Intro to grammar of graphics (ggplot2)

15 September 2020

Modern Research Methods

Artwork by @allison_horst

Michael Matta
@mikemattaUH

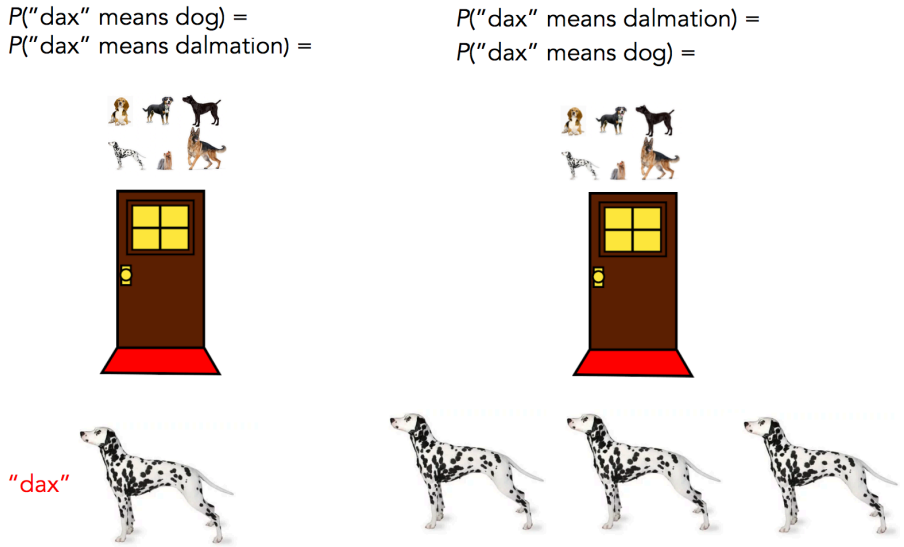
I believe that there are two kinds of people: those who space their R code and those who do not. Pretty clear which group I belong to #rstatsmemes #RStats

6:43 PM · Sep 11, 2021 · Twitter Web App

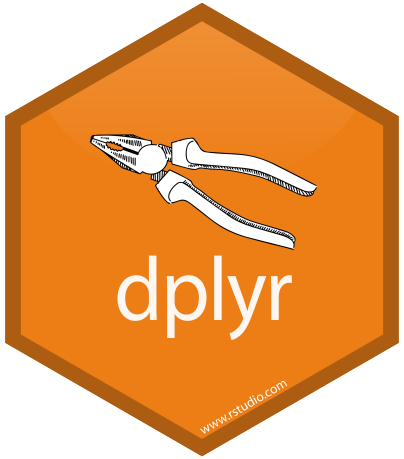
36 Retweets **5** Quote Tweets **389** Likes

Business

- Quiz 1:
  - Folks did very well
  - ”suspicious coincidence effect”
- Assignment 1 due tomorrow at noon
- Office hours today at 2:45 in Porter Hall
- Okay to share exemplary responses?



Last Time: dplyr verbs



Extract variables with **select()**

Extract cases with **filter()**

Arrange cases, with **arrange()**.

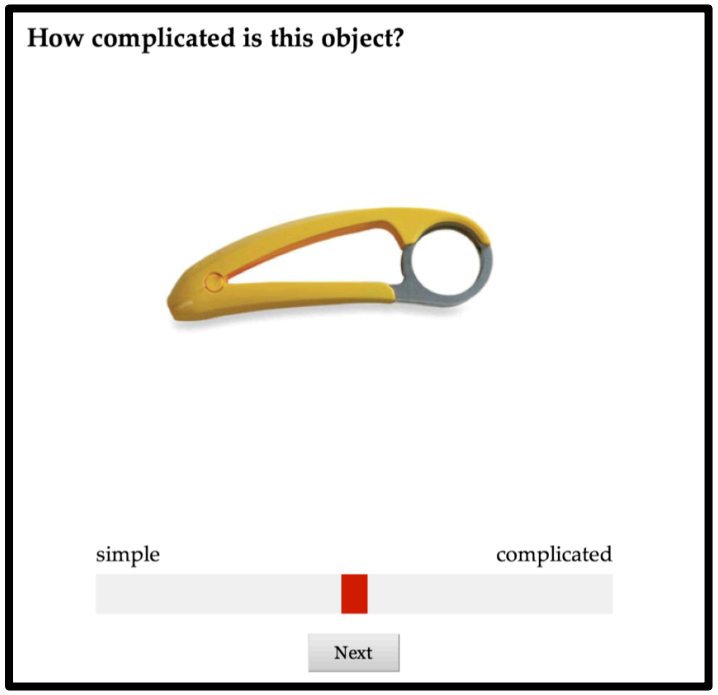
Make tables of summaries with **summarise()**.

Make new variables, with **mutate()**.

Measuring “Conceptual Complexity”

(Lewis & Frank, 2016)

Measuring "Conceptual Complexity"



