

Hands-On Data Visualization with {ggplot2}

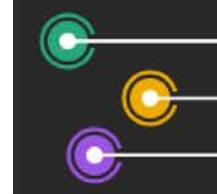
— Concepts —

Cédric Scherer // Pearson Live Training Session for O'Reilly // June 13, 2023



Welcome!





CÉDRIC SCHERER

Data Visualization & Information Design



Consulting

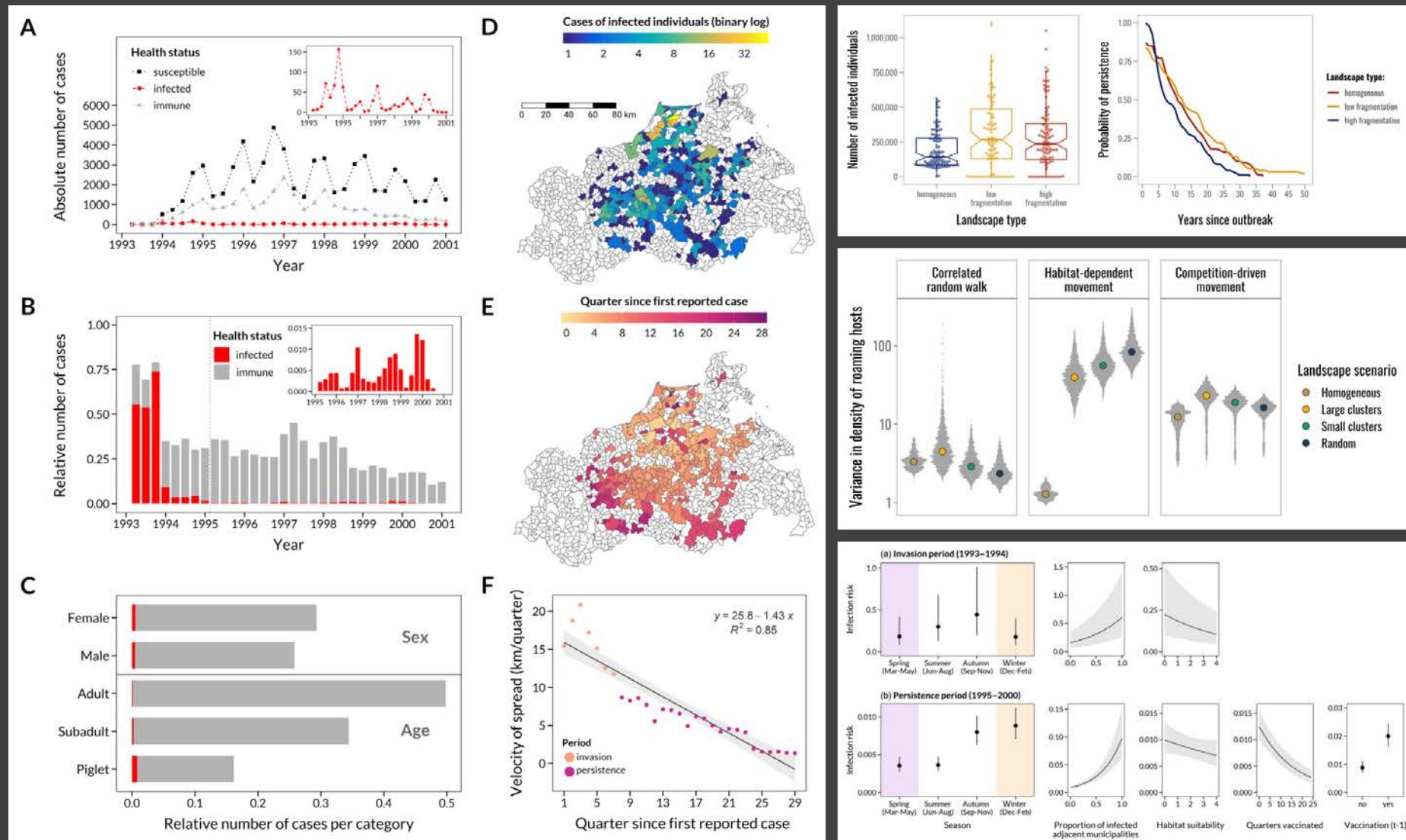


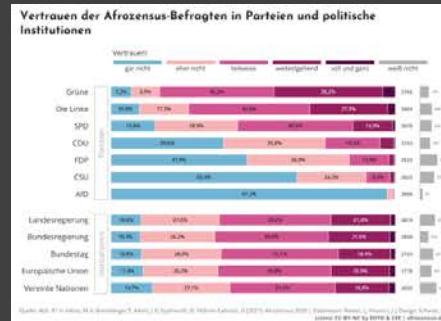
Coaching



Coding







**Politiker rechnen bald mit einer Fortsetzung der Fußball-Bundesliga.
Wenn auch nicht im Stadion, so ist es voraussichtlich bald wieder möglich
Fußball im Fernsehen zu sehen.**

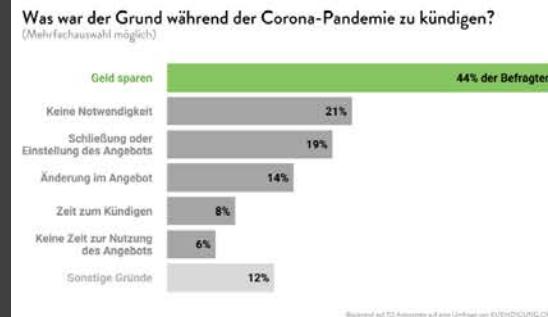
Ich habe **einen** Vertrag mit einem Anbieter für Sportübertragungen.



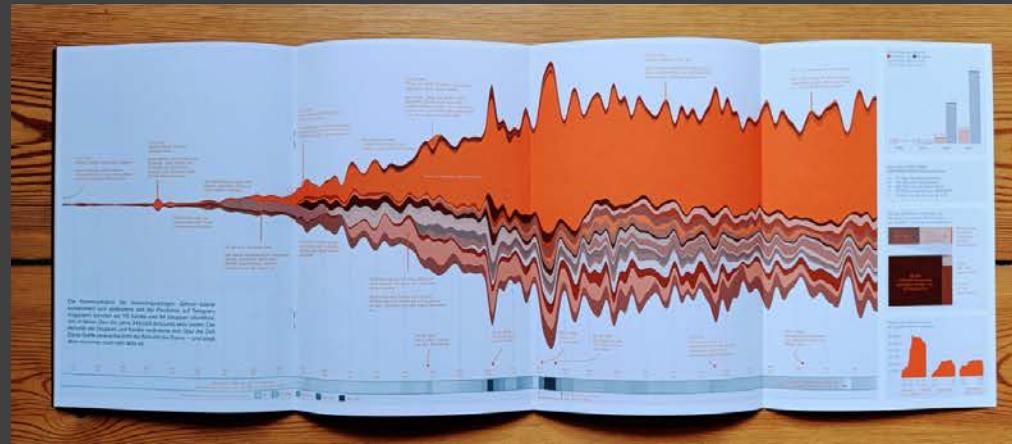
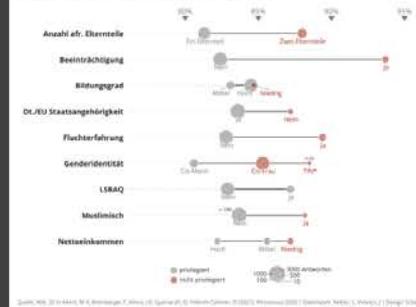
Ich habe **keinen** Vertrag mit einem Anbieter für Sportübertragungen.



Basiswert auf 1518 Befragten auf eine Umfrage von EUROTODAY.GOV



Häufigkeit von Diskriminierungserfahrungen entlang ausgewählter Vielfaltsdimensionen im Bereich „Arbeitsleben“



GRAPHIC SCIENCE

Text by Clara Moskowitz | Graphic by Cédric Scherer and Georgios Karamanis

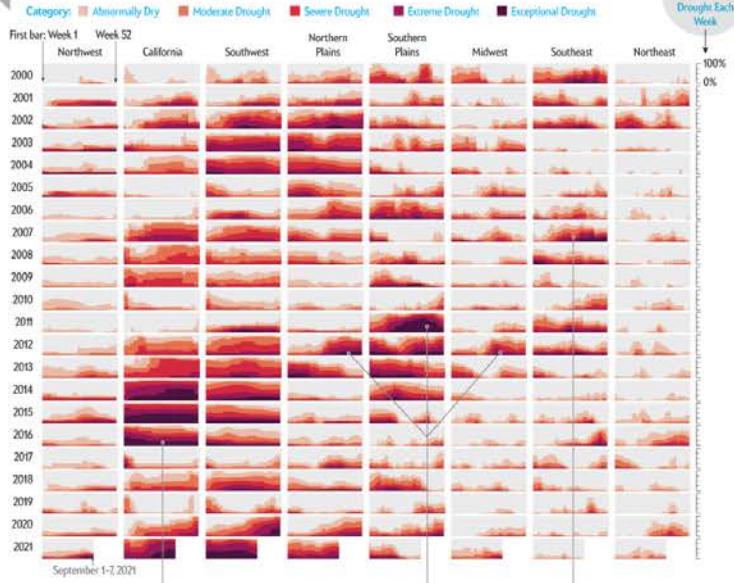
Escalating Drought

Climate change is intensifying periods of extreme dryness, particularly in the U.S. West

For more than 20 years the National Drought Mitigation Center (NDMC) has been monitoring dozens of indices of drought around the country, including satellite measurements of evaporation and color in vegetation, soil-moisture sensors, rainfall estimates, and river and streamflow levels. Although the agency's weekly assessments have identified periods of exceptional drought before, lately dryness has been ramping up. "The changing climate is definitely contributing to more natural disasters, drought being one of them," says Brian Fuchs, a climatologist who oversees the weekly report at the NDMC. "We're seeing more frequent and high-intensity episodes. This year some of these areas in the West have been in drought more than they have been without drought."



Drought Extent and Intensity by Region over Time



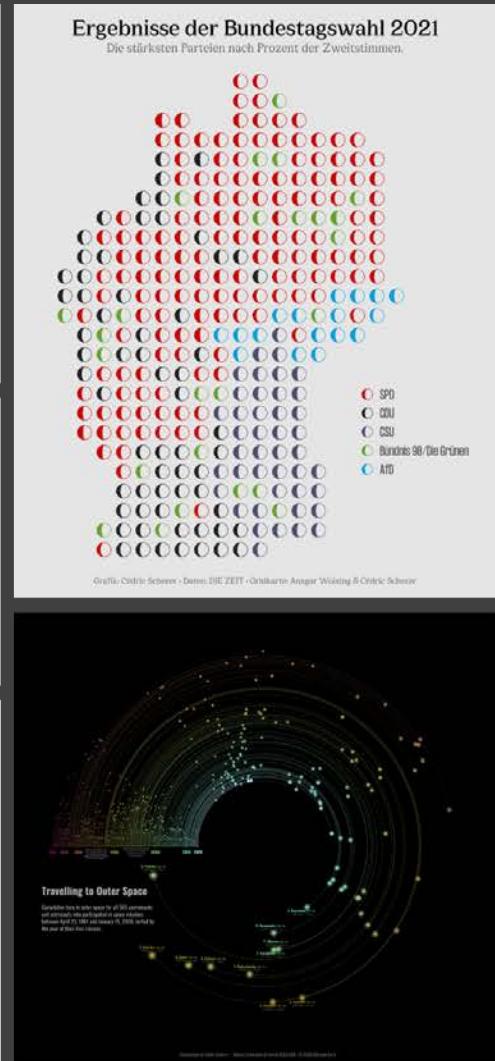
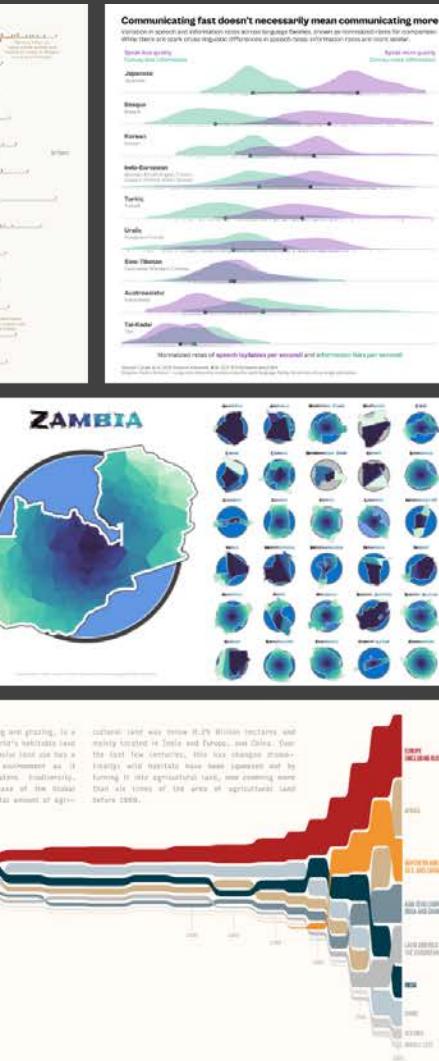
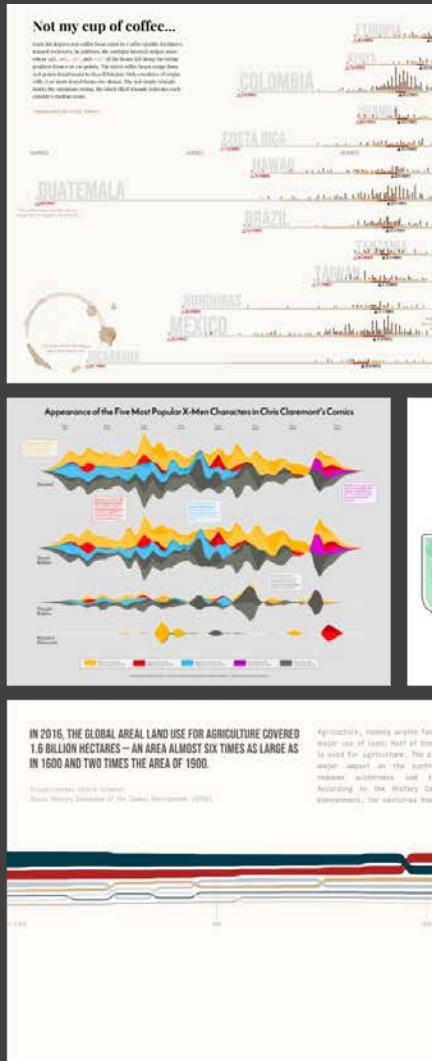
California experienced its hottest drought in recorded history from 2012 to 2016. A warming climate makes the atmosphere thirstier, which increases evaporation and boosts drought.

A drought that originated in the Southern Plains in 2011 eventually spread to the Midwest and Northern Plains when the moisture coming in from the Gulf of Mexico was absorbed by the parched South before it could reach the North.

The Southeast's driest year to date was 2007, when only 31.85 inches of rain fell in Atlanta, 62 percent of its average yearly rainfall.

74 Scientific American, November 2021

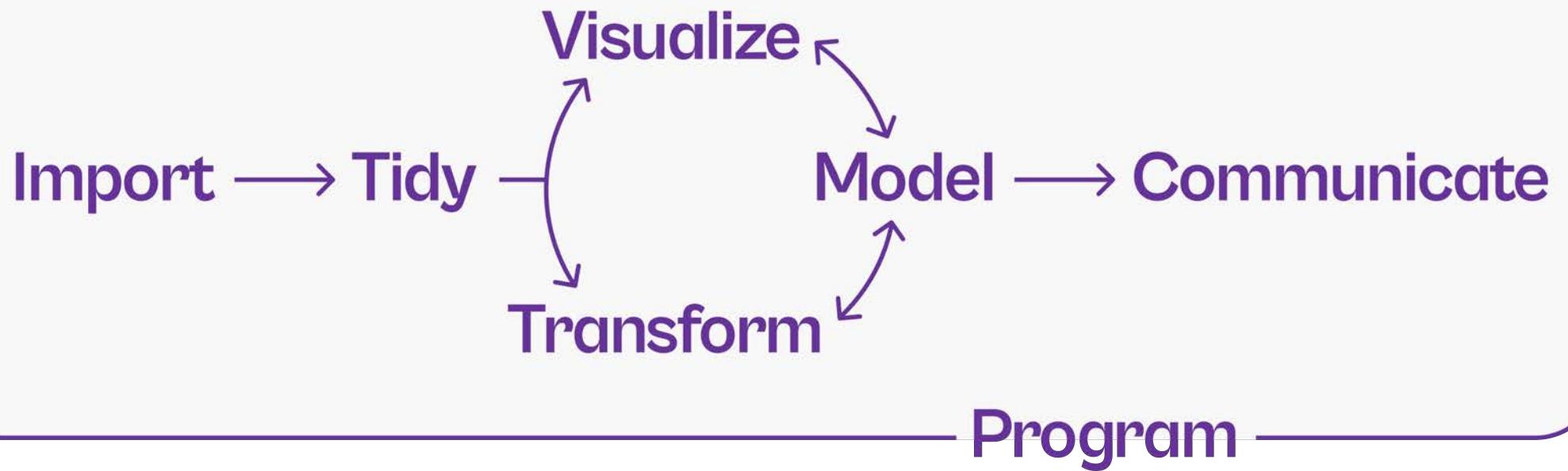




— What is the Course About? —



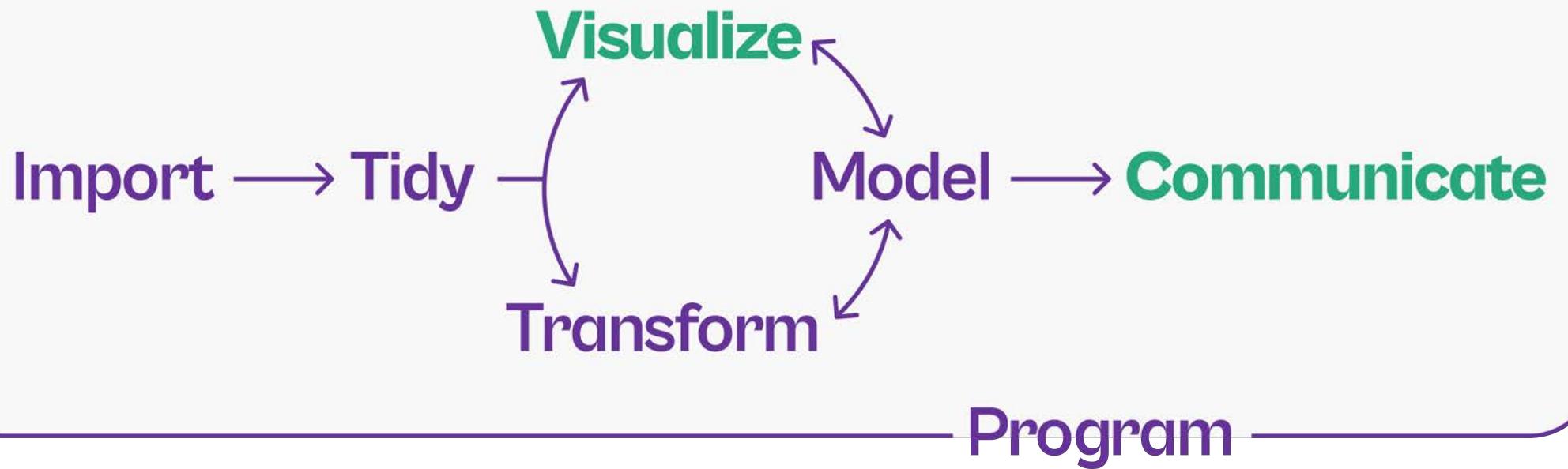
The Data Science Workflow



The data science workflow, modified from "[R for Data Science](#)"



The Data Science Workflow



The data science workflow, modified from “R for Data Science”





What is R?

R is a programming language and free software environment for statistical computing and graphics.

R was conceived in 1992 by **Ross Ihaka** and **Robert Gentleman** as an open source implementation of the **S** programming language and released in 1995.

Since then, R has outgrown its original purpose and is used to:

- run statistical analyses and data-science workflows
- design high-level, publication-ready visualizations
- generate automated reports
- develop stand-alone web applications
- create presentation slides, books, and web pages



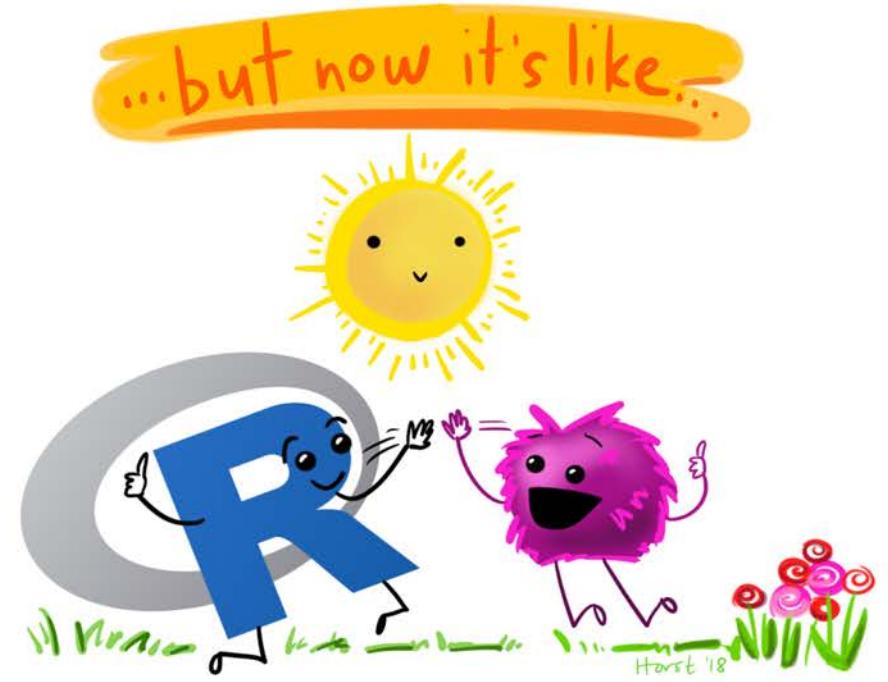
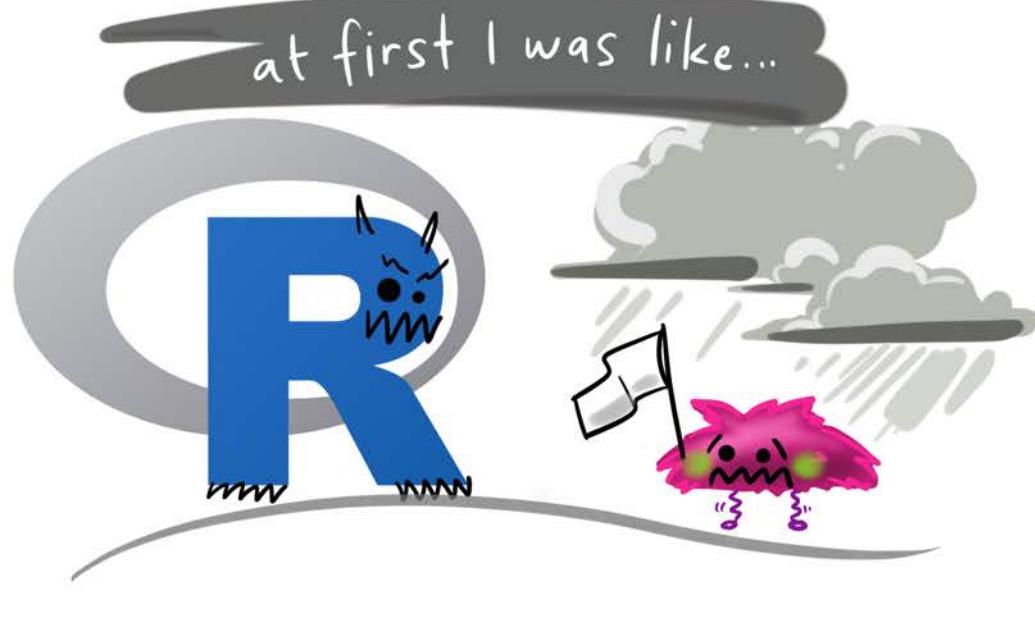


Illustration by [Allison Horst](#)



POLL

Do you use R already?

- Yes, on a regular basis.
- Yes, from time to time.
- Yes, but just starting.
- No.



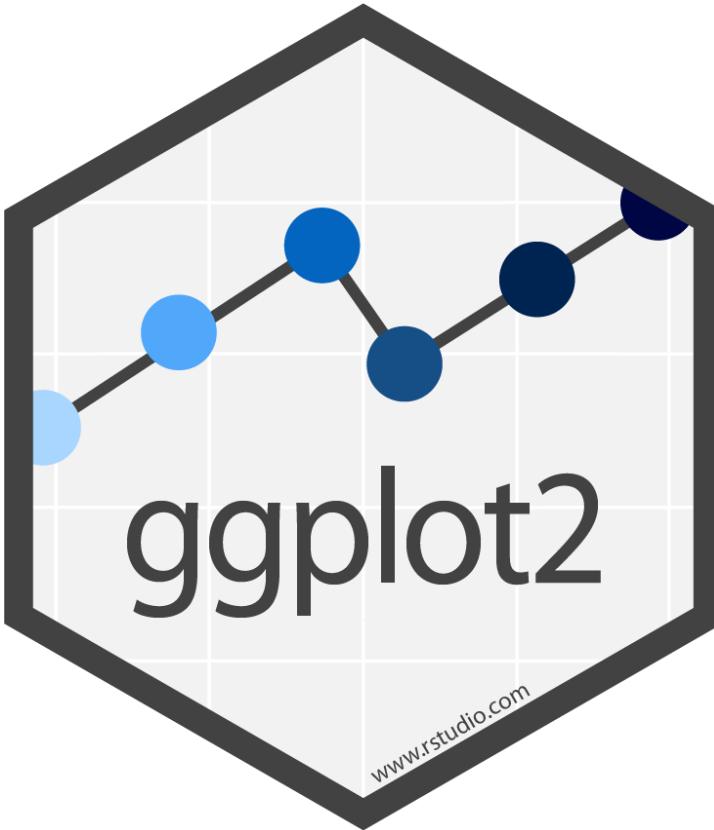
The Grammar of Graphics



The Grammar of Graphics

- was introduced by Lee Wilkinson in 2005
- can be used to concisely describe components of a graphic
- provides a set of structural rules
- consists of syntax and semantics
- can be used to define “uncommon” visualizations





{ggplot2} is a system for declaratively creating graphics,
based on “The Grammar of Graphics” (Wilkinson, 2005).

You provide the data, tell **{ggplot2}** how to map variables to aesthetics,
what graphical primitives to use, and it takes care of the details.



Advantages of {ggplot2}

- consistent underlying “grammar of graphics”
- code-first approach → reproducible and transparent workflow
- very flexible, layered plot specification
- theme system for polishing plot appearance
- lots of additional functionality thanks to extensions
- active and helpful community



POLL

Do you have created visualizations with {ggplot2} already?

- Yes, I can create publication-ready figures with it.
- Yes, but I use it in combination with other tools.
- Yes, but I would consider myself a beginner.
- No, I have never used it before.



Break
— 10 minutes —



A Basic ggplot

data



A Basic ggplot

**data +
mapping**



A Basic ggplot

**data +
mapping +
geometry**



A Polished ggplot

data +
mapping +
geometry +
statistics +
facets +
coordinate systems +
scales +
theme



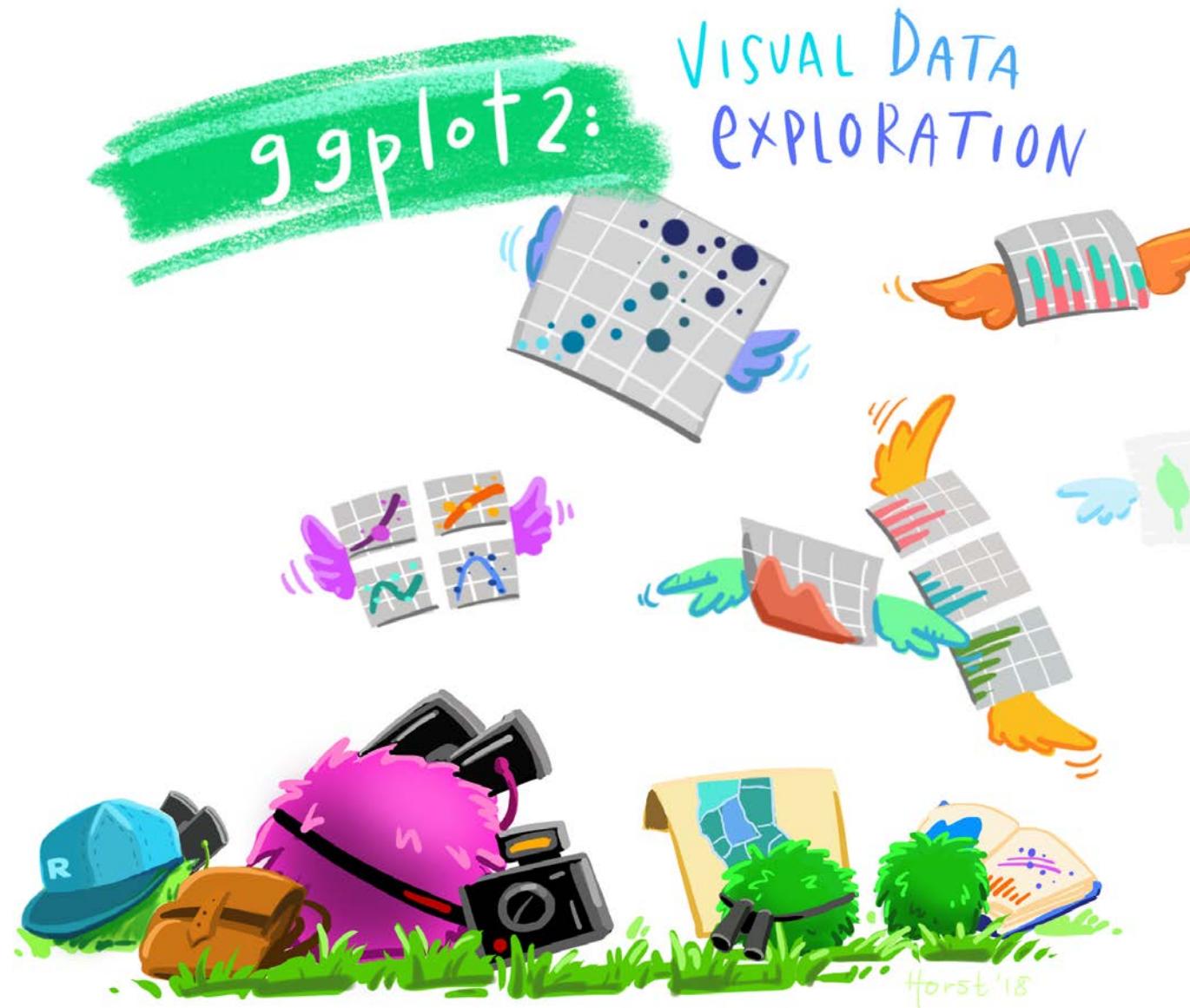
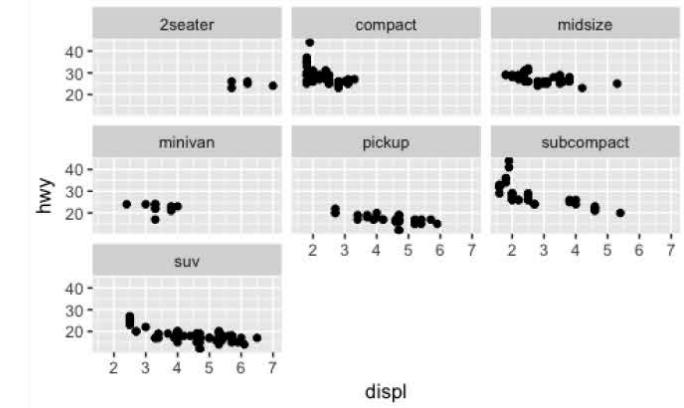
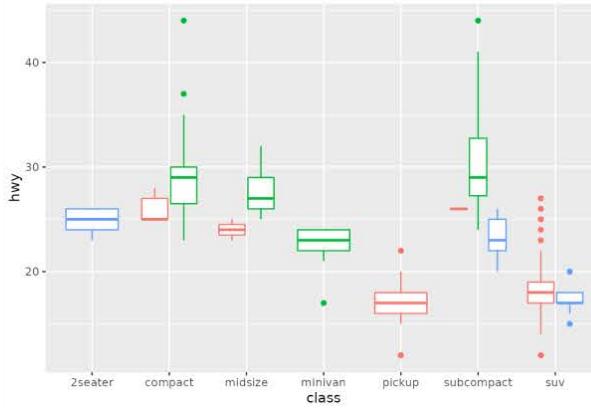
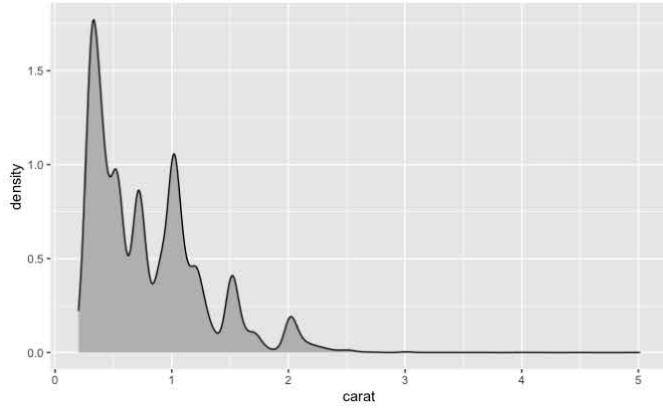
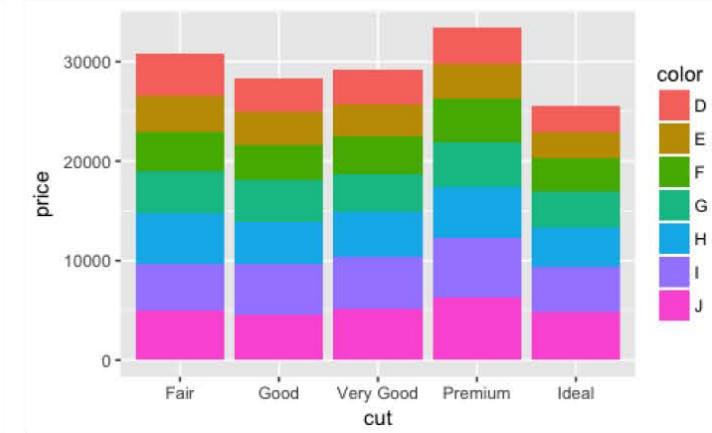
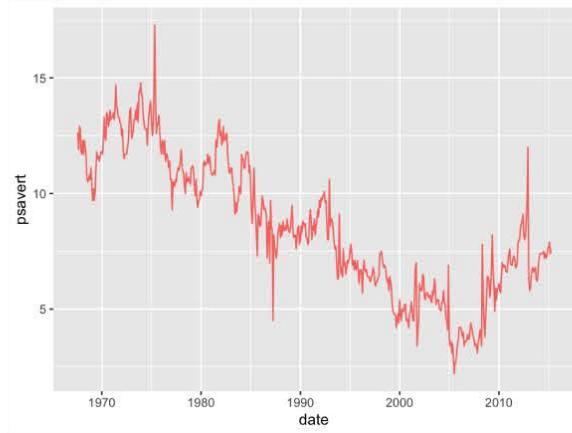
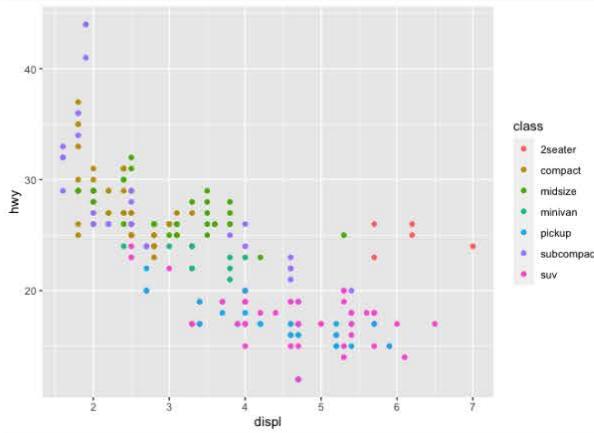
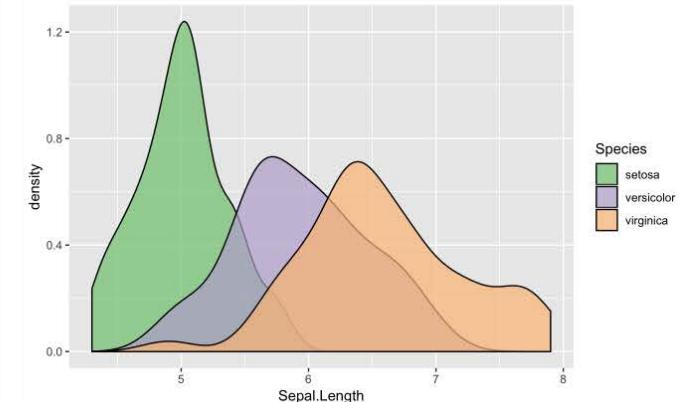
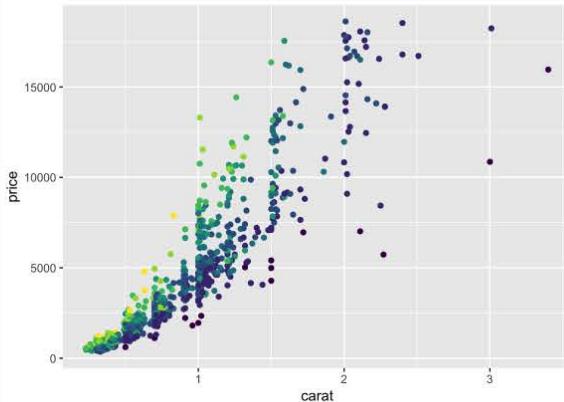
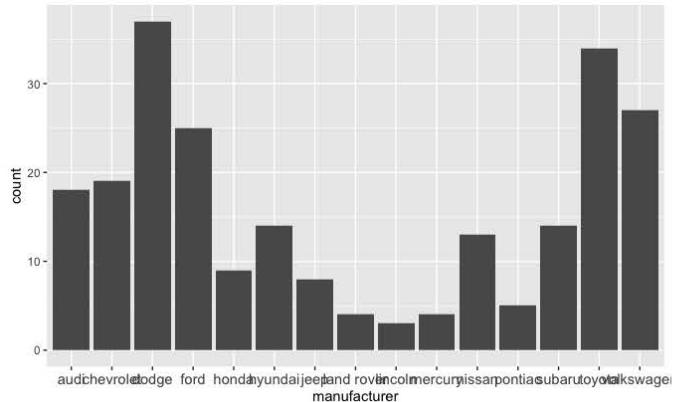


