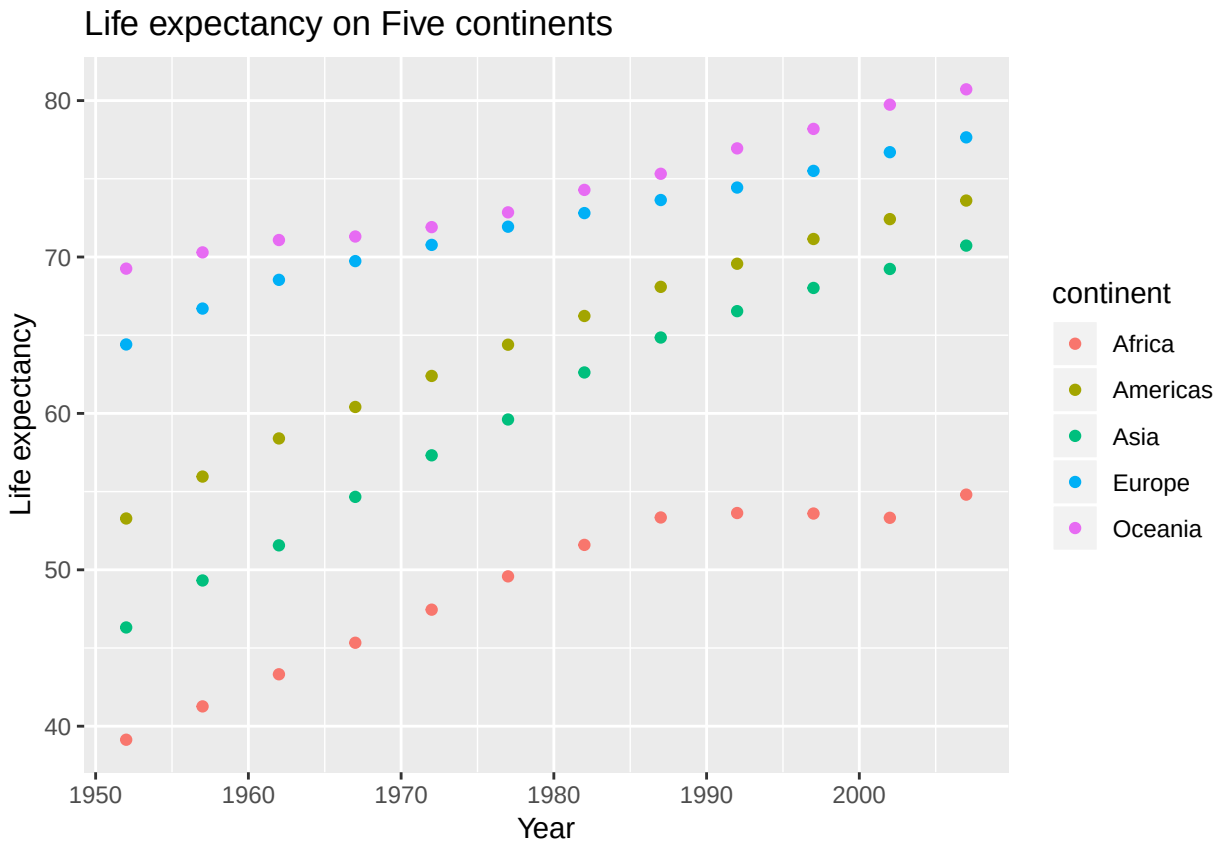



```
p + geom_line(aes(group = country), color = "gray70") +
  geom_smooth(size = 1.1, method = "loess", se = FALSE) +
  scale_y_log10(labels = scales::dollar) +
  facet_wrap(~continent, ncol = 5) + # for single categorical variable; for multiple categorical variab
  labs(x = "Year",
       y = "GDP per capita",
       title = "GDP per capita on Five continents") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

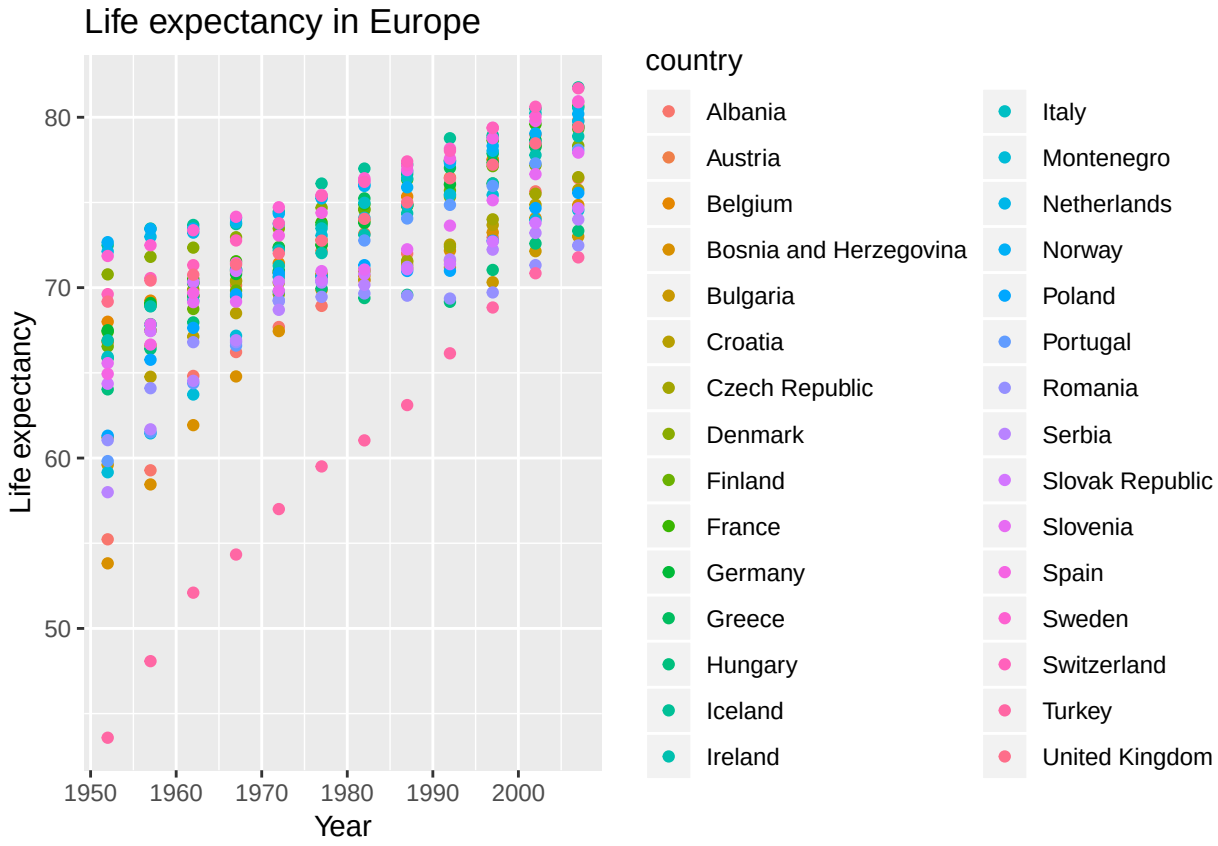


Use pipes to summarize data

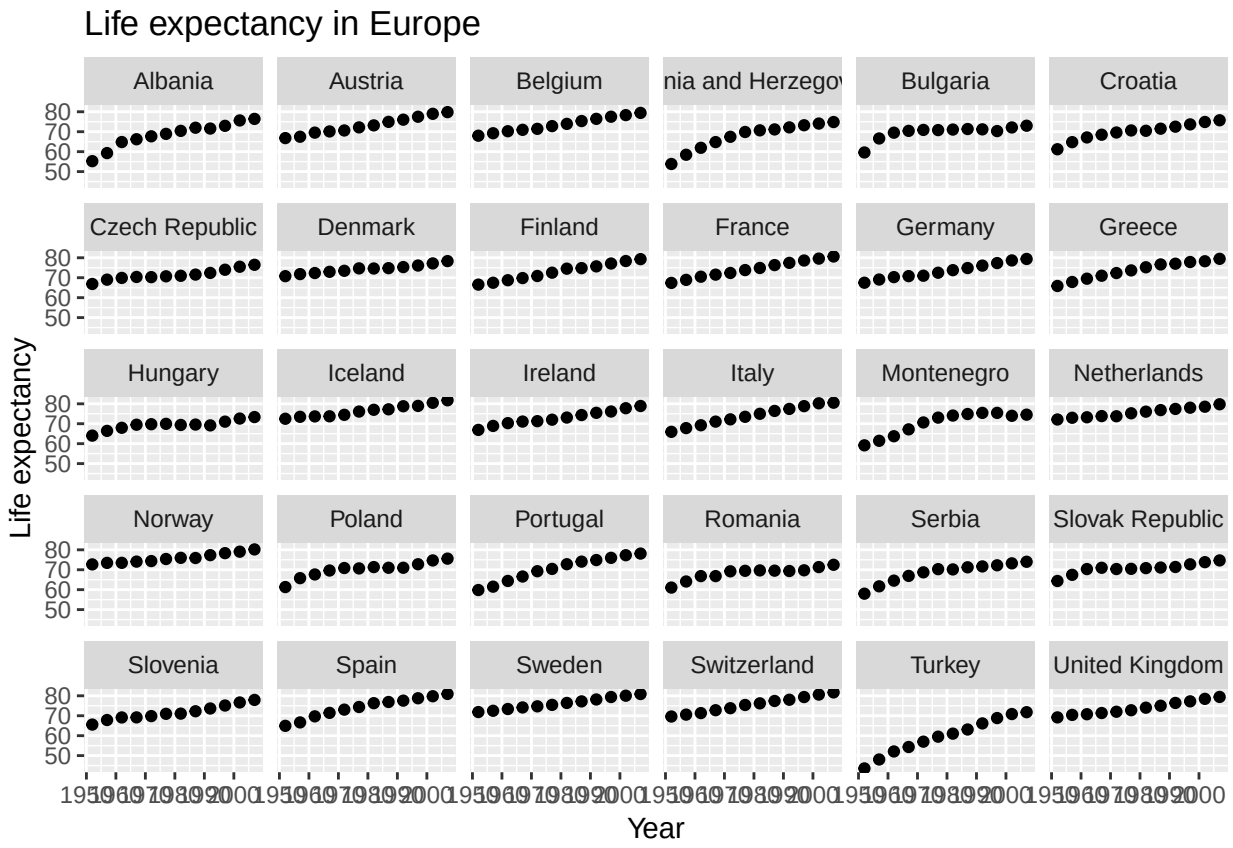
```
gapminder %>%
  group_by(continent, year) %>%