0 1

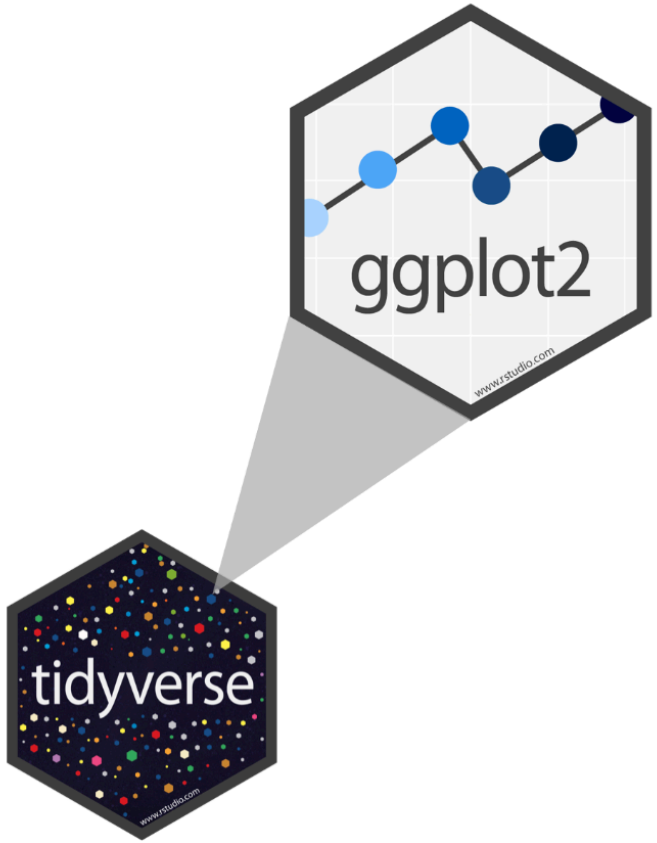
Design:

- Each participant rated an image of a ball and of a circuit first
- Then, random sample of 10 additional objects
- Ran the study twice each with 60 participants

sample	subjectid	objectid	rating
1	1	54	0.24331551
1	5	54	0.43315508
1	6	54	0.26470588
1	10	54	0.18181818
1	18	54	0.13636364
1	36	54	0.21122995
1	40	54	0.35828877
1	42	54	0.73529412
1	53	54	0.42780749
1	58	54	0.40641711
1	1	57	0.76203209

What are some analytical questions we could ask of this data?

ggplot2



- **ggplot2** is tidyverse's data visualization package
- The gg in “ggplot2" stands for Grammar of Graphics
- It is inspired by the book **Grammar of Graphics** by Leland Wilkinson
- A grammar of graphics is a tool that enables us to concisely describe the components of a graphic

What data do you want to plot?

1. Tidy Data

```
p ← ggplot(data = gapminder, ...
```

gdp	lifexp	pop	continent
340	65	31	Euro
227	51	200	Amer
909	81	80	Euro
126	40	20	Asia

Map variables to aesthetics (parts) of plot

What kind of plot do you want

1. Tidy Data

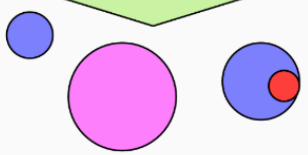
```
p ← ggplot(data = gapminder, ...
```

gdp	lifexp	pop	continent
340	65	31	Euro
227	51	200	Amer
909	81	80	Euro
126	40	20	Asia

2. Mapping

```
p ← ggplot(data = gapminder,
           mapping = aes(x = gdp,
           y = lifexp, size = pop,
           color = continent))
```

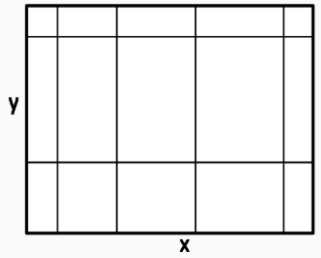
3. Geom



```
p + geom_point()
```

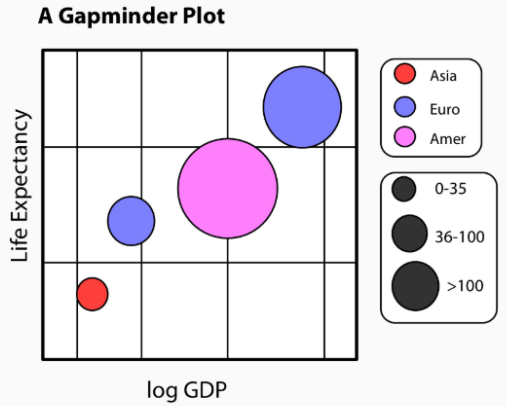
4. Co-Ordinates & Scales

```
p + coord_cartesian() +
    scale_x_log10()
```



5. Labels & Guides

```
p + labs(x = "log GDP",
         y = "Life Expectancy",
         title = "A Gapminder Plot")
```



Data: gapminder dataset

country	continent	year	life_exp	pop	gdp_percap
Afghanistan	Asia	1952	28.801	8425333	779.4453
Afghanistan	Asia	1957	30.332	9240934	820.8530
Afghanistan	Asia	1962	31.997	10267083	853.1007
Afghanistan	Asia	1967	34.020	11537966	836.1971
Afghanistan	Asia	1972	36.088	13079460	739.9811
Afghanistan	Asia	1977	38.438	14880372	786.1134
Afghanistan	Asia	1982	39.854	12881816	978.0114
Afghanistan	Asia	1987	40.822	13867957	852.3959
Afghanistan	Asia	1992	41.674	16317921	649.3414
Afghanistan	Asia	1997	41.763	22227415	635.3414

Analytic question:

What's the relationship between GDP and life expectancy?