Illustration by [Allison Horst](#)

Cédric Scherer // Hands-On Data Visualization with ggplot2: Concepts





ggplot2 Examples featured on ggplot2.tidyverse.org

Cédric Scherer // Hands-On Data Visualization with ggplot2: Concepts



ggplot2:

Build a data MASTERpiece



Illustration by [Allison Horst](#)

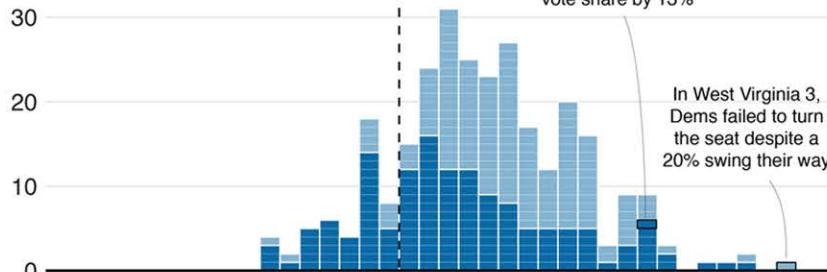
[Cédric Scherer](#) // Hands-On Data Visualization with ggplot2: Concepts



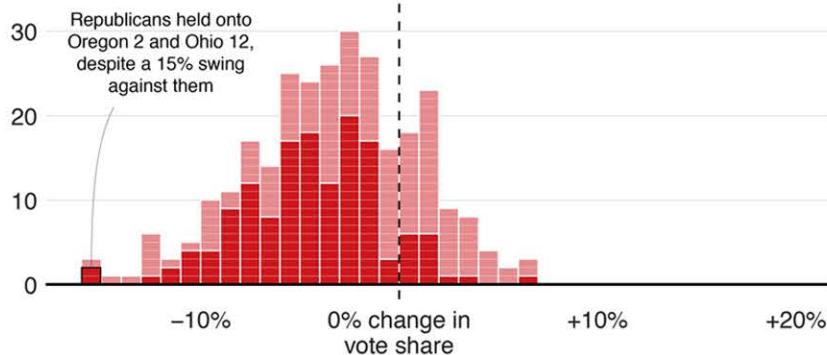
Blue wave

■ Won seat ■ Didn't win

Democrat candidates



Republican candidates

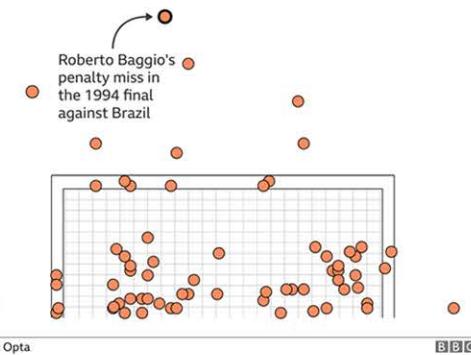


Source: AP, 19:01 ET

BBC

Where penalties are saved

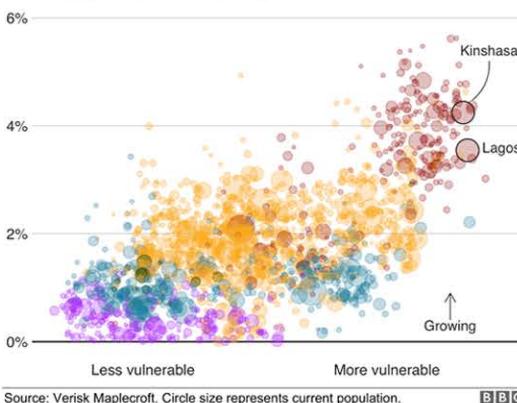
World Cup shootout misses and saves, 1982-2014



Fast-growing cities face worse climate risks

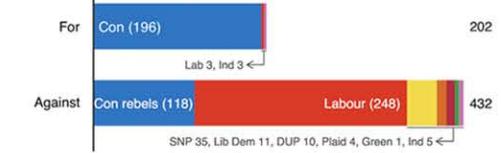
Population growth 2018-2035 over climate change vulnerability

■ Africa ■ Asia ■ Americas ■ Europe ■ Oceania



Source: Verisk Maplecroft. Circle size represents current population.

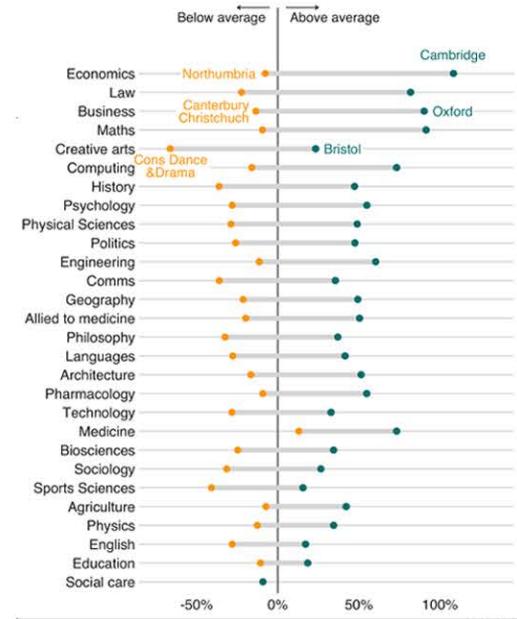
MPs rejected Theresa May's deal by 230 votes



Source: Commons Votes Services. Excludes 'tellers', the Speaker and deputies BBC

Earnings vary across units even within subjects

Impact on men's earnings relative to the average degree



Source: Institute for Fiscal Studies

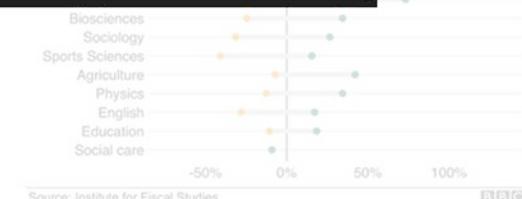
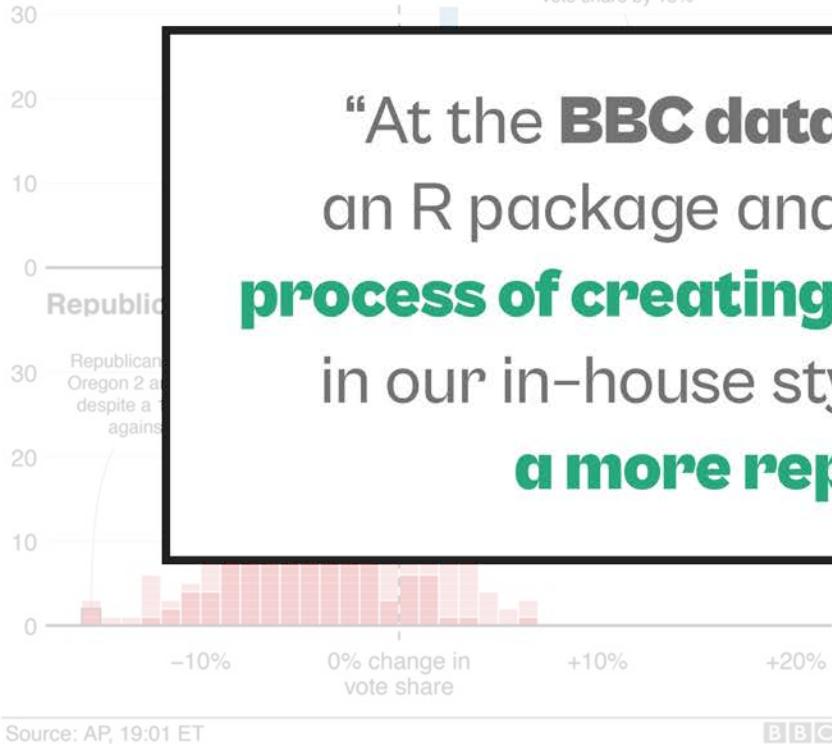


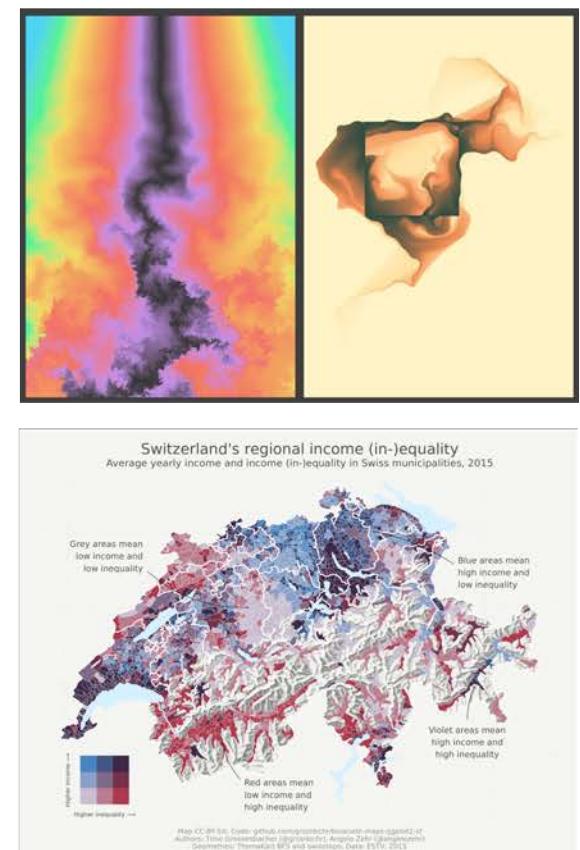
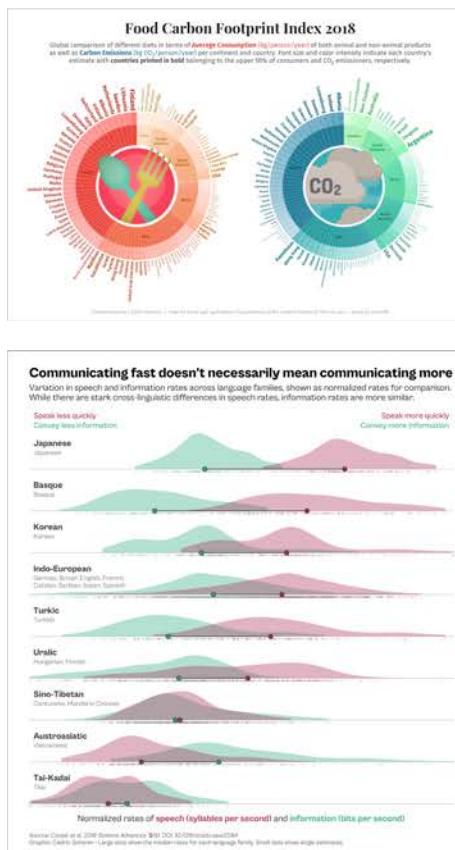
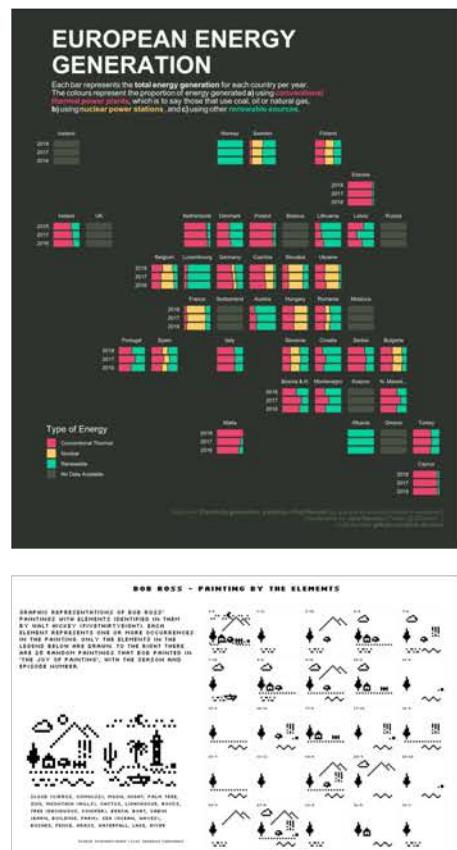
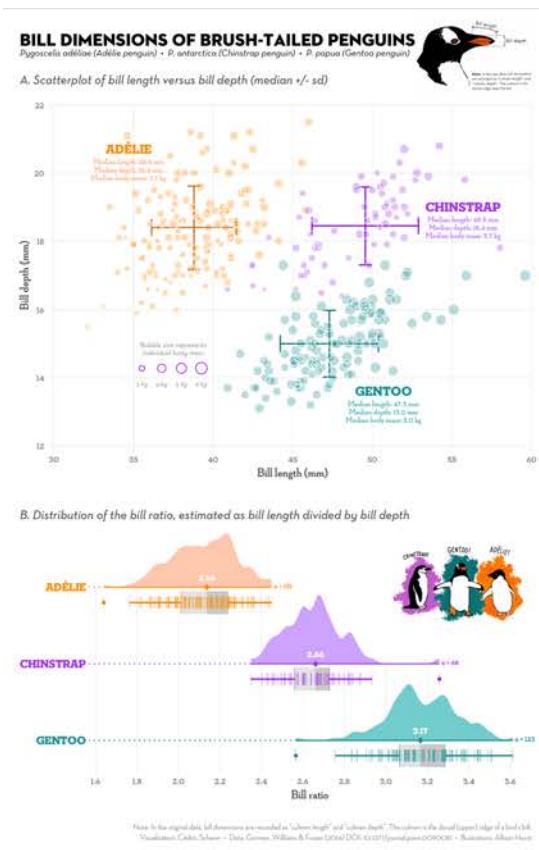
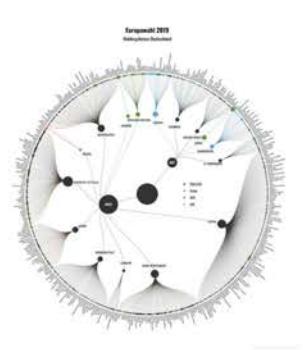
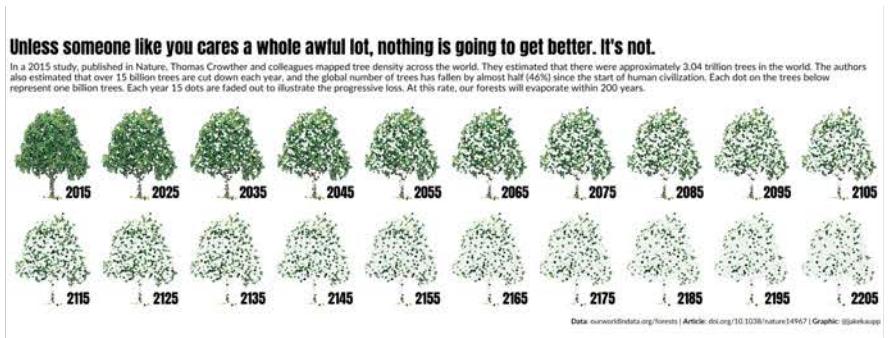
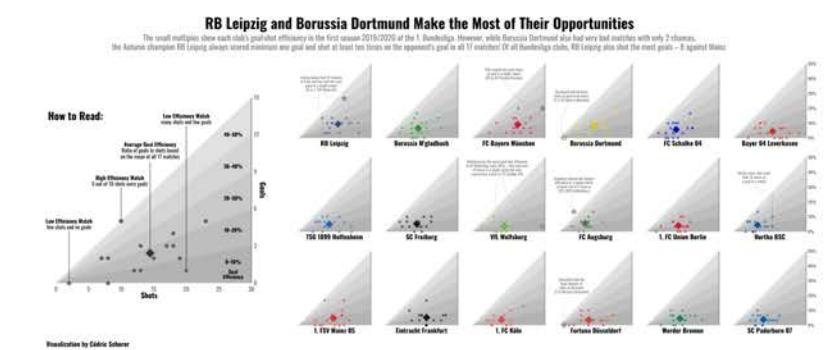
Blue wave

Democrat candidates



“At the **BBC data team**, we have developed an R package and an R package to make the **process of creating publication-ready graphics** in our in-house style using R’s ggplot2 library **a more reproducible process.**”

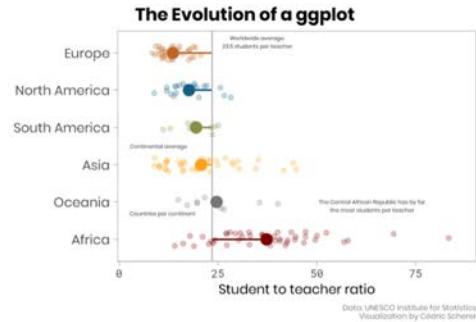
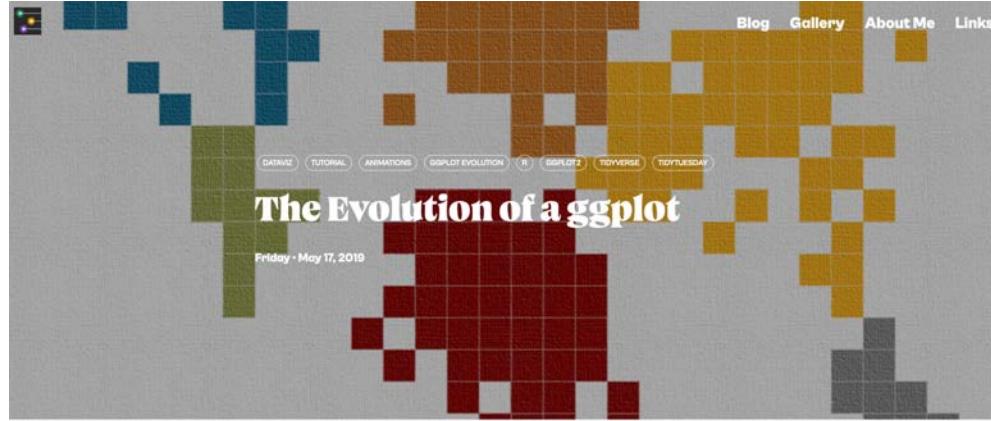




Selection of visualizations created 100% with ggplot2 by Thomas Linn Pedersen,
Georgios Karamanis, Timo Gossenbacher, Torsten Sprengler, Jake Kaupp, Jack Davison, and myself.

Cédric Scherer // Hands-On Data Visualization with ggplot2: Concepts





❖ Aim of this Tutorial

In this blog post, I aim to show you how to turn a default ggplot into a plot that visualizes information in an appealing and easily understandable way. The goal is to provide a step-by-step tutorial explaining how my visualization evolves from a typical basic ggplot. All plots are created with 100% `ggplot2` and 0% Inkscape.

We are going to transform a basic box plot into a compelling and self-explanatory combination of a jittered dot strip plot and a lollipop plot. I am going to use `data` provided by the UNESCO on global student to teacher ratios that was selected as





Header image by [Richard Strozyński](#)

Last update: 2023-05-05

Introductory Words

[I don't care, just show me the content!](#)

Back in 2016, I had to prepare my PhD introductory talk and I started using [\[ggplot2\]](#) to visualize my data. I never liked the syntax and style of base plots in R, so I was quickly in love with ggplot. Especially useful was its faceting utility. But because I was short on time, I plotted these figures by trial and error and with the help of lots of googling. The resource I came always back to was a blog entry called [Beautiful plotting in R: A ggplot2 cheatsheet](#) by Zev Ross, updated last in January 2016. After giving the talk which contained some decent plots thanks to the blog post, I decided to go through this tutorial step-by-step. I learned so much from it and directly started modifying the codes and over the time I added additional code snippets, chart types and resources.

Since the blog entry by Zev Ross was not updated for some years and step by step this became a unique version of a tutorial, I decided to host the updated version on my GitHub. Now it finds its proper place on this homepage! (Plus I added a ton of other updates—just to name a few: The fantastic [\[patchwork\]](#), [\[eggtext\]](#) and





2-Day Workshop on "Graphic Design with ggplot2" at rstudio::conf 2022

Tuesday - August 9, 2022

End of July, I had the honor to teach a 2-day, in-person workshop on "Graphic Design with ggplot2" at the rstudio::conf(2022) in Washington DC. Invited by RStudio (now named Posit), I developed a new course that covers the most important steps and helpful tips to create visually appealing, engaging and complex graphics with ggplot2. The course focused on the main concepts of the grammar of graphics and used hands-on examples to explore ggplot2's utility to create multi-layered, more complex graphs.
[All course resources are available as open-source material on the course page.](#)



The course webpage as well as the slide decks and the exercises and solutions were developed with the new open-source scientific and technical publishing system [Quarto](#). The new workshop development was a perfect opportunity to give it a try and the experience was overall wonderful—the [reveal.js](#) integration for the slides works perfect and allows for a lot of customization. Thanks to Marco Scilani for helping me setting up the course webpage which was, after learning about a few quirks, a smooth experience as well.



Cédric Scherer // Hands-On Data Visualization with ggplot2: Concepts



Screenshot of Cédric Scherer's GitHub profile page.

Profile Picture: A circular portrait of Cédric Scherer, a man with a beard and short hair, wearing a black t-shirt.

Name: Cédric Scherer

Handle: z3tt

Bio: Data Visualization Specialist • Computational Ecologist

Followers: 1.5k followers • 126 following

Location: Self-Employed • IZW Berlin, Berlin

Links: cedricscherer.com, @CedScherer

Achievements: YOLO (x3), 🎉 (x3), 🎊 (x2), 🎁 (x2)

Organizations: (Icons for various organizations)

GitHub Activity: A pie chart titled "My GitHub Activity aka 'How it feels to be a DataViz coder'" showing the distribution of commits:

- Commits trying to make *juicy interfaces* work: Yellow
- Commits that actually contain code changes: Green
- Commits fixing *-typographical-errors*: Green

What I Actually Do: A section titled "What do you need a DataViz coder?" listing services:

- I design effective graphics for custom needs
- I help you to find a suitable chart and design for your data visualization needs effectively
- using an efficient code-first approach
- I help you to create script-based workflows to automate reporting and easily update graphical

Pinned:

- TidyTuesday** Public: Contributions to the #TidyTuesday challenge.
- OutlierConf2021** Public: Slides and hands-on codes for my talk "ggplot Wizardry: My Favorite Tricks and Secrets for Beautiful Plots in R" at the 1st OutlierConf. February 4-7 2021.

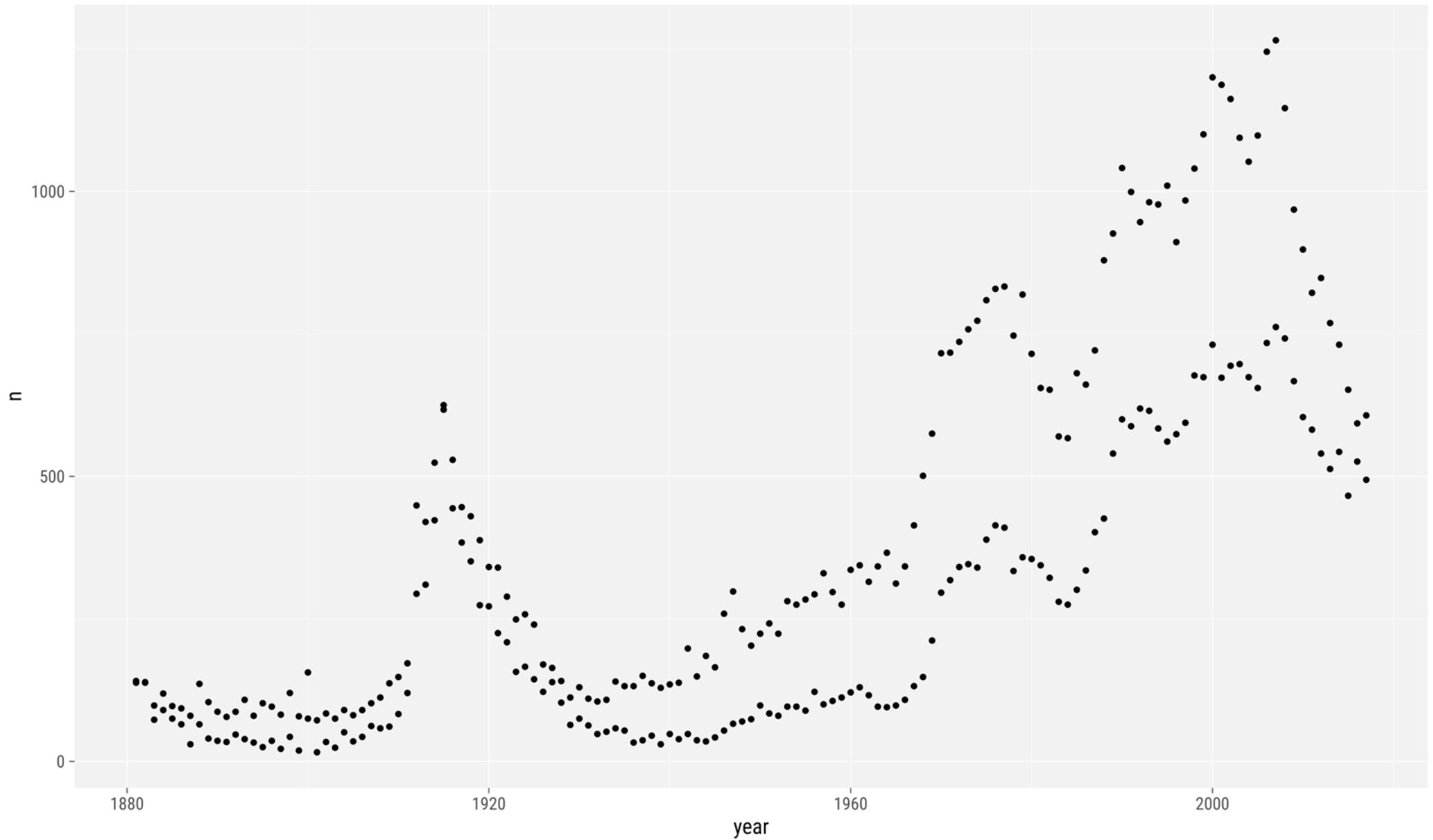
Customize your pins

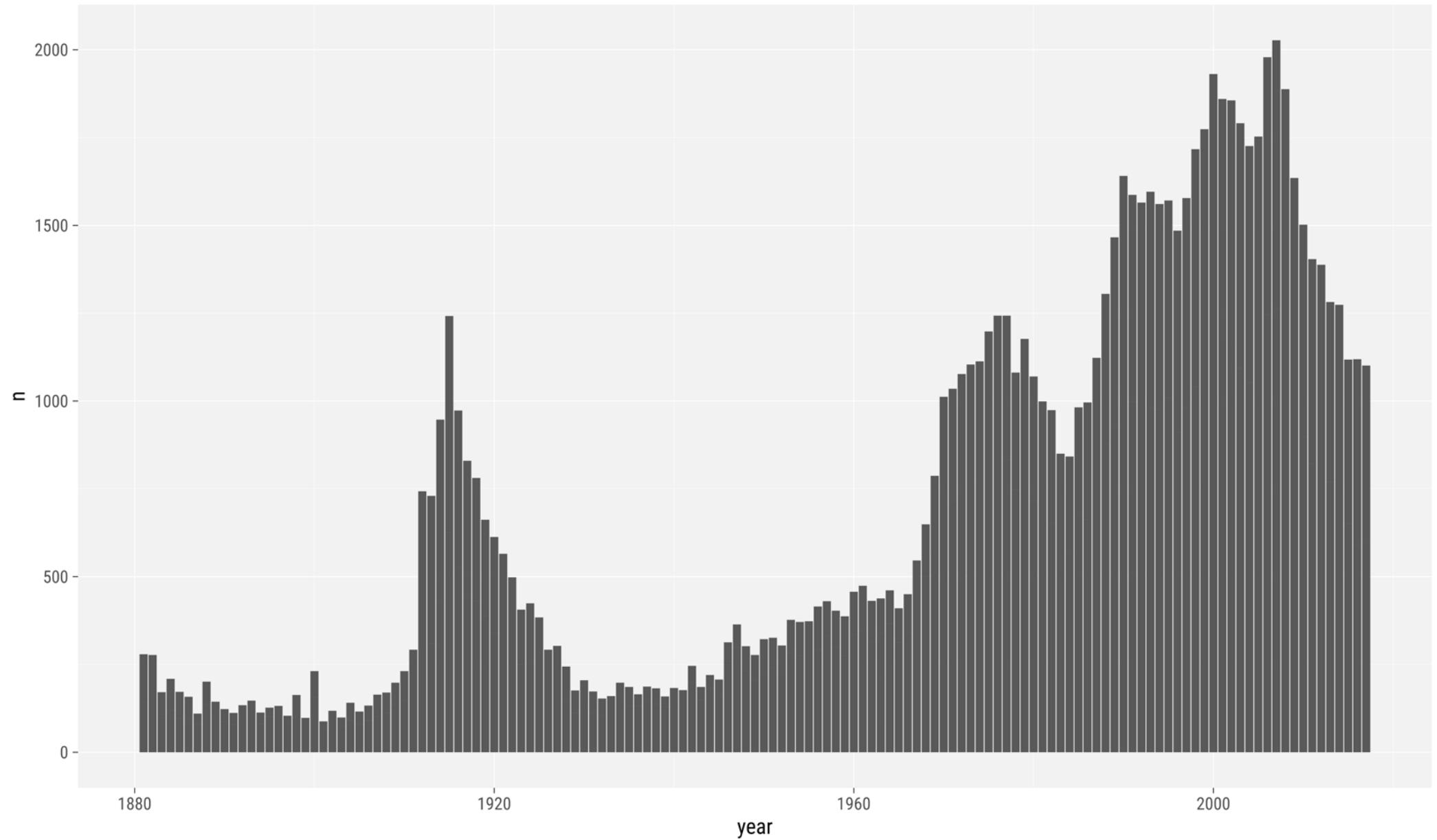
Cédric Scherer // Hands-On Data Visualization with ggplot2: Concepts