  summarize(gdp_mean = mean(gdpPercap),
            lifeExp_mean = mean(lifeExp)) %>%
  ggplot(aes(x = year, y = lifeExp_mean, color = continent)) +
  geom_point() +
    labs(x = "Year",
       y = "Life expectancy",
       title = "Life expectancy on Five continents")
```



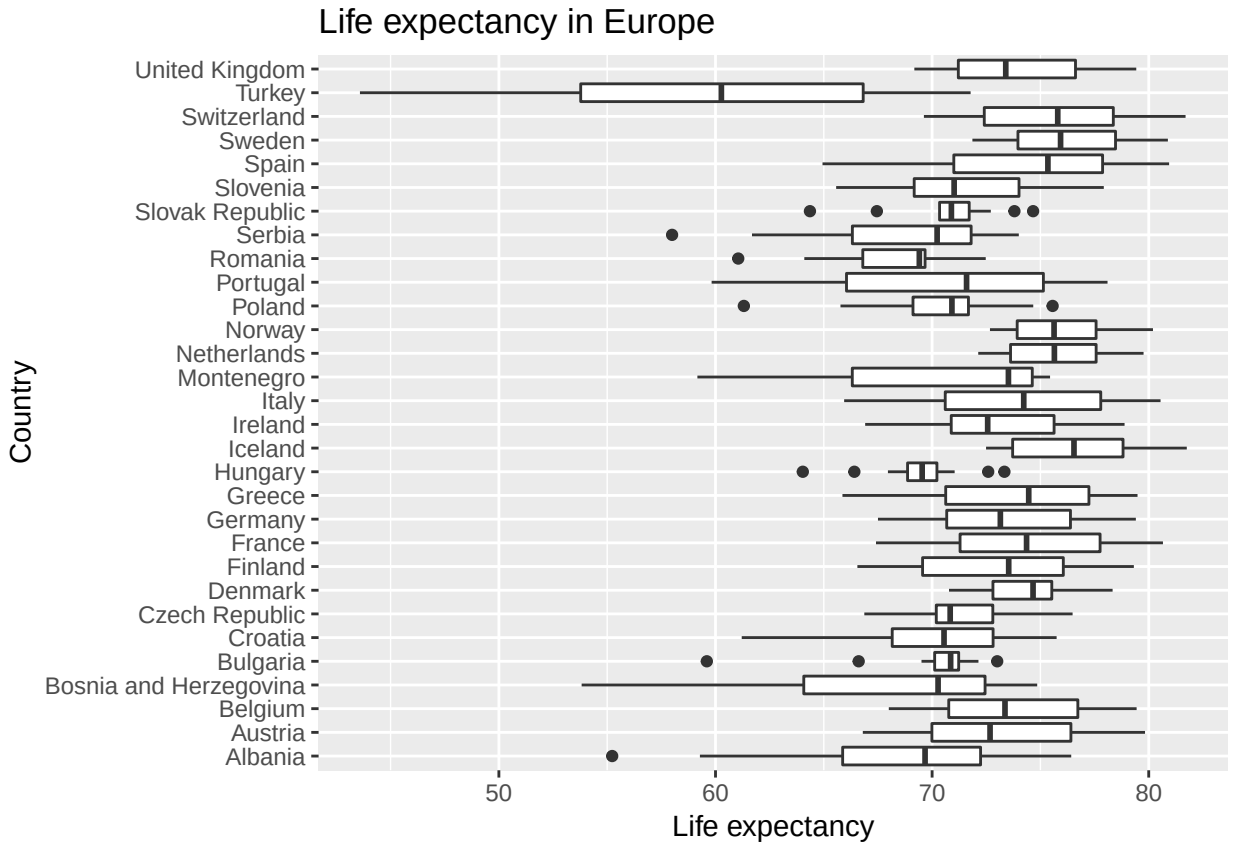

```
gapminder %>%
  filter(continent == "Europe") %>%
  group_by(country, year) %>%
  summarize(gdp_mean = mean(gdpPercap),
            lifeExp_mean = mean(lifeExp)) %>%
  ggplot(aes(x = year, y = lifeExp_mean, color = country)) +
  geom_point() +
    labs(x = "Year",
       y = "Life expectancy",