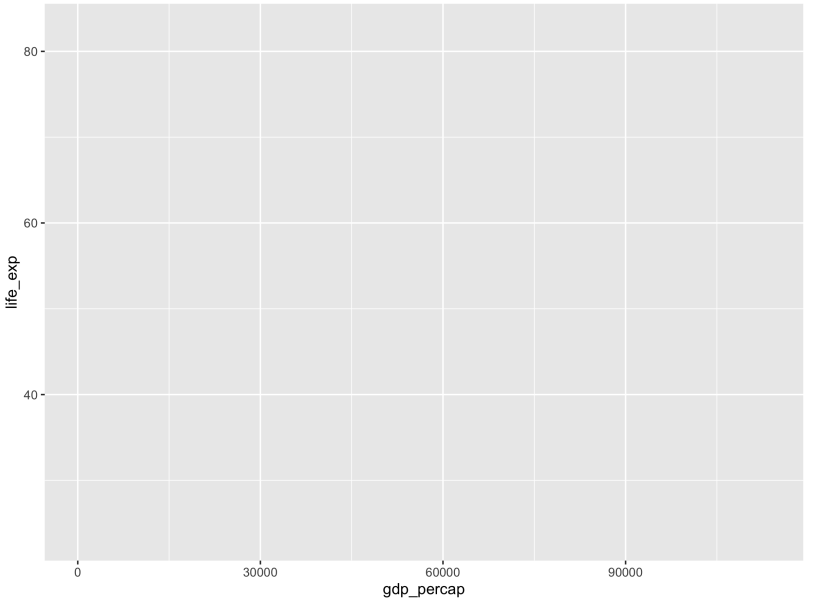
What plot could we make to explore that?

Let's start with an x-y scatter plot.

```
ggplot()
```

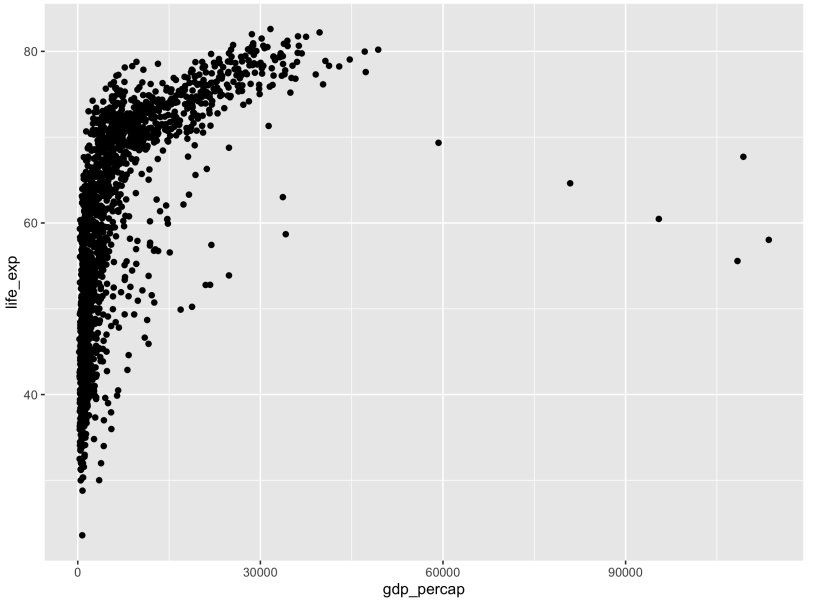
```
ggplot(data = gapminder)
```

```
ggplot(data = gapminder, mapping = aes(x = gdp_percap,
                                       y = life_exp))
```



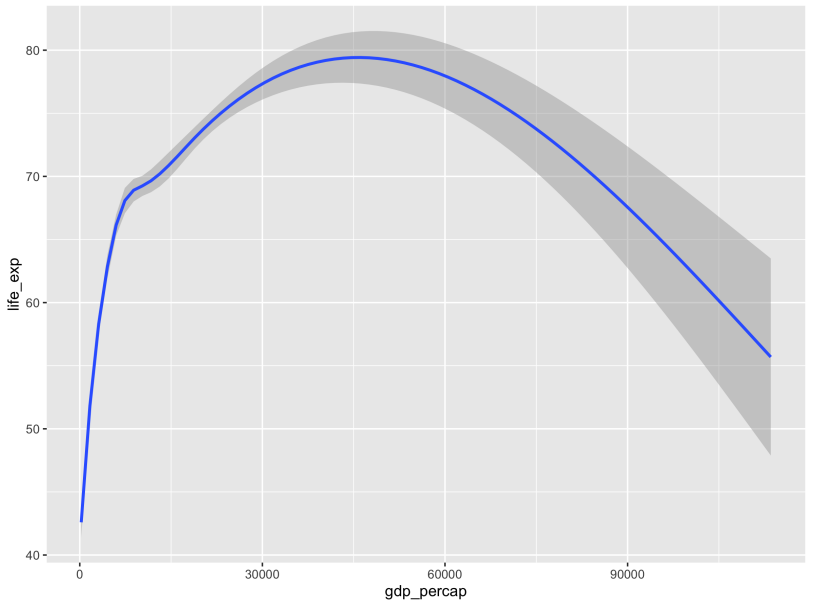
Let's add a geom (note the +).

```
ggplot(data = gapminder, mapping = aes(x = gdp_percap,
                                       y = life_exp)) +
  geom_point()
```



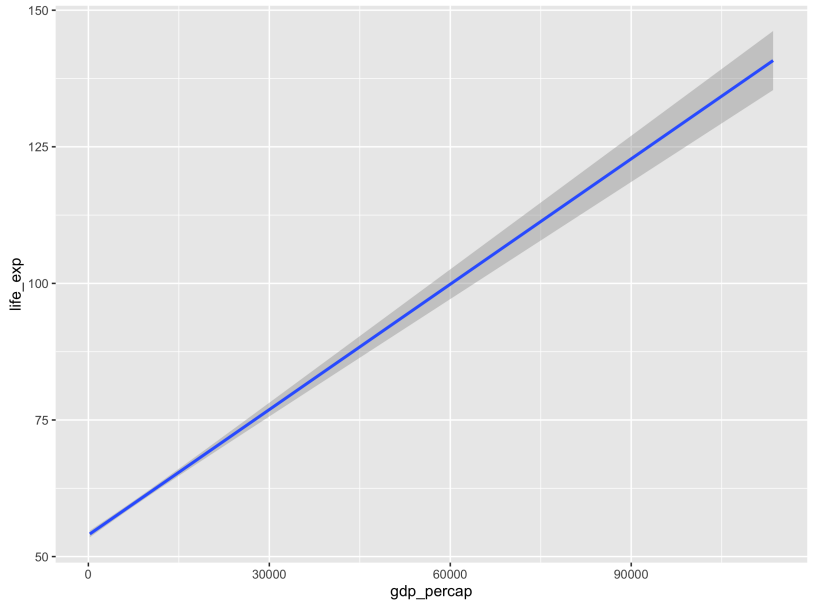
Let's try a different geom: a smoothed line.

```
ggplot(data = gapminder, mapping = aes(x = gdp_percap,
                                       y = life_exp)) +
  geom_smooth()
```