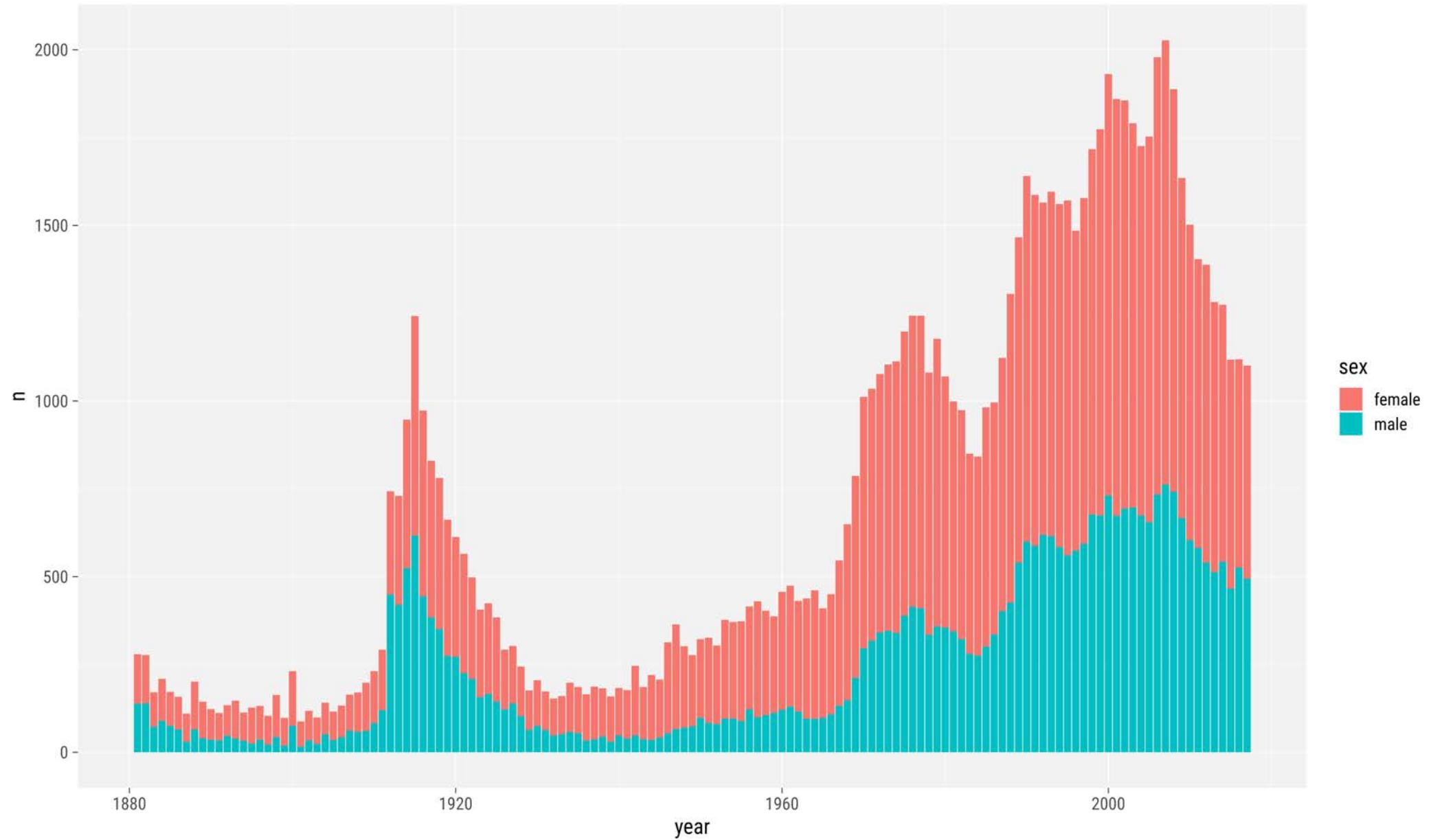


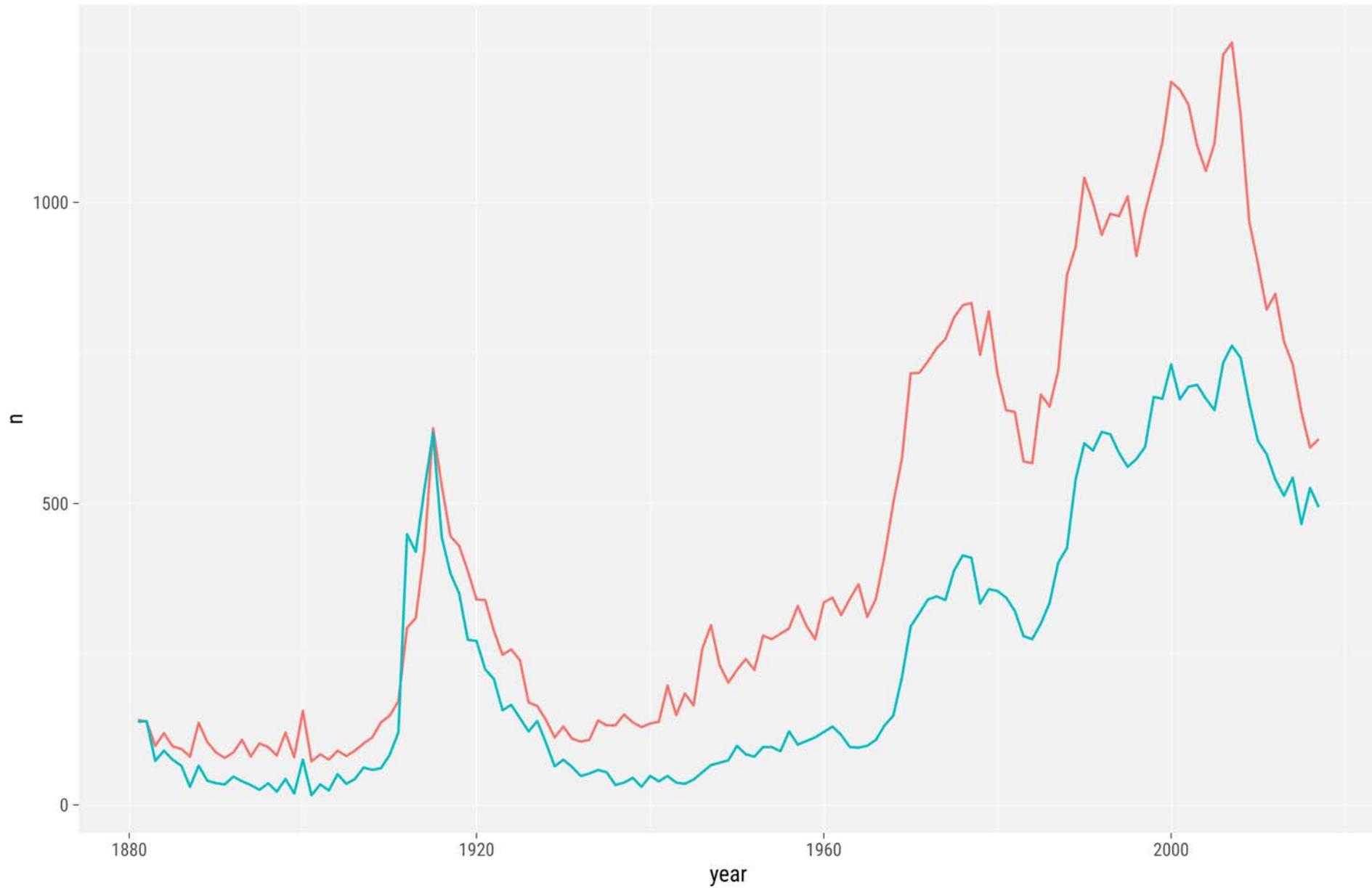
— A Motivational Example —

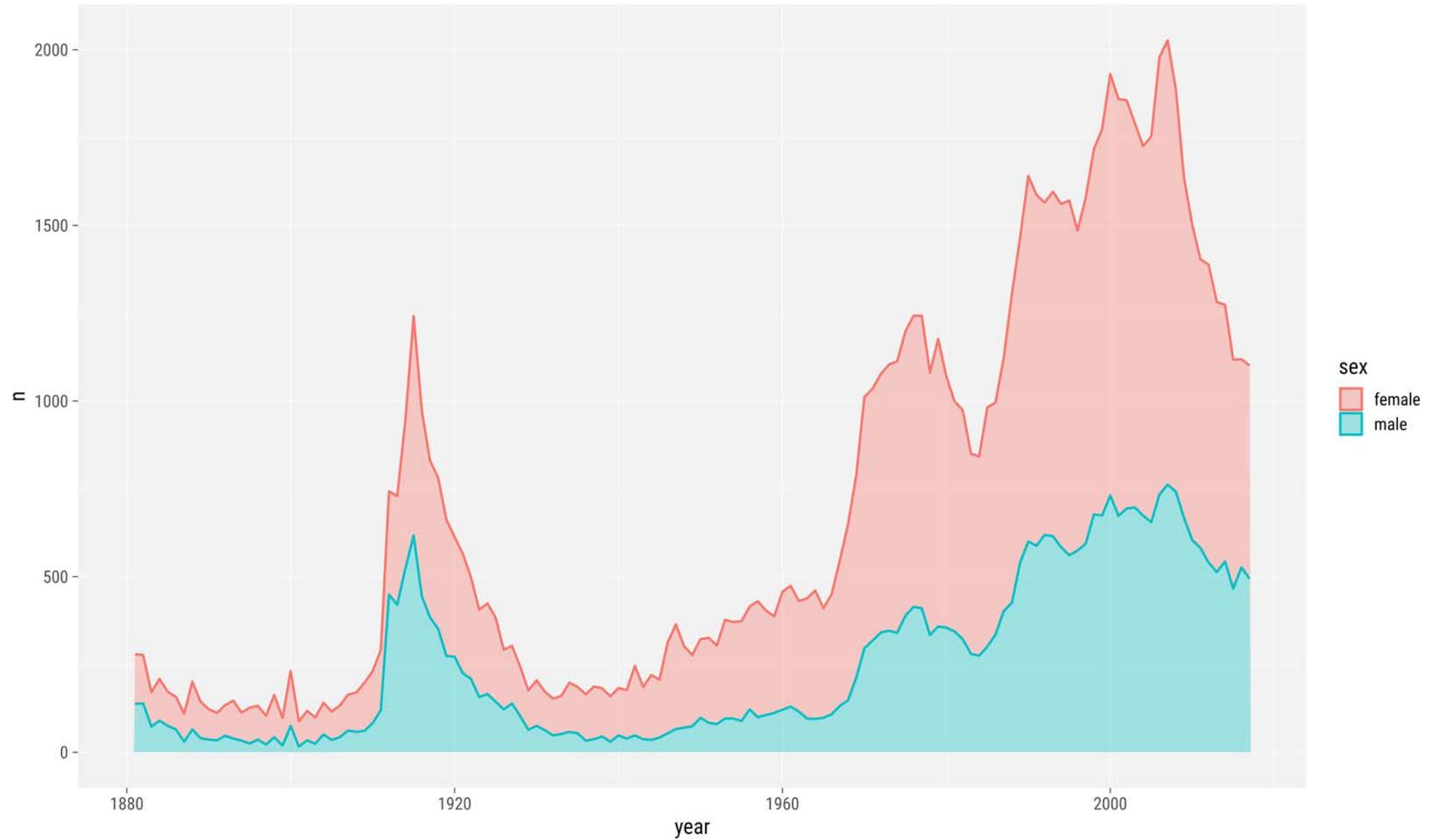


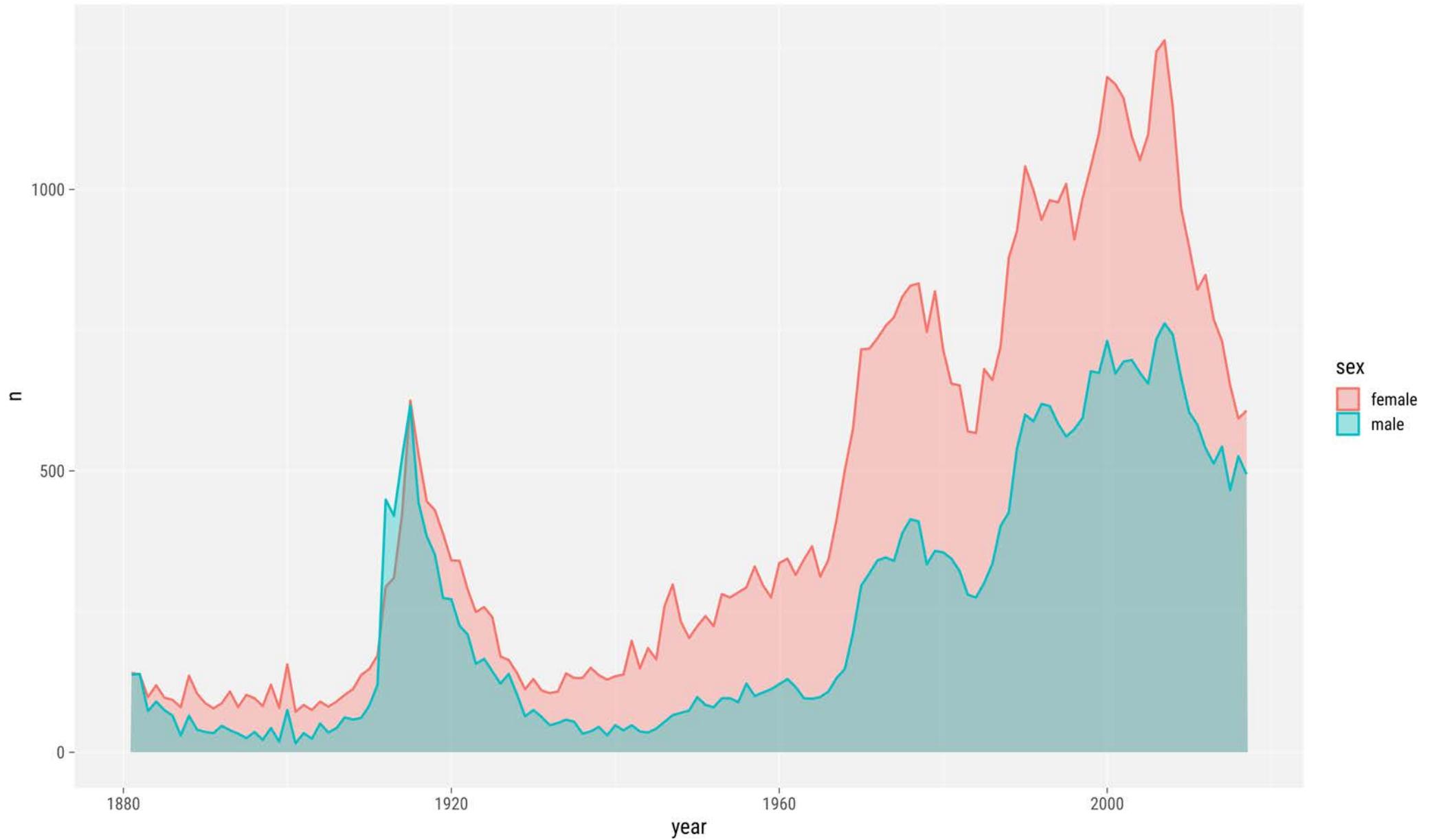




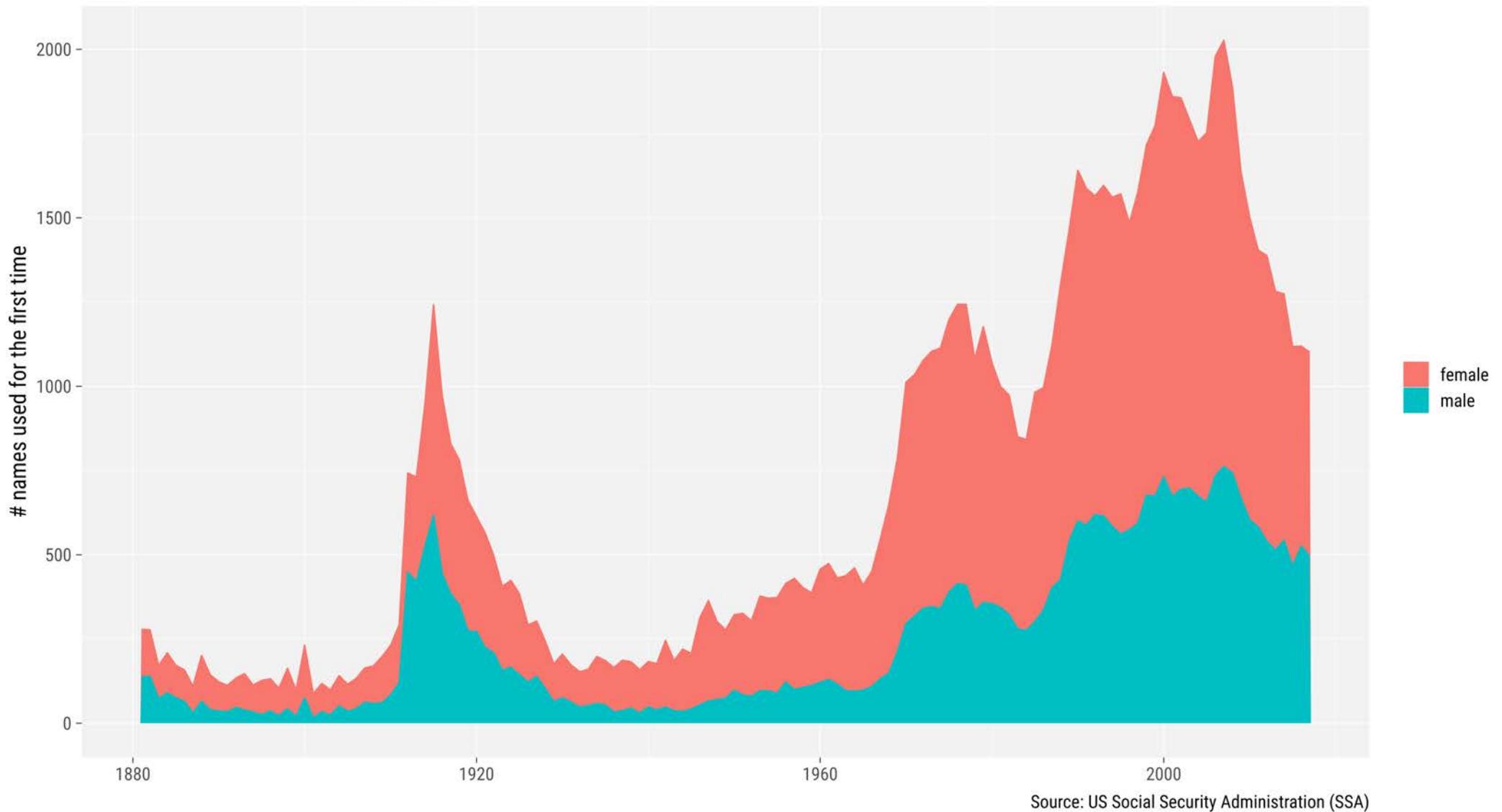








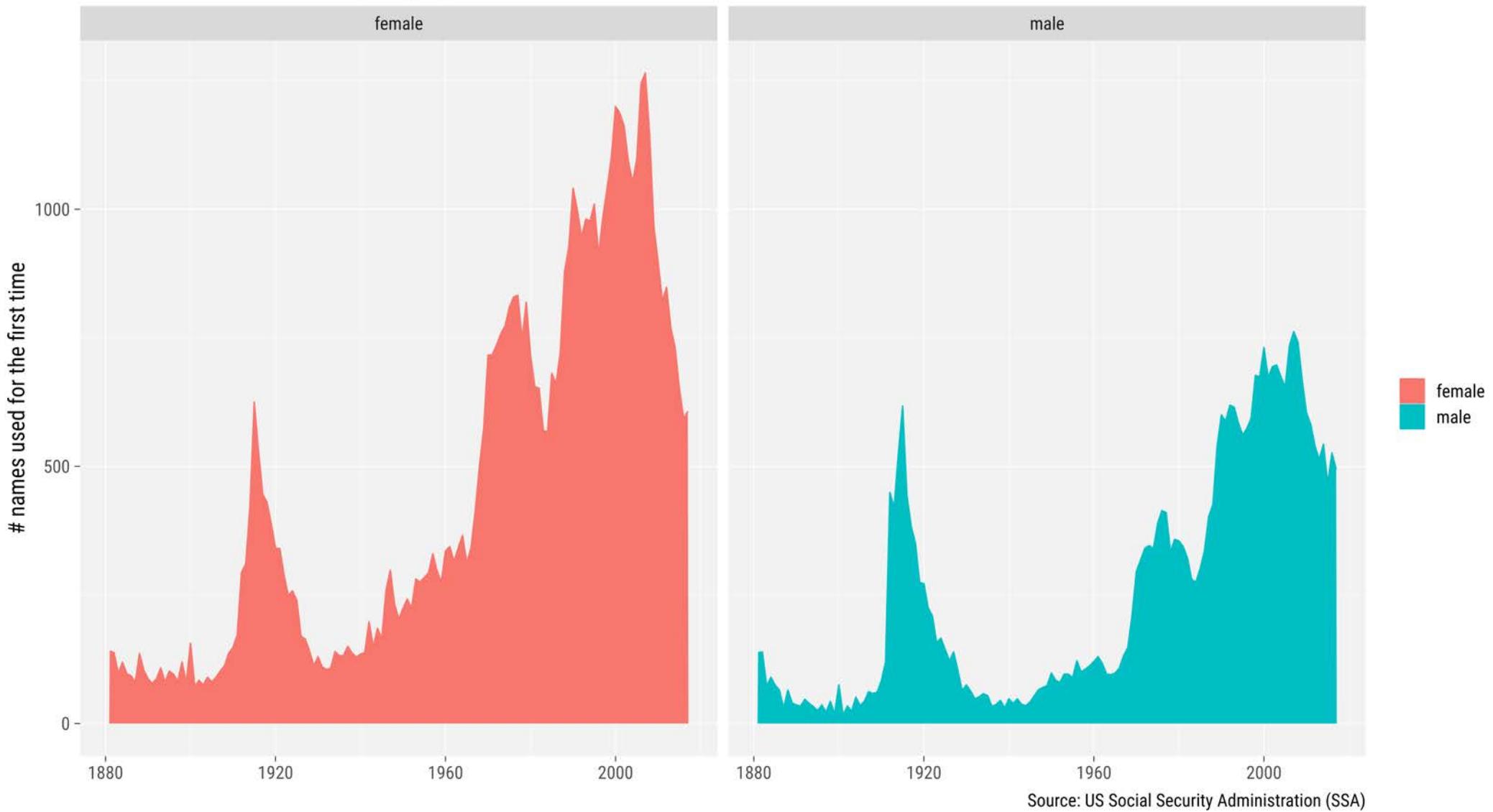
Every year, hundreds of babies in the US get a name that has not been given before
... but those numbers are mostly dropping since 2008 for both sexes



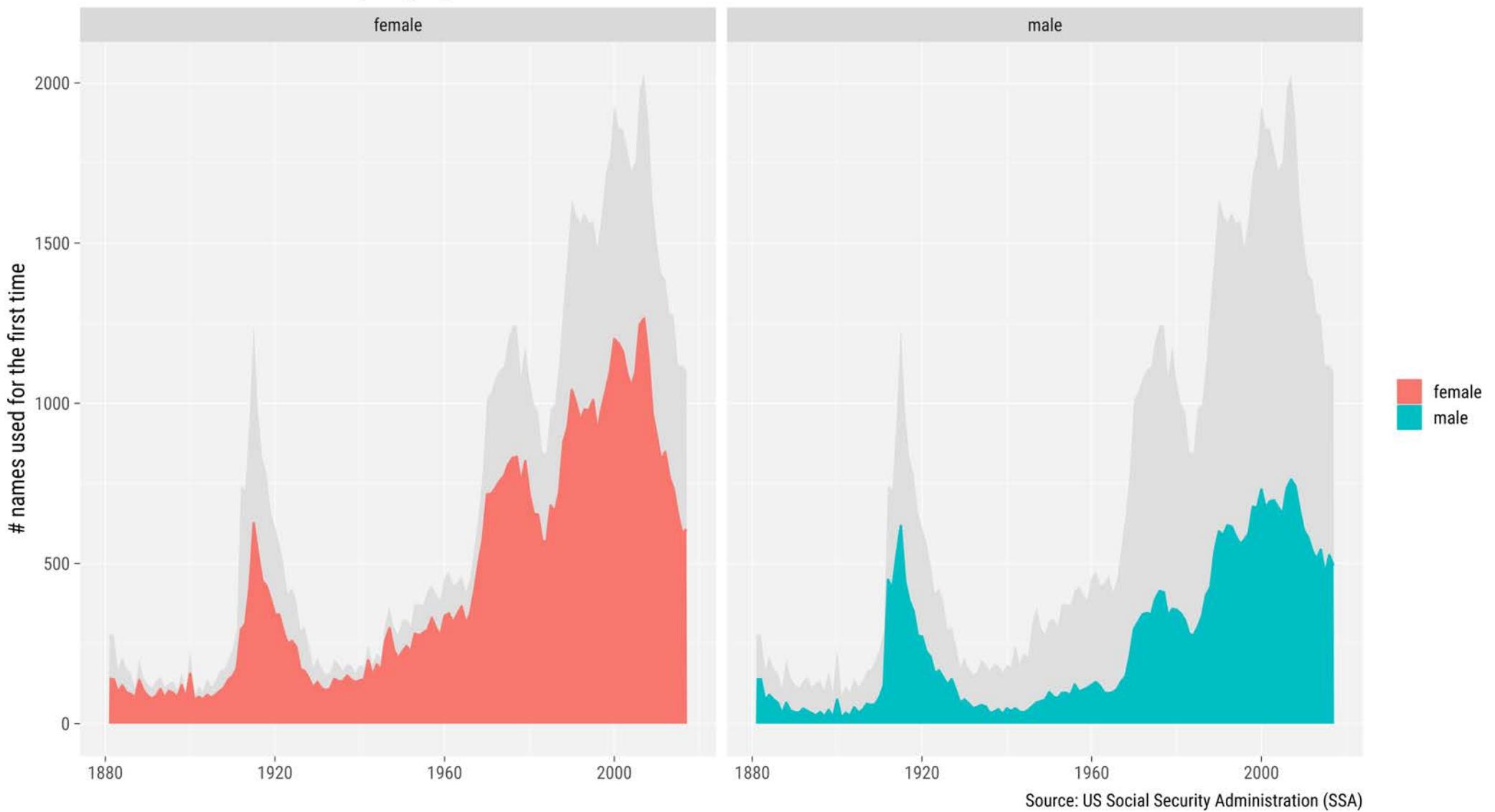
Source: US Social Security Administration (SSA)



Every year, hundreds of babies in the US get a name that has not been given before
... but those numbers are mostly dropping since 2008 for both sexes



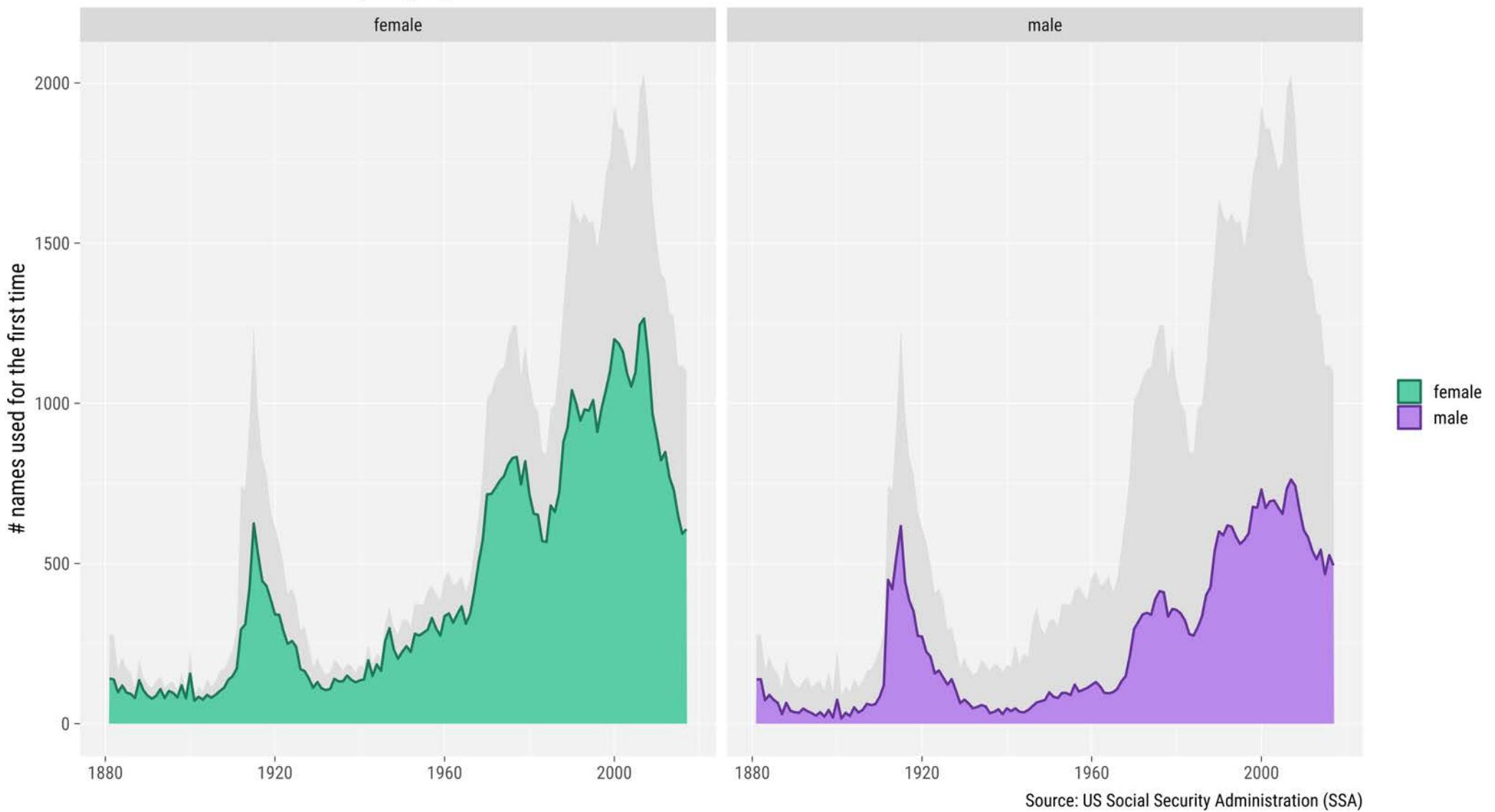
Every year, hundreds of babies in the US get a name that has not been given before
... but those numbers are mostly dropping since 2008 for both sexes



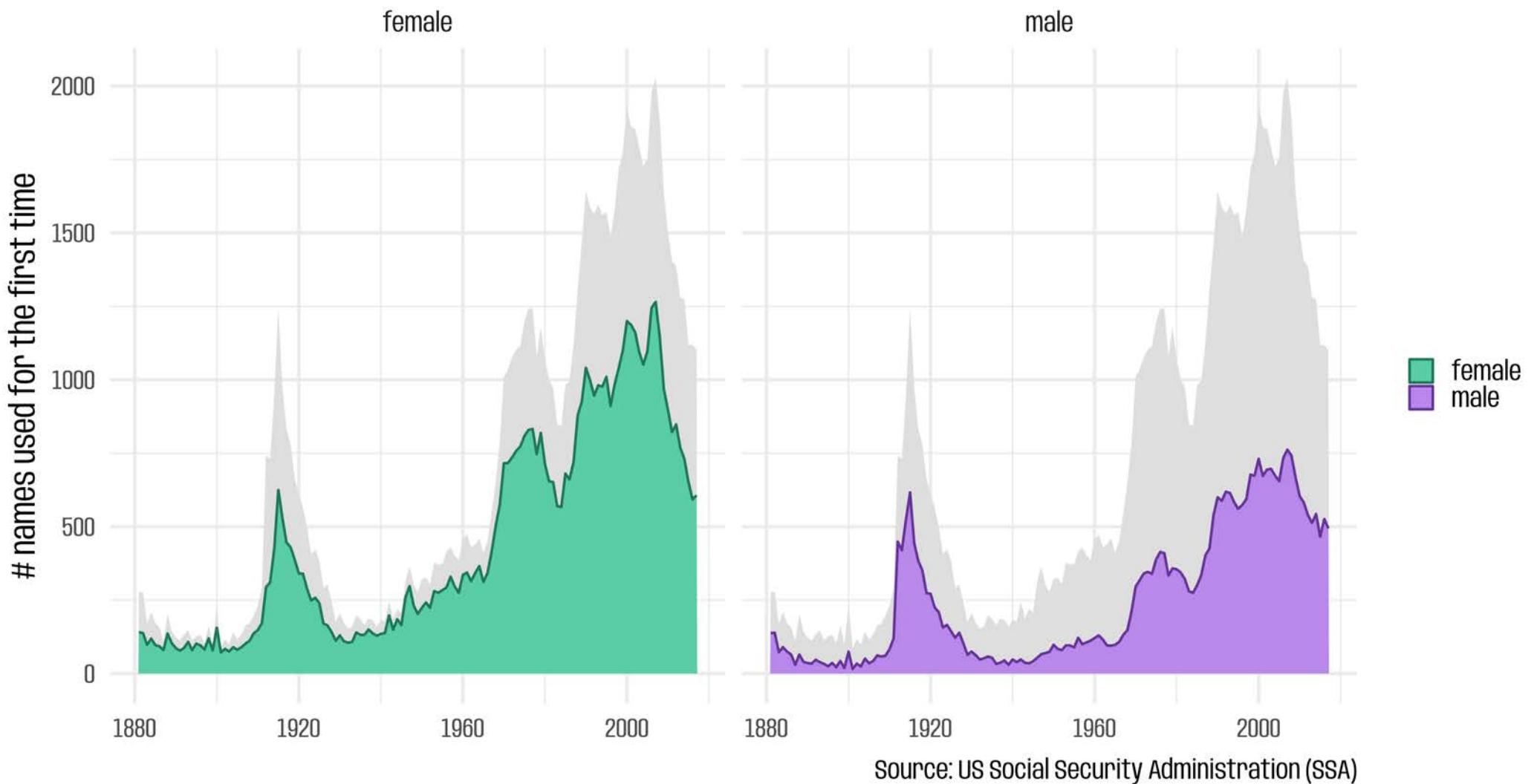
Source: US Social Security Administration (SSA)



Every year, hundreds of babies in the US get a name that has not been given before
... but those numbers are mostly dropping since 2008 for both sexes



Every year, hundreds of babies in the US get a name that has not been given before
... but those numbers are mostly dropping since 2008 for both sexes

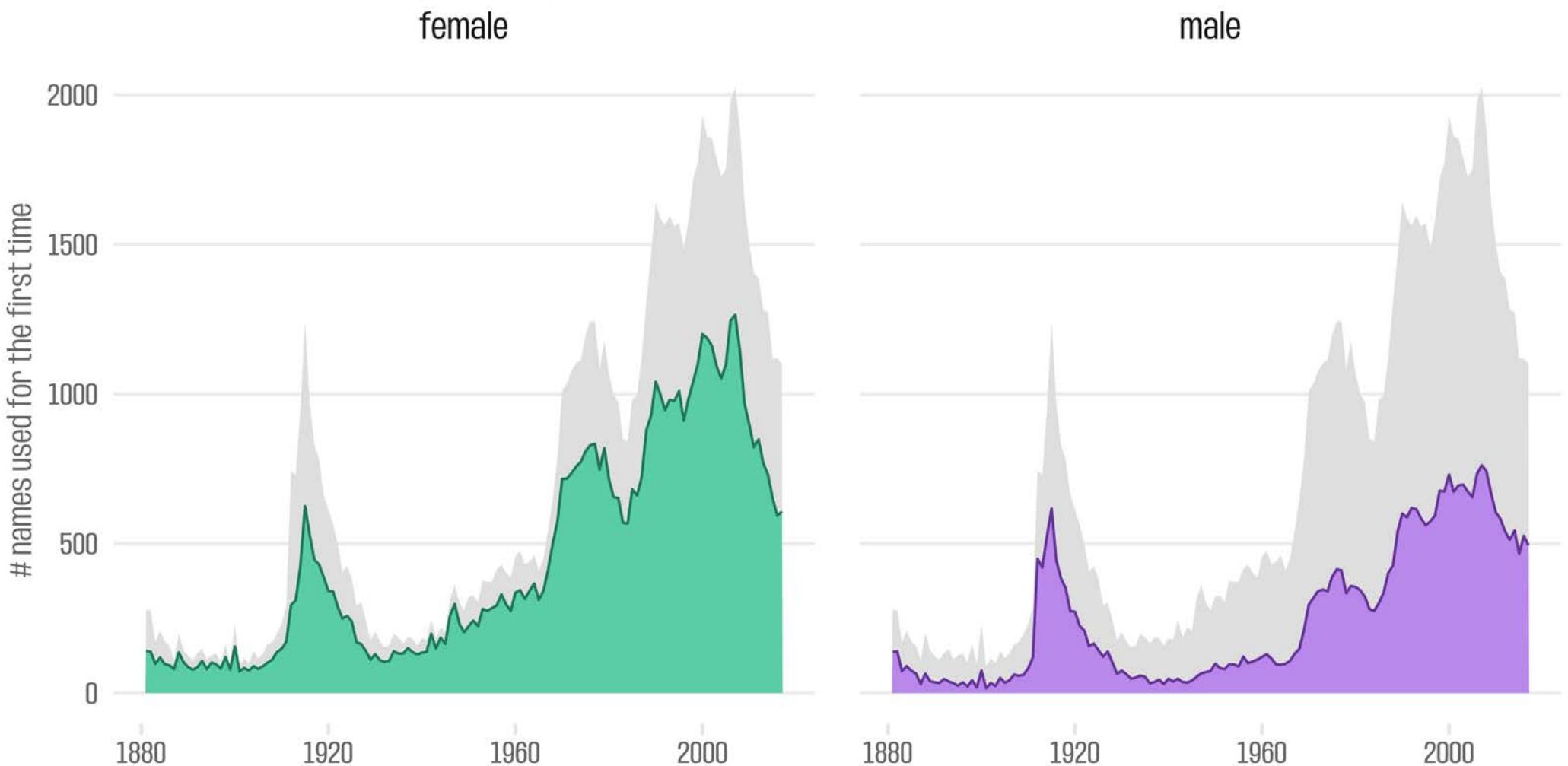


Source: US Social Security Administration (SSA)



Every year, hundreds of babies in the US get a name that has not been given before

... but those numbers are mostly dropping since 2008 for both sexes

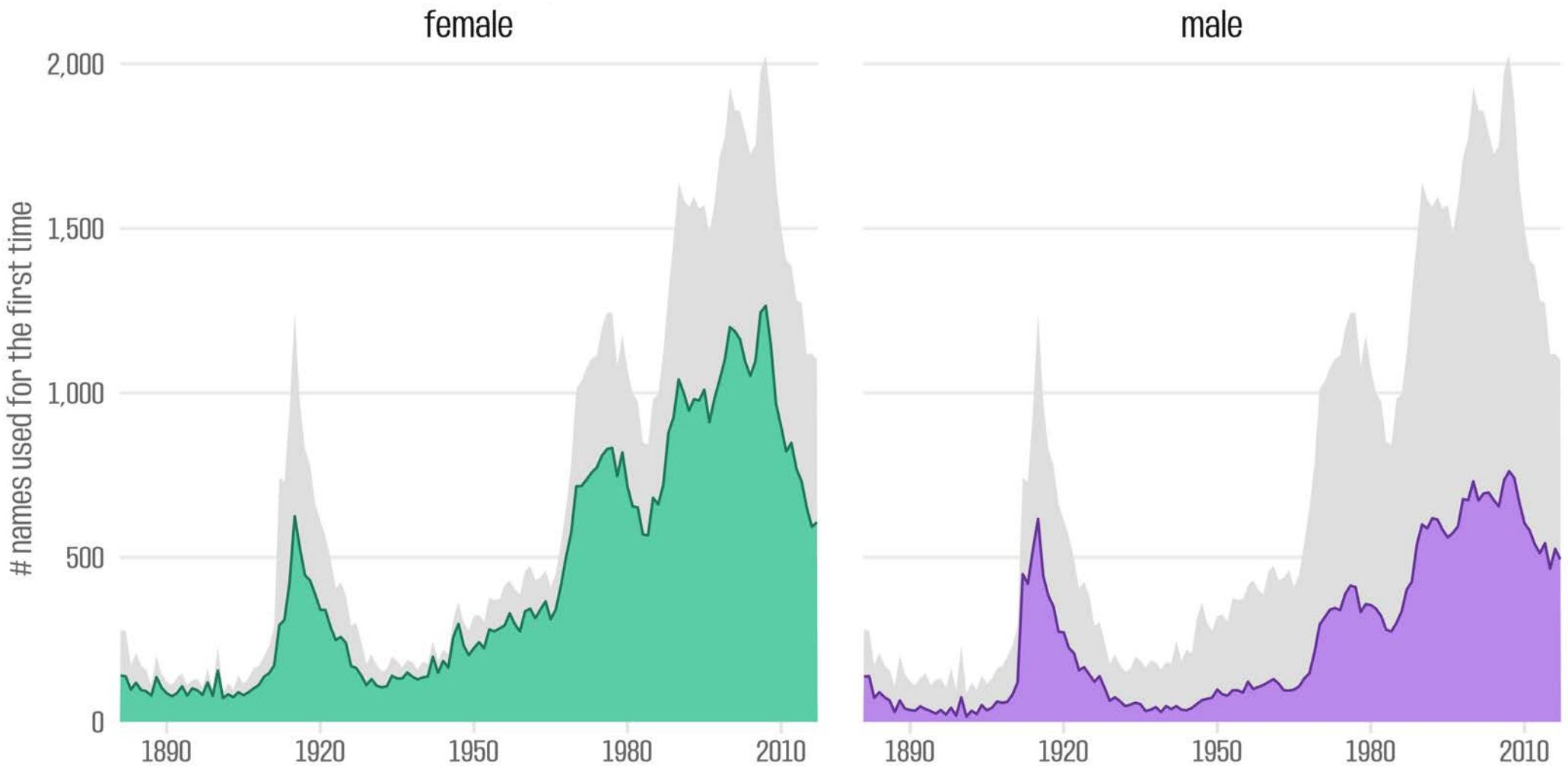


Source: US Social Security Administration (SSA)