       title = "Life expectancy in Europe")
```



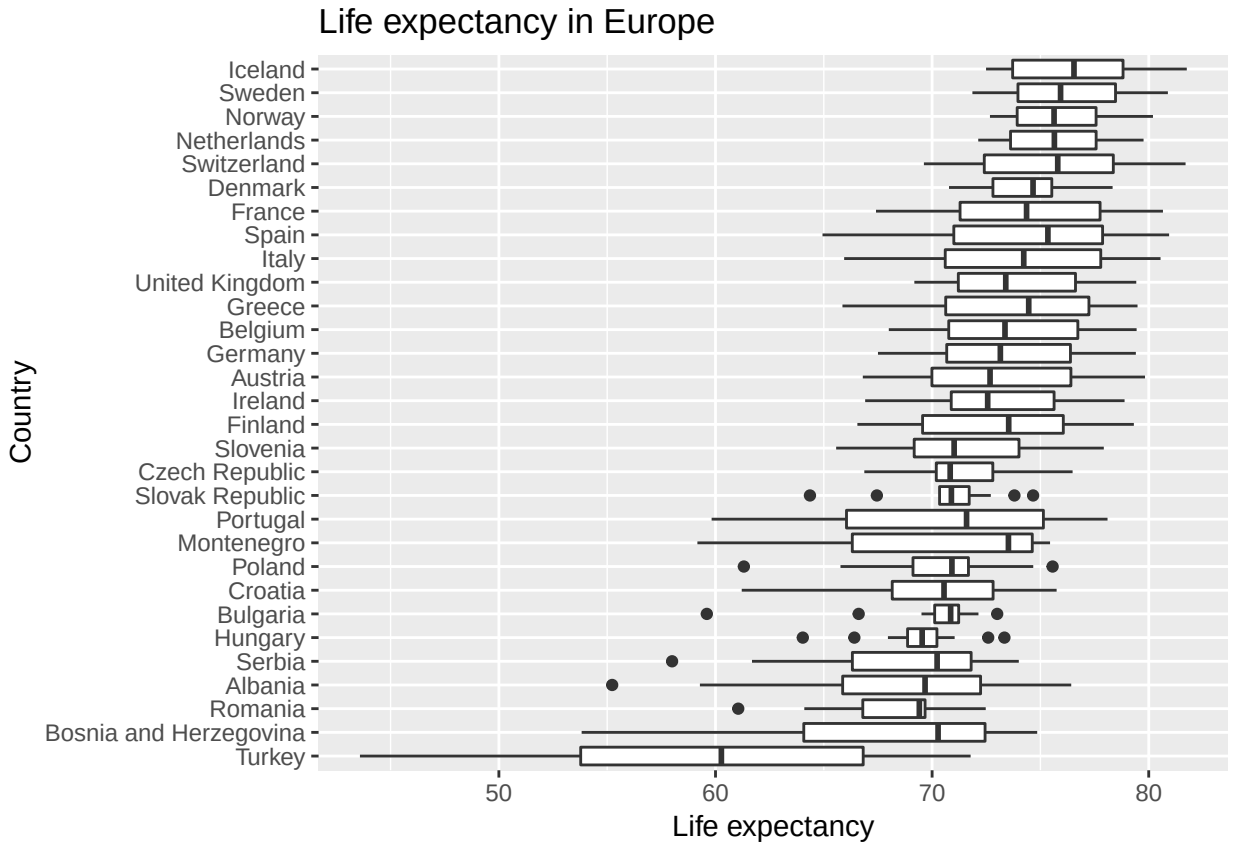

```
gapminder %>%
  filter(continent == "Europe") %>%
  group_by(country, year) %>%
  summarize(gdp_mean = mean(gdpPercap),
            lifeExp_mean = mean(lifeExp)) %>%
  ggplot(aes(x = year, y = lifeExp_mean)) +
  geom_point() +
    labs(x = "Year",
       y = "Life expectancy",
       title = "Life expectancy in Europe") +
  facet_wrap(~country)
```



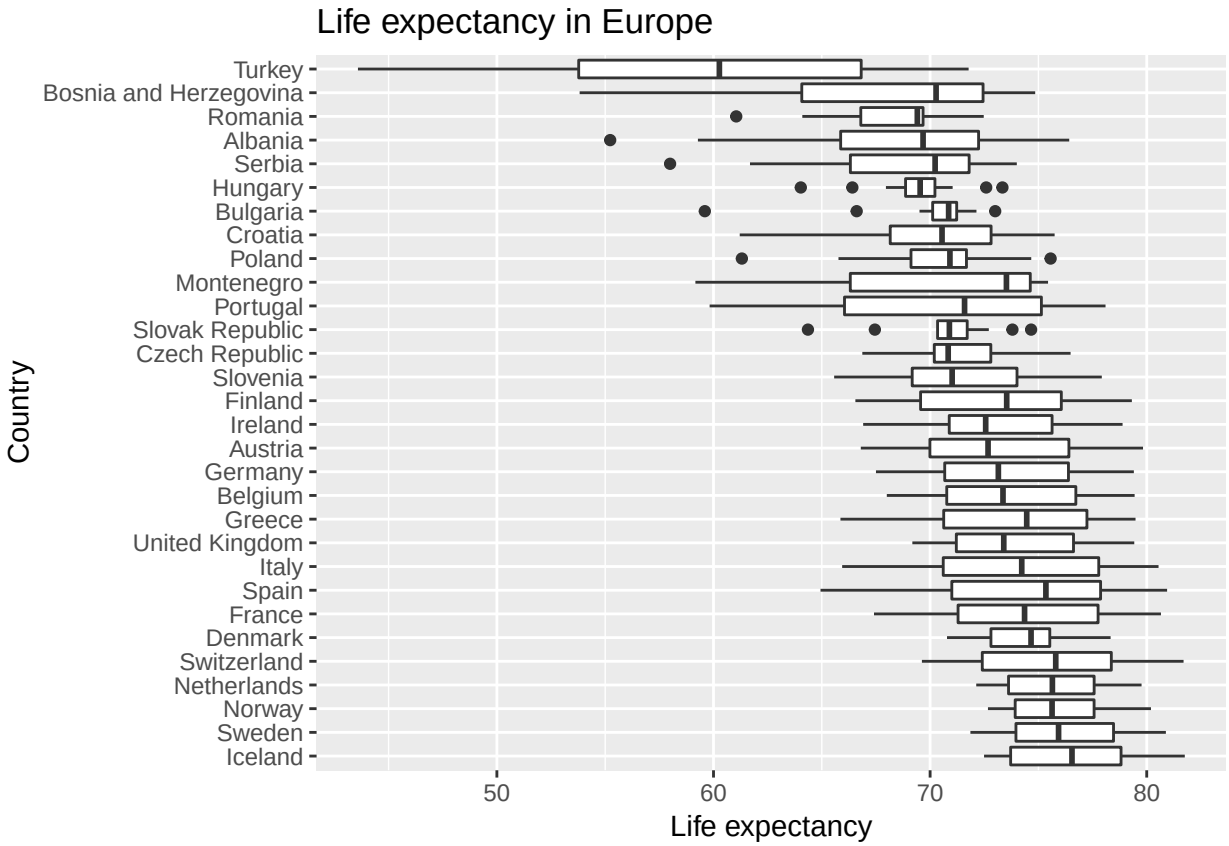

```
gapminder %>%
  filter(continent == "Europe") %>%
  group_by(country, year) %>%
  summarize(gdp_mean = mean(gdpPercap),
