

# Data visualization with ggplot2

*Jae Yeon Kim*

*24 December, 2018*

## Motivation

- The following material is adapted from Kieran Healy's wonderful book (2018) on data visualization.
- Why should we care?
- Sometimes, pictures are better tools than words in 1) exploring, 2) understanding, and 3) explaining data.

## Anscombe's quartet

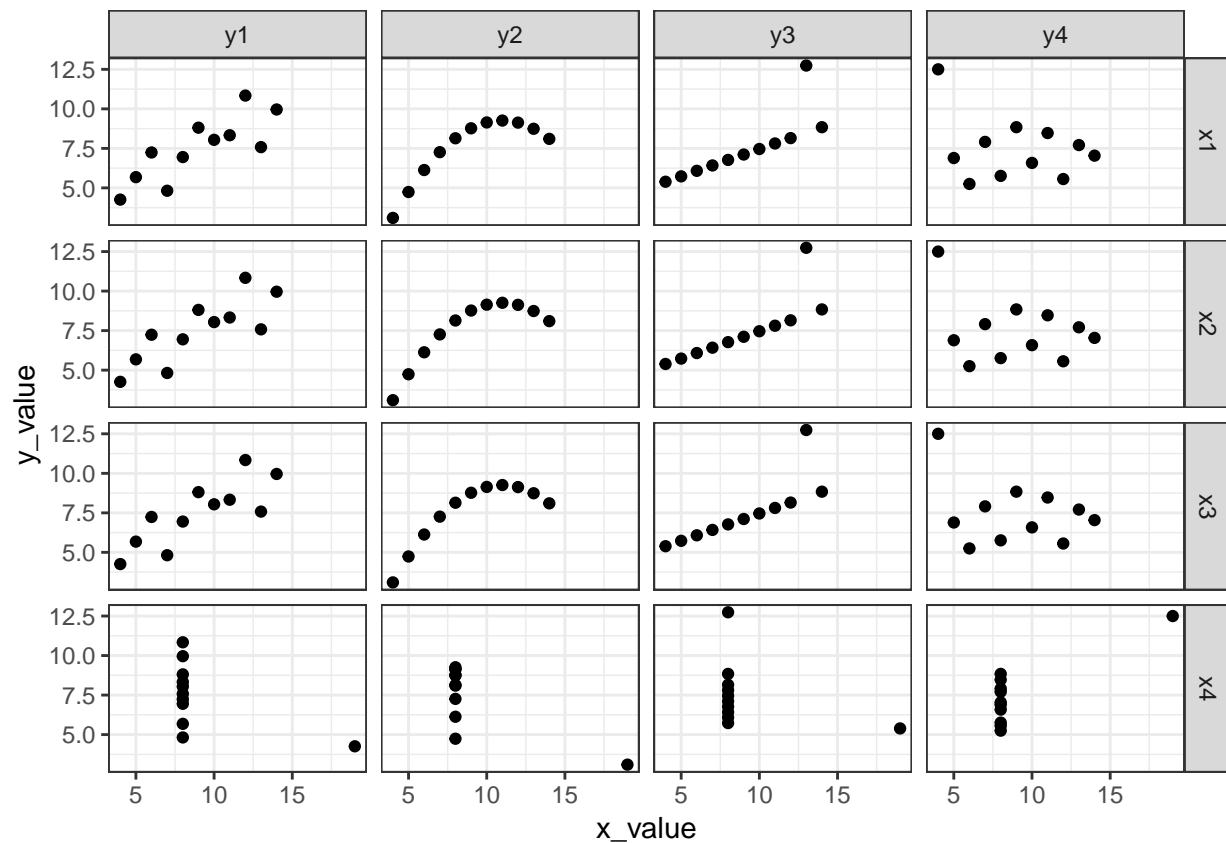
```
# data
anscombe

##      x1 x2 x3 x4      y1      y2      y3      y4
## 1   10 10 10  8   8.04 9.14   7.46  6.58
## 2    8  8  8  8   6.95 8.14   6.77  5.76
## 3   13 13 13  8   7.58 8.74  12.74  7.71
## 4    9  9  9  8   8.81 8.77   7.11  8.84
## 5   11 11 11  8   8.33 9.26   7.81  8.47
## 6   14 14 14  8   9.96 8.10   8.84  7.04
## 7    6  6  6  8   7.24 6.13   6.08  5.25
## 8    4  4  4 19   4.26 3.10   5.39 12.50
## 9   12 12 12  8  10.84 9.13   8.15  5.56
## 10   7  7  7  8   4.82 7.26   6.42  7.91
## 11   5  5  5  8   5.68 4.74   5.73  6.89

# correlation
cor(anscombe)[c(1:4),c(5:8)]

##              y1              y2              y3              y4
## x1  0.8164205  0.8162365  0.8162867 -0.3140467
## x2  0.8164205  0.8162365  0.8162867 -0.3140467
## x3  0.8164205  0.8162365  0.8162867 -0.3140467
## x4 -0.5290927 -0.7184365 -0.3446610  0.8165214

# plot
anscombe %>%
  gather(x_name, x_value, x1:x4) %>%
  gather(y_name, y_value, y1:y4) %>%
  ggplot(aes(x = x_value, y = y_value)) +
    geom_point() +
    facet_grid(x_name ~ y_name) +
    theme_bw()
```



## ggplot2 basics

- Workflow:
  1. Tidy data
  2. Mapping
  3. Geom
  4. Cor\_ordinates and scales
  5. Labels and guides
  6. Themes
  7. Save files

## Tidy data

- We covered tidy data in the previous sessions.

```
## # 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
```

What is the difference between `typeof` and `class`?

```
typeof(gapminder)
```

```
## [1] "list"
```

```
class(gapminder)
```

```
## [1] "tbl_df"      "tbl"        "data.frame"
```

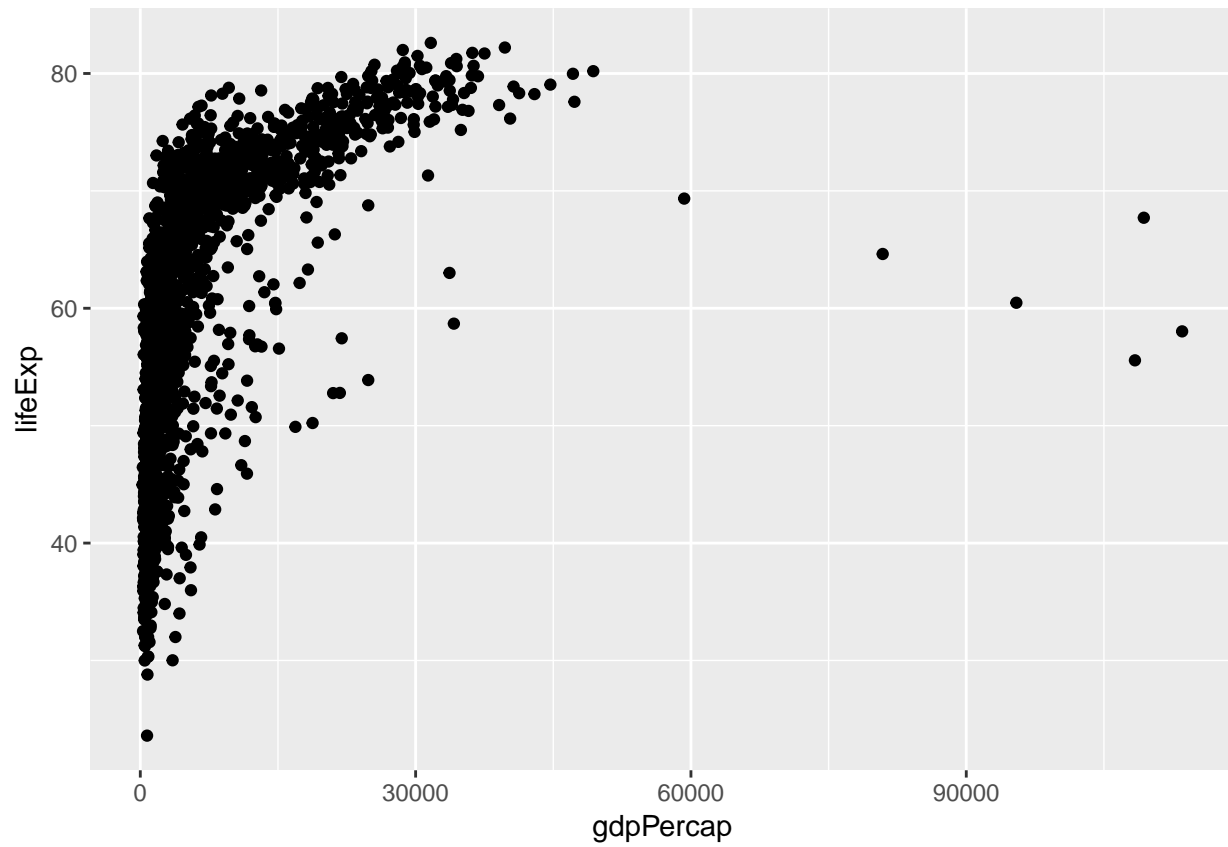
## Mapping and Geom

- `ggplot` tells what is your data
- `aes` (aesthetic mappings or aesthetics) tells what is your variables of interests in the data
- `geom_` tles the type of plot you are going to use

Basic `aes` (`x` , `y`)

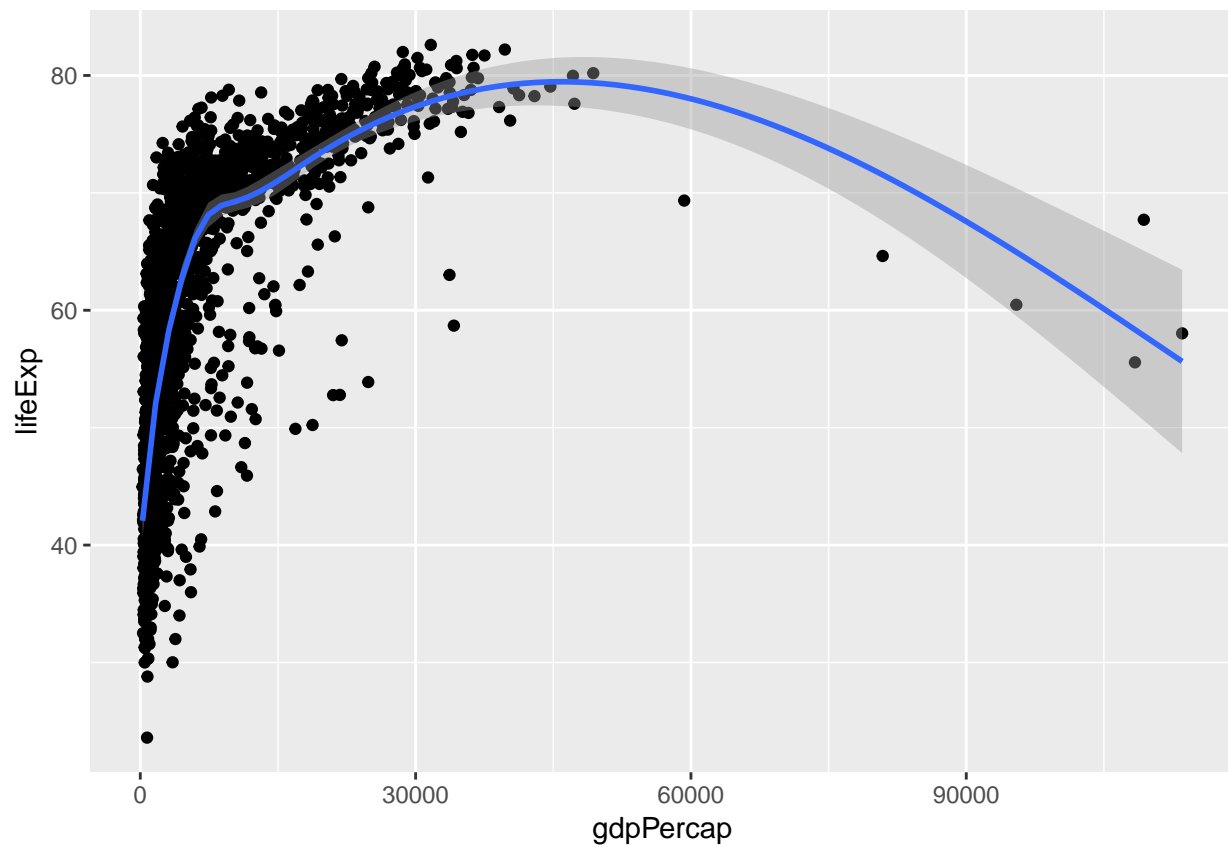
```
p <- ggplot(data = gapminder,  
            mapping = aes(x = gdpPercap, y = lifeExp))
```

```
p + geom_point()
```



```
p + geom_point() + geom_smooth()
```