Every year, hundreds of babies in the US get a name that has not been given before

... but those numbers are mostly dropping since 2008 for both sexes

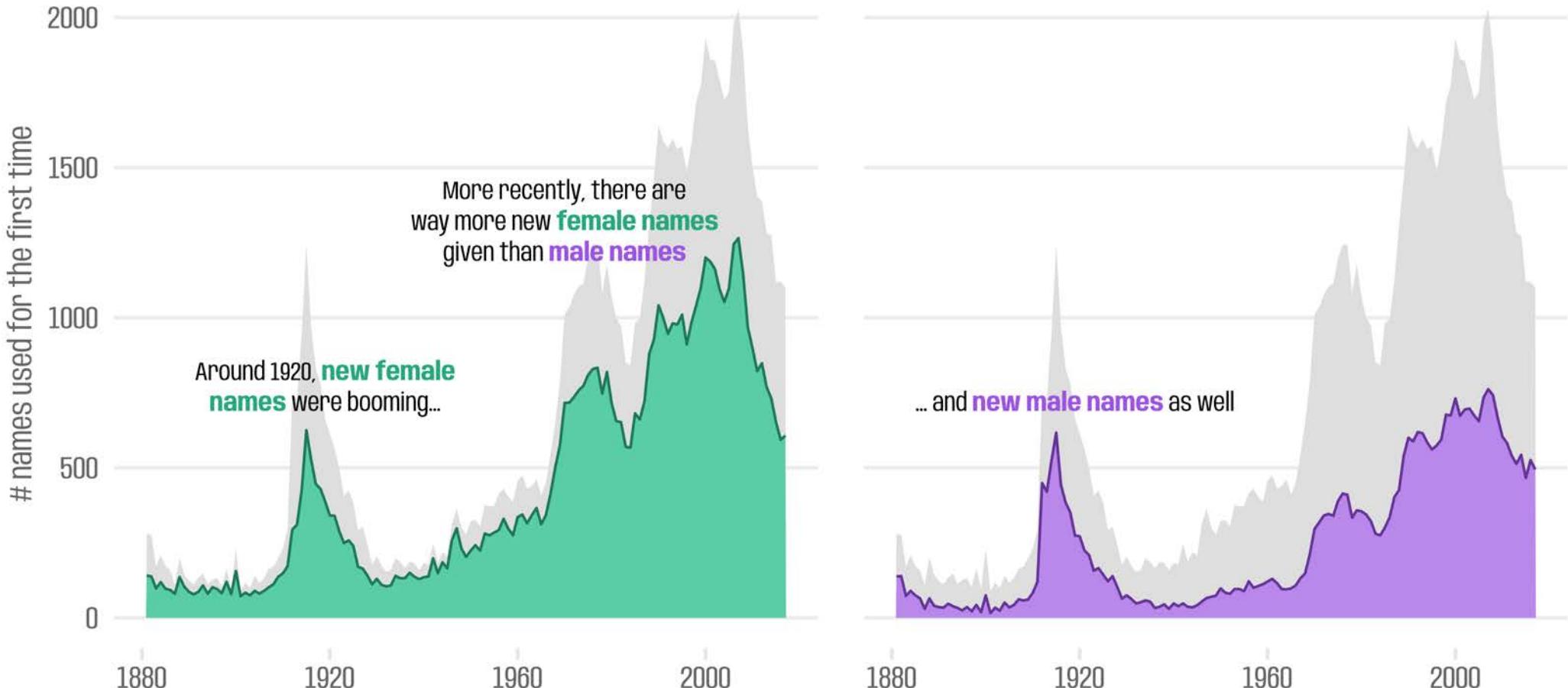


Source: US Social Security Administration (SSA)



Every year, hundreds of babies in the US get a name that has not been given before

... but those numbers are mostly dropping since 2008 for both sexes



Source: US Social Security Administration (SSA)



— Setup —



R Engine



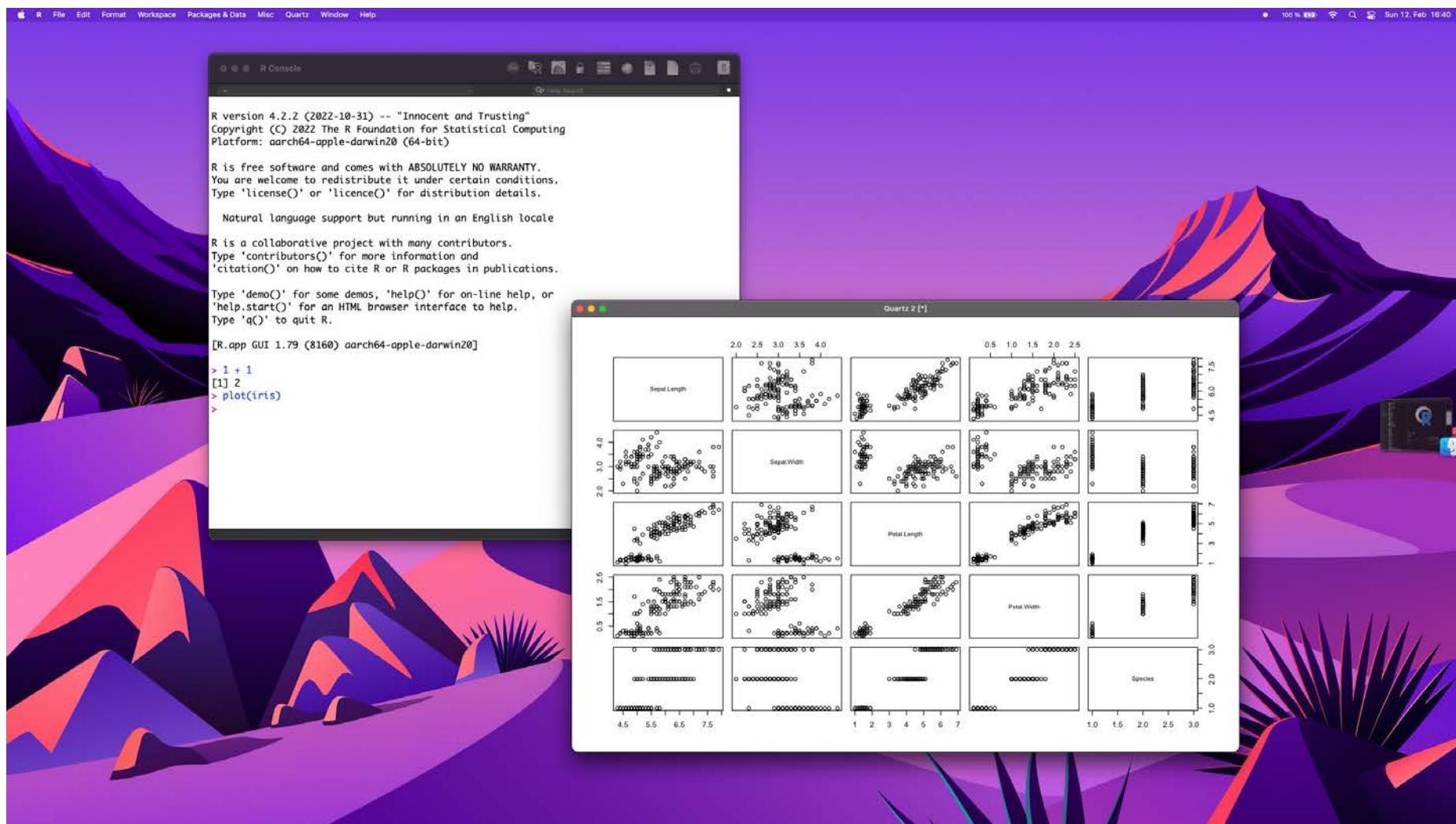
R Studio Dashboard

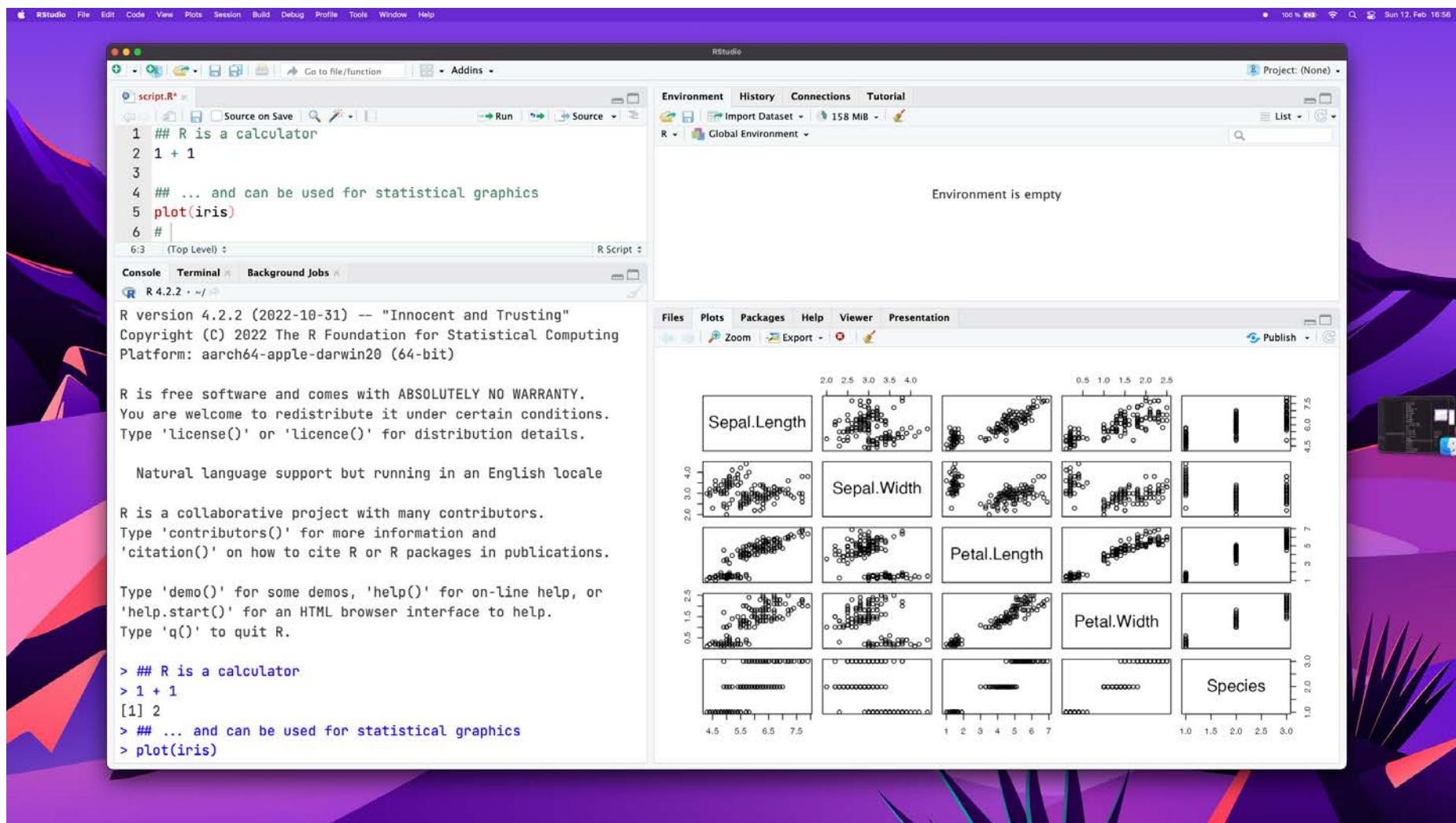


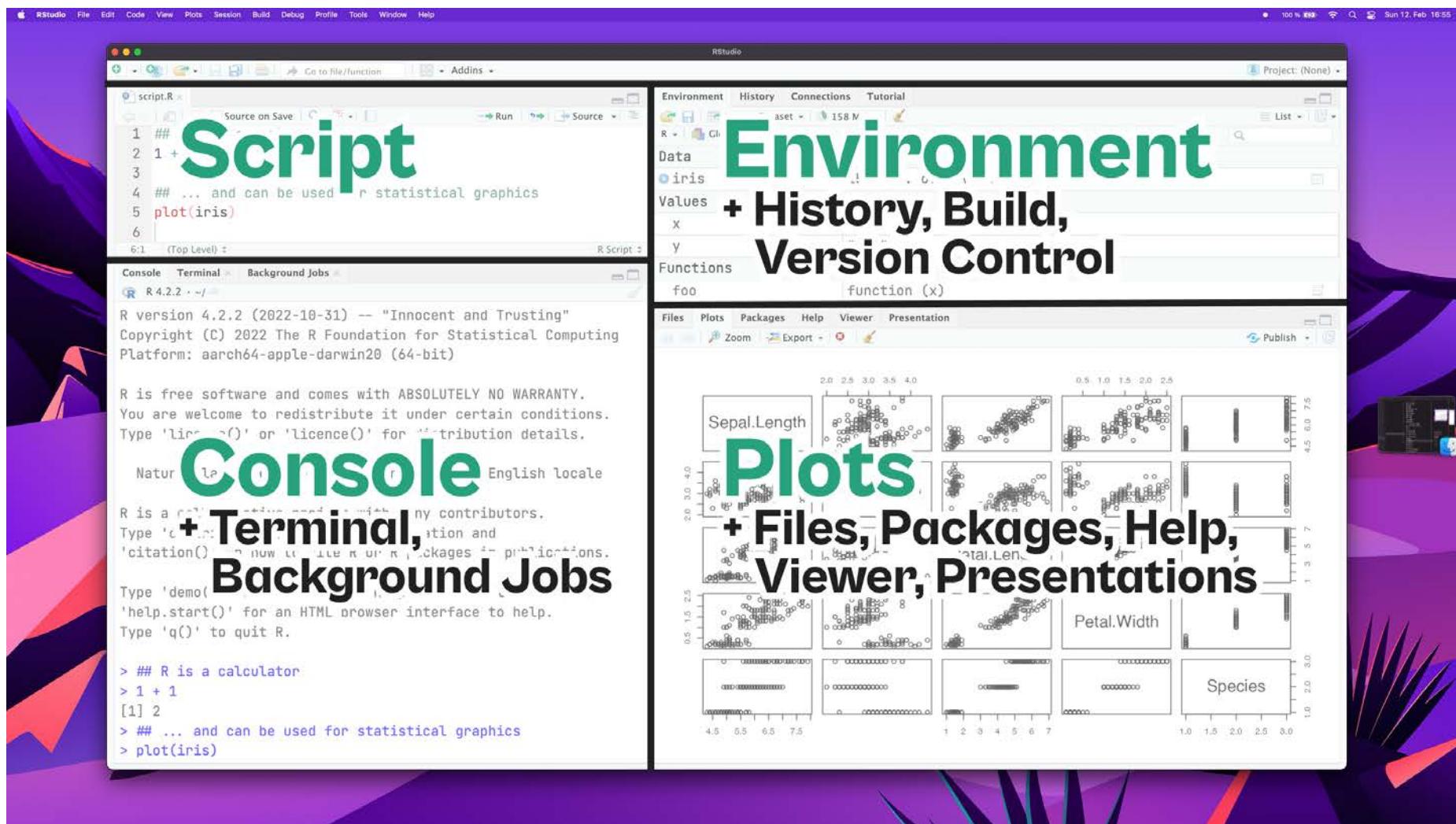
Modified from [ModernDive](#)

- **RStudio** is an open-source IDE (integrated development environment) for R
- many features + extensions to facilitate workflows (version control, toc, add-ins, ...)









Options

R General

-  Code
-  Console
-  Appearance
-  Pane Layout
-  Packages
-  R Markdown
-  Python
-  Sweave
-  Spelling
-  Git/SVN
-  Publishing
-  Terminal
-  Accessibility

Basic

Graphics

Advanced

Graphics Device

Backend: AGG

Antialiasing: (Default)

OK

Cancel

Apply



The ggplot2 Package

... is an **R package to visualize data** created by Hadley Wickham in 2005

```
1 library(ggplot2)
```

... can be loaded as part of the **{tidyverse}** package collection as well:

```
1 library(tidyverse)
```



Packages?!

A package is a **collection of functions, data, and other objects** that are designed to **perform specific tasks or solve certain problems**.

We need `{ggplot2}` but also a few other packages:

```
1 install.packages("ggplot2")
2 install.packages("dplyr")
3 install.packages("readr")
4 install.packages("forcats")
5 install.packages("stringr")
```

All of these packages are part of the `{tidyverse}` package collection:

```
1 install.packages("tidyverse")
```

We also need the following packages:

```
1 install.packages("scales")
2 install.packages("ragg")
3 install.packages("gapminder")
```



The Data

The famous **Gapminder** data set

- life expectancy, GDP per capita, and population by country
- contains complete data covering the years 1952-2007 in 5-year steps

```
1 library(gapminder)
```



The Data

The famous **Gapminder** data set

- life expectancy, GDP per capita, and population by country
- contains complete data covering the years 1952-2007 in 5-year steps

```
1 library(gapminder)
2 gapminder

# A tibble: 1,704 × 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.
# i 1,694 more rows
```



The Data

The famous **Gapminder** data set

- life expectancy, GDP per capita, and population by country
- contains complete data covering the years 1952-2007 in 5-year steps

```
1 gapminder <- readr::read_csv("./data/gapminder.csv")
```



The Data

The famous **Gapminder** data set

- life expectancy, GDP per capita, and population by country
- contains complete data covering the years 1952-2007 in 5-year steps

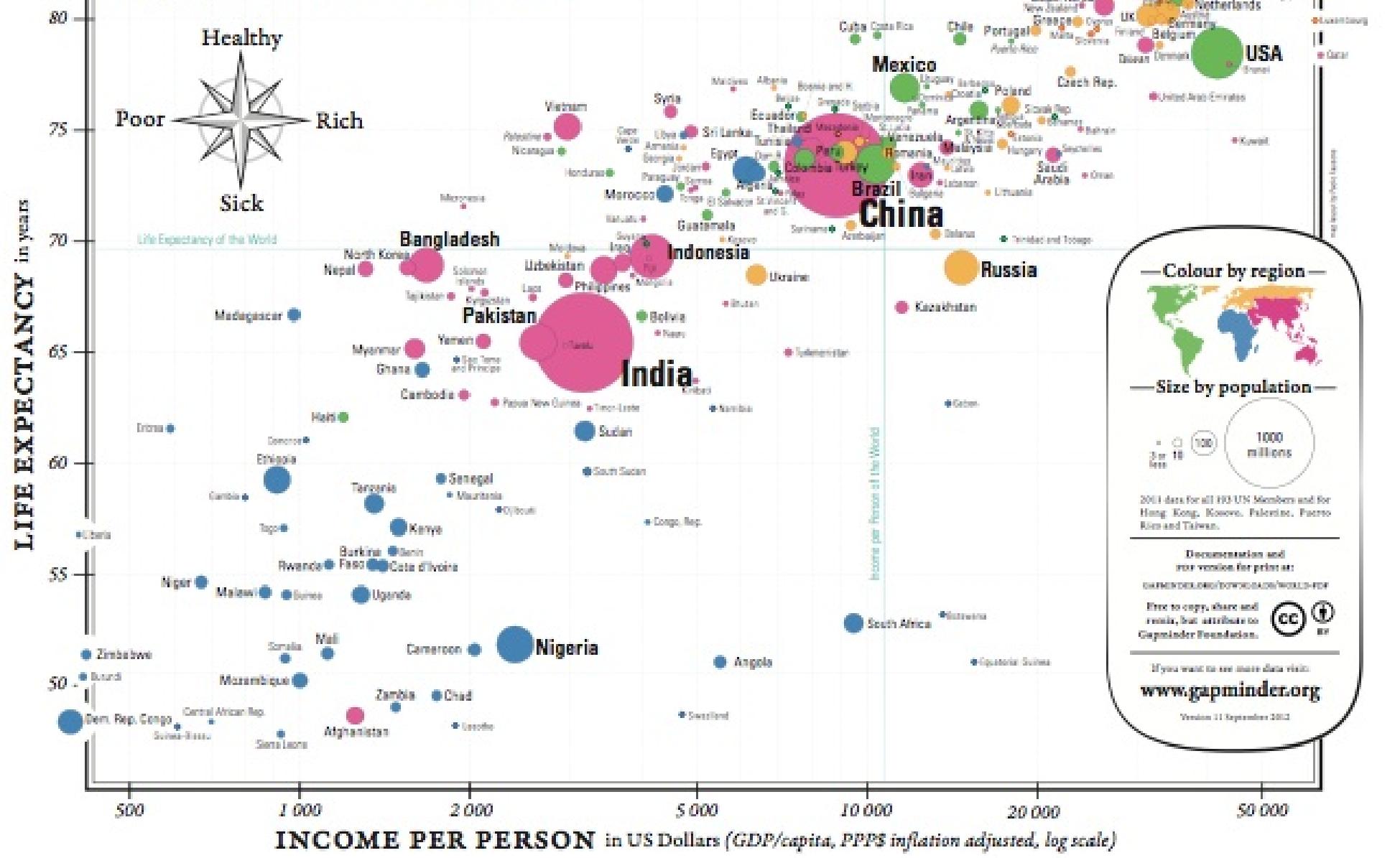
```
1 gapminder <- readr::read_csv("./data/gapminder.csv")
2 gapminder

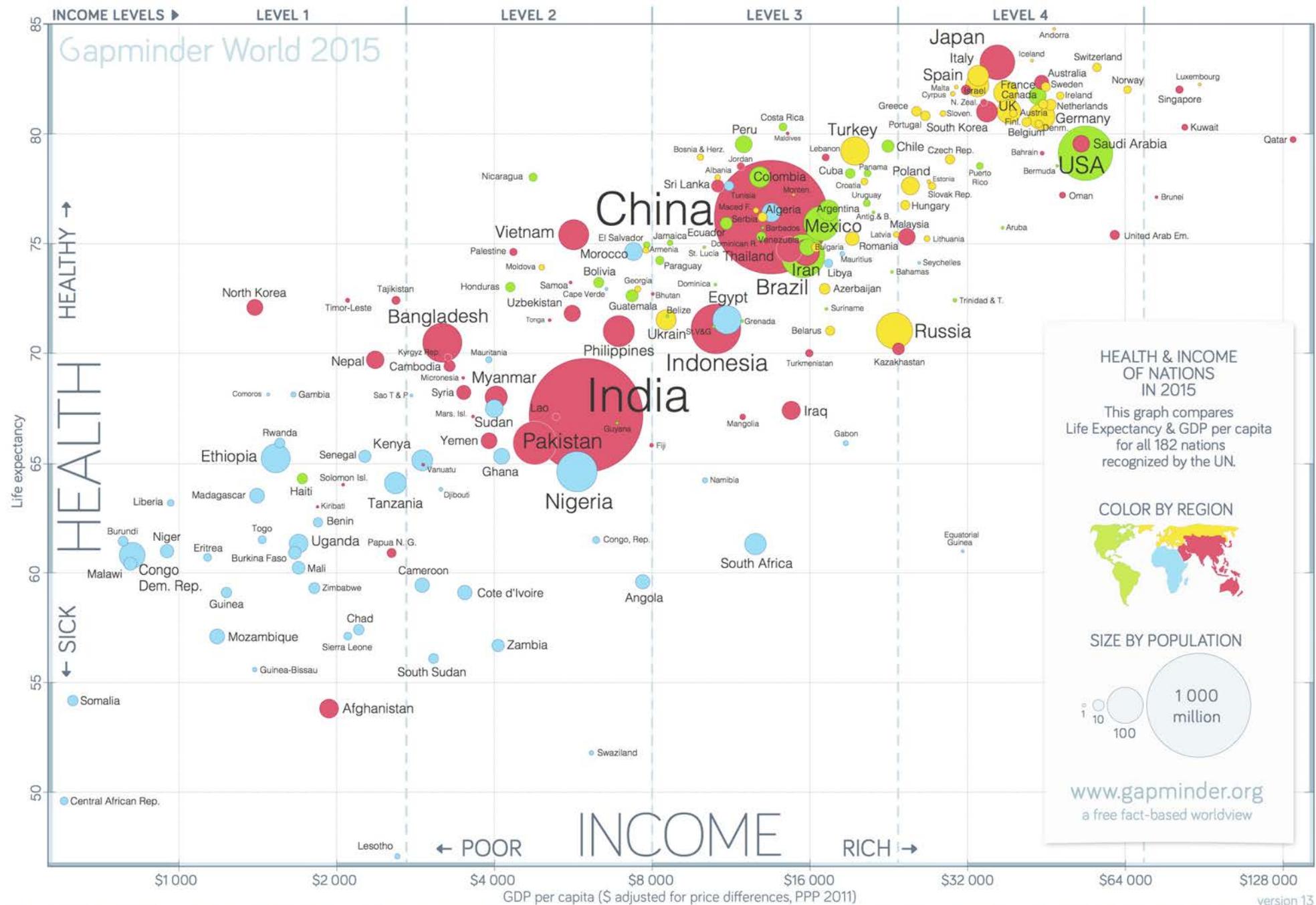
# A tibble: 1,704 × 6
  country continent year lifeExp      pop gdpPercap
  <chr>     <chr>   <dbl>    <dbl>    <dbl>       <dbl>
1 Afghanistan Asia     1952     28.8  8425333     779.
2 Afghanistan Asia     1957     30.3  9240934     821.
3 Afghanistan Asia     1962     32.0  10267083    853.
4 Afghanistan 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.
# i 1,694 more rows
```



GAPMINDER WORLD 2012

Mapping the Wealth and Health of Nations





DATA SOURCES—INCOME: World Bank's GDP per capita, PPP (2011 international \$), with a few additions by Gapminder. X-axis uses log-scale to make a doubling incomes show same distance on all levels. POPULATION: Numbers from UN Population Division. LIFE EXPECTANCY: IHME GBD-2015, as of Oct 2016. INTERACTIVE GRAPH: www.gapminder.org/tools which lets you animate historic data for hundreds of indicators. LICENSE: Our charts are freely available under Creative Commons Attribution License. Please copy, share, modify, integrate and even sell them, as long as you mention "Based on a free chart from www.gapminder.org".

Cédric Scherer // Hands-On Data Visualization with ggplot2: Concepts



Data Preparation

We are using a few subsets of the Gapminder data:

```
1 library(dplyr)  
2 gm2007 <- filter(gapminder, year == 2007)
```



Data Preparation

We are using a few subsets of the Gapminder data:

```
1 library(dplyr)
2 gm2007 <- filter(gapminder, year == 2007)
3 gm2007

# A tibble: 142 × 7
  country   continent   year lifeExp      pop gdpPercap income_lvl
  <chr>     <chr>     <dbl>   <dbl>    <dbl>    <dbl>    <chr>
1 Afghanistan Asia       2007     43.8 31889923     975.    1
2 Albania      Europe    2007     76.4 3600523      5937.   3
3 Algeria      Africa    2007     72.3 33333216     6223.   3
4 Angola       Africa    2007     42.7 12420476     4797.   3
5 Argentina    Americas  2007     75.3 40301927    12779.   4
6 Australia    Oceania   2007     81.2 20434176    34435.   4
7 Austria      Europe    2007     79.8 8199783     36126.   4
8 Bahrain      Asia      2007     75.6 708573      29796.   4
9 Bangladesh   Asia      2007     64.1 150448339    1391.   2
10 Belgium     Europe    2007     79.4 10392226    33693.   4
# i 132 more rows
```



The Data

We are using a few subsets of the Gapminder data:

```
1 library(dplyr)
2 gm_g7 <- filter(gapminder, country %in% c(
3   "United States", "Canada", "France", "Germany", "Italy", "Japan", "United Kingdom")
4 )
```



The Data

We are using a few subsets of the Gapminder data:

```
1 library(dplyr)
2 gm_g7 <- filter(gapminder, country %in% c(
3   "United States", "Canada", "France", "Germany", "Italy", "Japan", "United Kingdom")
4 )
5 gm_g7
```

A tibble: 84 × 7

	country	continent	year	lifeExp	pop	gdpPercap	income_lvl
	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>
1	Canada	Americas	1952	68.8	14785584	11367.	3
2	Canada	Americas	1957	70.0	17010154	12490.	3
3	Canada	Americas	1962	71.3	18985849	13462.	4
4	Canada	Americas	1967	72.1	20819767	16077.	4
5	Canada	Americas	1972	72.9	22284500	18971.	4
6	Canada	Americas	1977	74.2	23796400	22091.	4
7	Canada	Americas	1982	75.8	25201900	22899.	4
8	Canada	Americas	1987	76.9	26549700	26627.	4
9	Canada	Americas	1992	78.0	28523502	26343.	4
10	Canada	Americas	1997	78.6	30305843	28955.	4
# i 74 more rows							



— The Structure of a *ggplot* —



The Components of {ggplot2}

Component	Function	Explanation
Data	<code>ggplot(data)</code>	<i>The raw data that you want to visualise.</i>
Aesthetics	<code>aes()</code>	<i>Aesthetic mappings between variables and visual properties.</i>
Geometries	<code>geom_*</code> ()	<i>The geometric shapes representing the data.</i>



The Components of {ggplot2}

Component	Function	Explanation
Data	<code>ggplot(data)</code>	<i>The raw data that you want to visualise.</i>
Aesthetics	<code>aes()</code>	<i>Aesthetic mappings between variables and visual properties.</i>
Geometries	<code>geom_*</code> ()	<i>The geometric shapes representing the data.</i>
Statistics	<code>stat_*</code> ()	<i>The statistical transformations applied to the data.</i>
Scales	<code>scale_*</code> ()	<i>Maps between the data and the aesthetic dimensions.</i>
Coordinate System	<code>coord_*</code> ()	<i>Maps data into the plane of the data rectangle.</i>
Facets	<code>facet_*</code> ()	<i>The arrangement of the data into a grid of plots.</i>
Visual Themes	<code>theme()</code> and <code>theme_*</code> ()	<i>The overall visual defaults of a plot.</i>



A Basic *ggplot*



ggplot2::ggplot()

```
1 ?ggplot
```

ggplot: Create a new ggplot

Description

`'ggplot()'` initializes a ggplot object. It can be used to declare the input data frame for a graphic and to specify the set of plot aesthetics intended to be common throughout all subsequent layers unless specifically overridden.

Usage

```
ggplot(data = NULL, mapping = aes(), ..., environment = parent.frame())
```

Arguments

data Default dataset to use for plot. If not already a `data.frame`, will be converted to one by `'fortify()'`. If not specified, must be supplied in each layer added to the plot.

mapping Default list of aesthetic mappings to use for plot. If not specified, must be supplied in each layer added to the plot.

... Other arguments passed on to methods. Not currently used.

environment DEPRECATED. Used prior to tidy evaluation.