            lifeExp_mean = mean(lifeExp)) %>%
  ggplot(aes(x = country, y = lifeExp_mean)) +
  geom_boxplot() +
    labs(x = "Country",
       y = "Life expectancy",
       title = "Life expectancy in Europe") +
  coord_flip()
```




```
gapminder %>%
  filter(continent == "Europe") %>%
  group_by(country, year) %>%
  summarize(gdp_mean = mean(gdpPercap),
            lifeExp_mean = mean(lifeExp)) %>%
  ggplot(aes(x = reorder(country, lifeExp_mean), y = lifeExp_mean)) +
  geom_boxplot() +
    labs(x = "Country",
       y = "Life expectancy",
       title = "Life expectancy in Europe") +
  coord_flip()
```

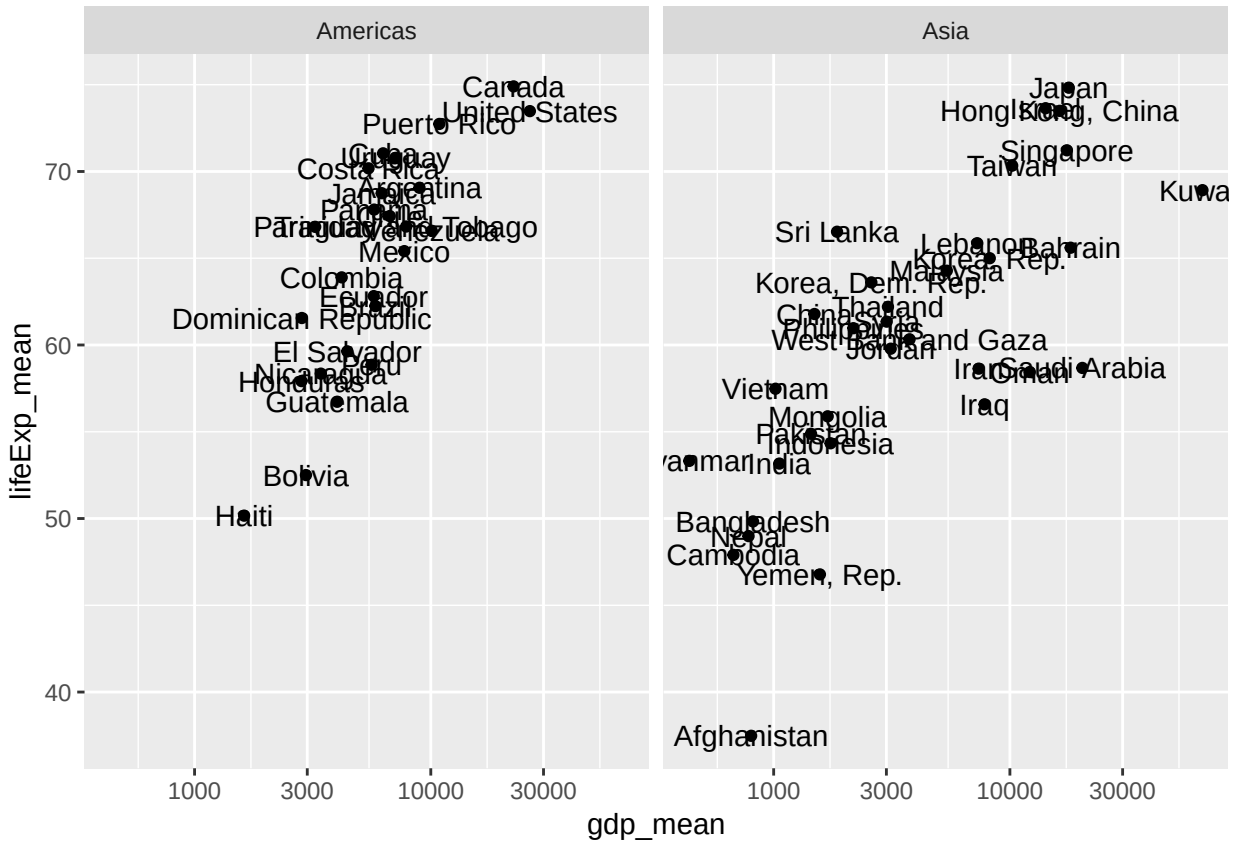



```
gapminder %>%
  filter(continent == "Europe") %>%
  group_by(country, year) %>%
  summarize(gdp_mean = mean(gdpPercap),
            lifeExp_mean = mean(lifeExp)) %>%
  ggplot(aes(x = reorder(country, -lifeExp_mean), y = lifeExp_mean)) +
  geom_boxplot() +
    labs(x = "Country",
       y = "Life expectancy",
       title = "Life expectancy in Europe") +
  coord_flip()
```



Plotting text

```
gapminder %>%
  filter(continent == "Asia" | continent == "Americas") %>%
  group_by(continent, country) %>%
  summarize(gdp_mean = mean(gdpPercap),
            lifeExp_mean = mean(lifeExp)) %>%