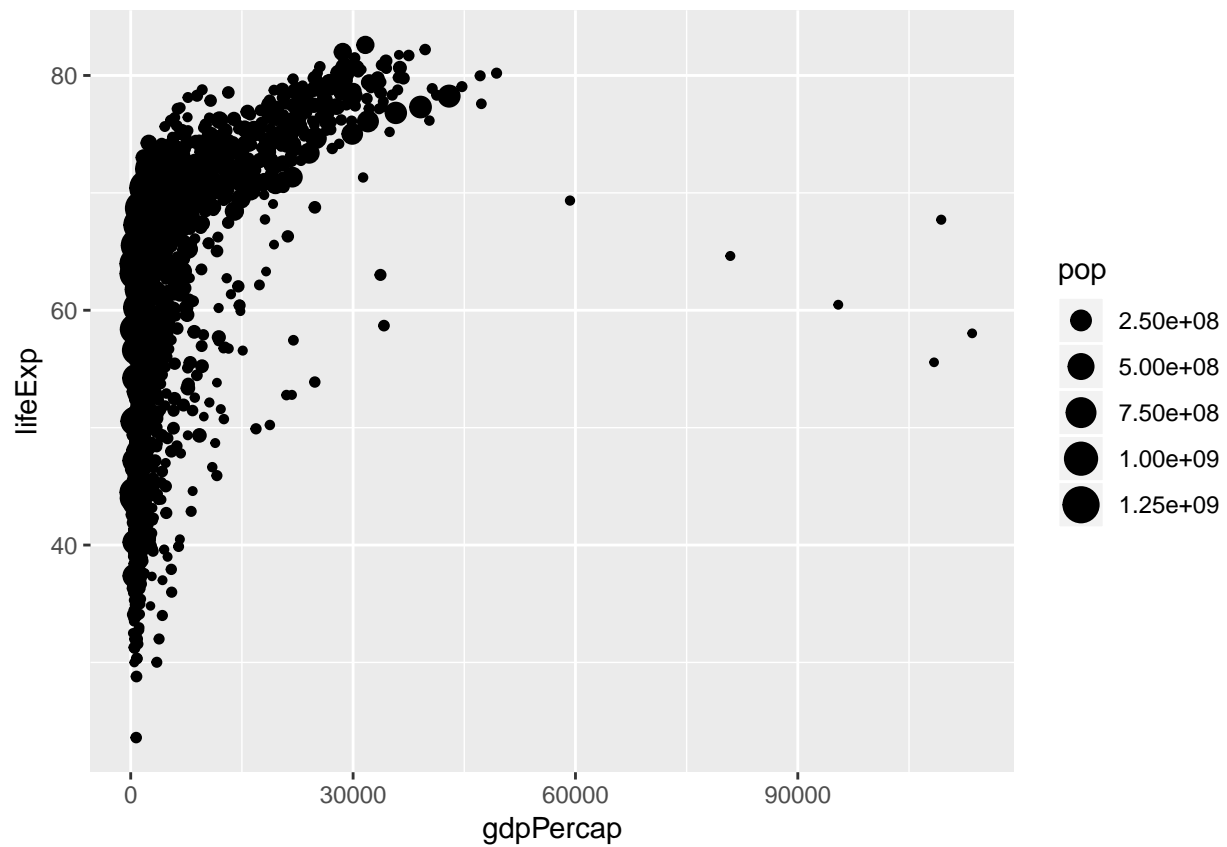
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



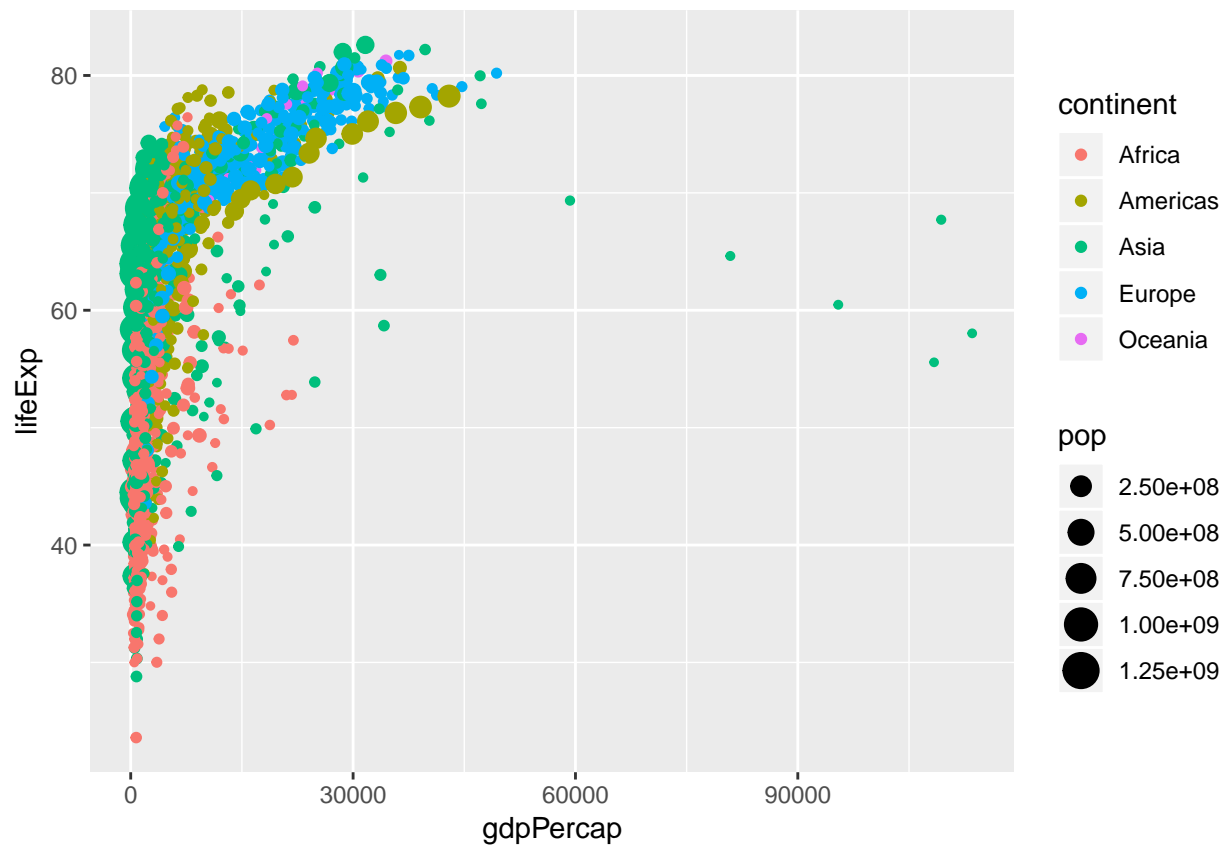
### Advanced aes (size, color)

- There's also fill argument (mostly used in `geom_bar()`).
- The property `size/color/fill` represents...

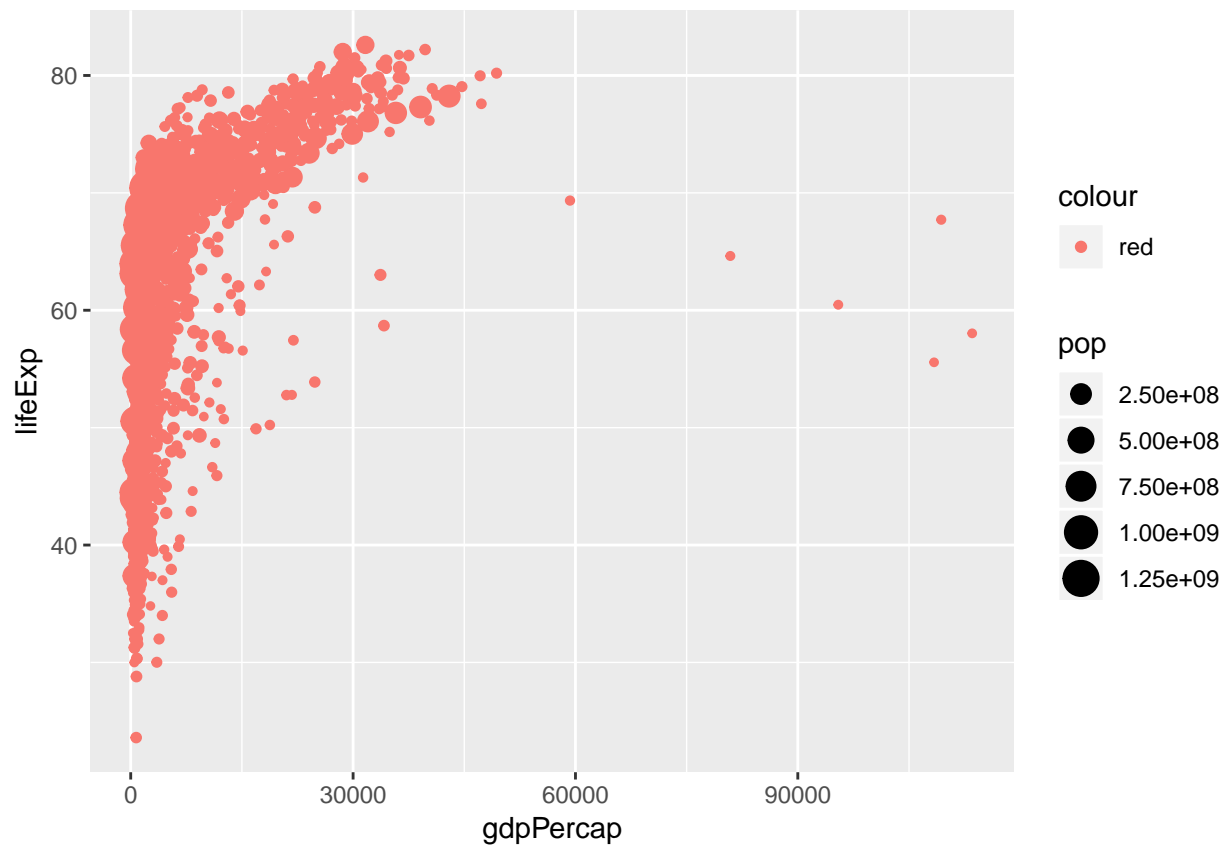
```
ggplot(data = gapminder,  
       mapping = aes(x = gdpPercap, y = lifeExp,  
                     size = pop)) +  
geom_point()
```



```
ggplot(data = gapminder,  
       mapping = aes(x = gdpPercap, y = lifeExp,  
                     size = pop,  
                     color = continent)) +  
geom_point()
```

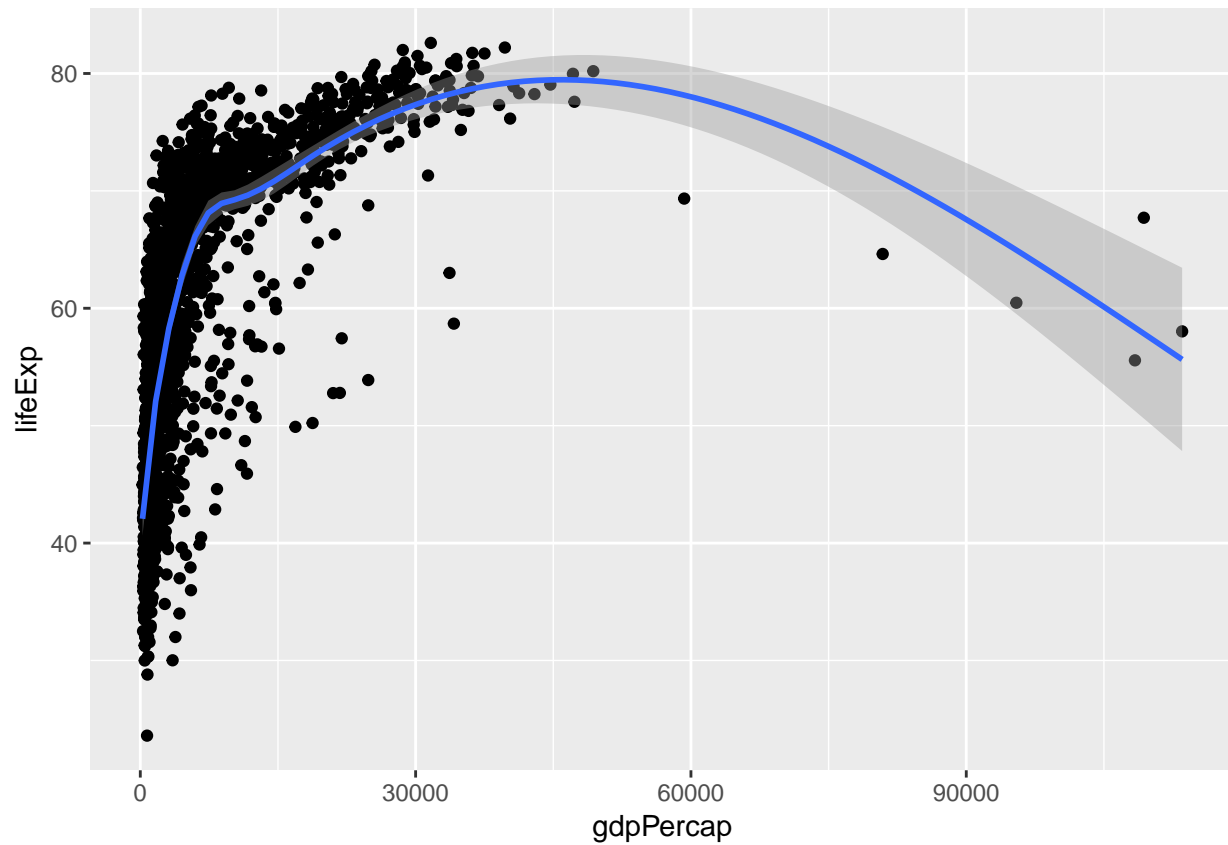


```
# try red instead of "red"  
ggplot(data = gapminder,  
        mapping = aes(x = gdpPercap, y = lifeExp,  
                      size = pop,  
                      color = "red")) +  
  geom_point()
```