Details

`'ggplot()'` is used to construct the initial plot object, and is almost always followed by `'+'` to add component to the plot. There are three common ways to invoke `'ggplot()'`:

- `'ggplot(df, aes(x, y, other aesthetics))'`
- `'ggplot(df)'`
- `'ggplot()'`



— Data —



Data

= define data object to visualize

- usually specified inside `ggplot()`
- can be specified for each layer as well



Data

```
1 ggplot(data = gm2007)
```



— Aesthetics —



Aesthetic Mapping

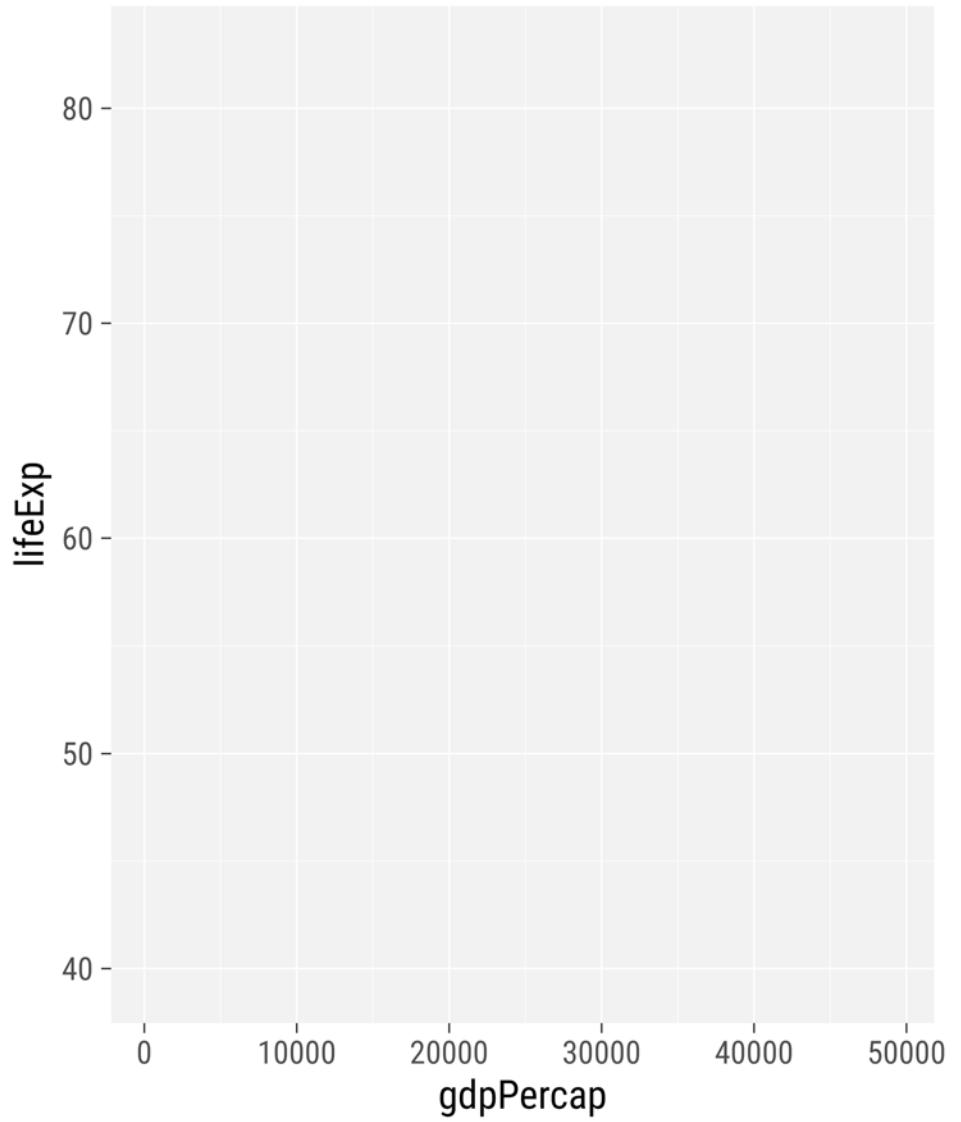
= link variables to graphical properties

- positions (`x, y`)
- colors (`color, fill`)
- shapes (`shape, linetype`)
- size (`size`)
- transparency (`alpha`)
- groupings (`group`)



Aesthetic Mapping

```
1 ggplot(data = gm2007) +  
2   aes(x = gdpPercap, y = lifeExp)
```



aesthetics

aes() outside as component

```
1 ggplot(data = gm2007) + aes(x = gdpPercap, y = lifeExp)
```

aes() inside, explicit matching

```
1 ggplot(data = gm2007, mapping = aes(x = gdpPercap, y = lifeExp))
```

aes() inside, implicit matching

```
1 ggplot(gm2007, aes(gdpPercap, lifeExp))
```

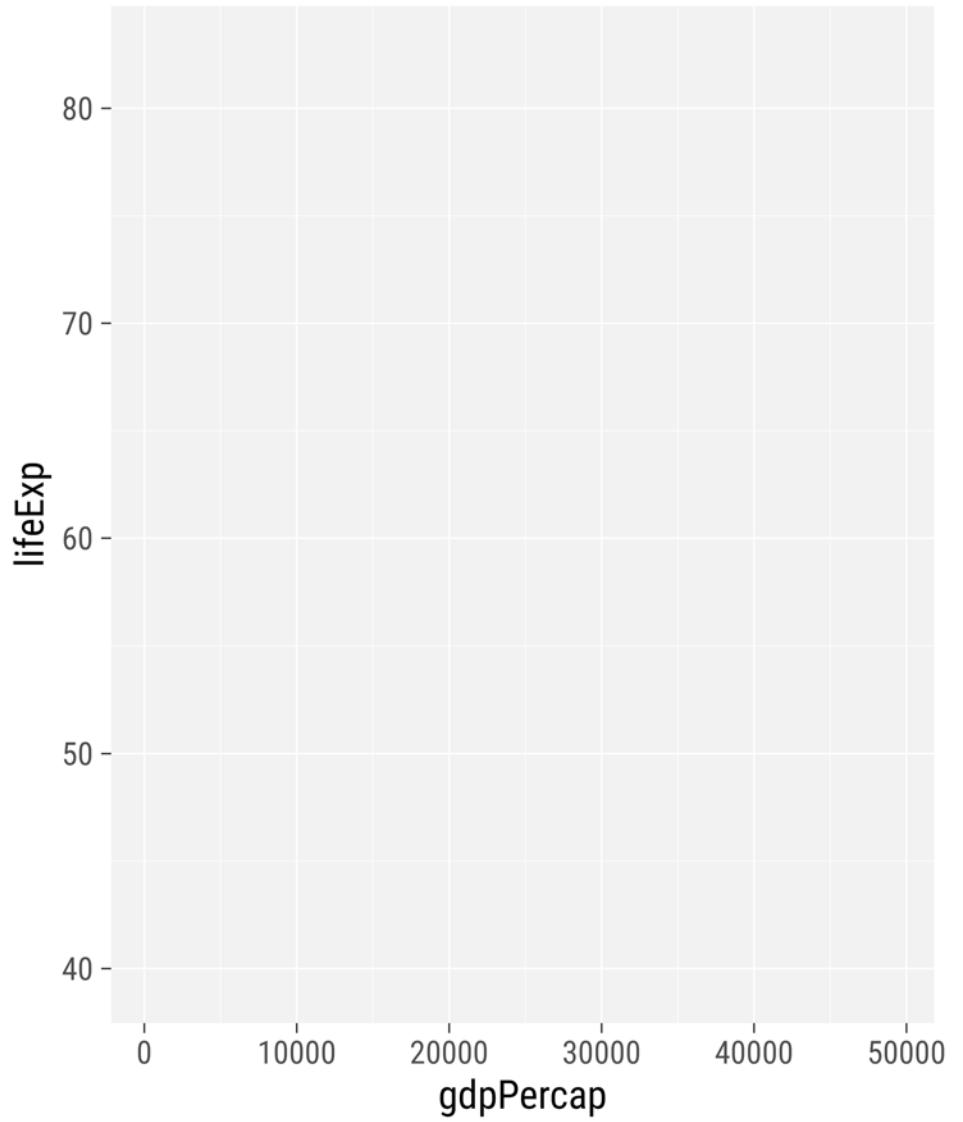
aes() inside, mixed matching

```
1 ggplot(data = gm2007, aes(x = gdpPercap, y = lifeExp))
```



Aesthetic Mapping

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp))  
4 )
```



— Geometrical Layers —



Geometrical Layers

= interpret aesthetics as graphical representations

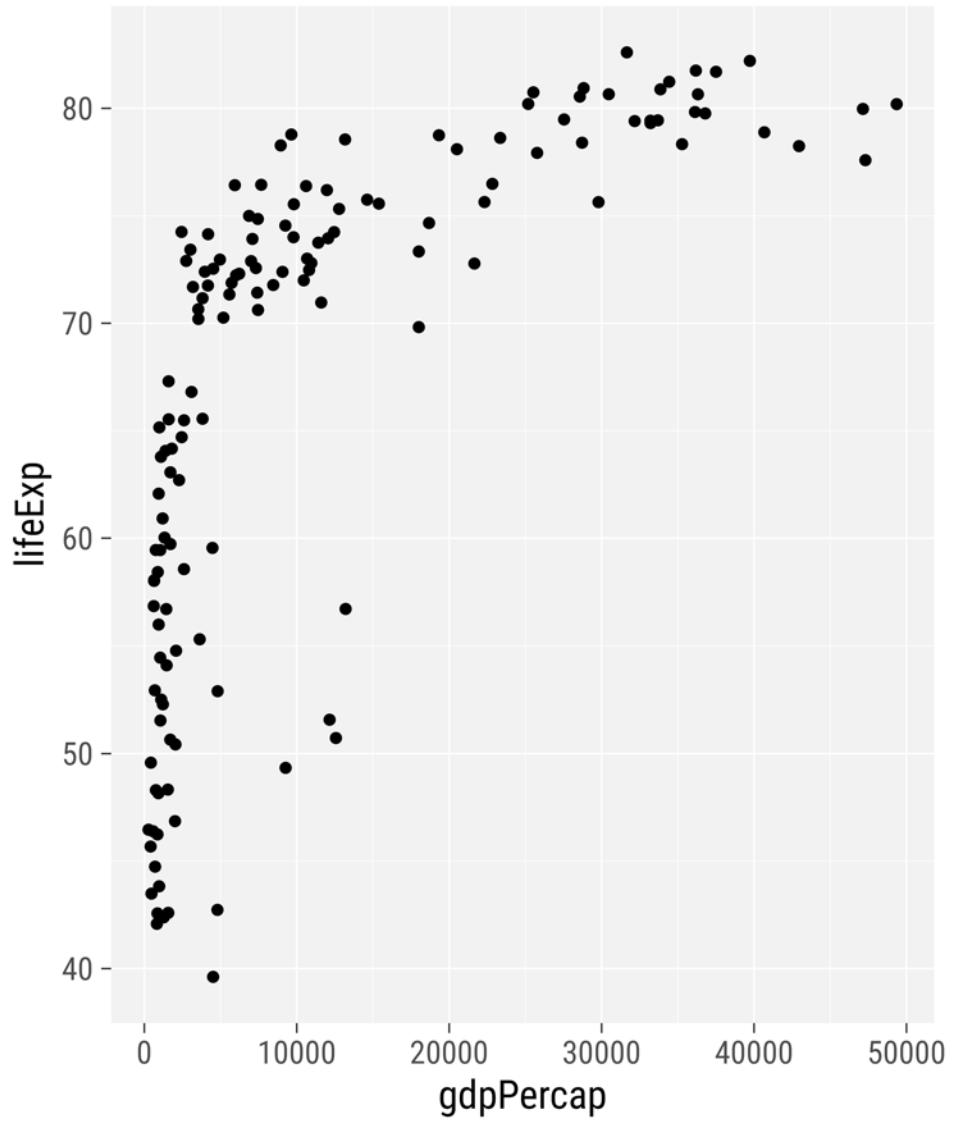
- points
- lines
- polygons
- text labels
- ...

Data-related variables are mapped via **aes()**, all other set outside **aes()**.



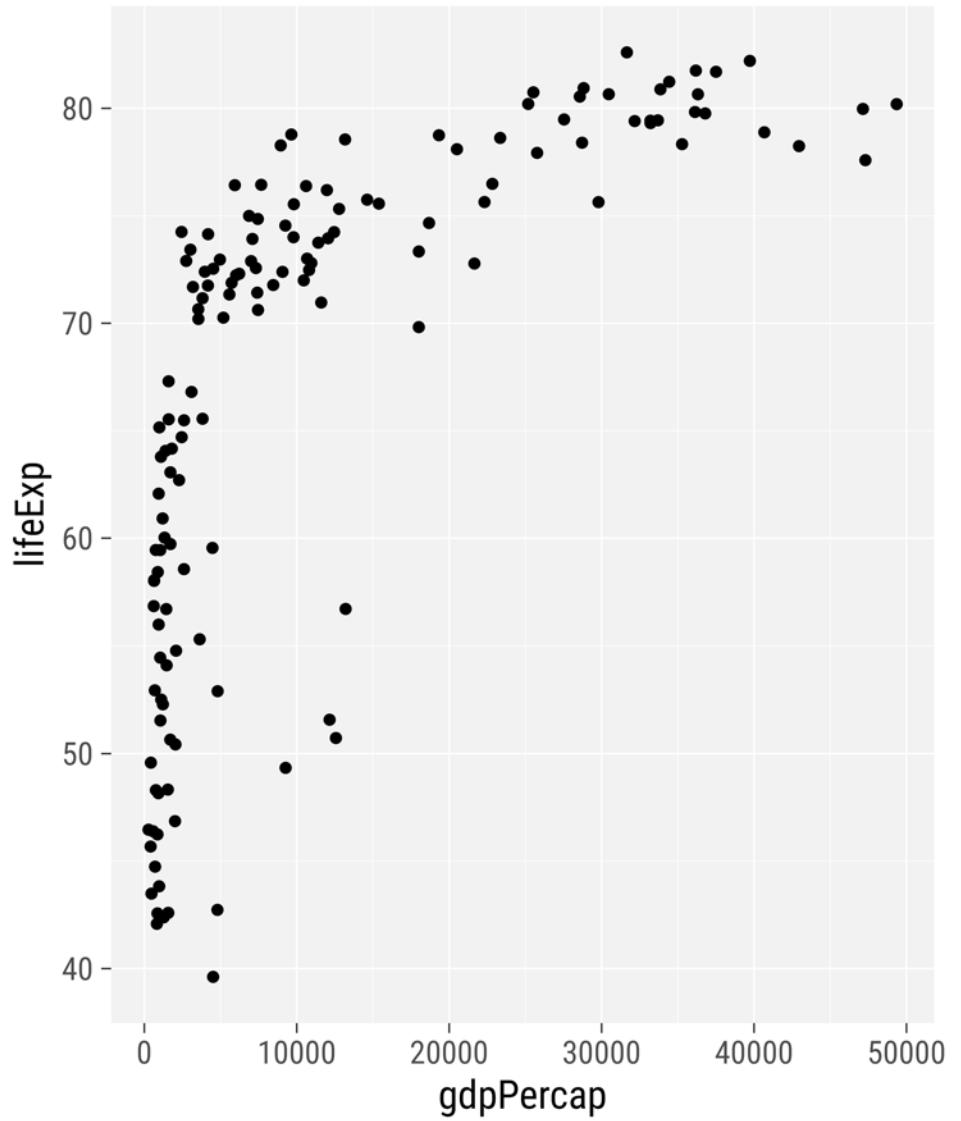
Geometrical Layers

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp)  
4 ) +  
5 geom_point()
```



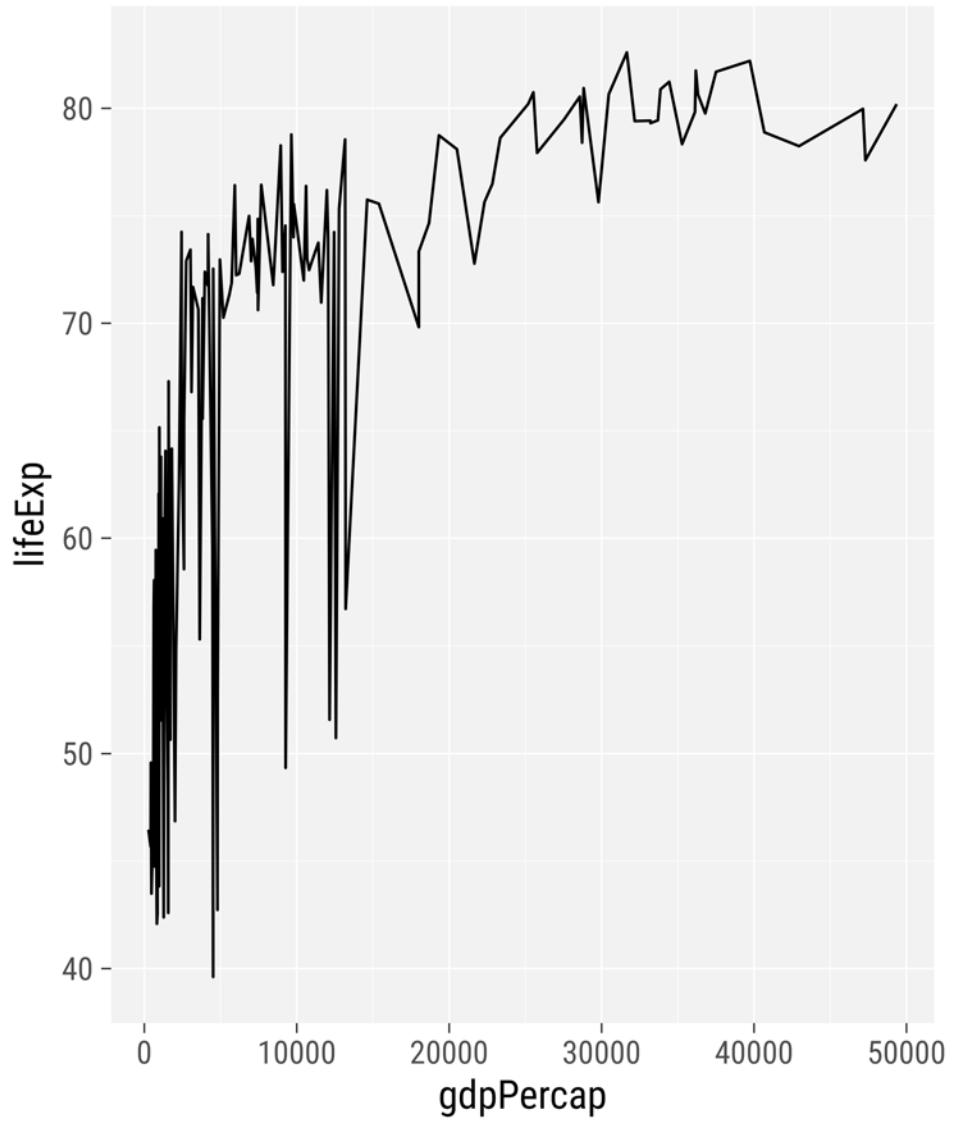
Layers

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp)  
4 ) +  
5 layer(  
6   geom = "point",  
7   stat = "identity", ## do nothing  
8   position = "identity" ## do nothing  
9 )
```



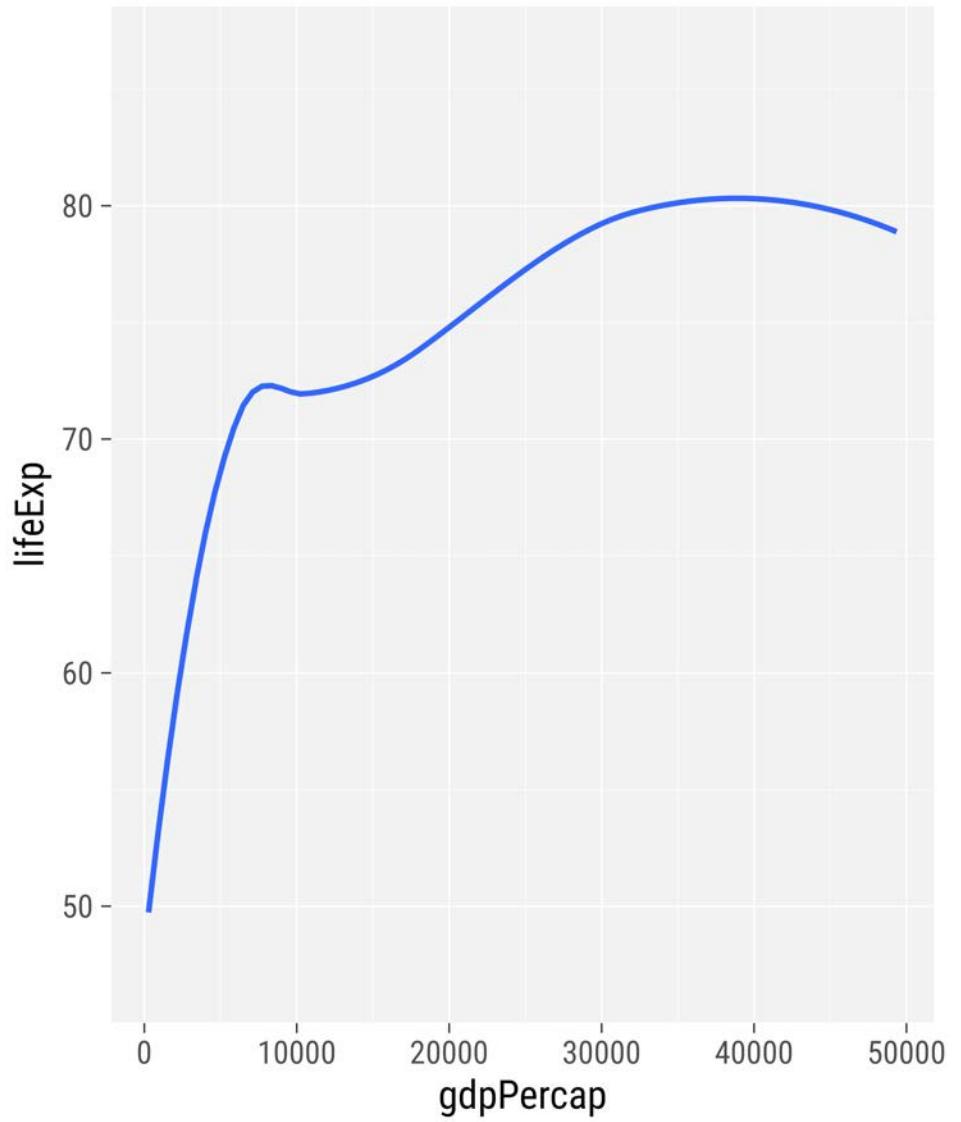
Layers

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp)  
4 ) +  
5 layer(  
6   geom = "line",  
7   stat = "identity", ## do nothing  
8   position = "identity" ## do nothing  
9 )
```



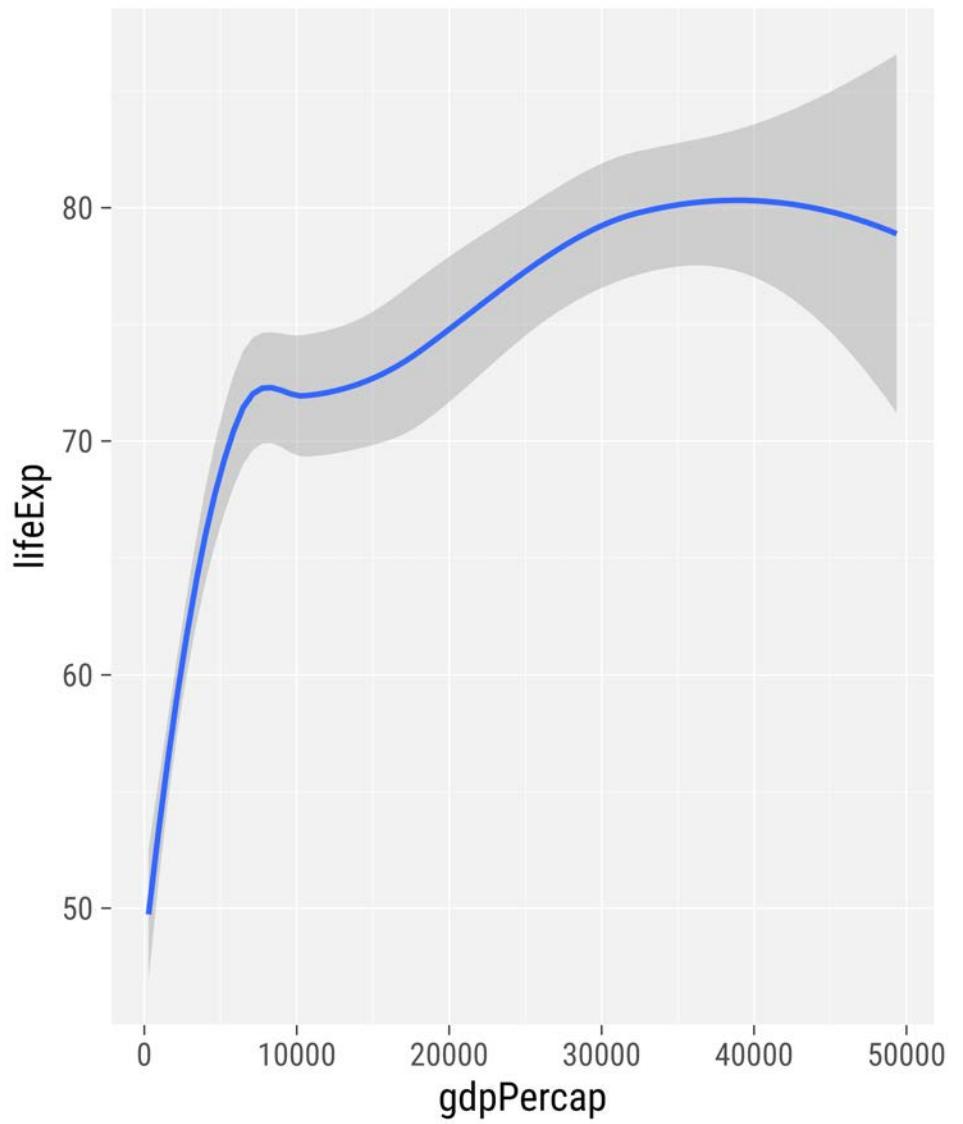
Layers

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp)  
4 ) +  
5 layer(  
6   geom = "smooth",  
7   stat = "smooth",  
8   position = "identity" ## do nothing  
9 )
```



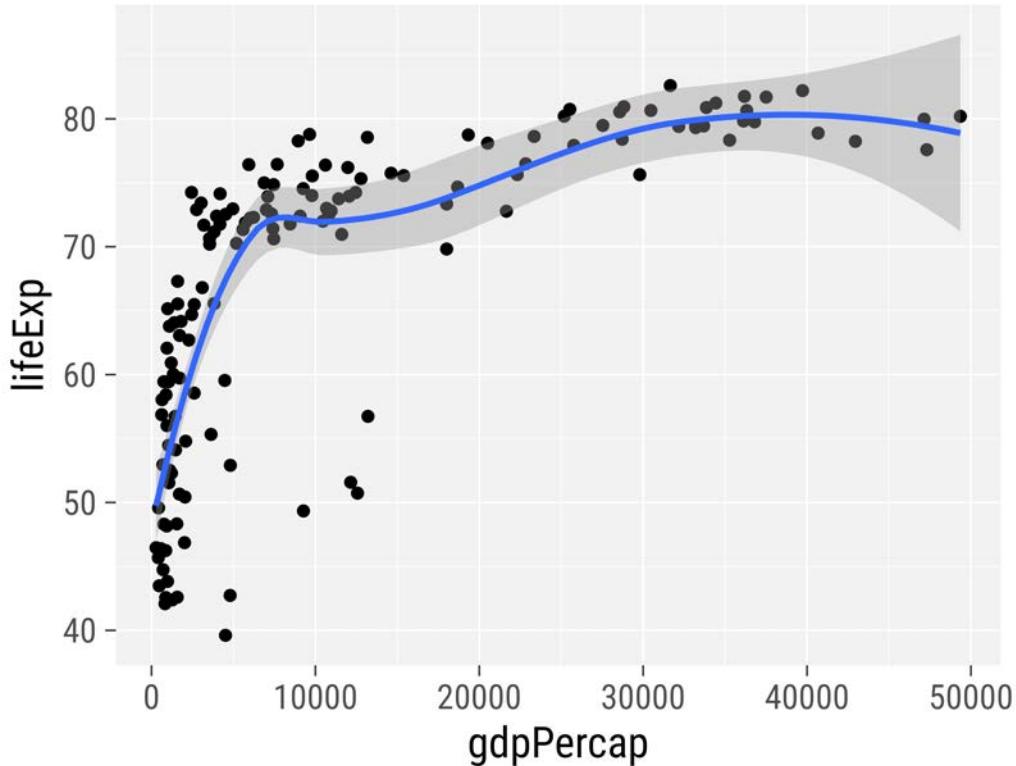
Layers

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp)  
4 ) +  
5 geom_smooth()
```

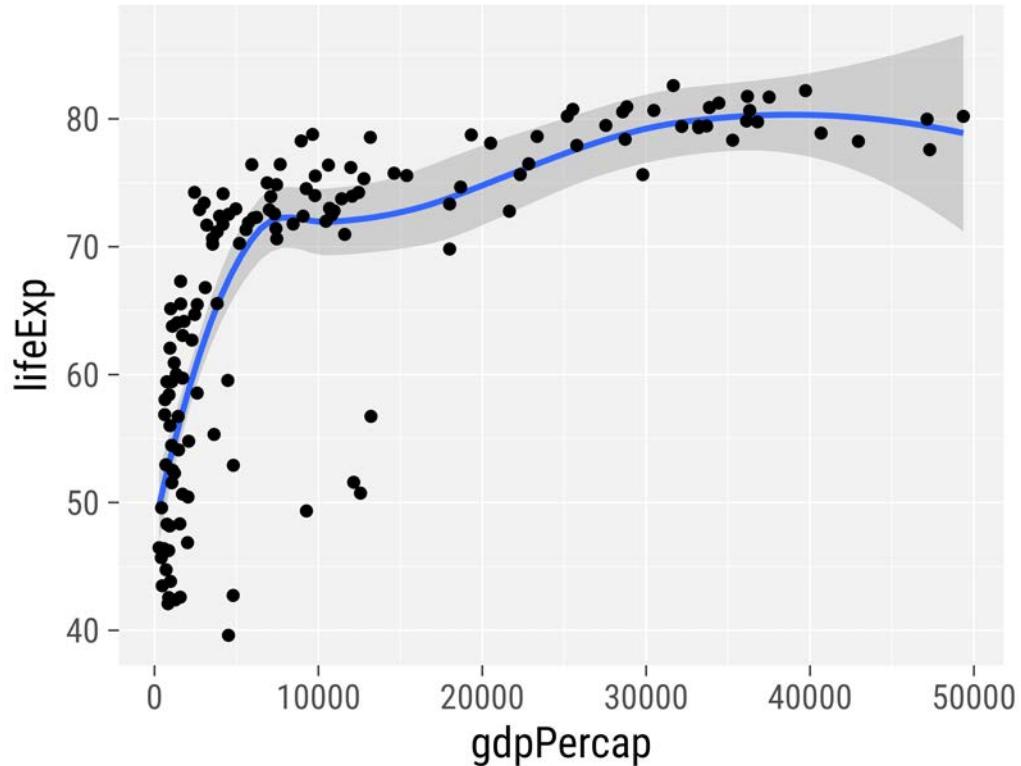


Combine Layers

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp)  
4 ) +  
5 geom_point() +  
6 geom_smooth()
```

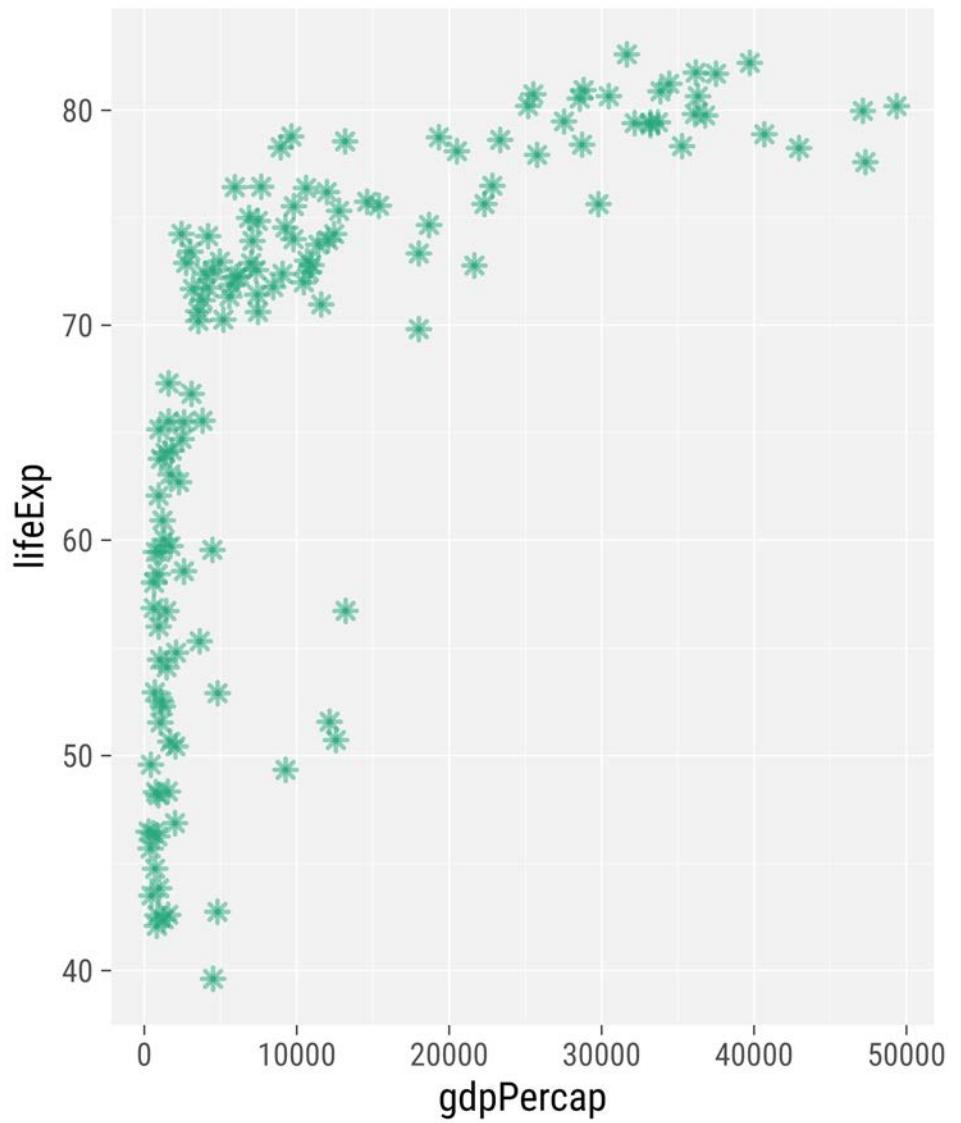


```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp)  
4 ) +  
5 geom_smooth() +  
6 geom_point()
```



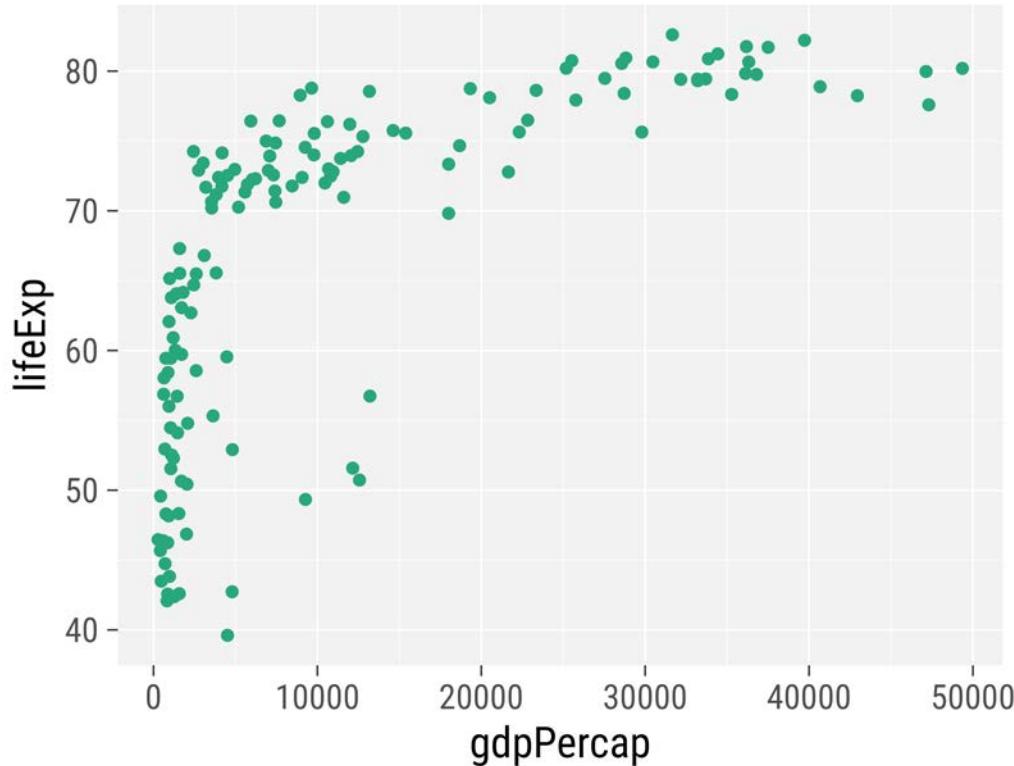
Visual Properties of Layers

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp)  
4 ) +  
5 geom_point(  
6   color = "#28a87d",  
7   alpha = .5,  
8   shape = 8,  
9   stroke = 1.2,  
10  size = 2  
11 )
```

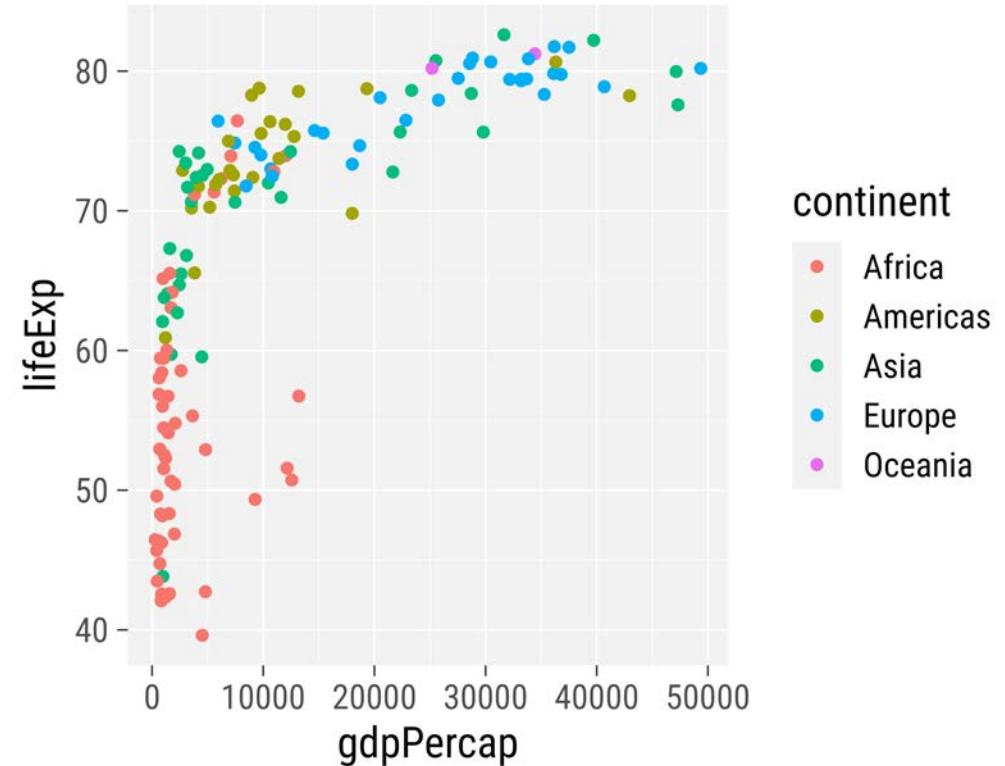


Setting vs Mapping of Visual Properties

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp)  
4 ) +  
5 geom_point(  
6   color = "#28a87d"  
7 )
```

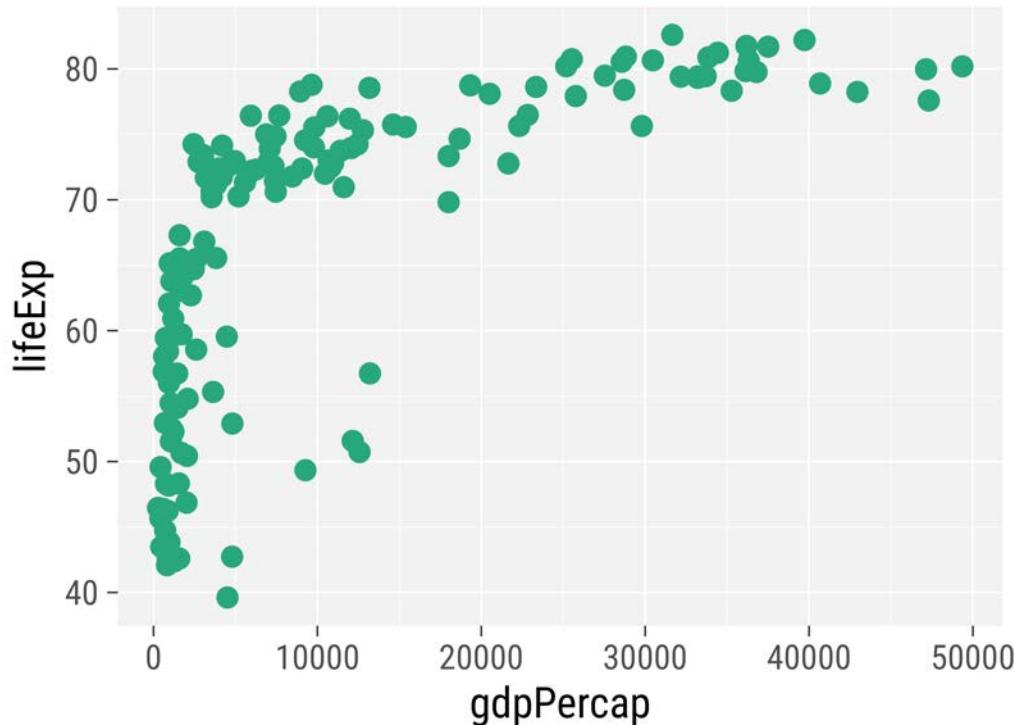