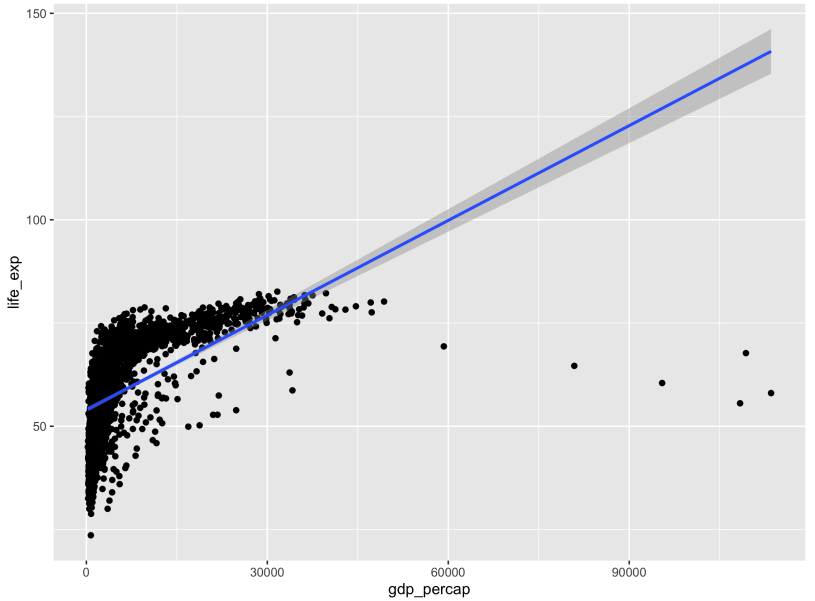
Maybe we want a linear line.

```
ggplot(data = gapminder, mapping = aes(x = gdp_percap,
                                       y = life_exp)) +
  geom_smooth(method = "lm")
```



It might be nice to see the raw data WITH the line. We can combine geoms!

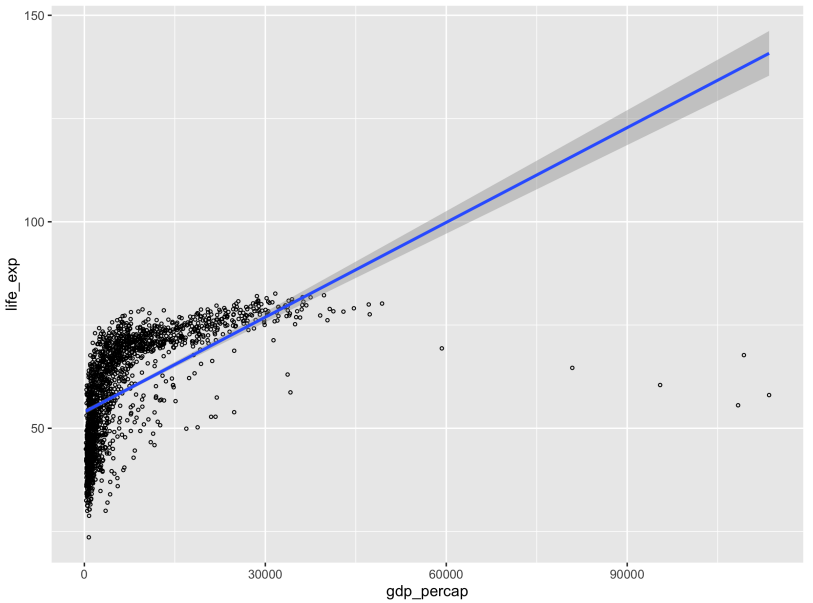
```
ggplot(data = gapminder, mapping = aes(x = gdp_percap,
                                       y = life_exp)) +
  geom_point() +
  geom_smooth(method = "lm")
```



Those points are too big. Let's make them smaller. Let's also change the shape.

```
ggplot(data = gapminder, mapping = aes(x = gdp_percap,
                                       y = life_exp)) +
  geom_point(size = .8, shape = 1) +
  geom_smooth(method = "lm")
```

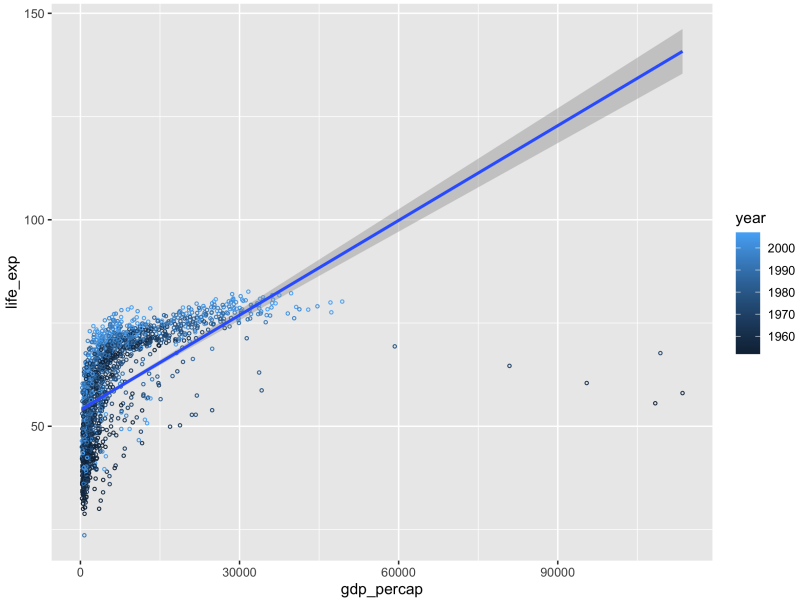
Why don't these changes go in aesthetics?