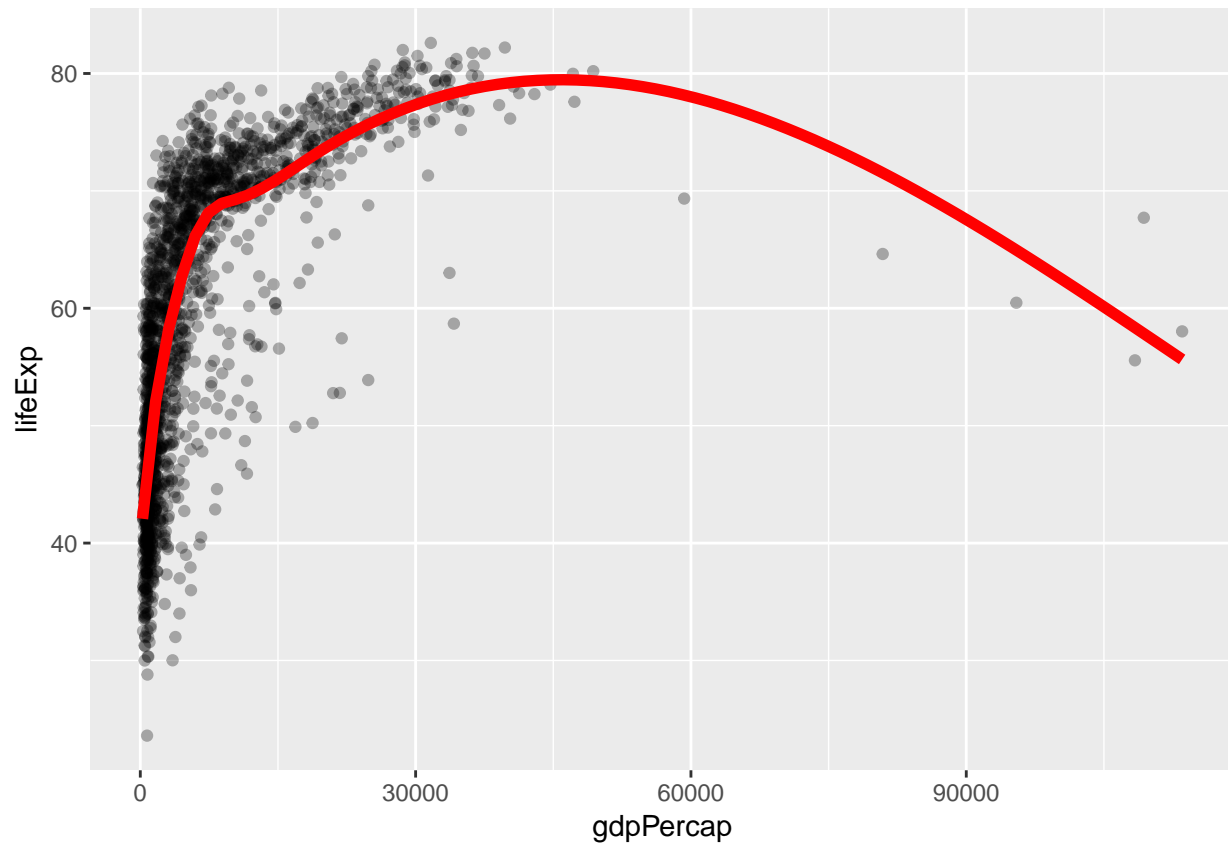
```
p + geom_point() +  
  geom_smooth()  
  
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```





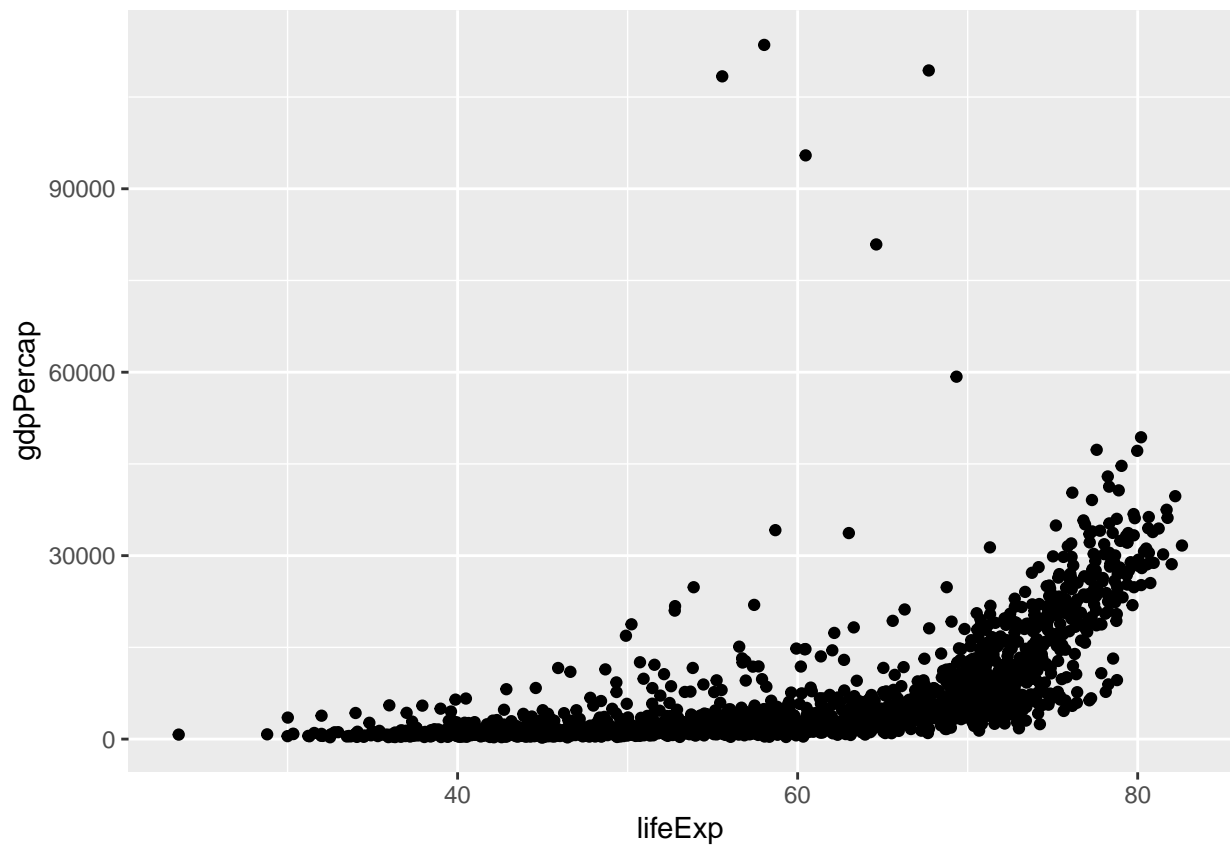
```
p + geom_point(alpha = 0.3) + # alpha controls transparency
  geom_smooth(color = "red", se = FALSE, size = 2)

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

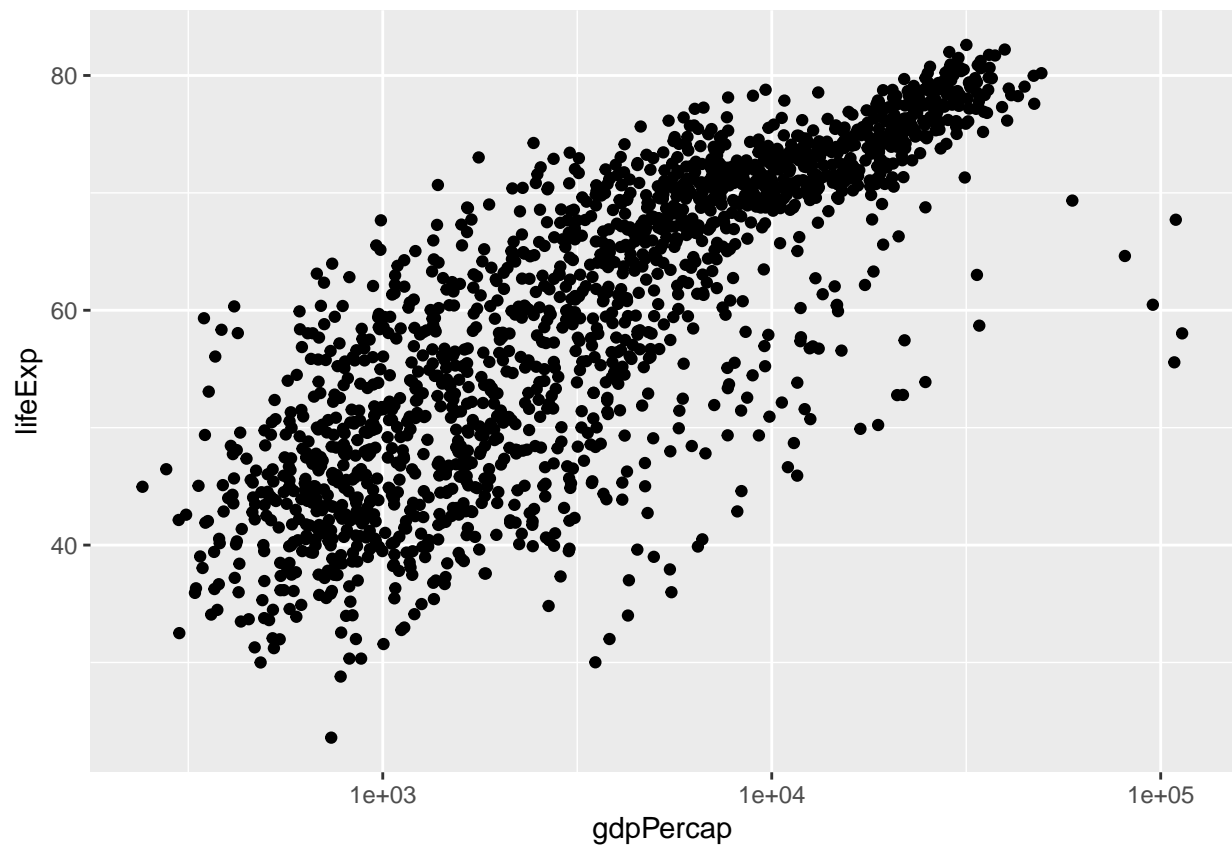


Co-ordinates and scales

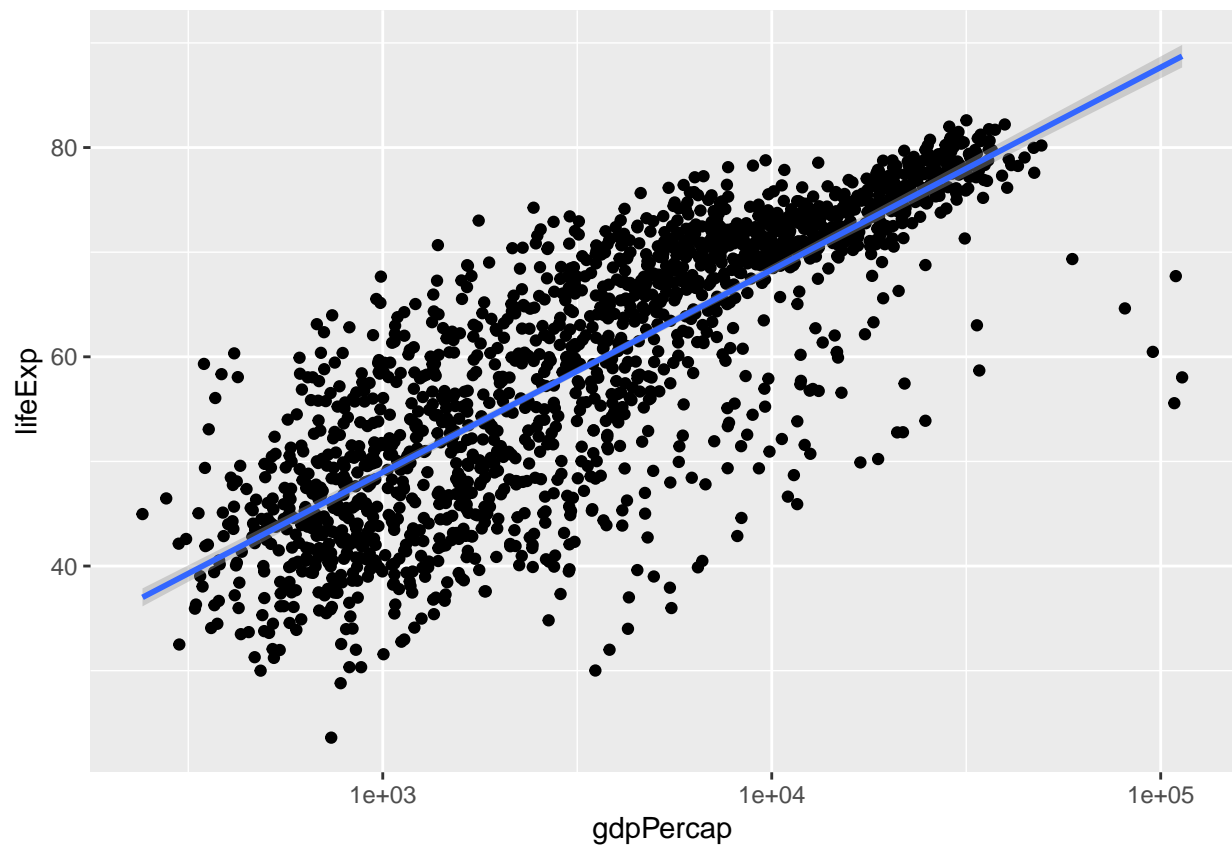
```
p + geom_point() +  
  coord_flip() # coord_type
```