```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp)  
4 ) +  
5 geom_point(  
6   aes(color = continent)  
7 )
```

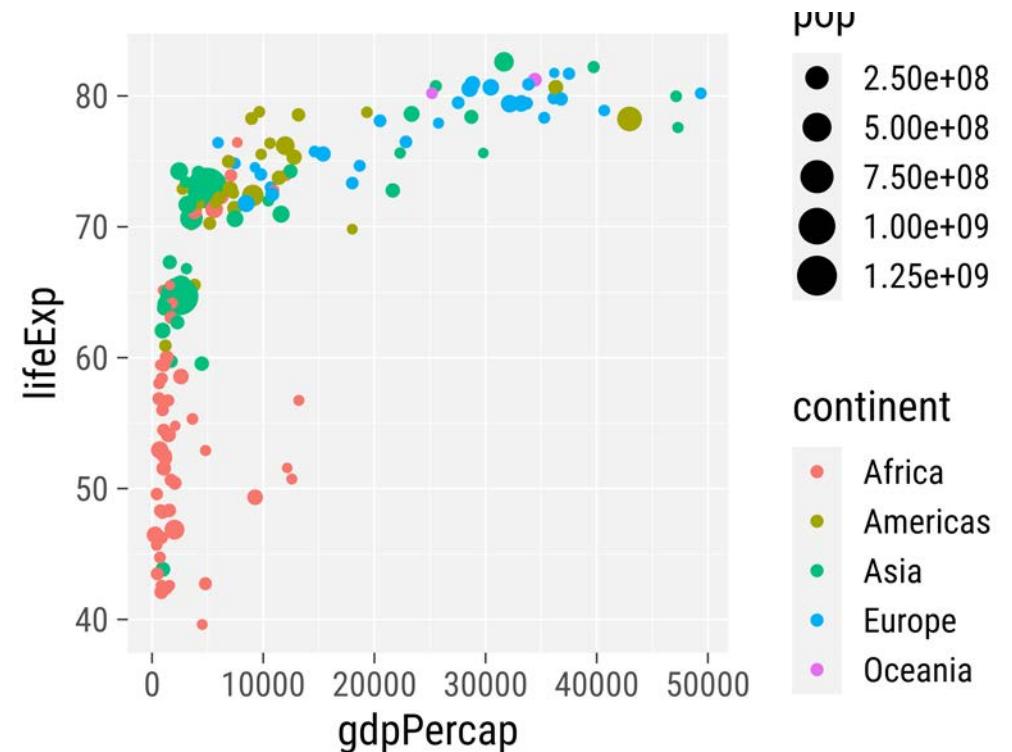


Setting vs Mapping of Visual Properties

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp)  
4 ) +  
5 geom_point(  
6   color = "#28a87d",  
7   size = 3  
8 )
```

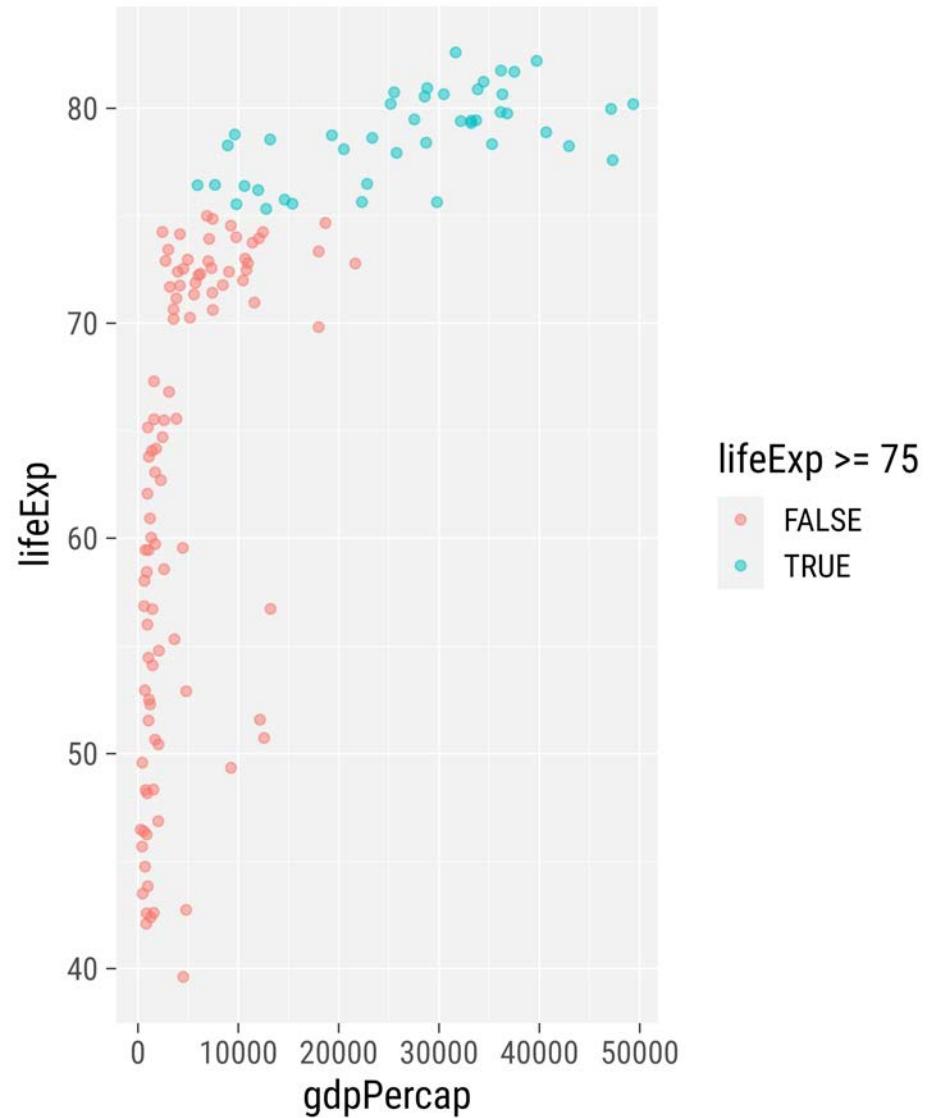


```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp)  
4 ) +  
5 geom_point(  
6   aes(color = continent,  
7   size = pop)  
8 )
```



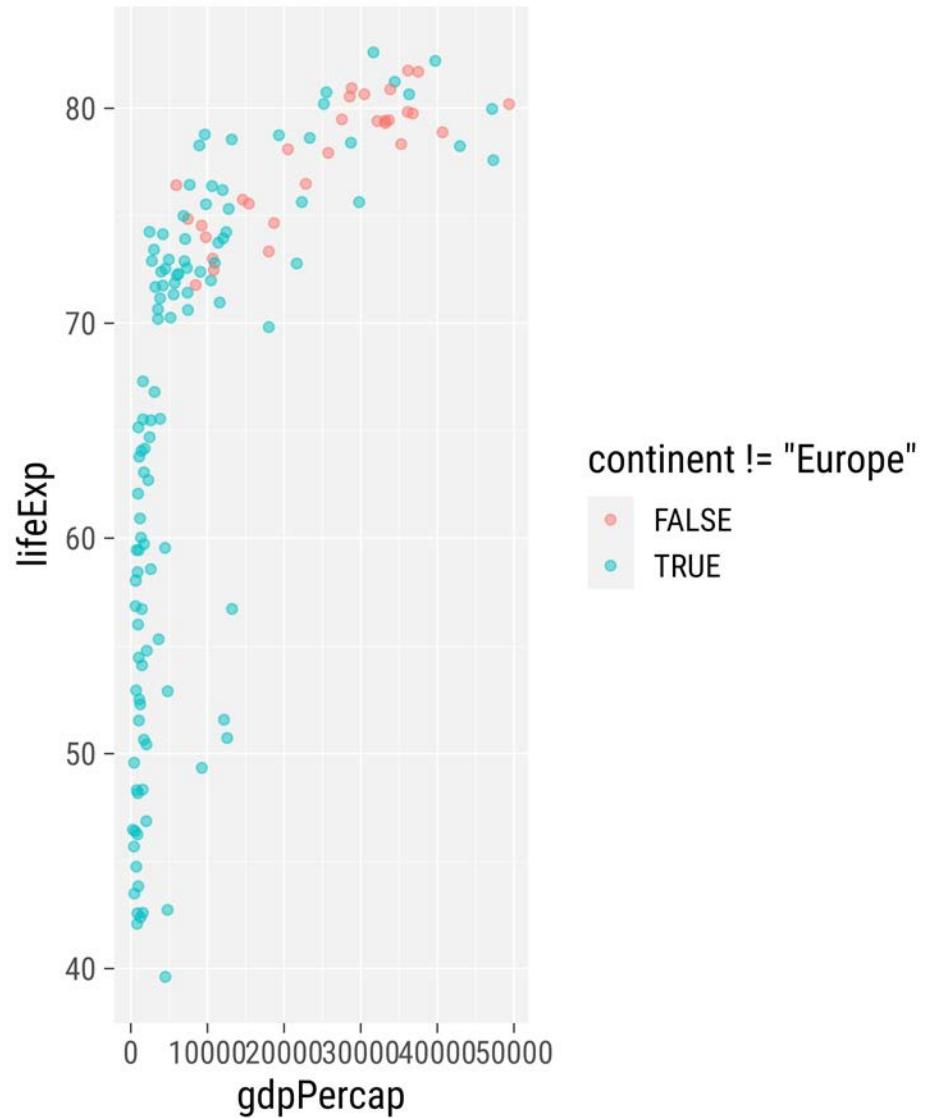
Mapping Expressions

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp)  
4 ) +  
5 geom_point(  
6   aes(color = lifeExp >= 75),  
7   alpha = .5  
8 )
```



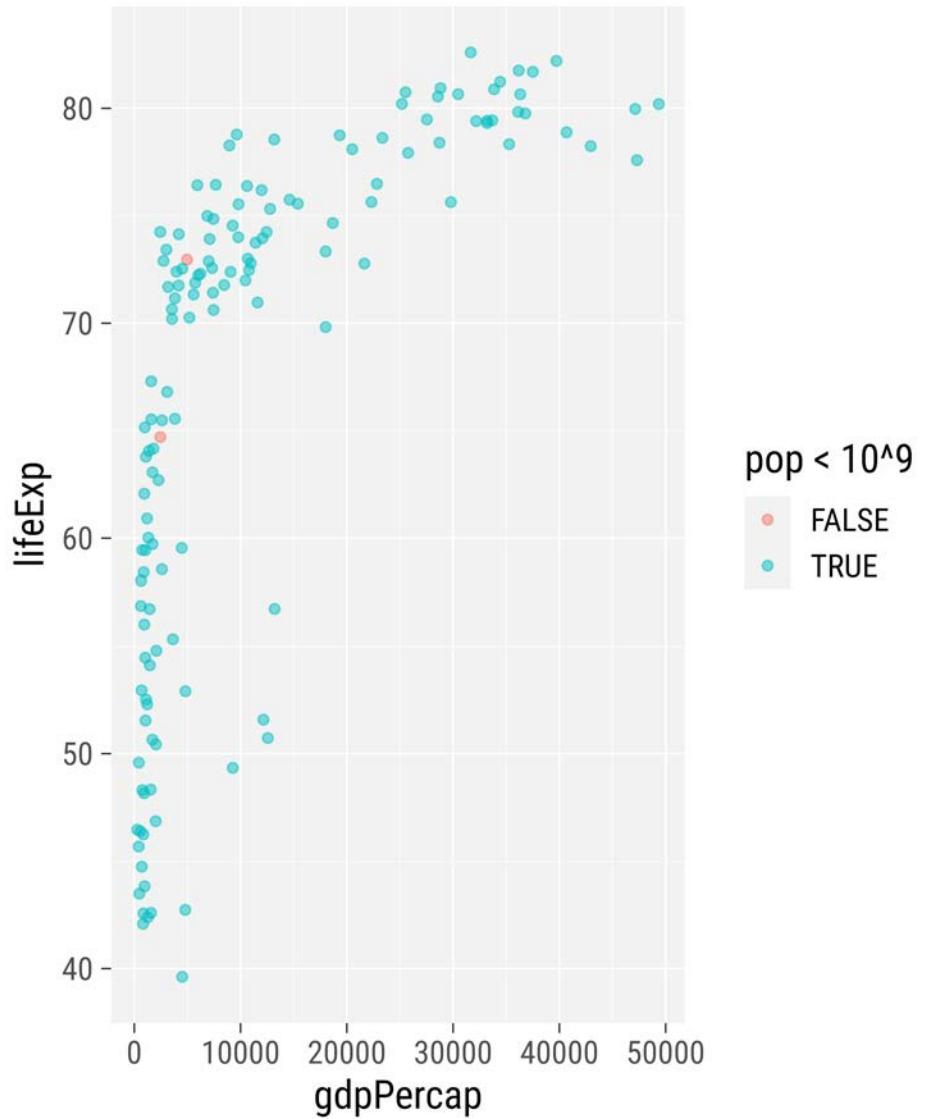
Mapping Expressions

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp)  
4 ) +  
5 geom_point(  
6   aes(color = continent != "Europe"),  
7   alpha = .5  
8 )
```



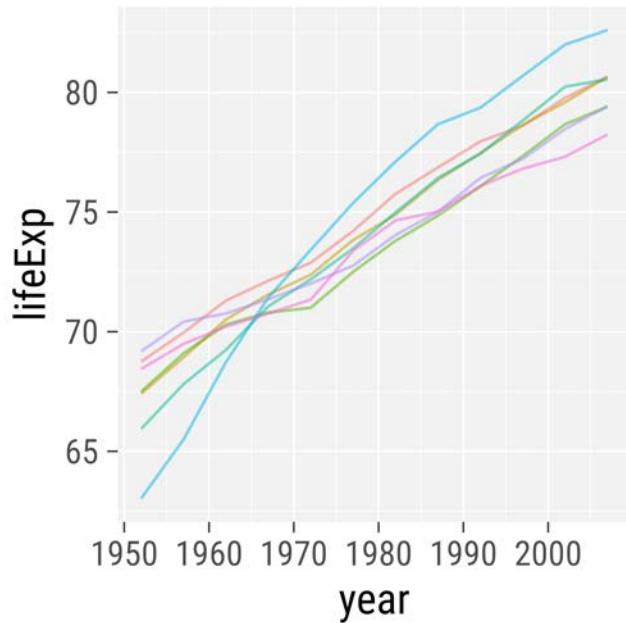
Mapping Expressions

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp)  
4 ) +  
5 geom_point(  
6   aes(color = pop < 10^9),  
7   alpha = .5  
8 )
```

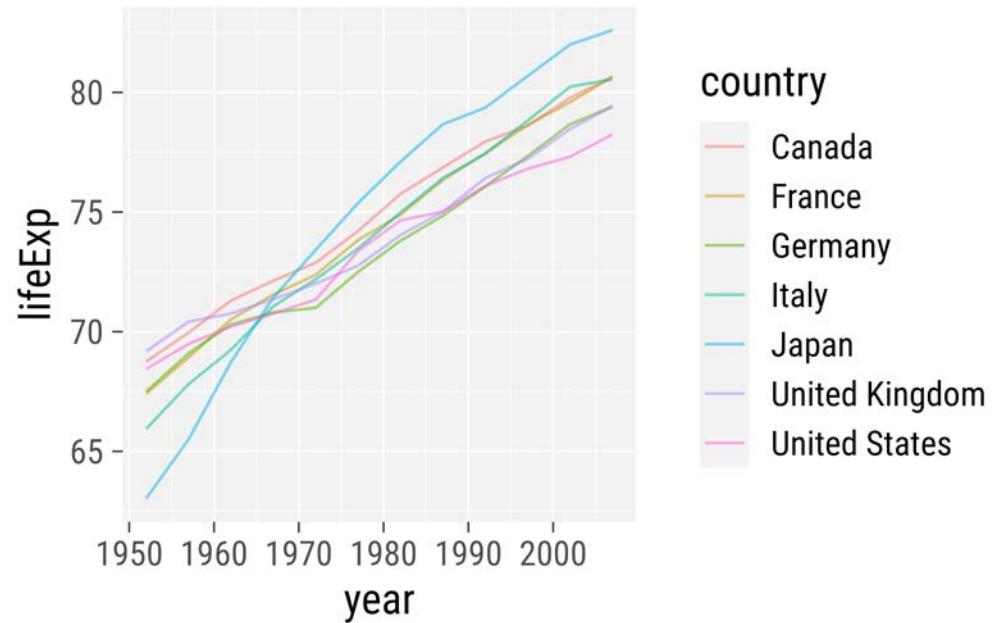


Local vs Global Encoding

```
1 ggplot(  
2   data = gm_g7,  
3   aes(x = year, y = lifeExp)  
4 ) +  
5 geom_line(  
6   aes(color = country),  
7   alpha = .5  
8 )
```

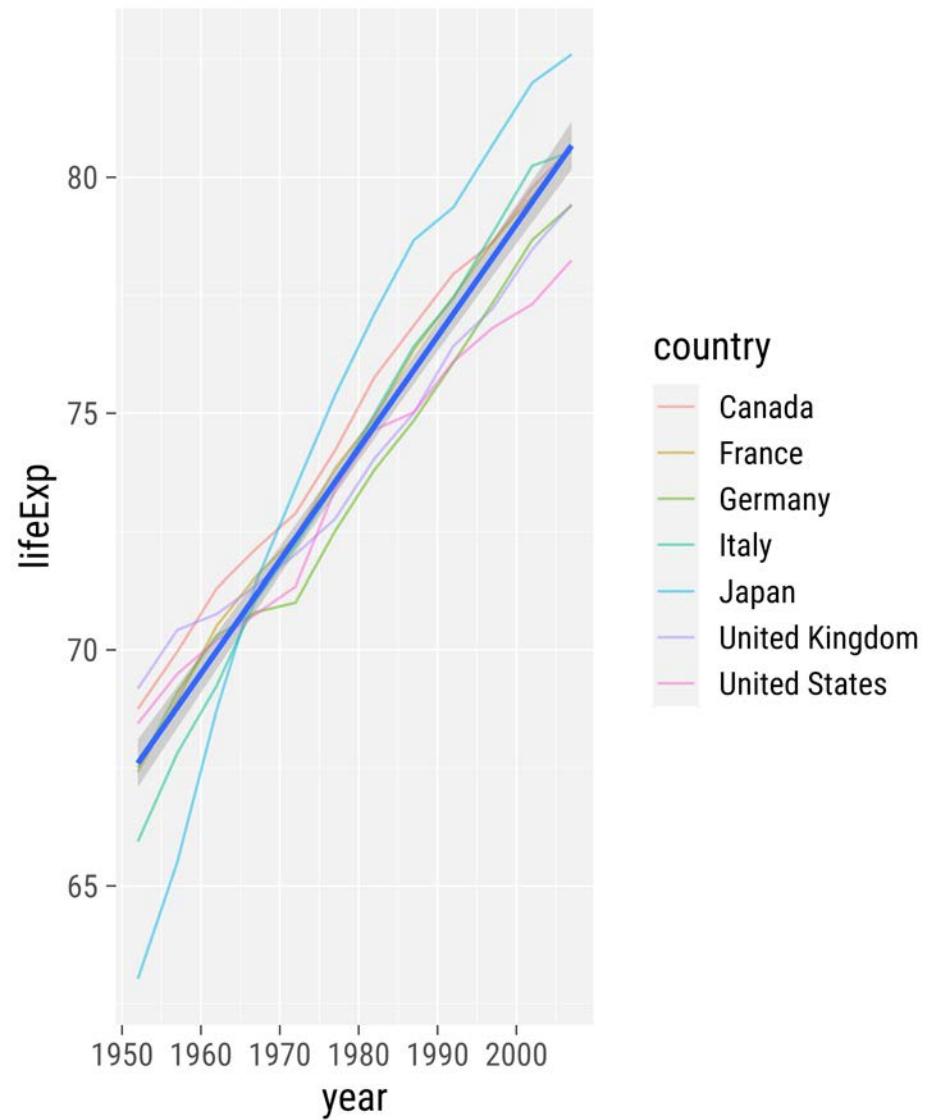


```
1 ggplot(  
2   data = gm_g7,  
3   aes(x = year, y = lifeExp,  
4        color = country)  
5 ) +  
6 geom_line(  
7   alpha = .5  
8 )
```



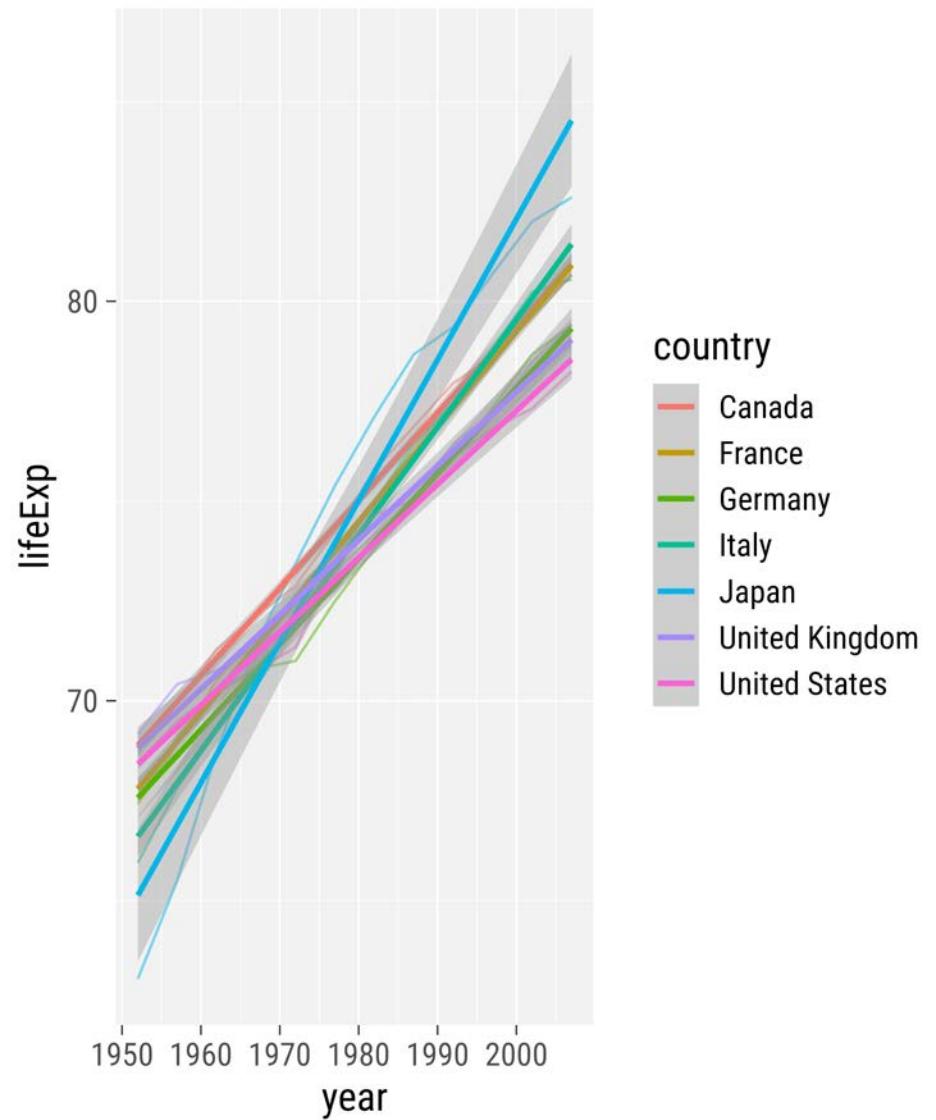
Adding More Layers: Local Encoding

```
1 ggplot(  
2   data = gm_g7,  
3   aes(x = year, y = lifeExp)  
4 ) +  
5   geom_line(  
6   aes(color = country),  
7   alpha = .5  
8 ) +  
9   geom_smooth(  
10  method = "lm"  
11 )
```



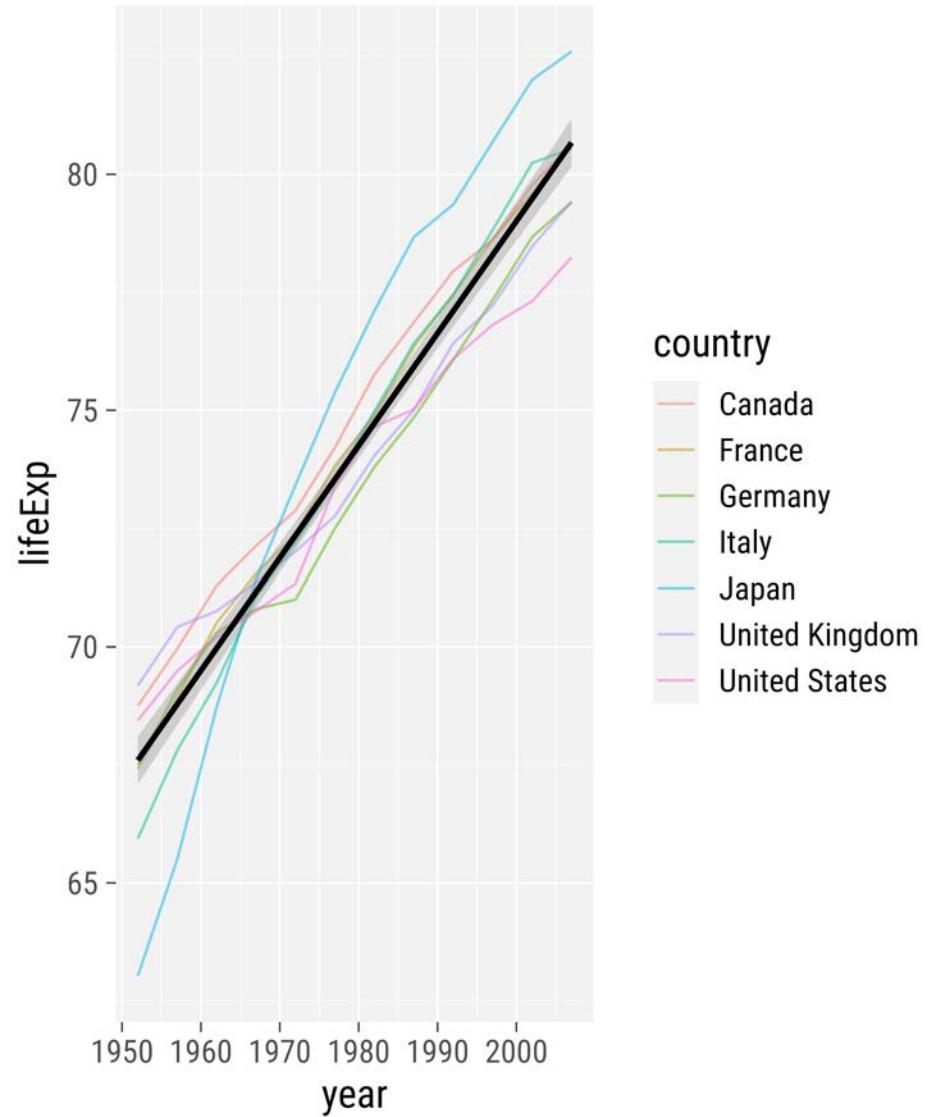
Adding More Layers: Global Encoding

```
1 ggplot(  
2   data = gm_g7,  
3   aes(x = year, y = lifeExp,  
4        color = country)  
5 ) +  
6 geom_line(  
7   alpha = .5  
8 ) +  
9 geom_smooth(  
10  method = "lm"  
11 )
```



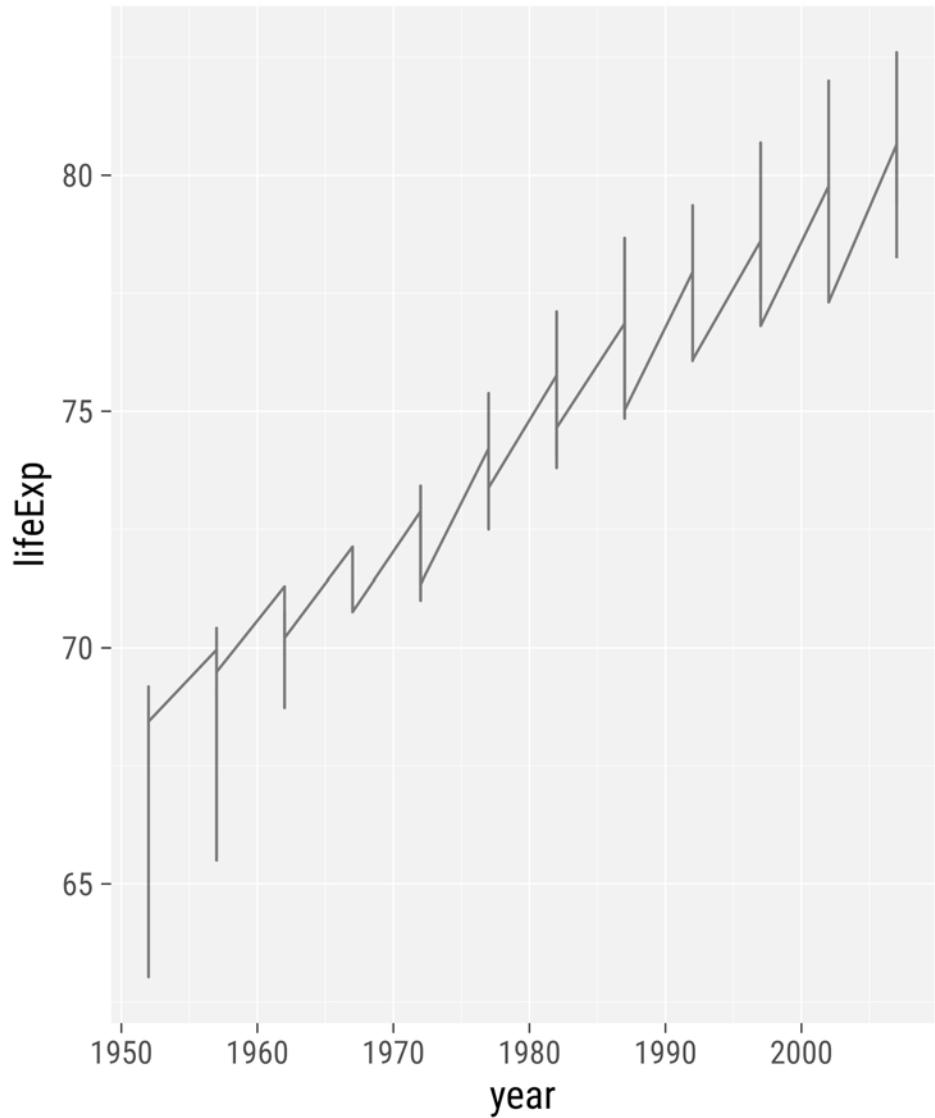
Overwrite Global Aesthetics

```
1 ggplot(  
2   data = gm_g7,  
3   aes(x = year, y = lifeExp,  
4        color = country)  
5 ) +  
6   geom_line(  
7     alpha = .5  
8 ) +  
9   geom_smooth(  
10    method = "lm",  
11    color = "black"  
12 )
```



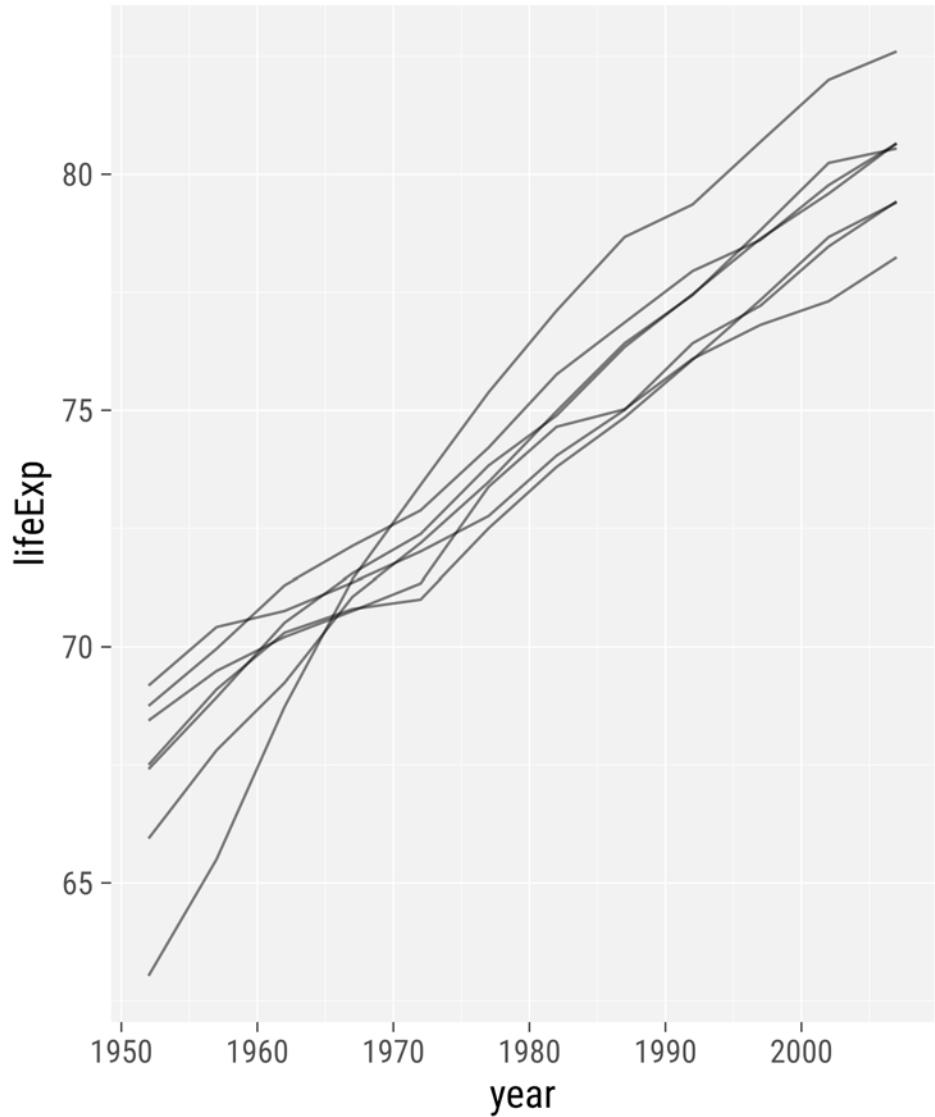
The group Aesthetic