How could we add information about year?

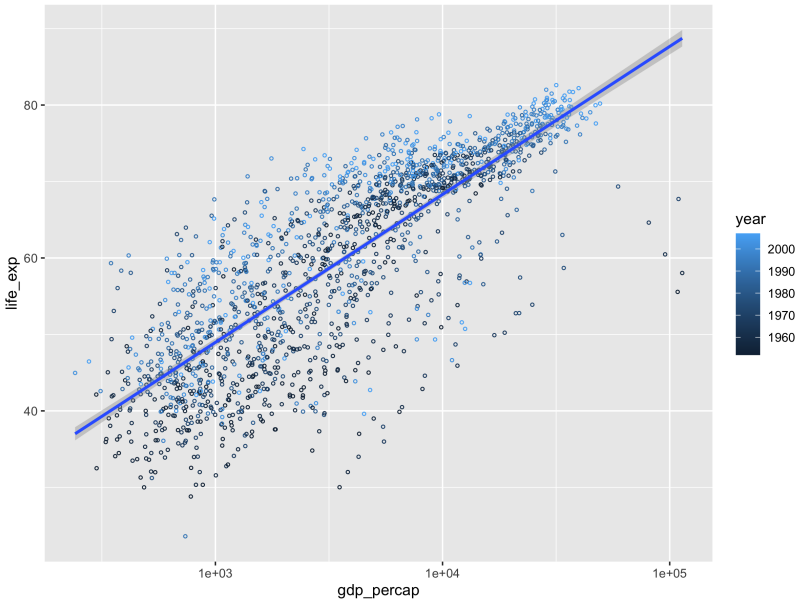
Through the color aesthetic...

```
ggplot(data = gapminder, mapping = aes(x = gdp_percap,
                                       y = life_exp,
                                       color = year)) +
  geom_point(size = .8, shape = 1) +
  geom_smooth(method = "lm")
```



The data are all squished on the x axis. We can fix that by changing the scale on the x-axis to the be the log of x.

```
ggplot(data = gapminder, mapping = aes(x = gdp_percap,
                                       y = life_exp,
                                       color = year)) +
  geom_point(size = .8, shape = 1) +
  geom_smooth(method = "lm") +
  scale_x_log10()
```



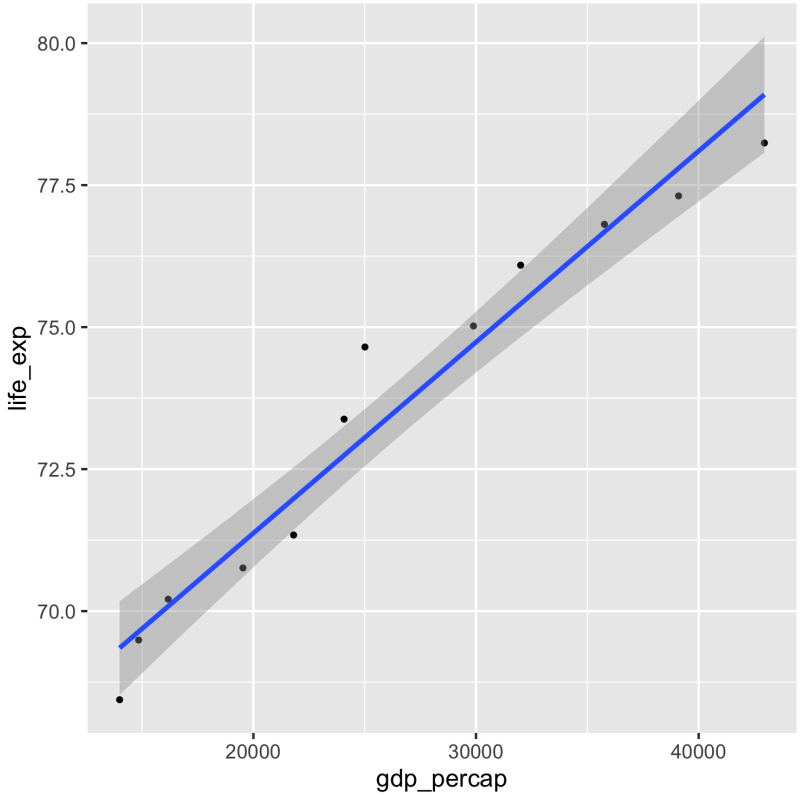
What if wanted to look at just data for the United States?