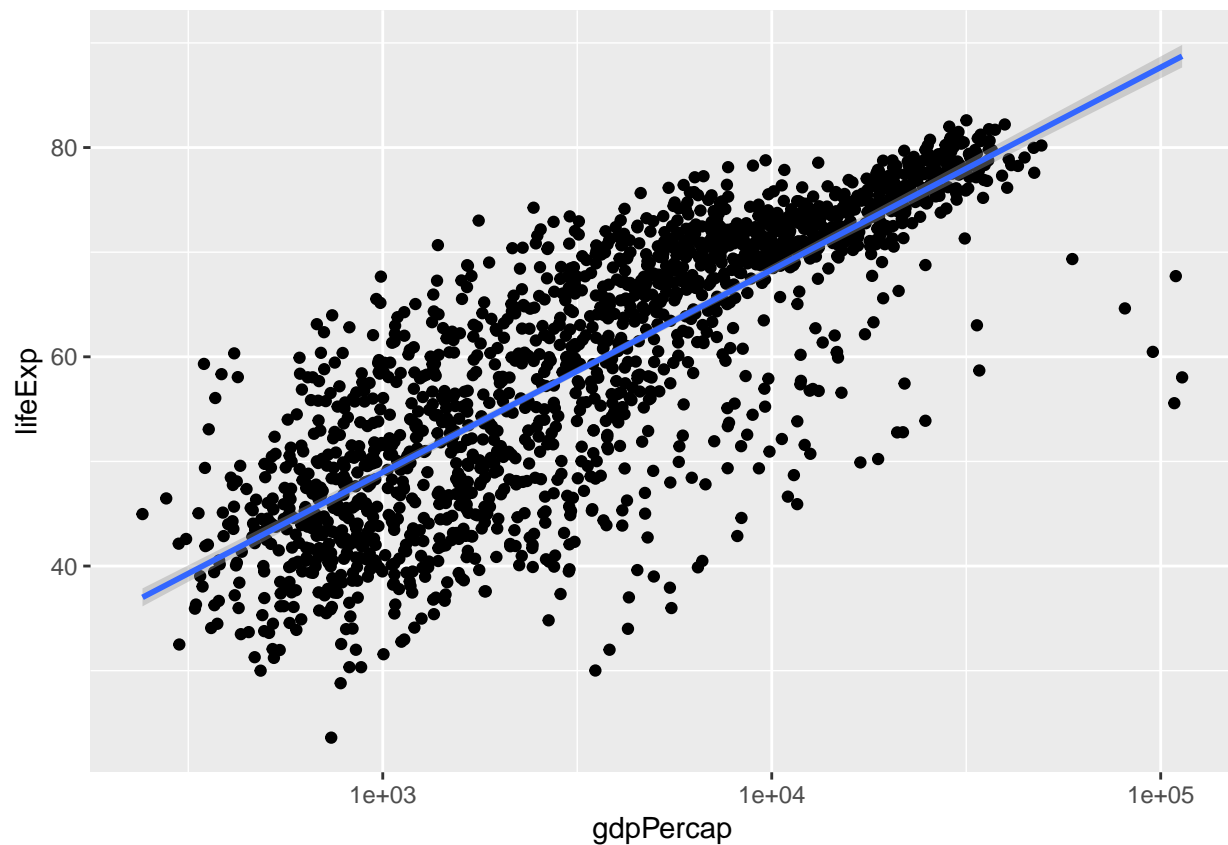
```
p + geom_point() +  
  scale_x_log10() # scale_mapping_type
```



```
p + geom_point() +  
  geom_smooth(method = "lm") +  
  scale_x_log10()
```



```
p + geom_point() +  
  geom_smooth(method = "lm") +  
  scale_x_log10()
```

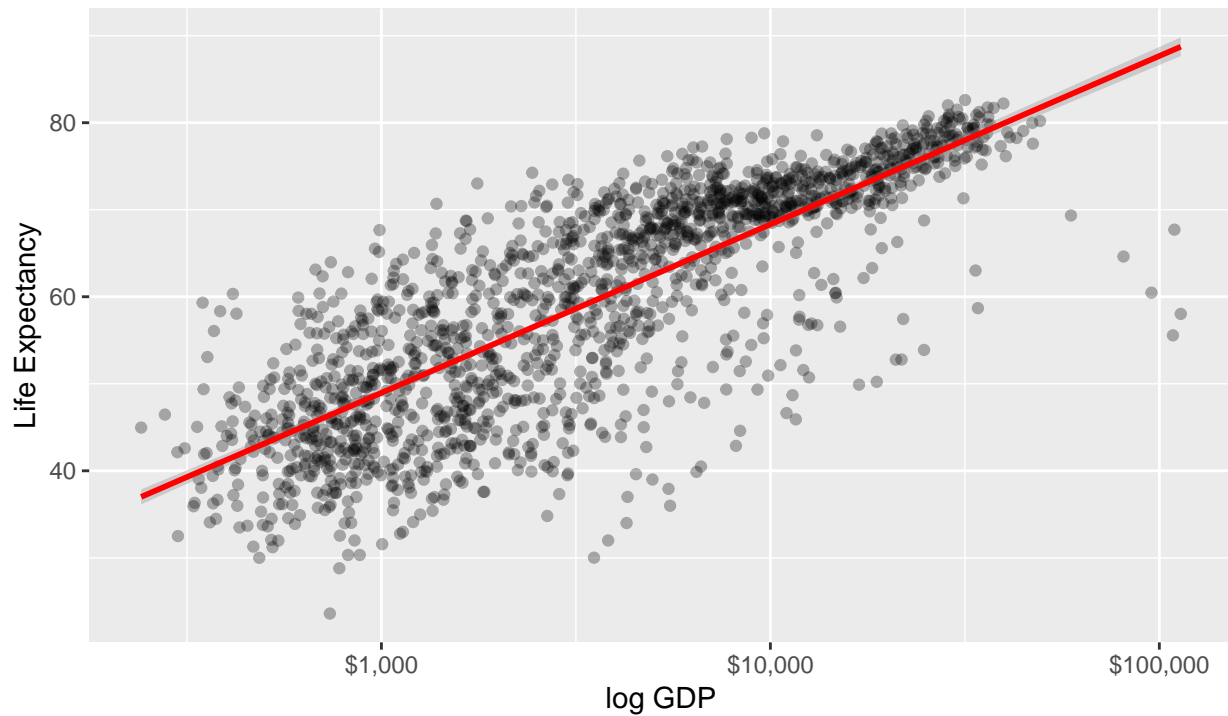


## Labels and guides

```
p + geom_point(alpha = 0.3) +  
  geom_smooth(method = "gam", color = "red") +  
  scale_x_log10(labels = scales::dollar) +  
  labs( x = "log GDP",  
        y = "Life Expectancy",  
        title = "A Gapminder Plot",  
        subtitle = "Data points are country-years",  
        caption = "Source: Gapminder")
```

## A Gapminder Plot

Data points are country-years

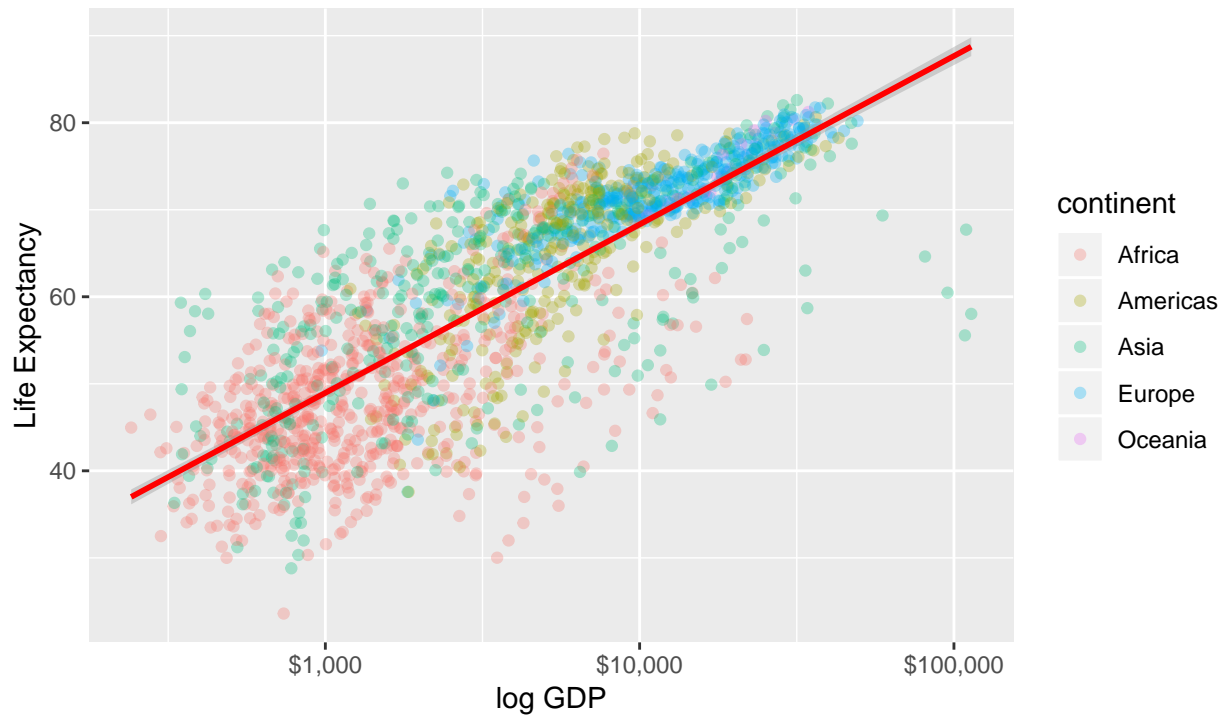


Source: Gapminder

```
ggplot(data = gapminder,
       mapping = aes(x = gdpPercap, y = lifeExp,
                     color = continent)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "gam", color = "red") +
  scale_x_log10(labels = scales::dollar) +
  labs(x = "log GDP",
       y = "Life Expectancy",
       title = "A Gapminder Plot",
       subtitle = "Data points are country-years",
       caption = "Source: Gapminder")
```