```
1 ggplot(  
2   data = gm_g7,  
3   aes(x = year, y = lifeExp)  
4 ) +  
5 geom_line(  
6   alpha = .5  
7 )
```



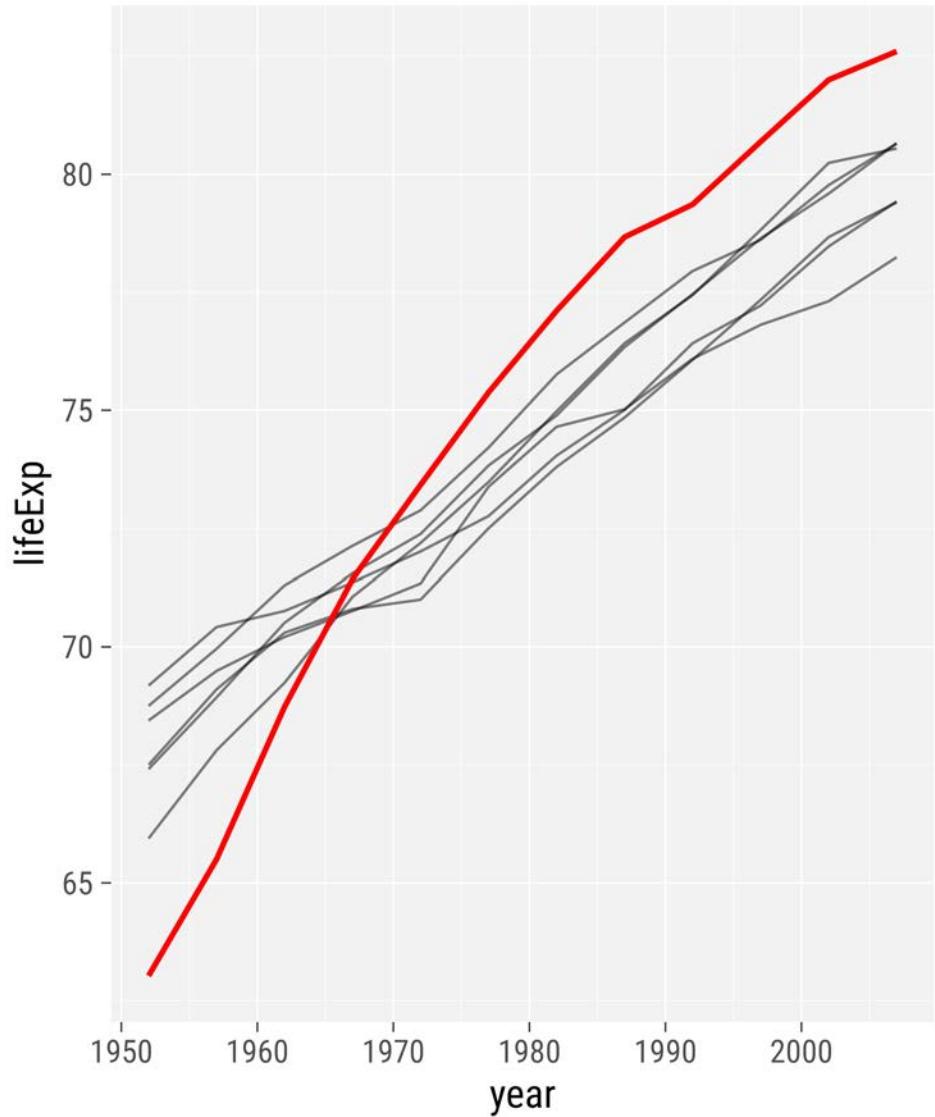
The group Aesthetic

```
1 ggplot(  
2   data = gm_g7,  
3   aes(x = year, y = lifeExp,  
4        group = country)  
5 ) +  
6 geom_line(  
7   alpha = .5  
8 )
```



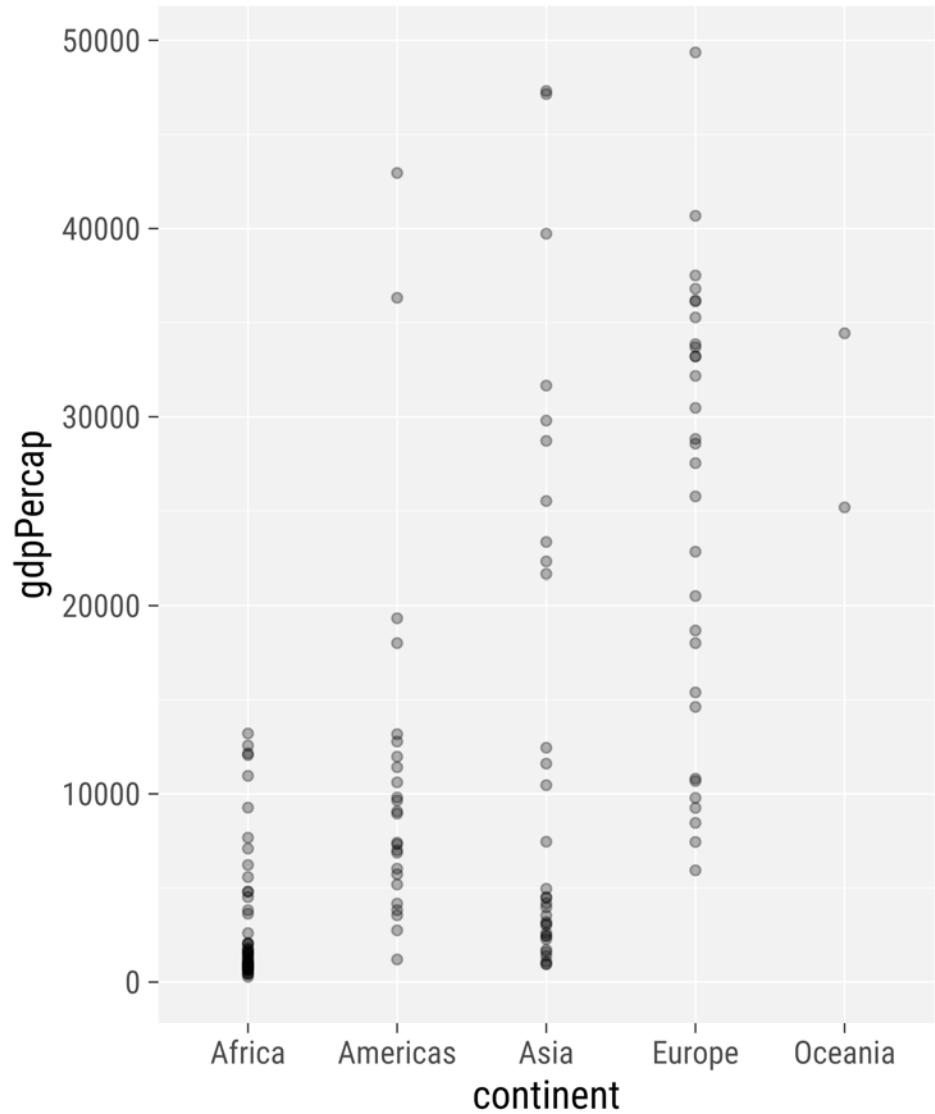
The group Aesthetic

```
1 ggplot(  
2   data = gm_g7,  
3   aes(x = year, y = lifeExp,  
4        group = country)  
5 ) +  
6 geom_line(  
7   alpha = .5  
8 ) +  
9 geom_line(  
10  data = filter(gm_g7, country == "Japan"),  
11  color = "red",  
12  size = 1  
13 )
```



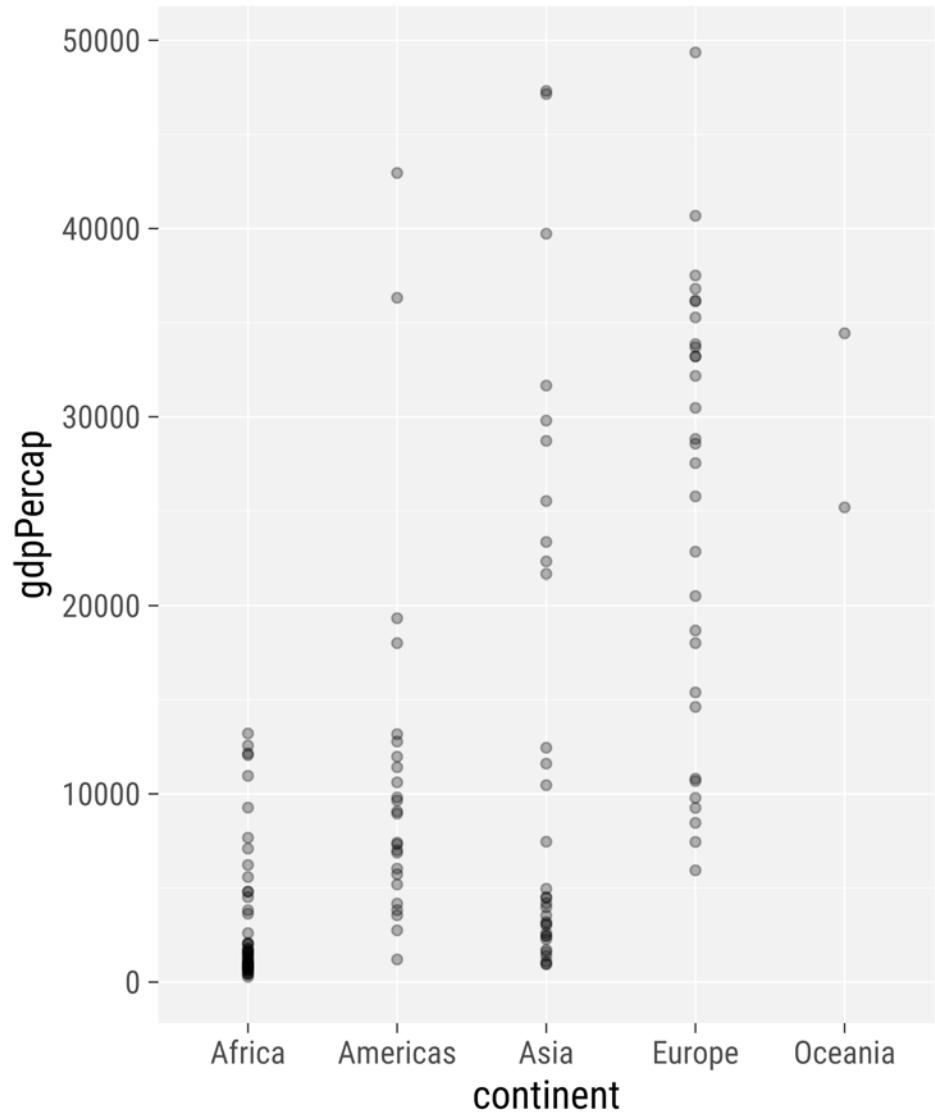
The position Argument

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = continent, y = gdpPercap)  
4 ) +  
5 geom_point(  
6   alpha = .3  
7 )
```



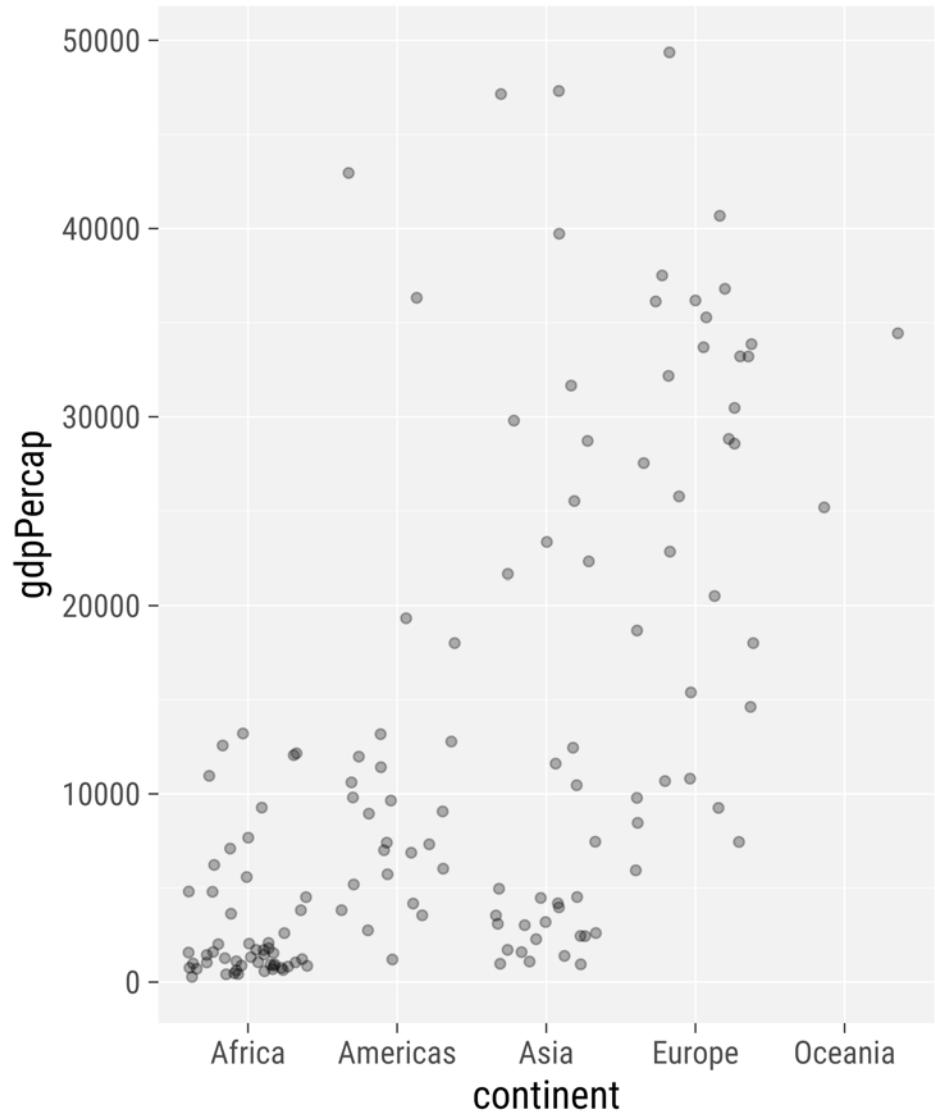
The position Argument

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = continent, y = gdpPercap)  
4 ) +  
5 geom_point(  
6   position = "identity",  
7   alpha = .3  
8 )
```



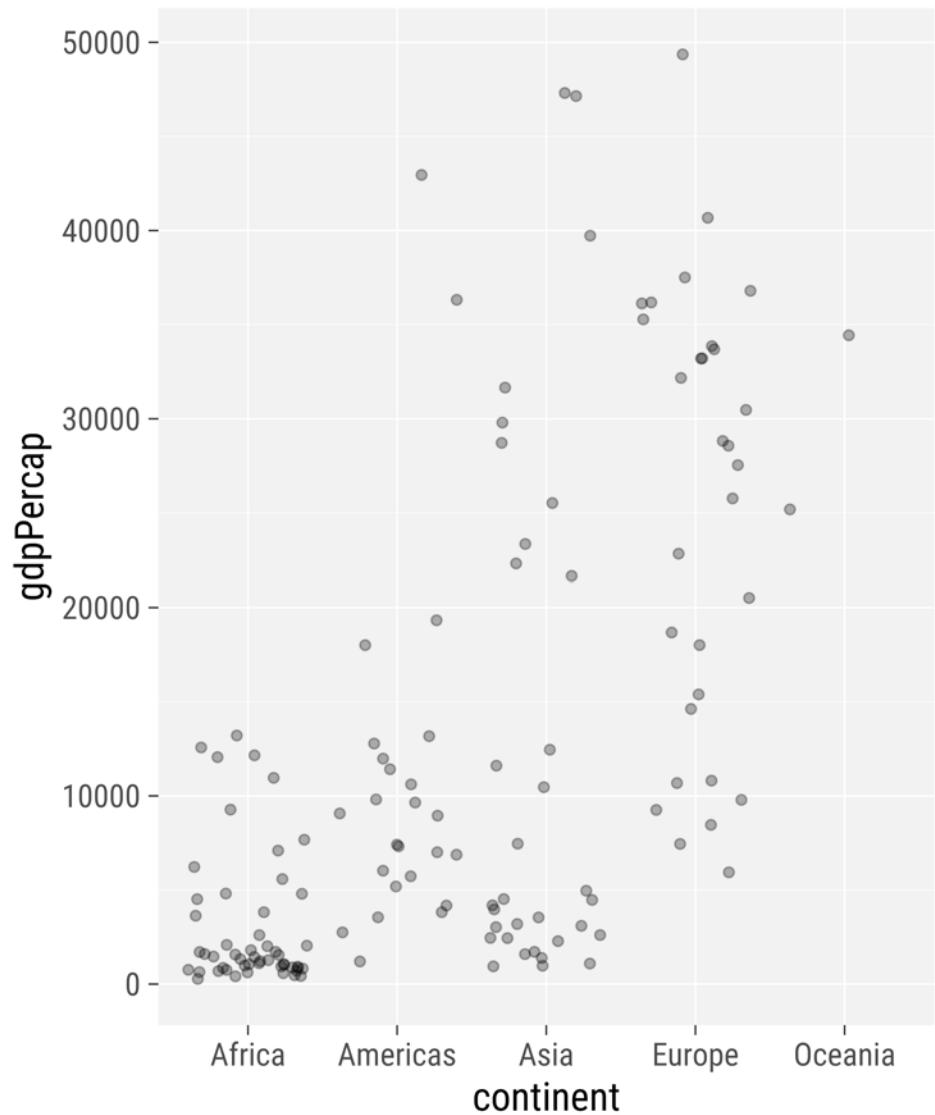
The position Argument

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = continent, y = gdpPercap)  
4 ) +  
5 geom_point(  
6   position = "jitter",  
7   alpha = .3  
8 )
```



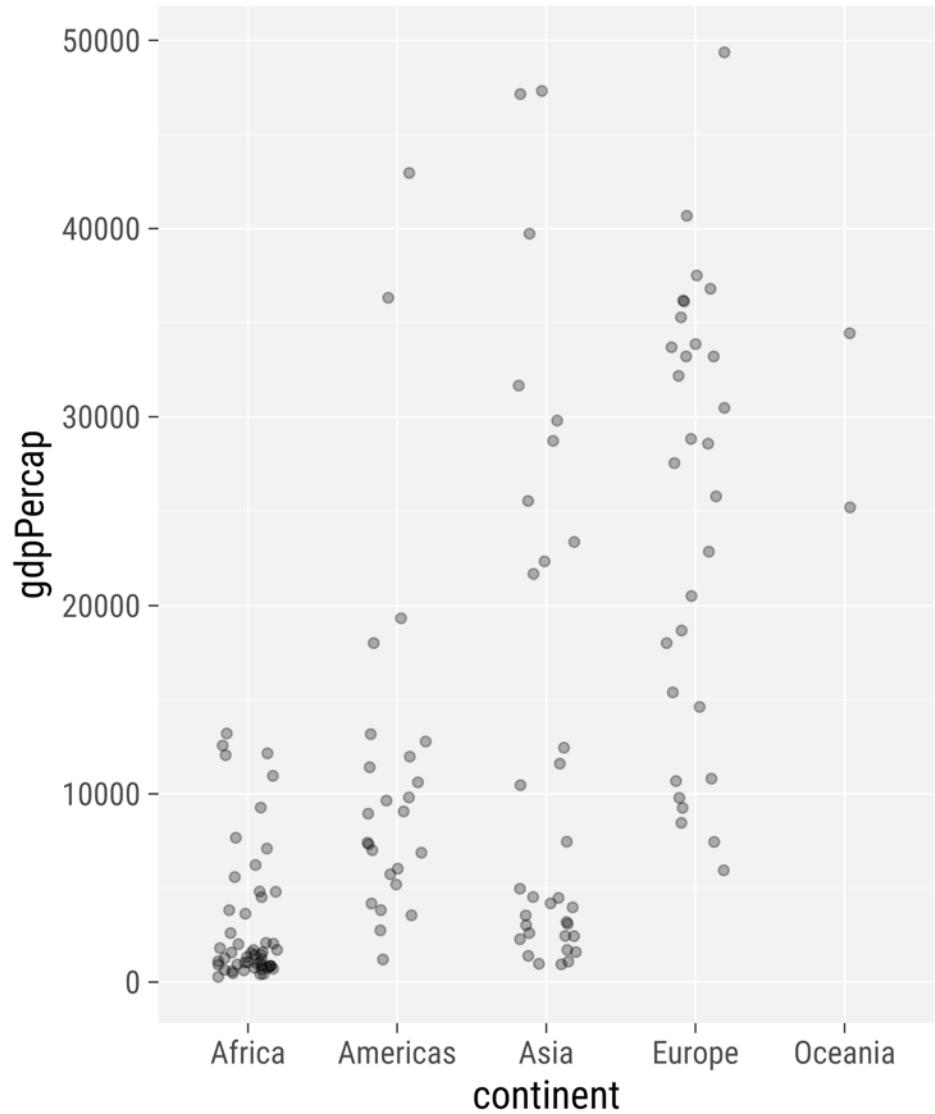
The position Argument

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = continent, y = gdpPercap)  
4 ) +  
5 geom_point(  
6   position = position_jitter(),  
7   alpha = .3  
8 )
```



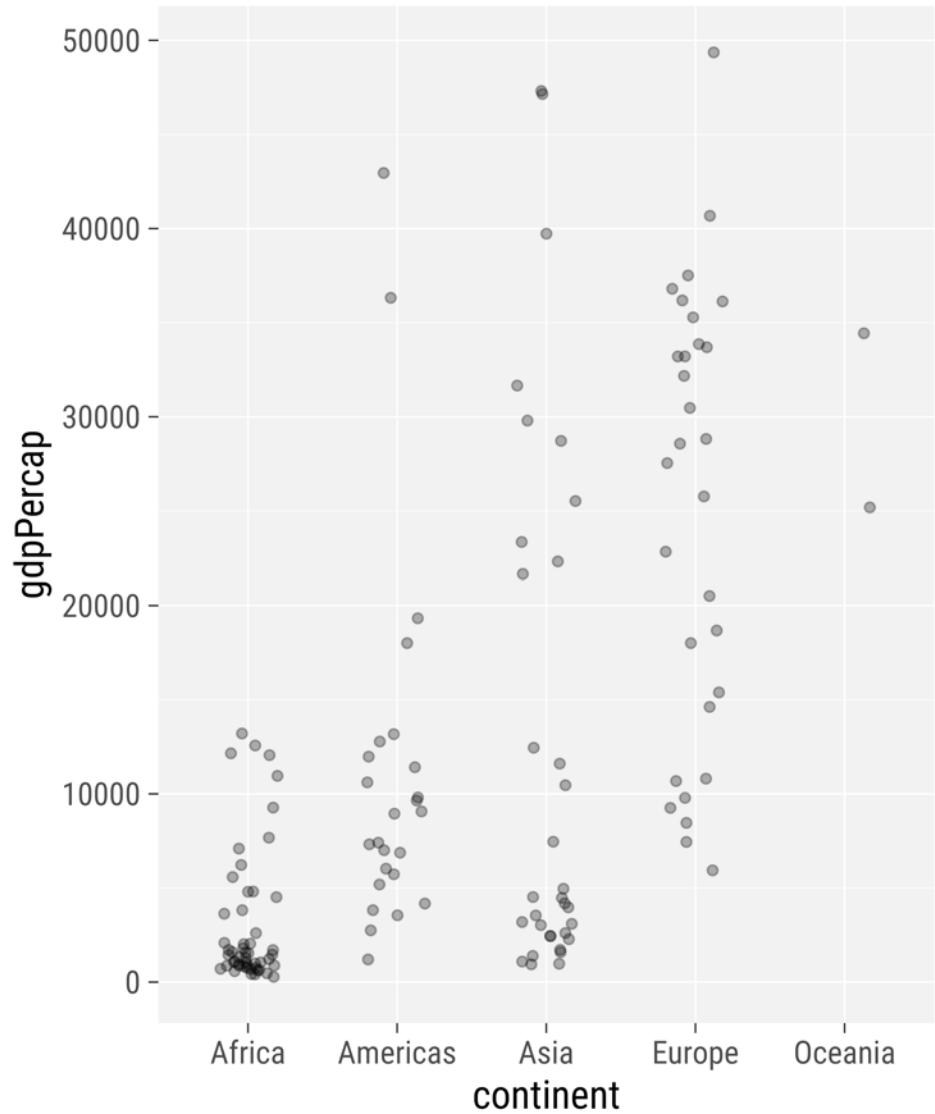
The position Argument

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = continent, y = gdpPercap)  
4 ) +  
5 geom_point(  
6   position = position_jitter(  
7     width = .2  
8   ),  
9   alpha = .3  
10 )
```



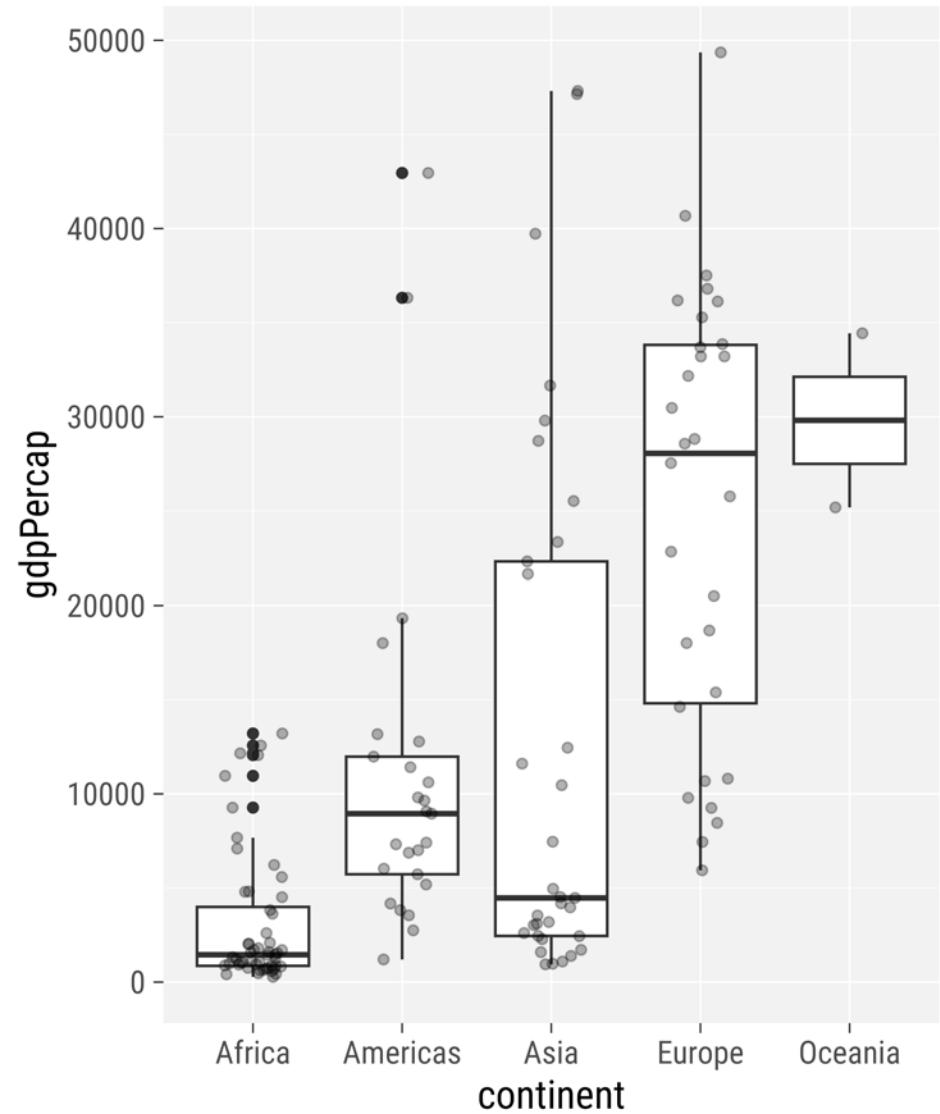
The position Argument

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = continent, y = gdpPercap)  
4 ) +  
5   geom_jitter(  
6     width = .2,  
7     alpha = .3  
8 )
```



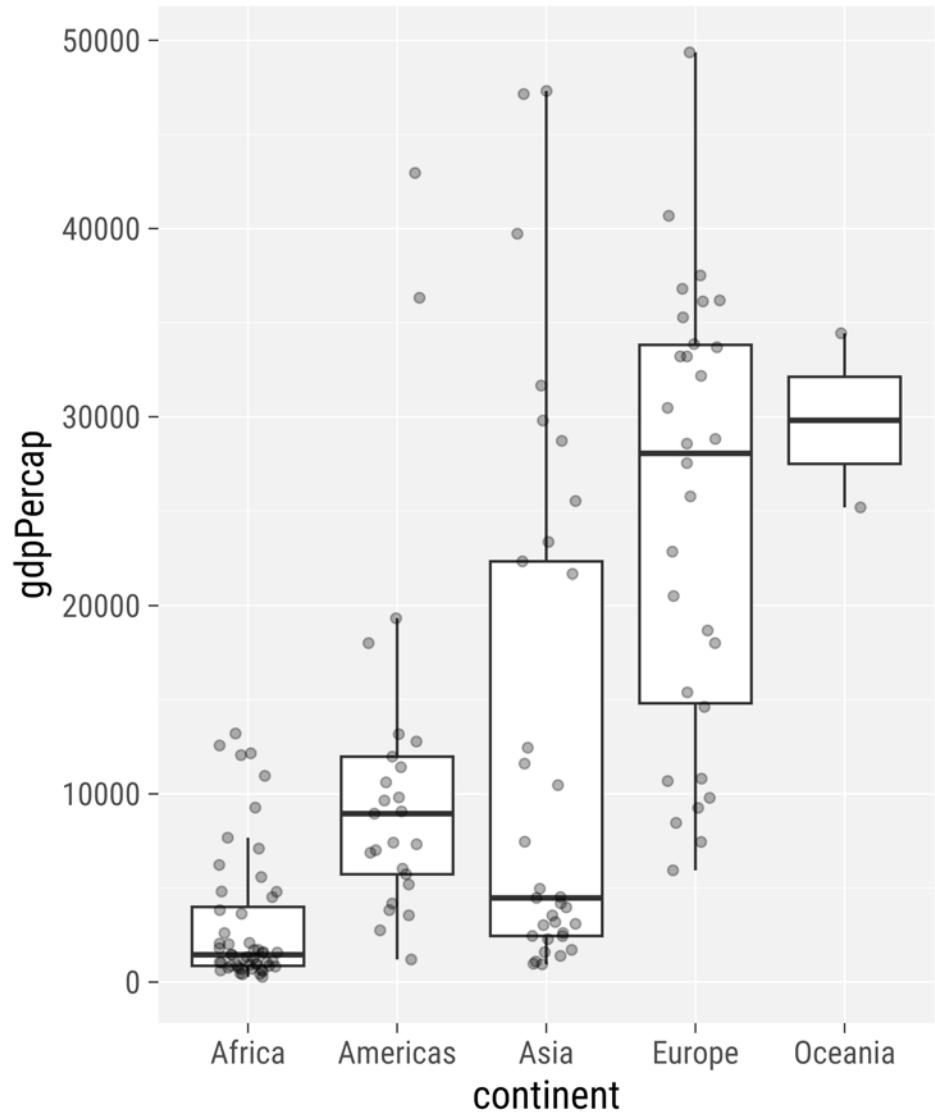
The position Argument

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = continent, y = gdpPercap)  
4 ) +  
5   geom_boxplot() +  
6   geom_jitter(  
7     width = .2,  
8     alpha = .3  
9 )
```



The position Argument

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = continent, y = gdpPercap)  
4 ) +  
5   geom_boxplot(  
6     outlier.shape = NA  
7 ) +  
8   geom_jitter(  
9     width = .2,  
10    alpha = .3  
11 )
```



The position Argument

Prepare data for a bar graph example:

```
1 gm_bars <- filter(  
2   gapminder,  
3   country %in% c("United States", "Canada"),  
4   year > 1999  
5 )
```



The position Argument

Prepare data for a bar graph example

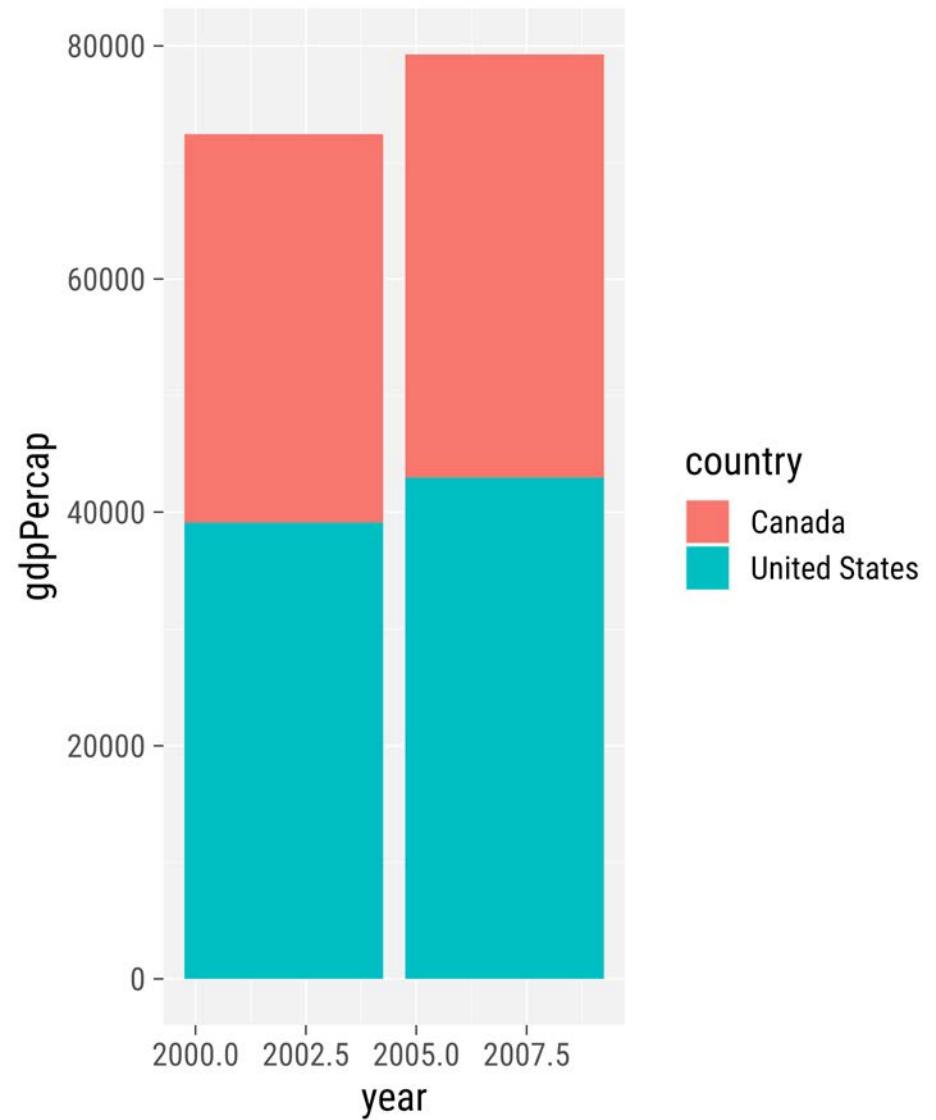
```
1 gm_bars <- filter(  
2   gapminder,  
3   country %in% c("United States", "Canada"),  
4   year > 1999  
5 )  
6  
7 gm_bars
```

```
# A tibble: 4 × 7  
country continent year lifeExp      pop gdpPercap income_lvl  
<chr>    <chr>    <dbl>    <dbl>      <dbl>     <dbl> <chr>  
1 Canada   Americas  2002     79.8  31902268    33329.  4  
2 Canada   Americas  2007     80.7  33390141    36319.  4  
3 United States Americas  2002     77.3  287675526   39097.  4  
4 United States Americas  2007     78.2  301139947   42952.  4
```



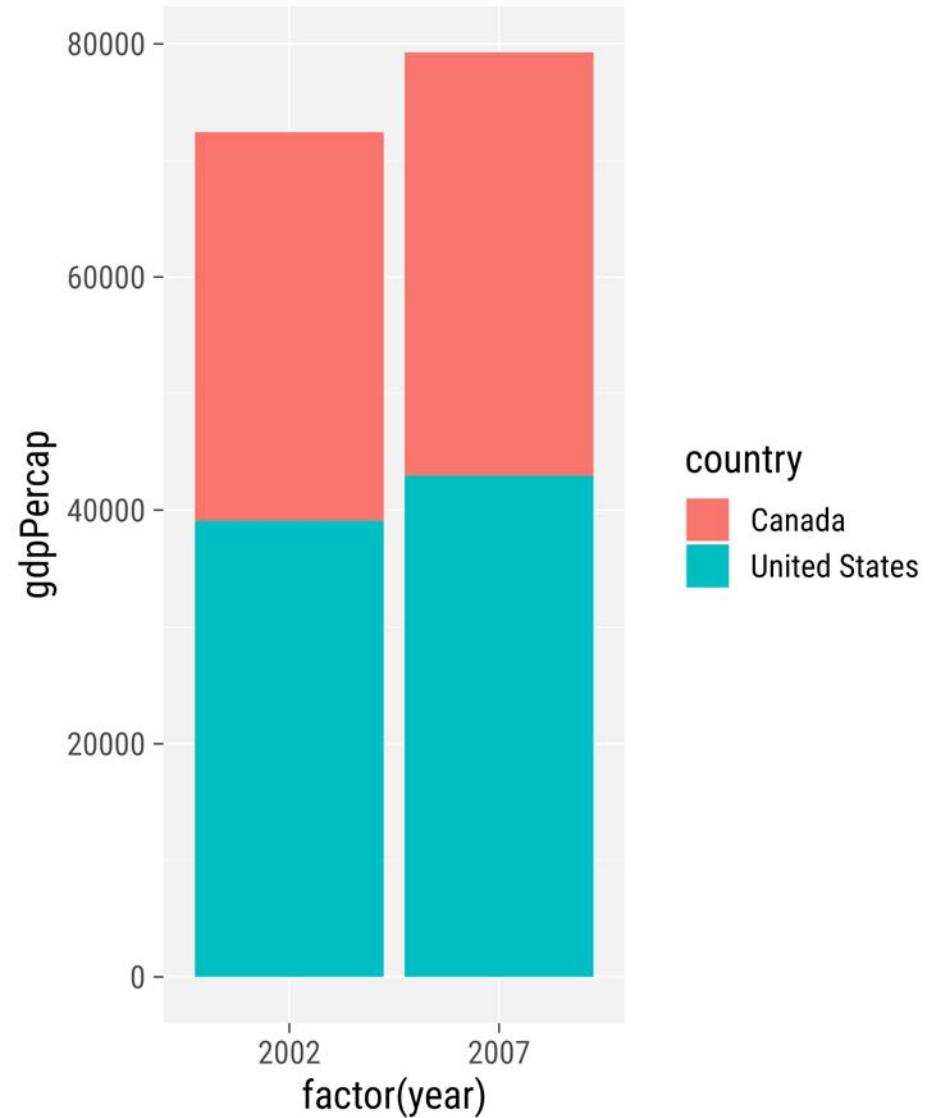
The position Argument