  ggplot(aes(x = gdp_mean, y = lifeExp_mean)) +
  geom_point() +
  geom_text(aes(label = country)) +
  scale_x_log10() +
  facet_grid(~continent)
```



Ploting models

In plotting models, we use David Robinson's broom package in R extensively. The idea is to transform model outputs (i.e., predictions and estimations) into tidy objects so that we can combine, separate, and visualize these elements easily.

```
# regression model
out <-lm(formula = lifeExp ~ gdpPercap + pop + continent,
         data = gapminder)
```

Tidy is a method in broom package. It "constructs a dataframe that summarizes the model's statistical findings". As the description states, tidy is a function that can be used generally for various models. For instance, a tidy can extract following information from a regression model.

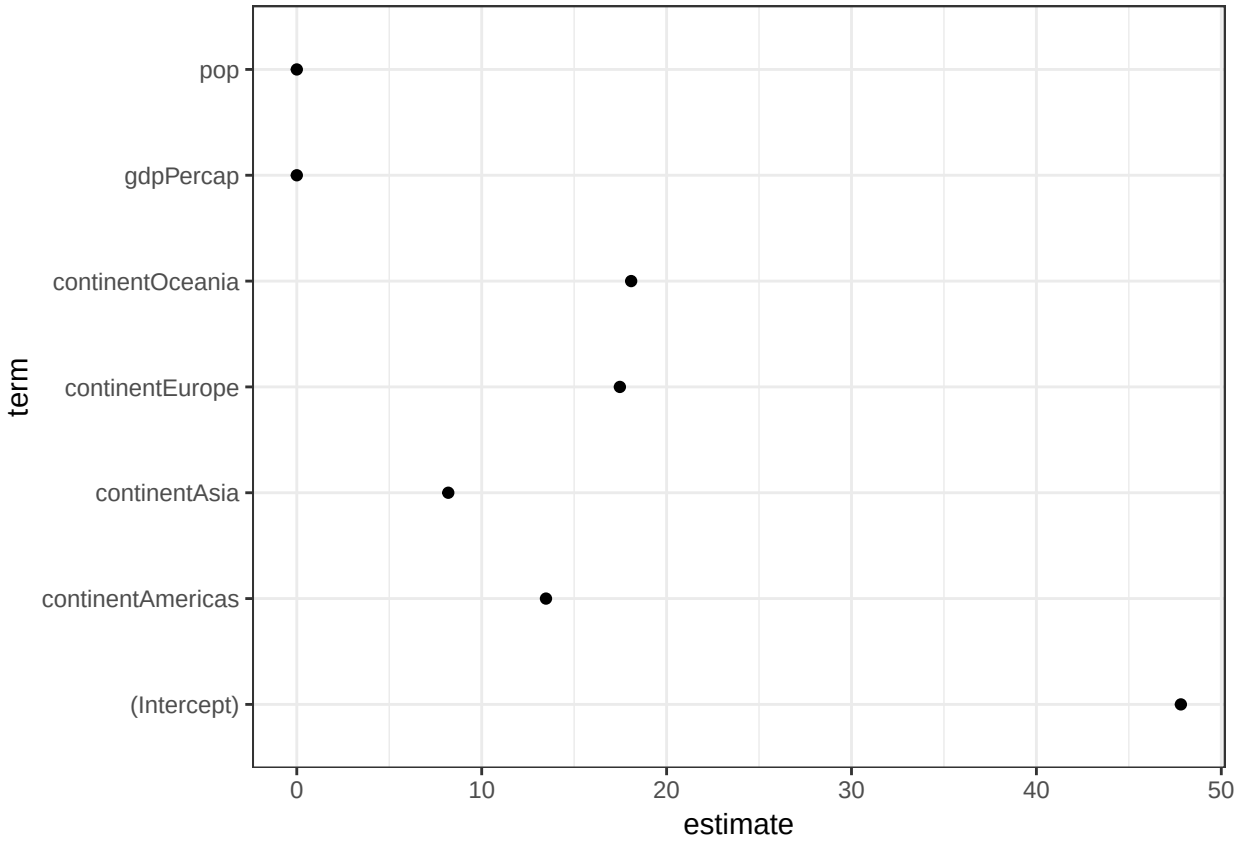
- Term: a term being estimated
- p.value
- statistic: a test statistic used to compute p-value
- estimate
- conf.low: the low end of a confidence interval
- conf.high: the high end of a confidence interval
- df: degrees of freedom

Challege

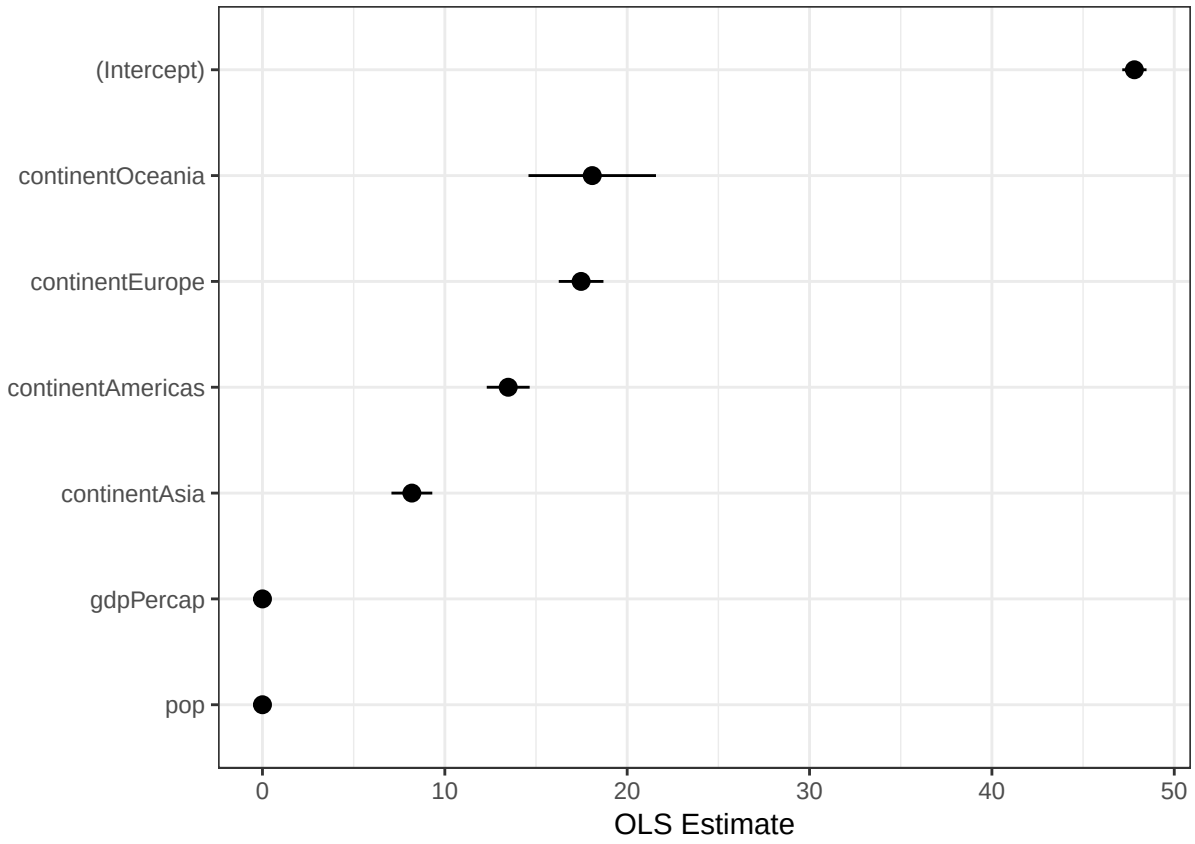
Try glance(out), what did you get from these commands? If you're curious, you can try ?glance.