## A Gapminder Plot

Data points are country-years



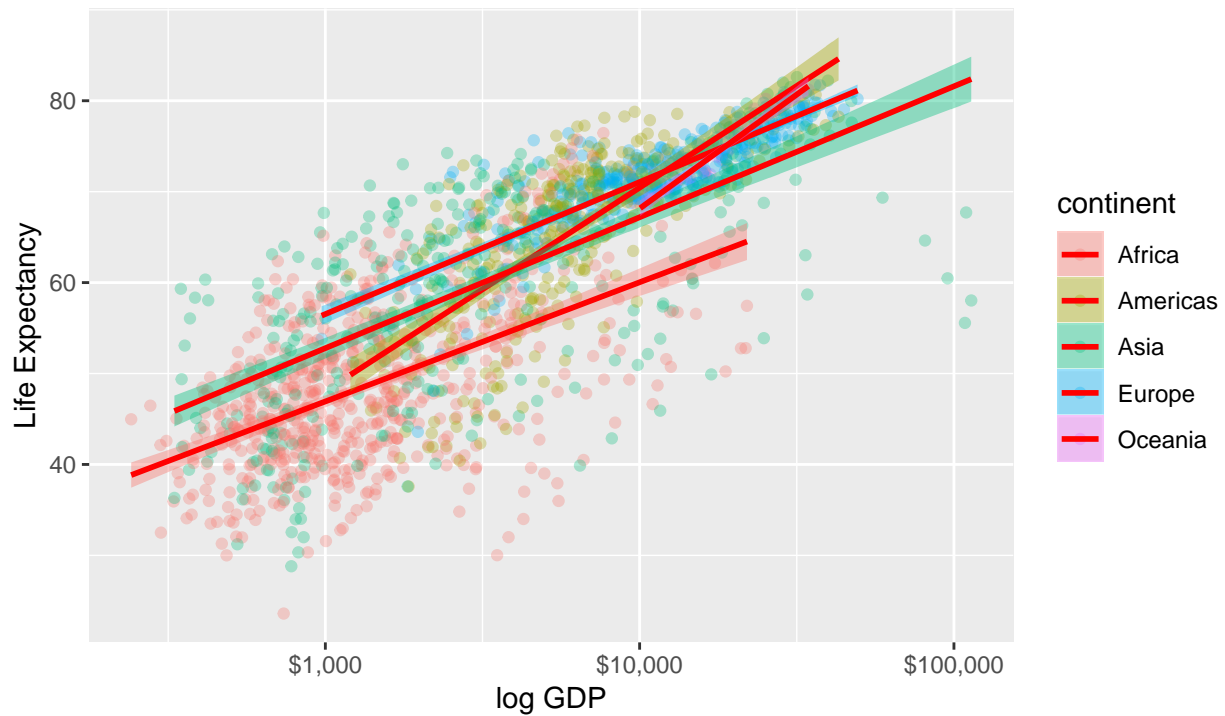
Source: Gapminder

```
ggplot(data = gapminder,
       mapping = aes(x = gdpPercap, y = lifeExp,
                     color = continent,
                     fill = continent)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "gam", color = "red") +
  scale_x_log10(labels = scales::dollar) +
  labs(x = "log GDP",
       y = "Life Expectancy",
       title = "A Gapminder Plot",
       subtitle = "Data points are country-years",
       caption = "Source: Gapminder")
```



## A Gapminder Plot

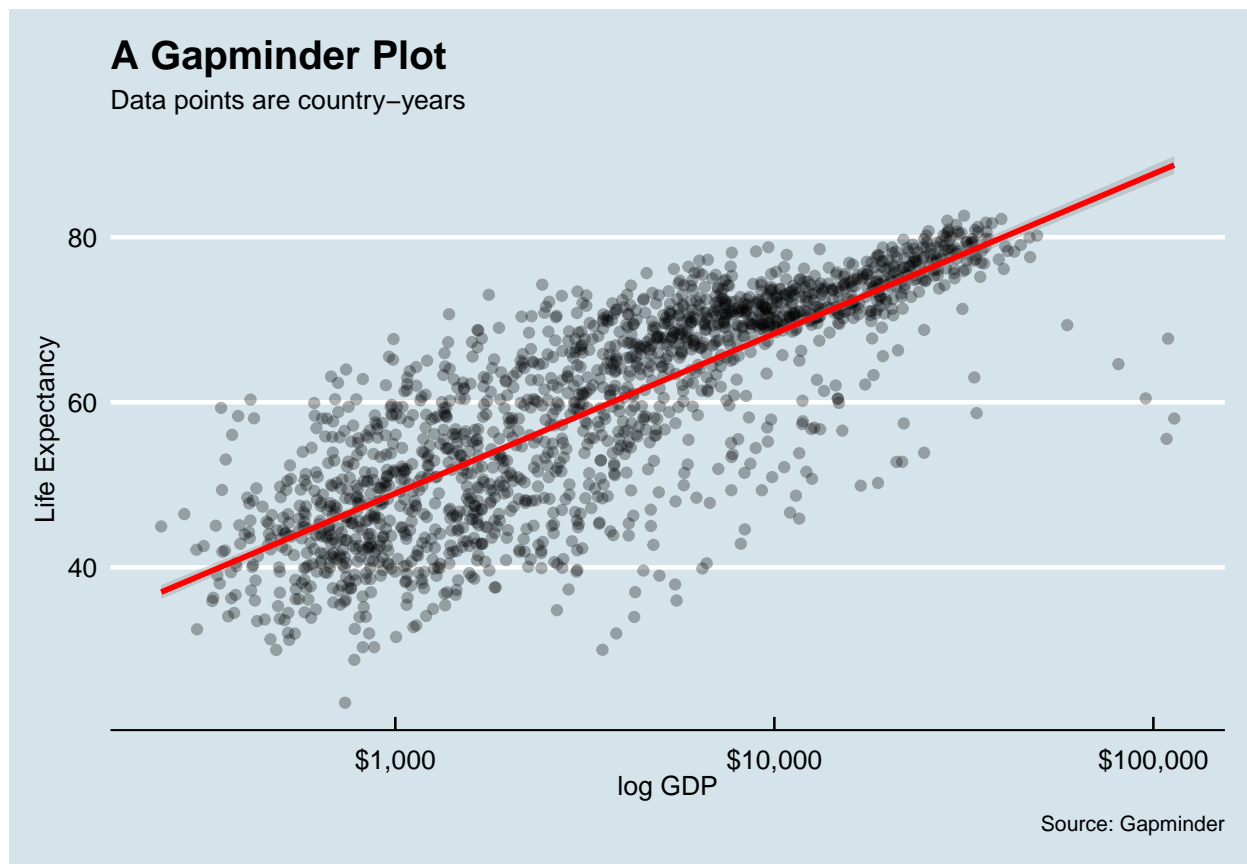
Data points are country-years



Source: Gapminder

### 6. Themes

```
p + geom_point(alpha = 0.3) +  
  geom_smooth(method = "gam", color = "red") +  
  scale_x_log10(labels = scales::dollar) +  
  labs( x = "log GDP",  
        y = "Life Expectancy",  
        title = "A Gapminder Plot",  
        subtitle = "Data points are country-years",  
        caption = "Source: Gapminder") +  
  theme_economist()
```



## Save files

- I highly recommend to save your file in a subdirectory named output or figures or something like that.

```
figure_example <- p + geom_point(alpha = 0.3) +
  geom_smooth(method = "gam", color = "red") +
  scale_x_log10(labels = scales::dollar) +
  labs(x = "log GDP",
       y = "Life Expectancy",
       title = "A Gapminder Plot",
       subtitle = "Data points are country-years",
       caption = "Source: Gapminder") +
  theme_economist()

ggsave(filename = "figure_example.png", plot = figure_example)
```

## Saving 6.5 x 4.5 in image

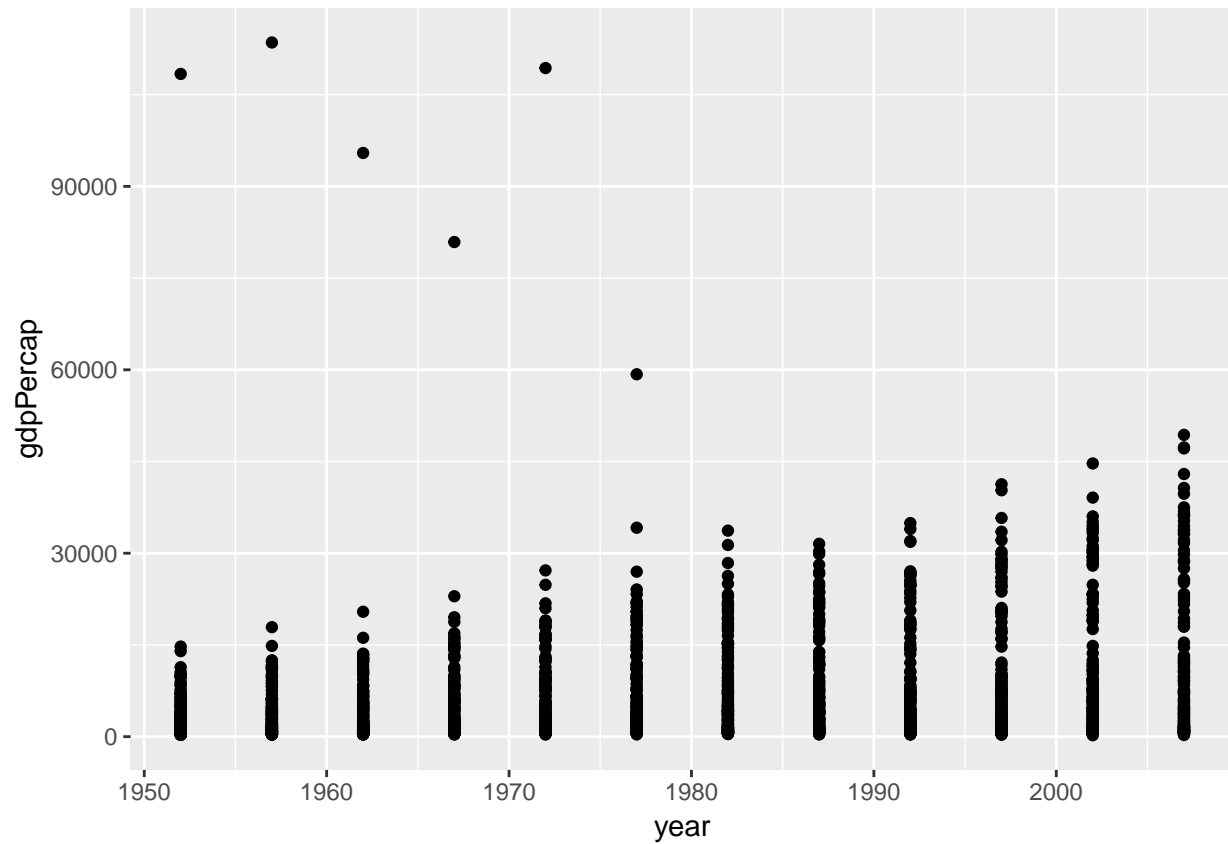
```
ggsave(filename = "figure_example_modified.png", plot = figure_example,
       height = 8,
       width = 10,
       units = "in")
```

## ggplot2 intermediates

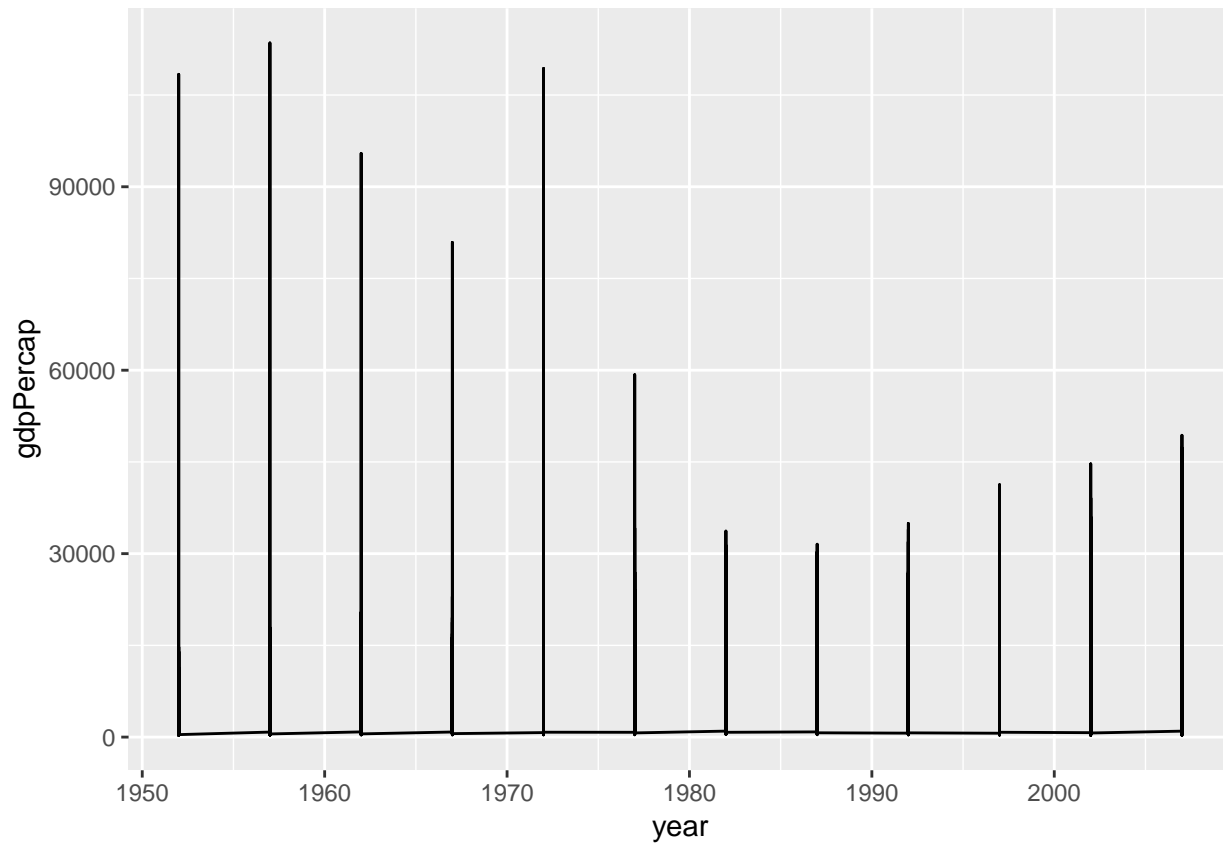
### Grouping and facetting

- Can you guess what's wrong?