Filter!

```
us_gapminder <- gapminder %>%
  filter(country == "United States")
```

```
## # A tibble: 5 × 6
##   country       continent  year life_exp       pop gdp_percap
##   <fct>         <fct>     <int>    <dbl>     <int>      <dbl>
## 1 United States Americas   1952     68.4 157553000     13990.
## 2 United States Americas   1957     69.5 171984000     14847.
## 3 United States Americas   1962     70.2 186538000     16173.
## 4 United States Americas   1967     70.8 198712000     19530.
## 5 United States Americas   1972     71.3 209896000     21806.
```

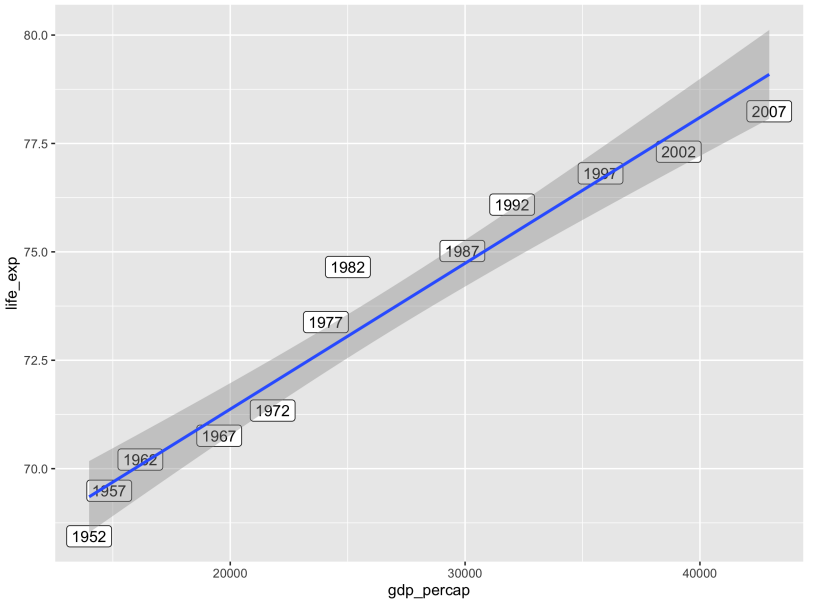
```
ggplot(us_gapminder, mapping = aes(x = gdp_percap, y = life_exp)) +
geom_point(size = .8) +
geom_smooth(method = "lm")
```



What if wanted to label the points with years?

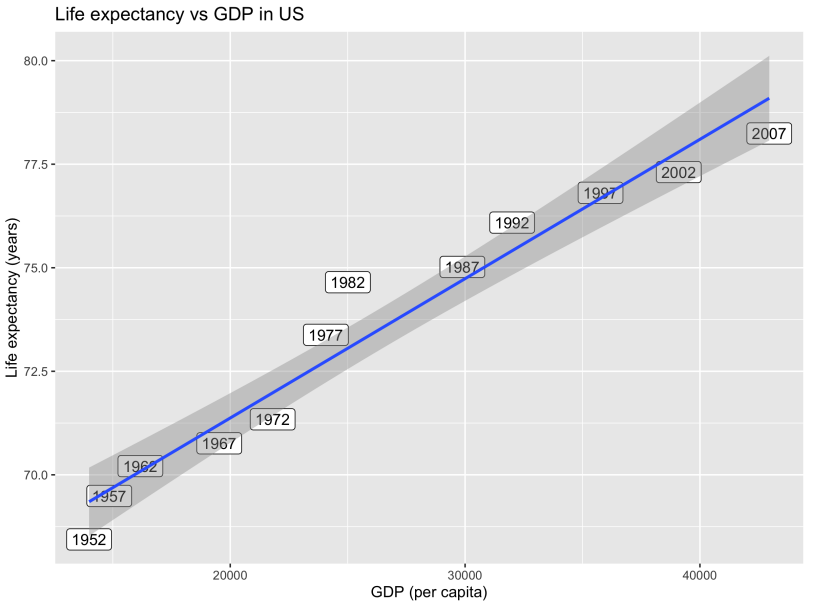
`geom_label()`

```
ggplot(us_gapminder, mapping = aes(x = gdp_percap, y = life_exp, label = year
geom_label() +
geom_smooth(method = "lm")
```



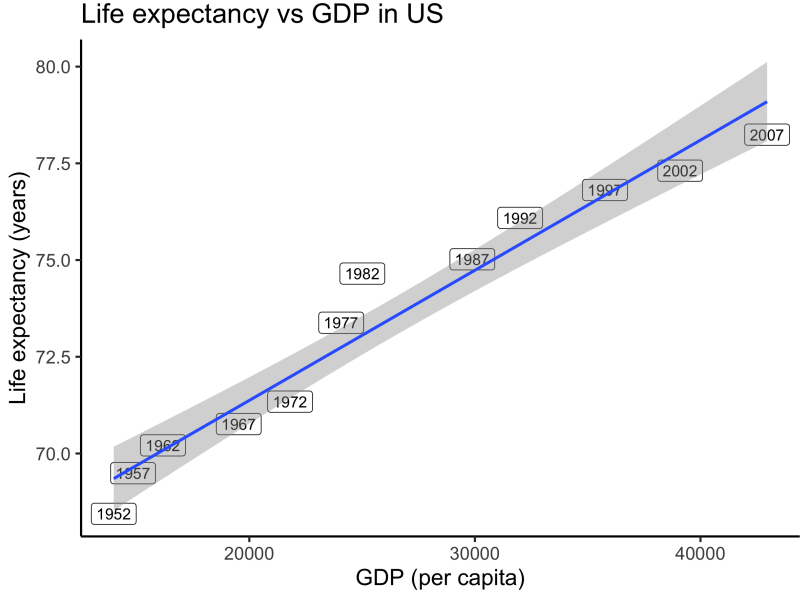
Let's do a few things to make our plot more readable: Add clearer axis labels and a title.

```
ggplot(data = us_gapminder,
       mapping = aes(x = gdp_percap,
                     y = life_exp,
                     label = year)) +
geom_label() +
geom_smooth(method = "lm")  +
xlab("GDP (per capita)") +
ylab("Life expectancy (years)") +
ggtitle("Life expectancy vs GDP in US")
```