```
# estimates
out_comp <- tidy(out)
```

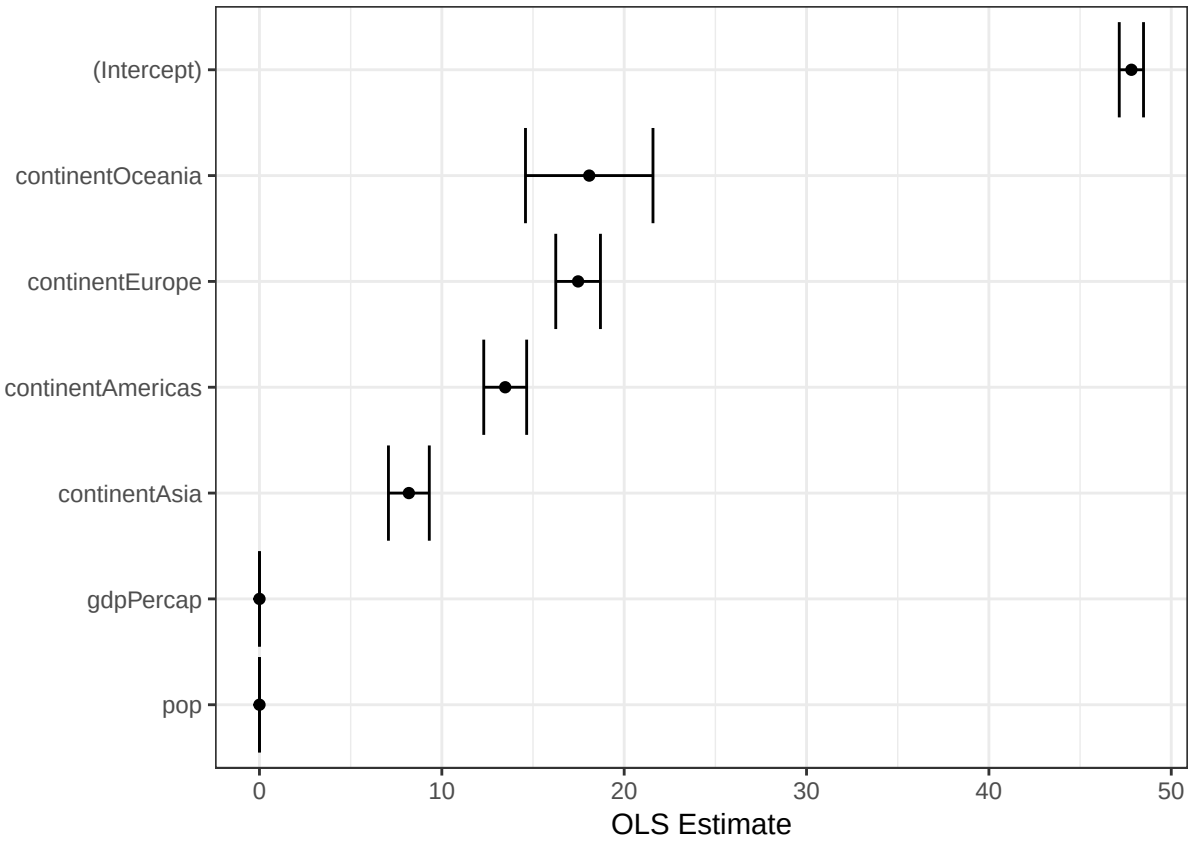
```r
p <- out_comp %>%
  ggplot(aes(x = term, y = estimate))

p + geom_point() +
  coord_flip() +
  theme_bw()
```




```r
# plus confidence intervals
out_conf <- tidy(out, conf.int = TRUE)

# plotting coefficients using ggplot2 (pointrange)
out_conf %>%
  ggplot(aes(x = reorder(term, estimate), y = estimate, ymin = conf.low, ymax = conf.high)) +
  geom_pointrange() + coord_flip() + labs(x = "", y = "OLS Estimate") +
  theme_bw()
```




```
# another way to do it (errorbar)
out_conf %>%
  ggplot(aes(x = estimate, y = reorder(term, estimate))) +
  geom_point() +
  geom_errorbarh(aes(xmin = conf.low, xmax = conf.high)) +
  labs(y = "", x = "OLS Estimate") +
  theme_bw()
```



Challenge

1. If we only want to visualize a certain subset of variables, let's say gdpPercap and pop, how can you do that? Also, gdpPercap might be not very informative. What's the best way to change the value name?
2. broom is a great package for running split-and-combine regressions. See the following example and write down your workflow for visualize it.

```
gapminder %>%
  group_by(continent) %>%
    do(tidy(lm(gdpPercap ~ lifeExp, data = .), conf.int = TRUE))
```

```
## # A tibble: 10 x 8
## # Groups:   continent [5]
##    continent term  estimate std.error statistic  p.value conf.low conf.high
##    <fct>     <chr>    <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
##  1 Africa    (Int~   -4234.     557.      -7.59 1.14e-13   -5329.    -3139.
##  2 Africa    life~     132.      11.2     11.7  7.60e-29     110.      154.
##  3 Americas  (Int~  -17577.    2149.      -8.18 8.35e-15  -21806.   -13348.
##  4 Americas  life~     382.      32.9     11.6  5.45e-26     317.      447.
##  5 Asia      (Int~  -19264.    3374.      -5.71 2.24e- 8  -25897.   -12630.
##  6 Asia      life~     452.      55.1      8.21 3.29e-15     344.      561.
##  7 Europe    (Int~  -82198.    4100.     -20.0  1.77e-60  -90261.   -74135.
##  8 Europe    life~    1344.      56.9     23.6  4.05e-75    1233.     1456.
##  9 Oceania   (Int~ -100481.    7757.     -13.0  9.03e-12 -116568.   -84394.
## 10 Oceania   life~    1602.     104.      15.4  2.99e-13    1386.     1819.
```