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



```
p + geom_line()
```



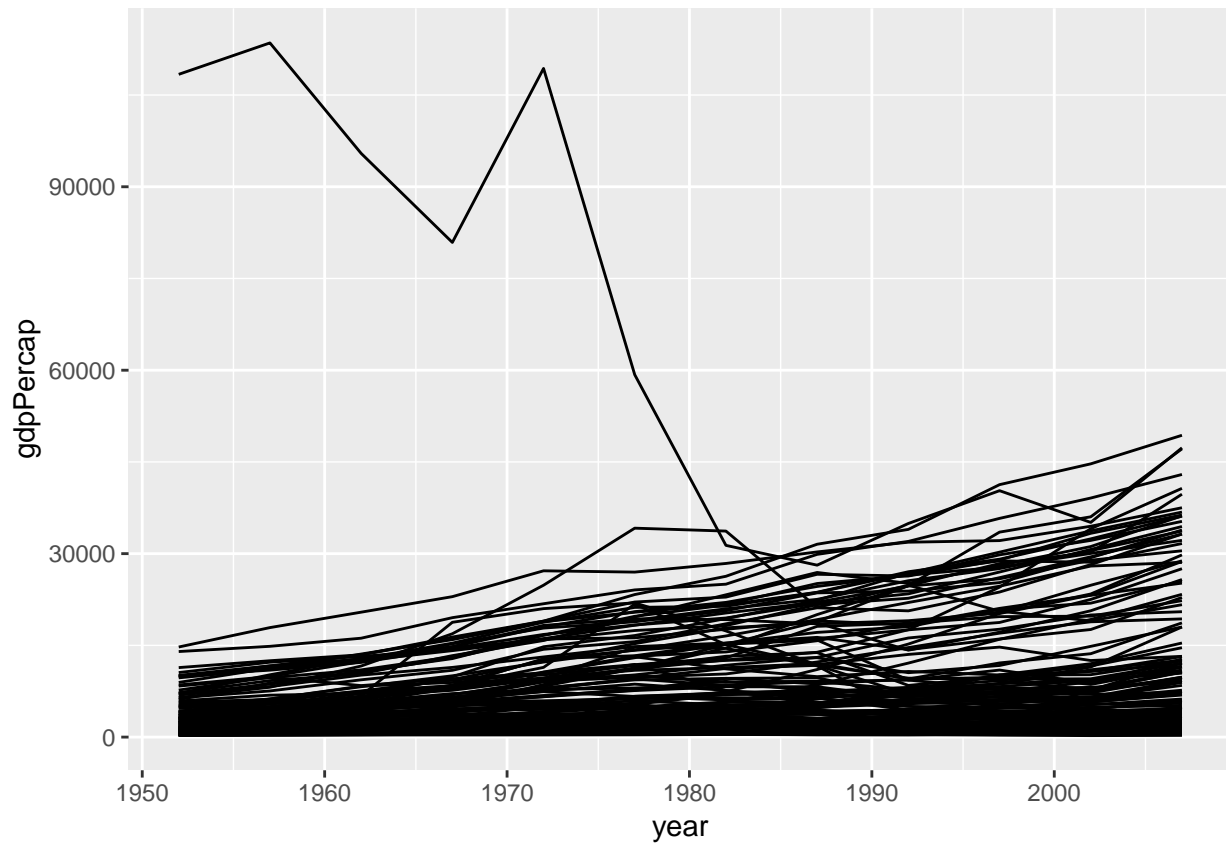
```
gapminder
```

```
## # 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
```

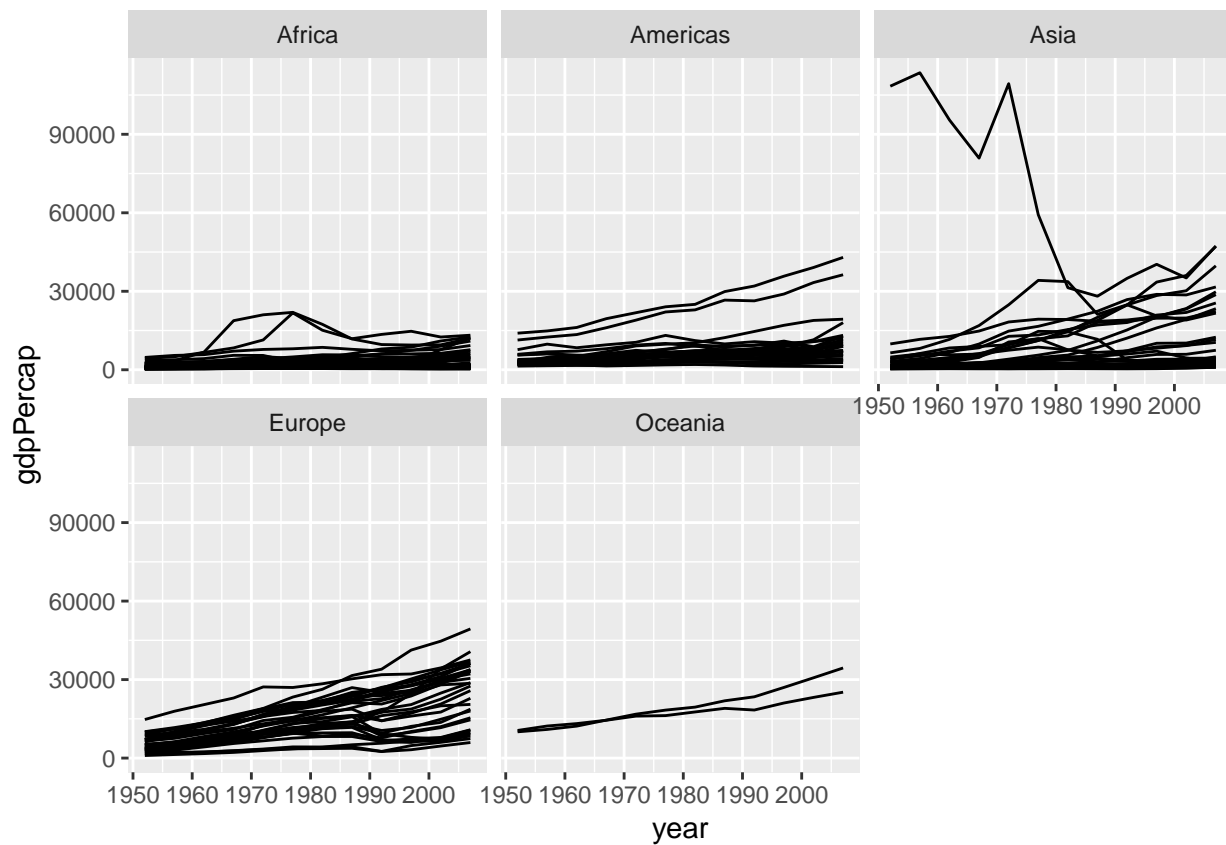
- Use grouping and facetting to clarify

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

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

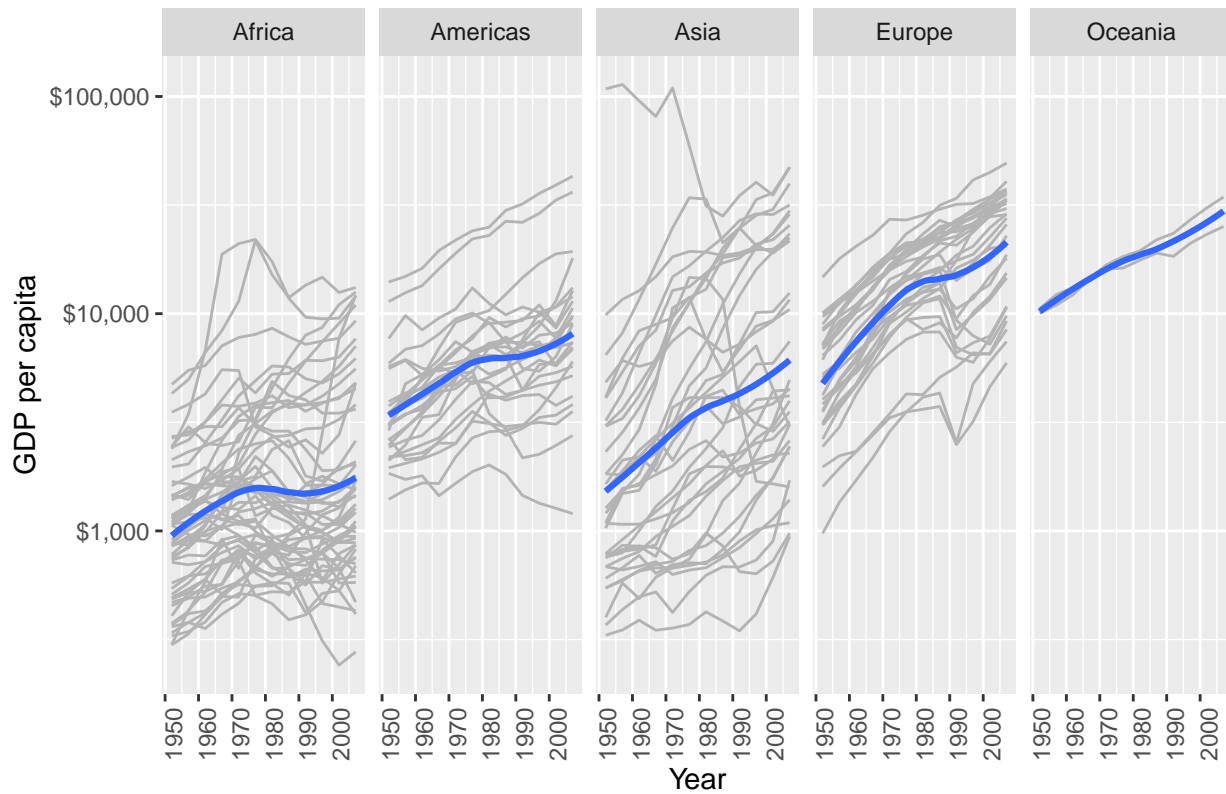


```
p + geom_line(aes(group = country)) + facet_wrap(~continent) # facetting
```