We can also change the *theme*. I like `theme_classic`.

```
ggplot(data = us_gapminder,
       mapping = aes(x = gdp_percap,
                     y = life_exp,
                     label = year)) +
geom_label() +
geom_smooth(method = "lm")  +
xlab("GDP (per capita)") +
ylab("Life expectancy (years)") +
ggtitle("Life expectancy vs GDP in US") +
theme_classic(base_size = 16)
```

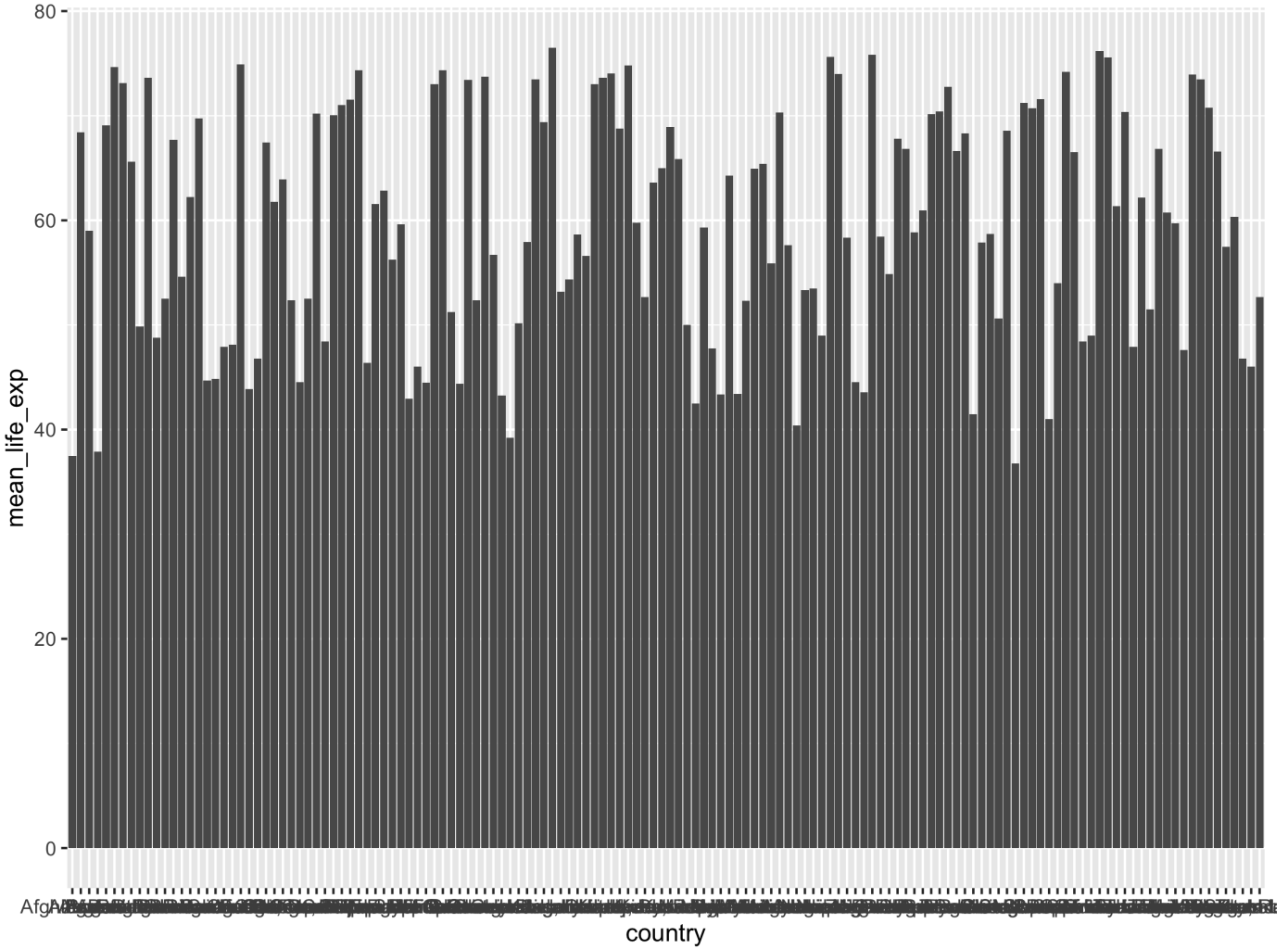


So far we've just been working with the raw data in tidy format. We can also *summarize* our data before we plot it.

What if we wanted to know which country had the highest life expectancy? The second highest?

```
life_exp_by_country <- gapminder %>%
  group_by(country) %>%
  summarize(mean_life_exp = mean(life_exp))

ggplot(life_exp_by_country, mapping = aes(x = country,
                                          y = mean_life_exp)) +
  geom_bar(stat = "identity")
```



Yowza! That's a disaster!

One way to make this plot more attractive is to reorder the bars.

To do that, we need to know about **factors** in R.

R uses factors to handle qualitative variables (variables that have a fixed and known set of possible values)

```
x <- c("Monday", "Tuesday", "Wednesday", "Monday")
x
```

```
## [1] "Monday"    "Tuesday"   "Wednesday" "Monday"
```

```
class(x)
```

```
## [1] "character"
```

```
y <- as.factor(x)
y
```

```
## [1] Monday    Tuesday   Wednesday Monday
## Levels: Monday Tuesday Wednesday
```

```
class(y)
```

```
## [1] "factor"
```

Factors are like a character (level labels) and an integer (level numbers) glued together. They are ORDERED character levels.

```
glimpse(y)
```

```
##  Factor w/ 3 levels "Monday","Tuesday",..: 1 2 3 1
```

When you print a dataframe it will tell you each variable type, including factors.

```
slice(gapminder, 1:4)
```

```
## # A tibble: 4 × 6
##   country     continent  year life_exp      pop gdp_percap
##   <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.
```

The **forcats** package in the tidyverse is helpful for working with factors.