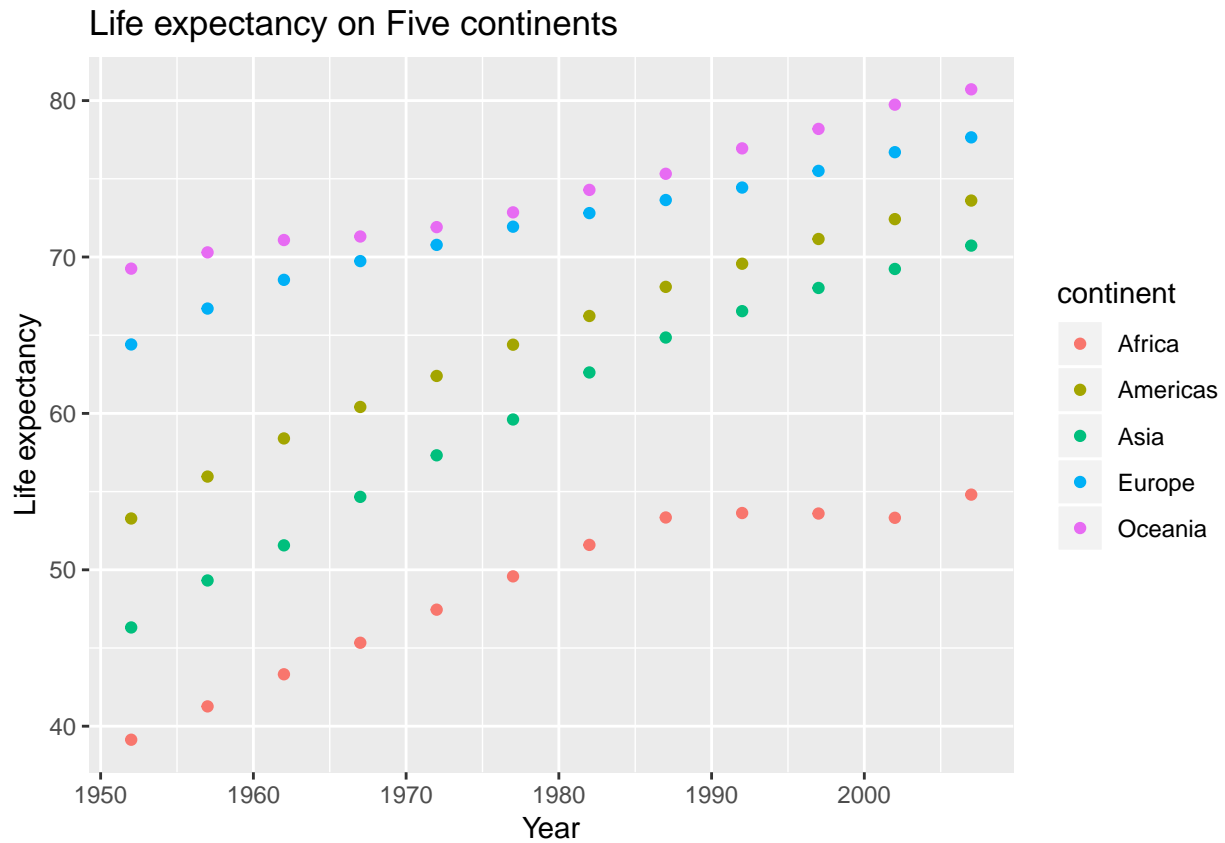
```
p + geom_line(aes(group = country), color = "gray70") +
  geom_smooth(size = 1.1, method = "loess", se = FALSE) +
  scale_y_log10(labels = scales::dollar) +
  facet_wrap(~continent, ncol = 5) + # for single categorical variable; for multiple categorical variab
  labs(x = "Year",
       y = "GDP per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

## GDP per capita on Five continents



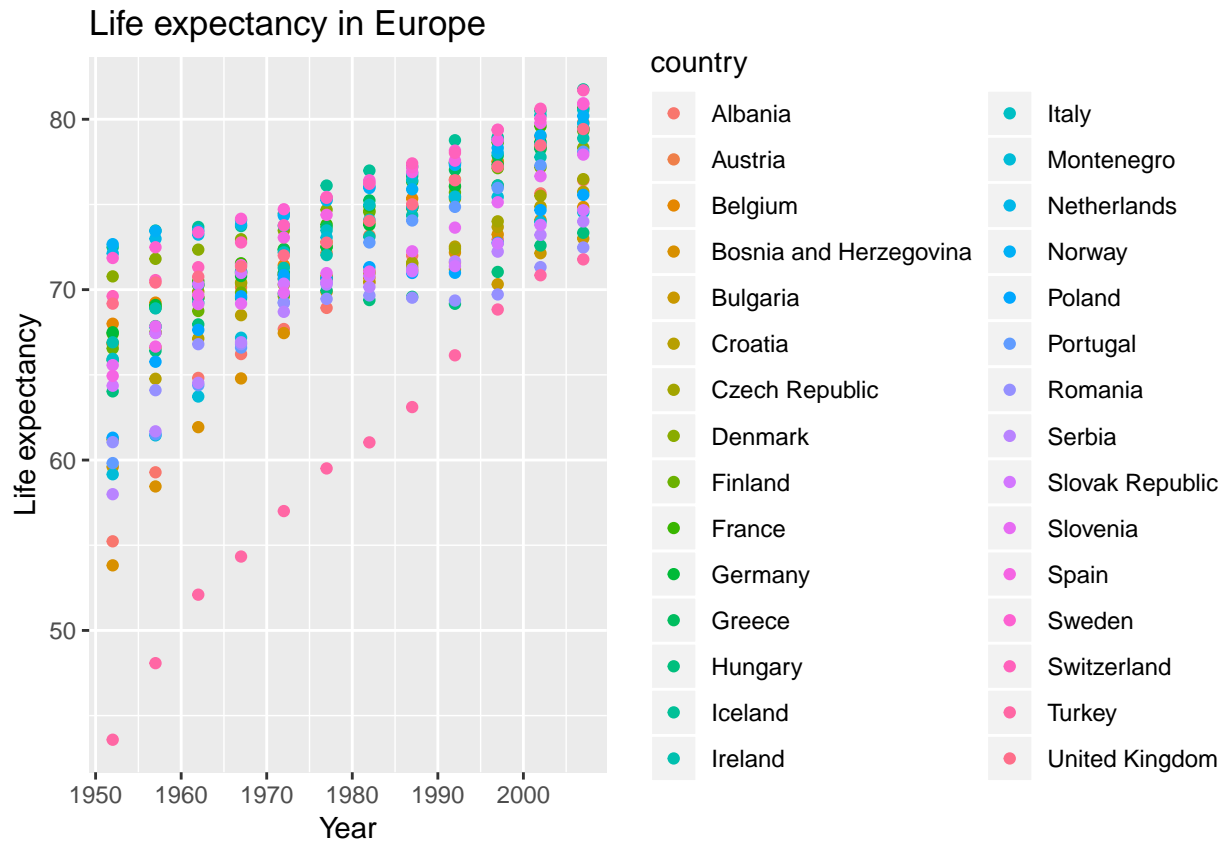
## Use pipes to summarize data

```
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")
```



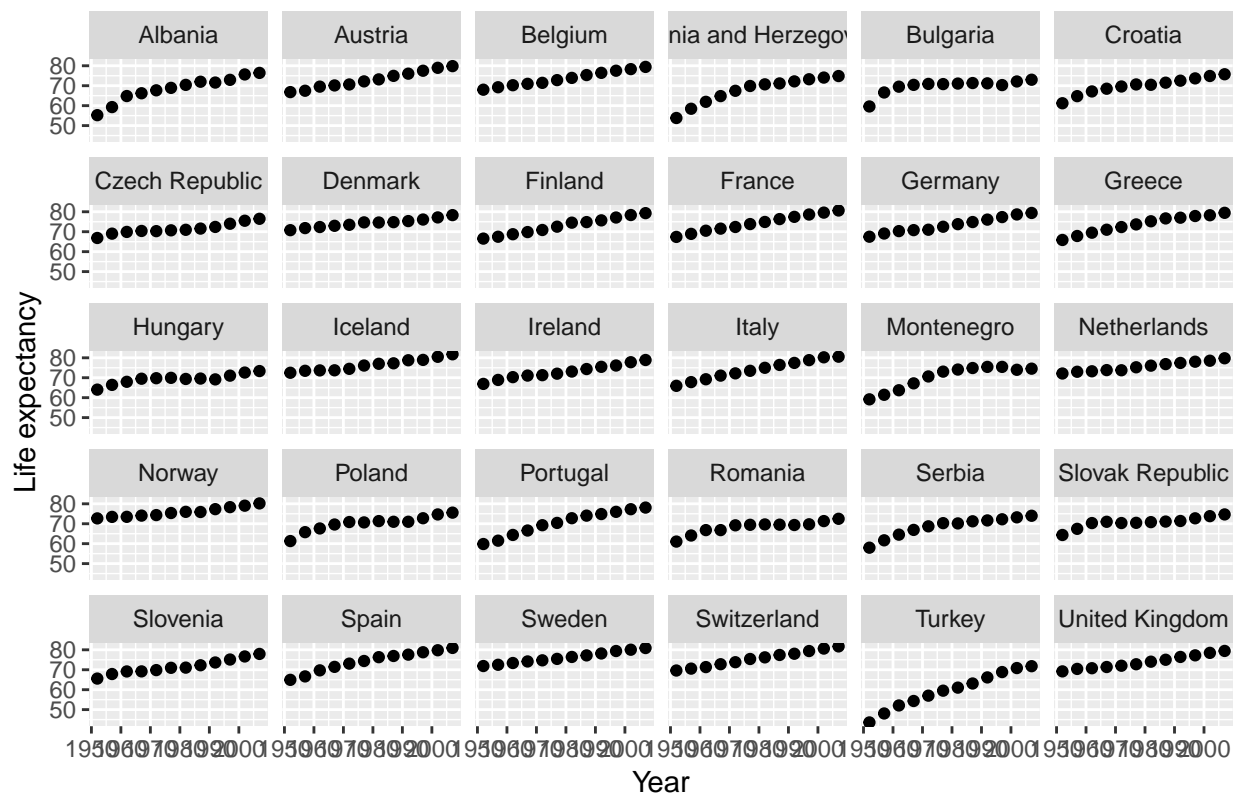
```
gapminder %>%
  filter(continent == "Europe") %>%
  group_by(country, year) %>%
  summarize(gdp_mean = mean(gdpPercap),
             lifeExp_mean = mean(lifeExp)) %>%
  ggplot(aes(x = year, y = lifeExp_mean, color = country)) +
  geom_point() +
  labs(x = "Year",
       y = "Life expectancy",
  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



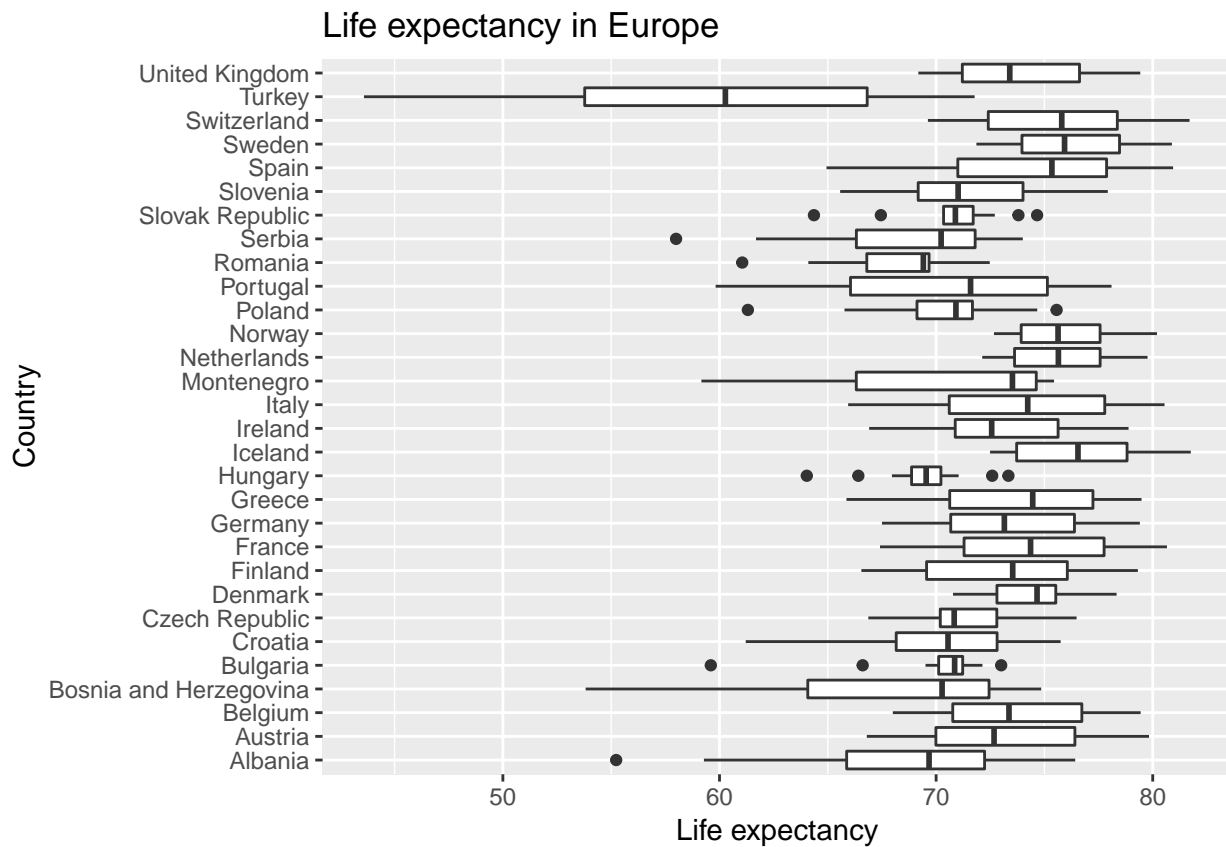


```
gapminder %>%
  filter(continent == "Europe") %>%
  group_by(country, year) %>%
  summarize(gdp_mean = mean(gdpPercap),
             lifeExp_mean = mean(lifeExp)) %>%
  ggplot(aes(x = year, y = lifeExp_mean)) +
  geom_point() +
  labs(x = "Year",
       y = "Life expectancy",
       title = "Life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

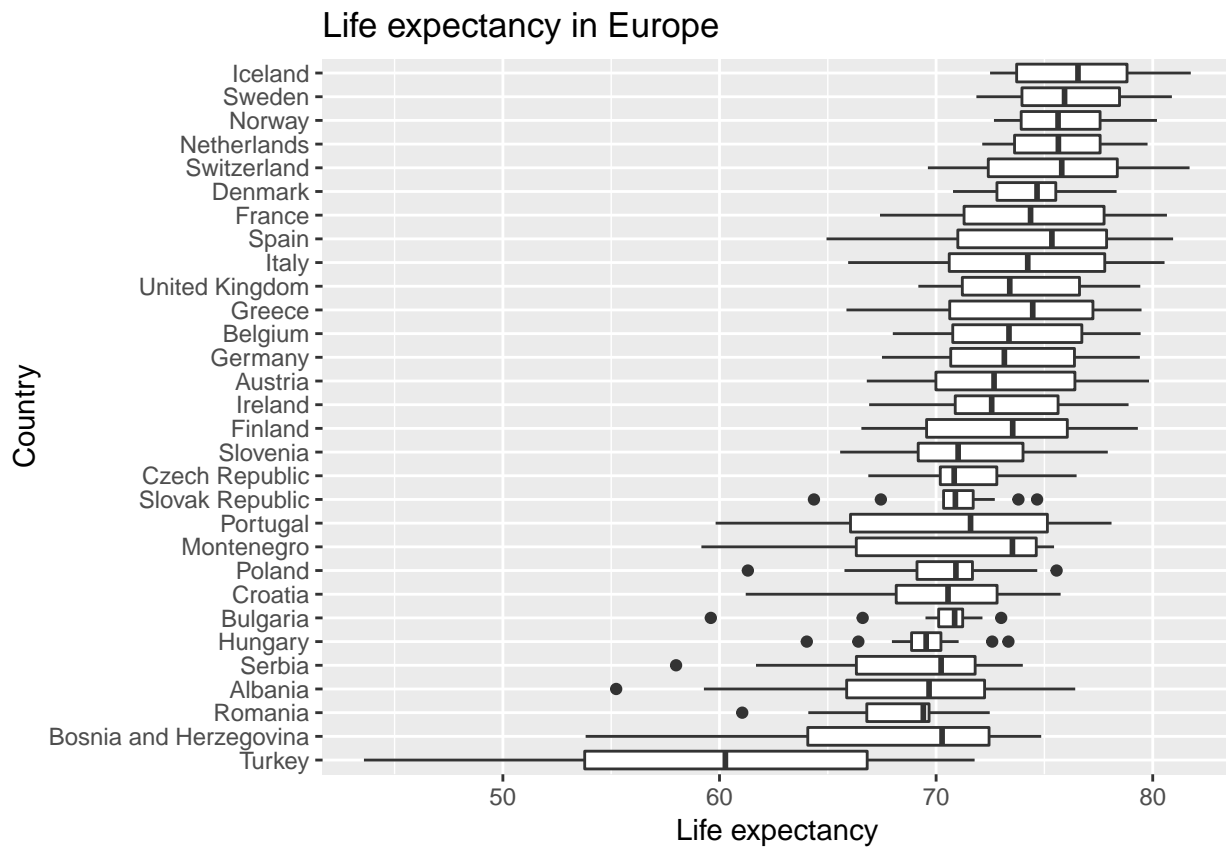
## Life expectancy in Europe



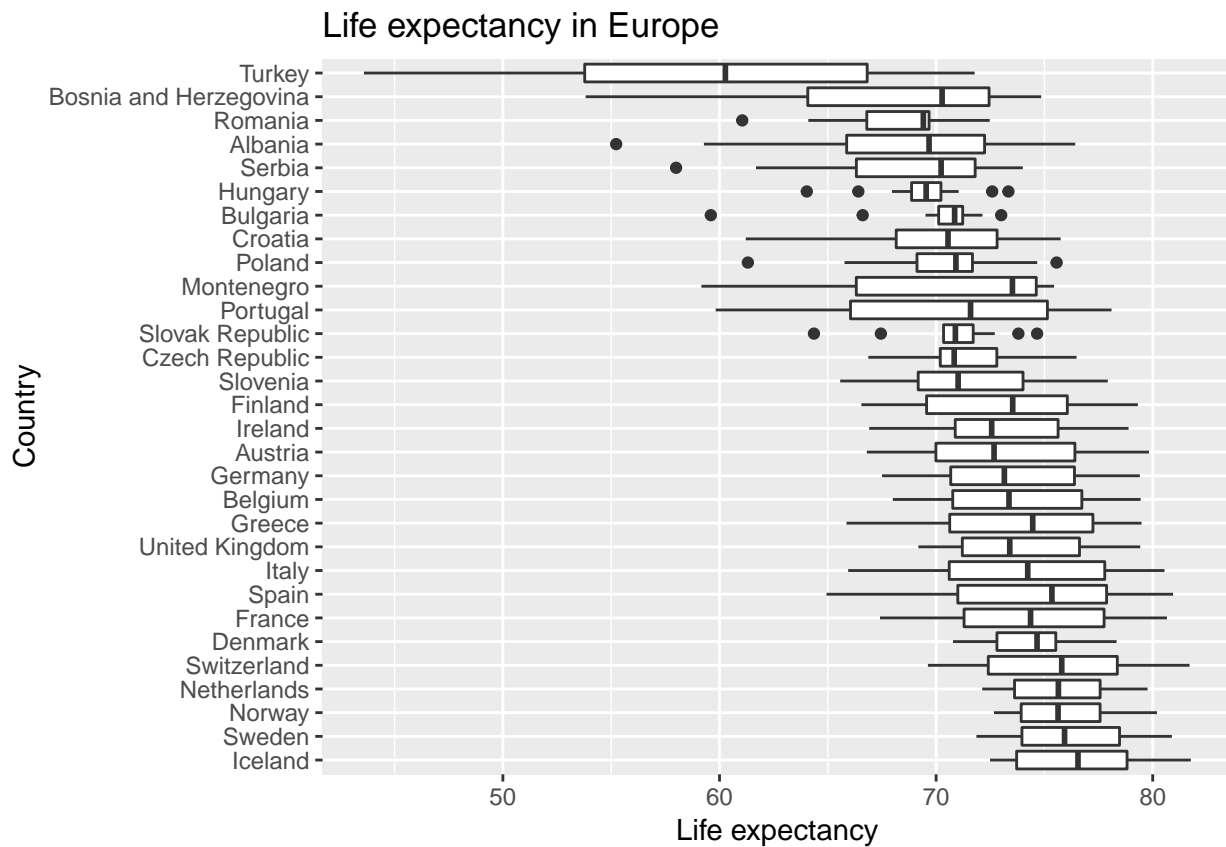
```
gapminder %>%
  filter(continent == "Europe") %>%
  group_by(country, year) %>%
  summarize(gdp_mean = mean(gdpPercap),
             lifeExp_mean = mean(lifeExp)) %>%
  ggplot(aes(x = country, y = lifeExp_mean)) +
  geom_boxplot() +
  labs(x = "Country",
       y = "Life expectancy",
       title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

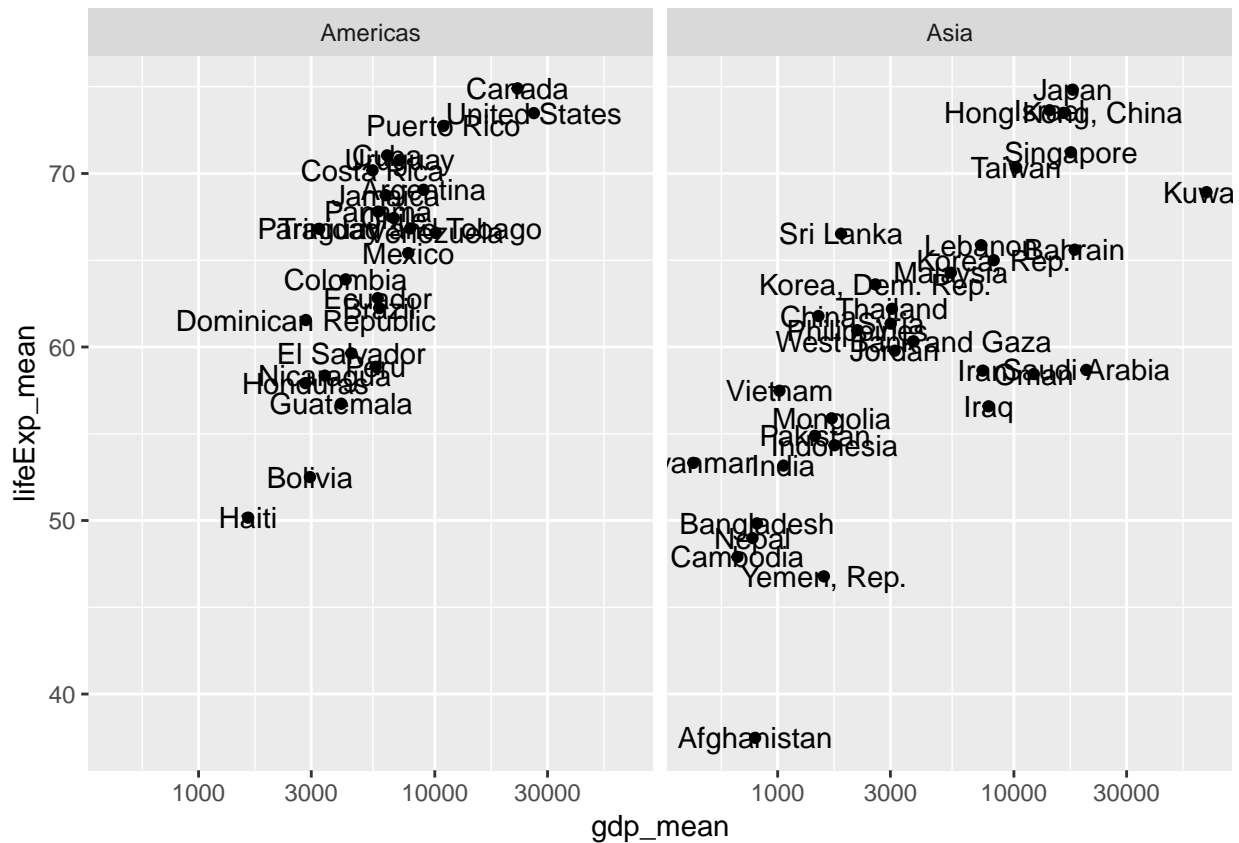


```
gapminder %>%
  filter(continent == "Europe") %>%
  group_by(country, year) %>%
  summarize(gdp_mean = mean(gdpPercap),
            lifeExp_mean = mean(lifeExp)) %>%
  ggplot(aes(x = reorder(country, -lifeExp_mean), y = lifeExp_mean)) +
  geom_boxplot() +
  labs(x = "Country",
       y = "Life expectancy",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



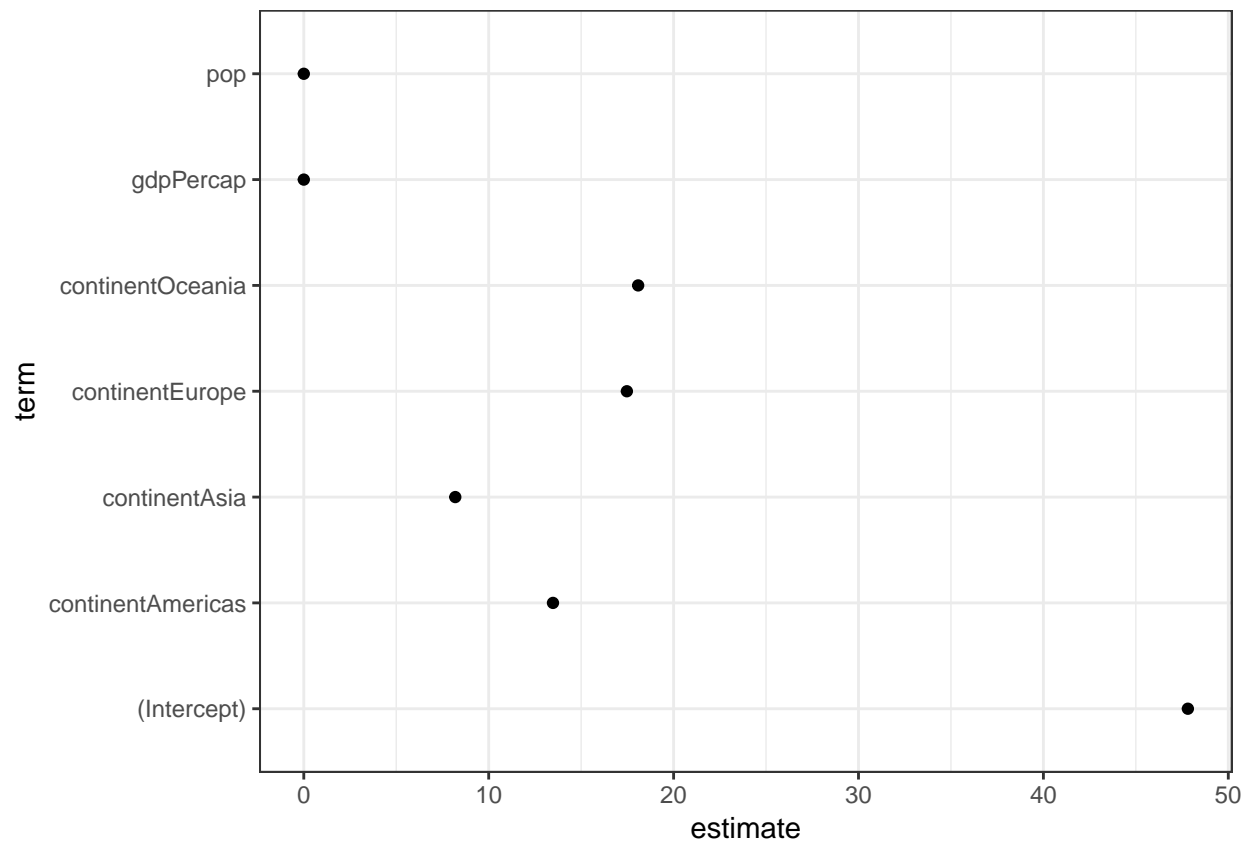
## Plotting models

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

# estimates
out_comp <- tidy(out)

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

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



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

p <- out_conf %>%
  ggplot(aes(x = reorder(term, estimate), y = estimate, ymin = conf.low, ymax = conf.high))

p + geom_pointrange() + coord_flip() + labs(x = "", y = "OLS Estimate") +
  theme_bw()
```