`fct_reorder()`: Reordering a factor by another variable.

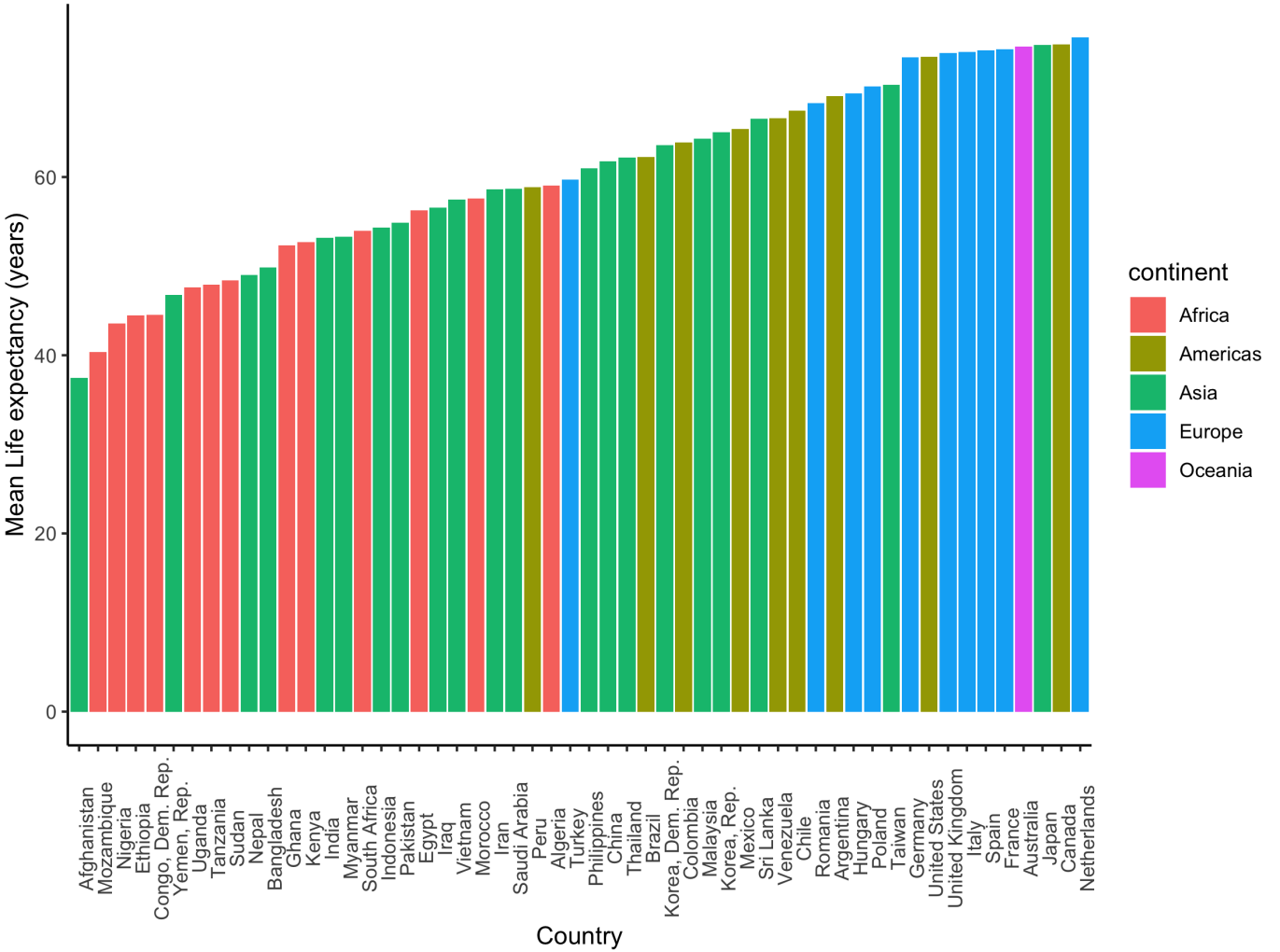
`fct_relevel()`: Changing the order of factor levels by hand.

`fct_recode()`: Changing factor level names by hand

Here's a prettier version of the earlier plot:

```
life_exp_by_country <- gapminder %>%
  group_by(continent, country) %>%
  summarize(mean_life_exp = mean(life_exp),
            mean_pop = mean(pop)) %>%
  filter(mean_pop > 10000000)


ggplot(life_exp_by_country, aes(x = fct_reorder(country, mean_life_exp) ,
                                        y = mean_life_exp,
                                        fill = continent)) +
  geom_bar(stat = "identity") +
  ylab("Mean Life expectancy (years)") +
  xlab("Country") +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 90))
```


Mean Life expectancy (years)
0
20
40
60
Afghanistan
Mozambique
Nigeria
Ethiopia
Congo, Dem. Rep.
Yemen, Rep.
Uganda
Tanzania
Sudan
Nepal
Bangladesh
Ghana
Kenya
India
Myanmar
South Africa
Indonesia
Pakistan
Egypt
Iraq
Vietnam
Morocco
Iran
Saudi Arabia
Peru
Algeria
Turkey
Philippines
China
Thailand
Brazil
Korea, Dem. Rep.
Colombia
Malaysia
Korea, Rep.
Mexico
Sri Lanka
Venezuela
Chile
Romania
Argentina
Hungary
Poland
Taiwan
Germany
United States
United Kingdom
Italy
Spain
France
Australia
Japan
Canada
Netherlands
Country
continent
Africa
Americas
Asia
Europe
Oceania