```
1 ggplot(  
2   data = gm_bars,  
3   aes(x = year, y = gdpPercap)  
4 ) +  
5 geom_col(  
6   aes(fill = country)  
7 )
```



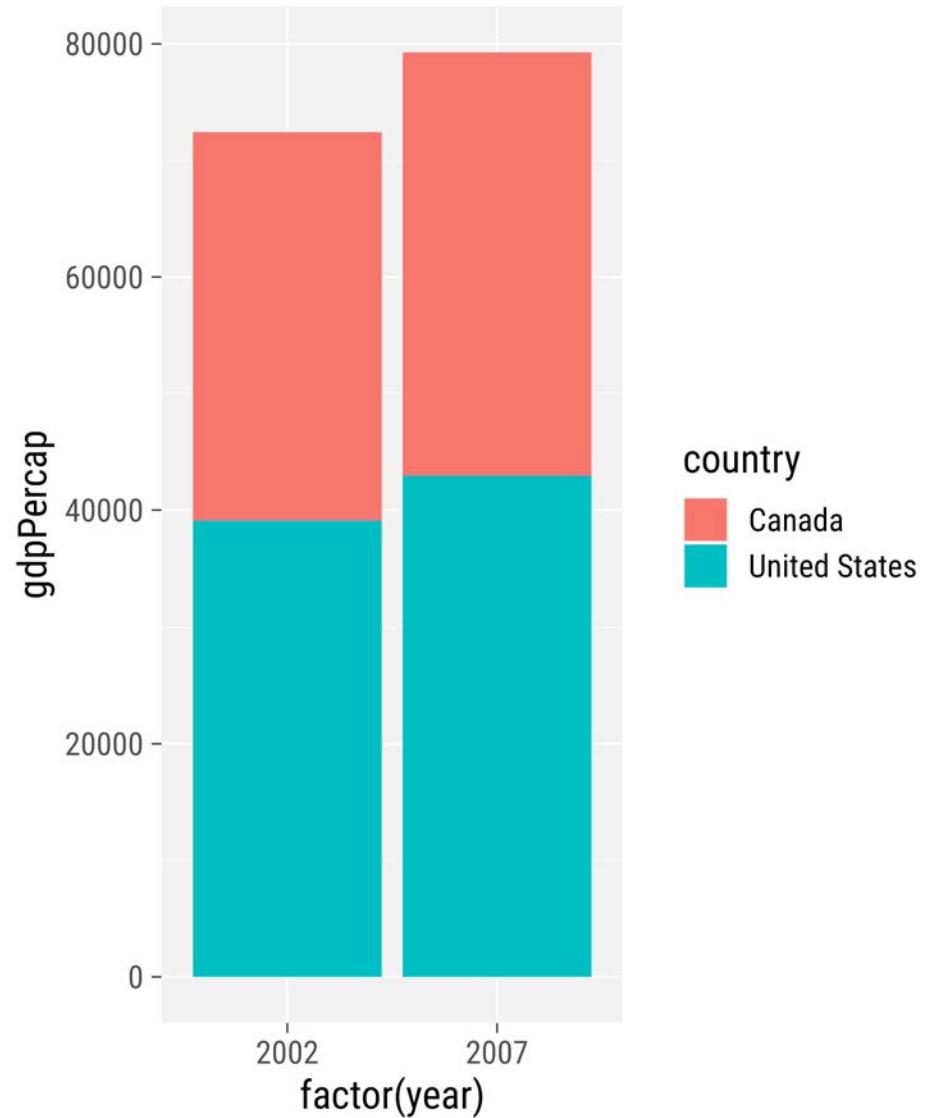
The position Argument

```
1 ggplot(  
2   data = gm_bars,  
3   aes(x = factor(year), y = gdpPerCap)  
4 ) +  
5 geom_col(  
6   aes(fill = country)  
7 )
```



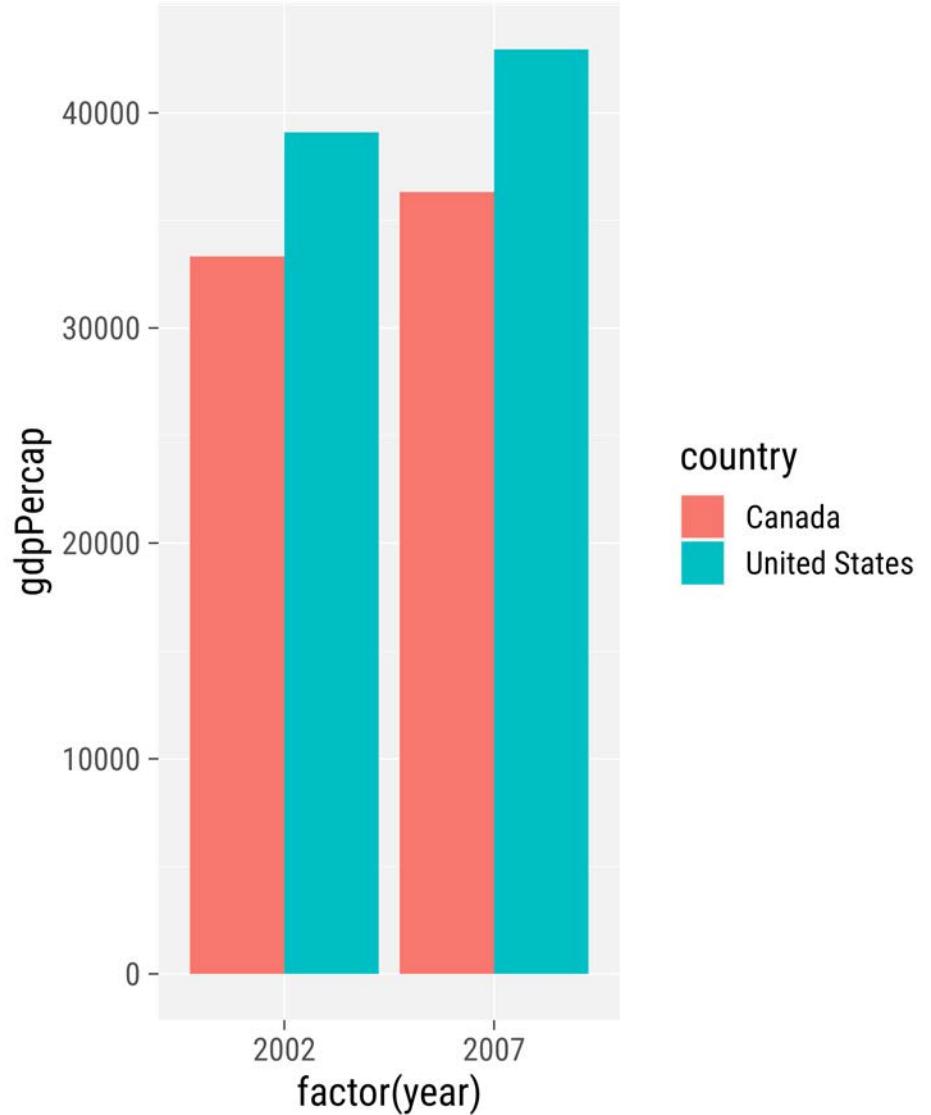
The position Argument

```
1 ggplot(  
2   data = gm_bars,  
3   aes(x = factor(year), y = gdpPerCap)  
4 ) +  
5 geom_col(  
6   aes(fill = country),  
7   position = "stack"  
8 )
```



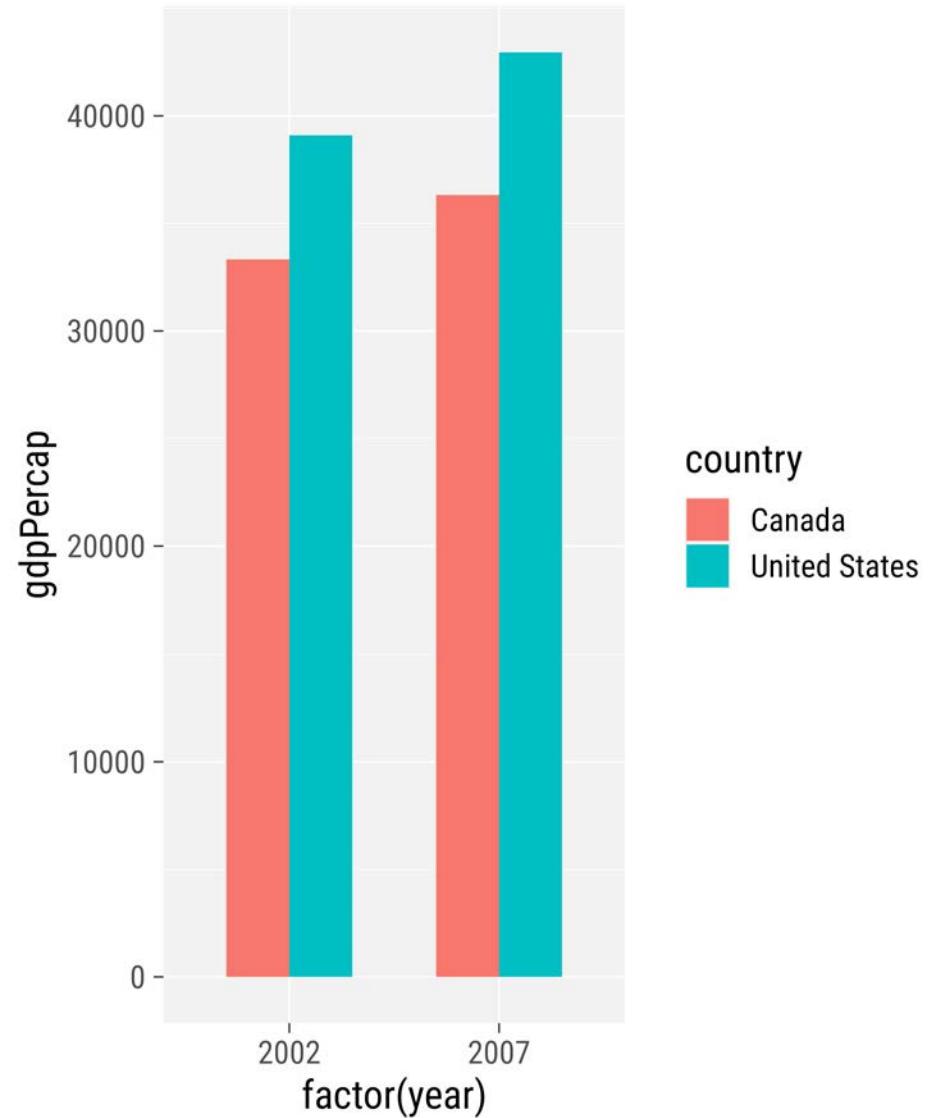
The position Argument

```
1 ggplot(  
2   data = gm_bars,  
3   aes(x = factor(year), y = gdpPerCap)  
4 ) +  
5 geom_col(  
6   aes(fill = country),  
7   position = "dodge"  
8 )
```



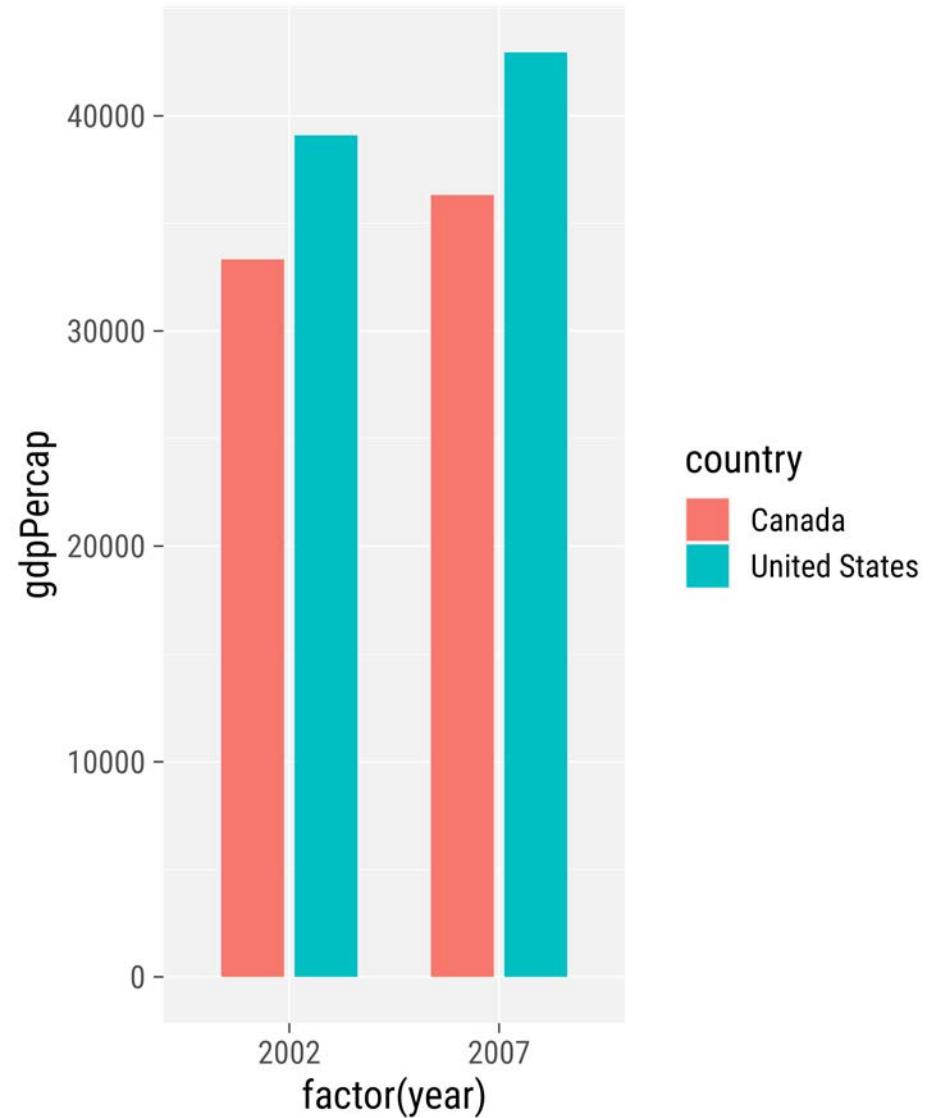
The position Argument

```
1 ggplot(  
2   data = gm_bars,  
3   aes(x = factor(year), y = gdpPerCap)  
4 ) +  
5 geom_col(  
6   aes(fill = country),  
7   position = "dodge",  
8   width = .6  
9 )
```



The position Argument

```
1 ggplot(  
2   data = gm_bars,  
3   aes(x = factor(year), y = gdpPerCap)  
4 ) +  
5 geom_col(  
6   aes(fill = country),  
7   position = position_dodge(  
8     width = .7  
9   ),  
10  width = .6  
11 )
```

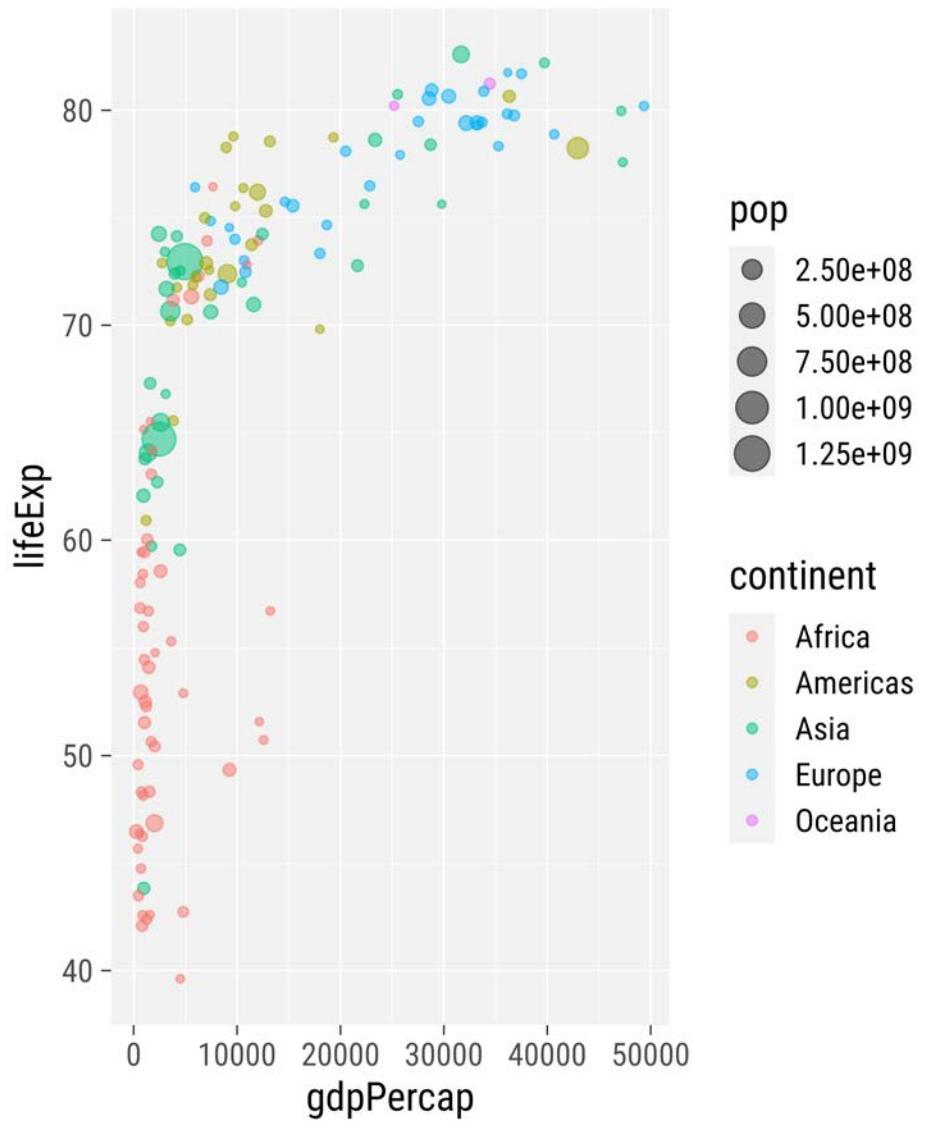


— Labels —



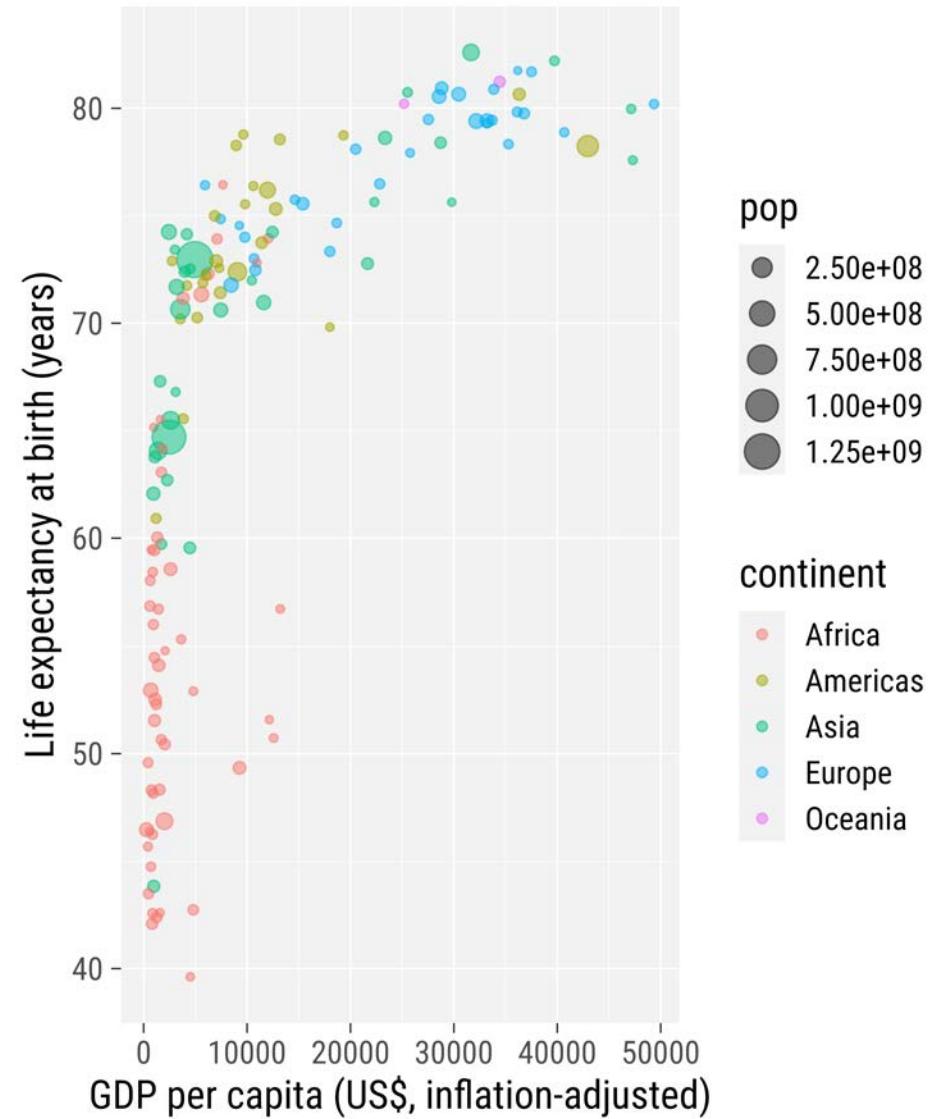
Labels

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp,  
4       color = continent,  
5       size = pop)  
6 ) +  
7   geom_point(  
8     alpha = .5  
9 )
```



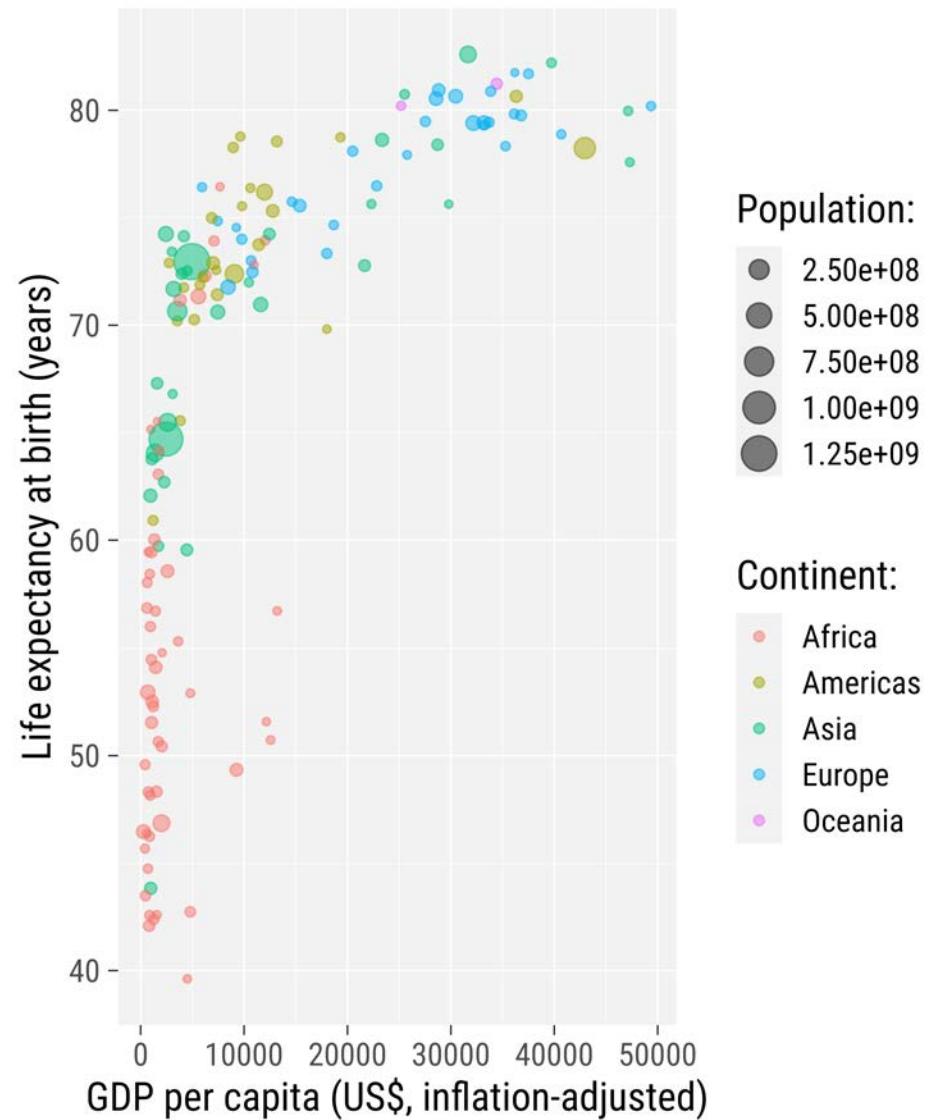
Overwrite Axis Titles

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp,  
4       color = continent,  
5       size = pop)  
6 ) +  
7   geom_point(  
8     alpha = .5  
9   ) +  
10  labs(  
11    x = "GDP per capita (US$, inflation-adjusted)",  
12    y = "Life expectancy at birth (years)"  
13  )
```



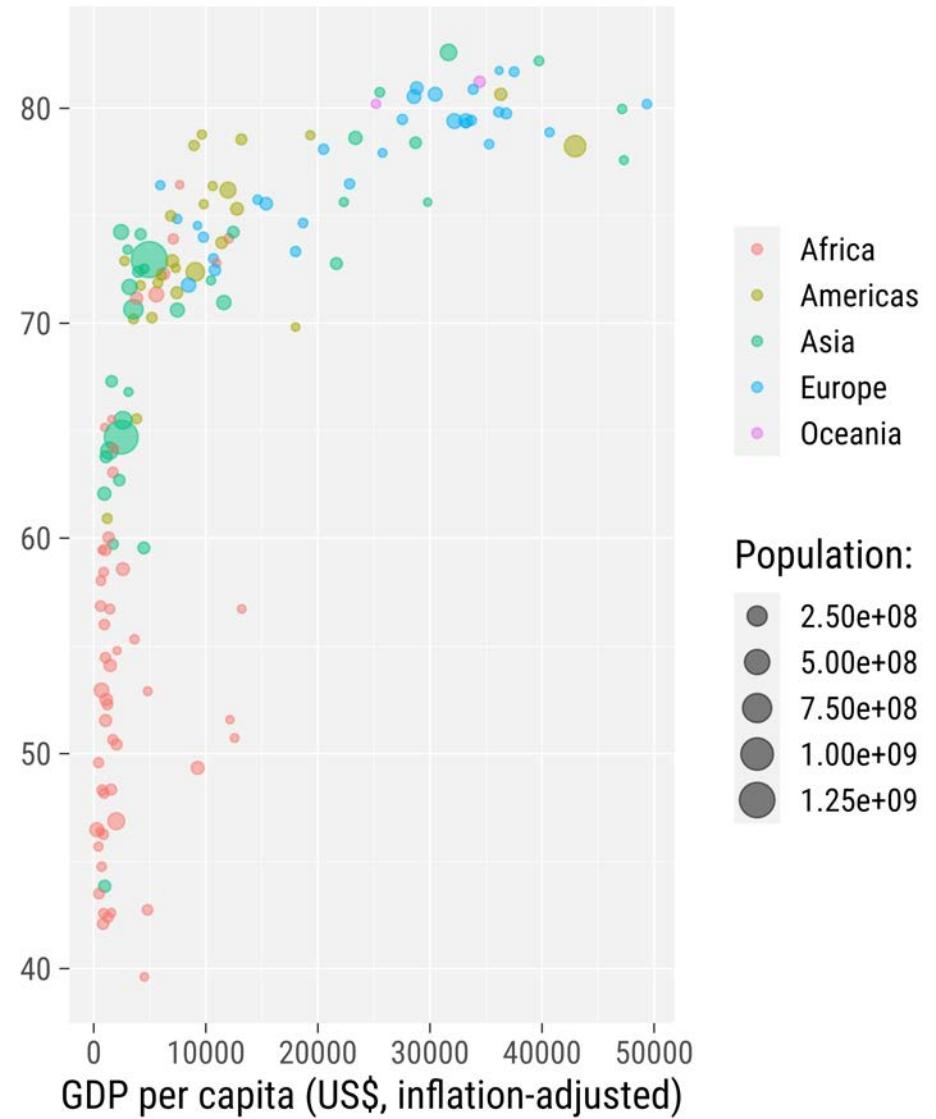
Overwrite Legend Titles

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp,  
4       color = continent,  
5       size = pop)  
6 ) +  
7   geom_point(  
8     alpha = .5  
9   ) +  
10  labs(  
11    x = "GDP per capita (US$, inflation-adjusted)",  
12    y = "Life expectancy at birth (years)",  
13    color = "Continent:",  
14    size = "Population:"  
15  )
```



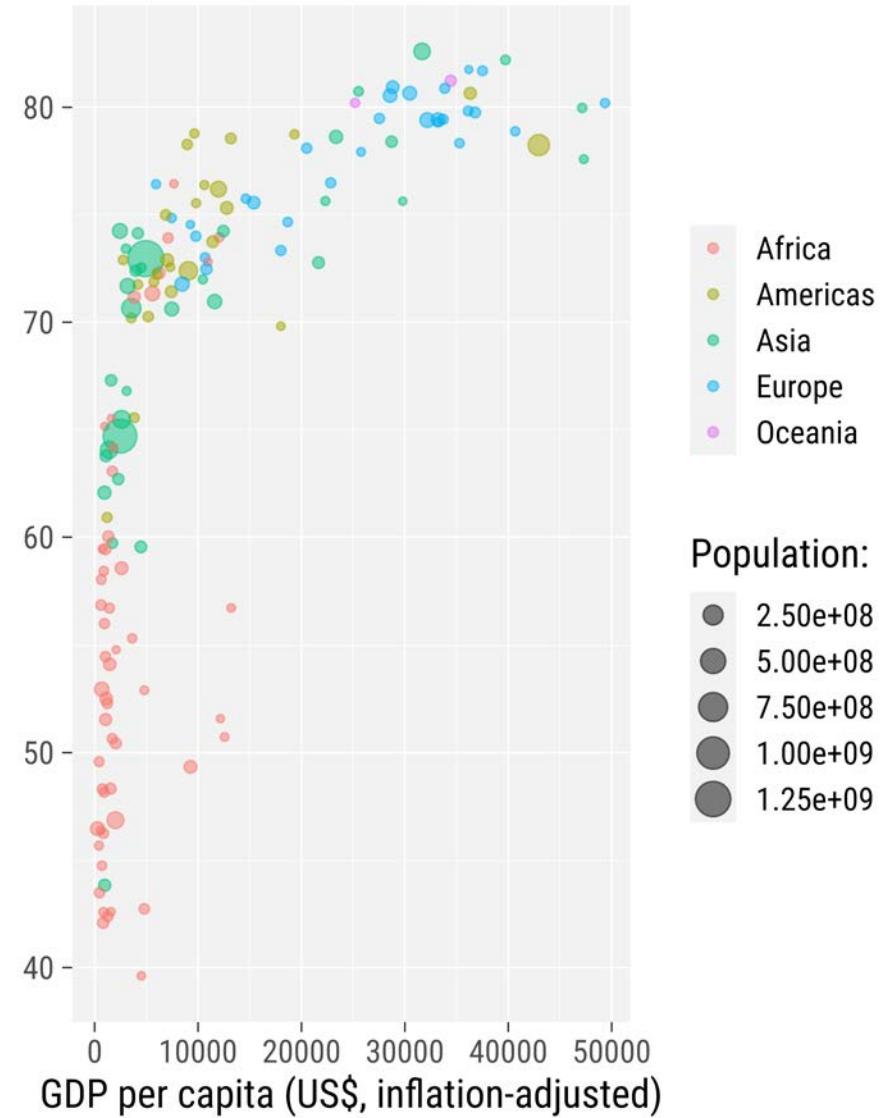
Remove Titles

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp,  
4       color = continent,  
5       size = pop)  
6 ) +  
7   geom_point(  
8     alpha = .5  
9   ) +  
10  labs(  
11    x = "GDP per capita (US$, inflation-adjusted)",  
12    y = NULL,  
13    color = NULL,  
14    size = "Population:"  
15  )
```



Remove Titles

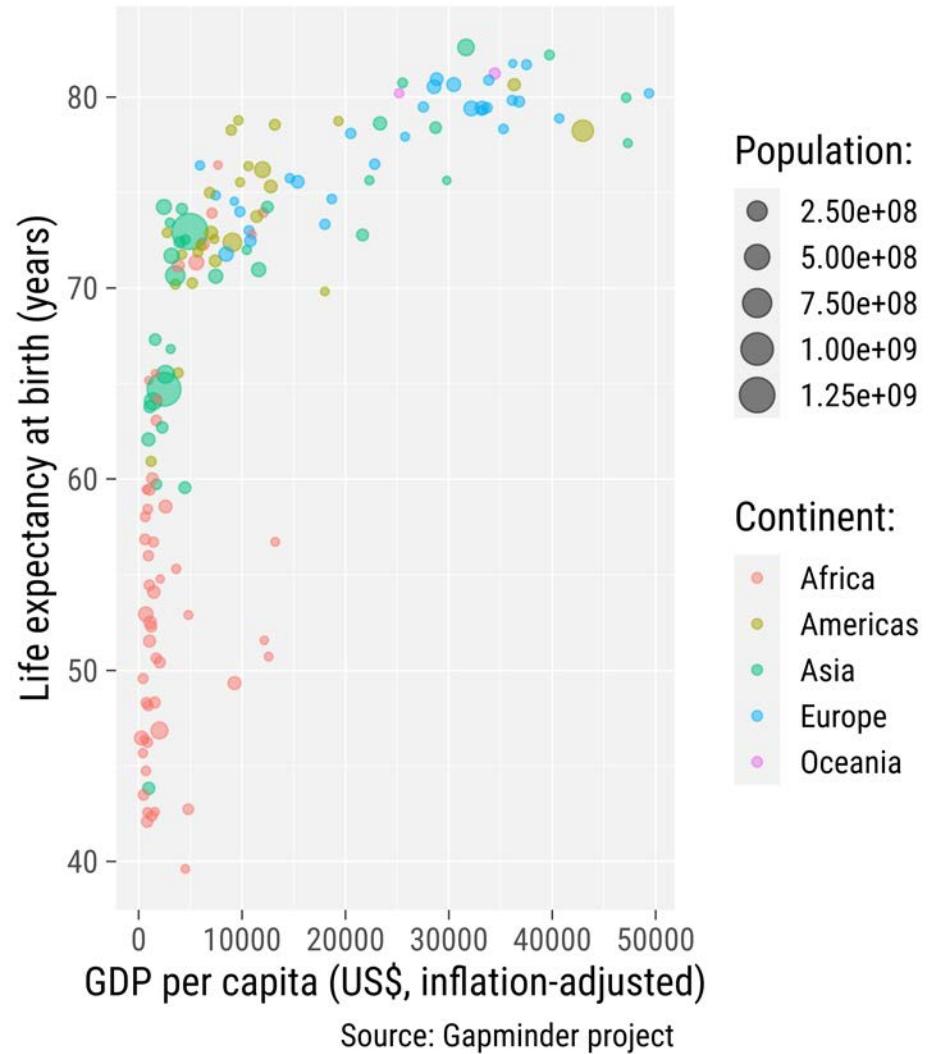
```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp,  
4       color = continent,  
5       size = pop)  
6 ) +  
7   geom_point(  
8     alpha = .5  
9   ) +  
10  labs(  
11    x = "GDP per capita (US$, inflation-adjusted)",  
12    y = "",  
13    color = NULL,  
14    size = "Population:"  
15  )
```



Add Plot Title + Caption

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp,  
4       color = continent,  
5       size = pop)  
6 ) +  
7   geom_point(  
8     alpha = .5  
9   ) +  
10  labs(  
11    x = "GDP per capita (US$, inflation-adjusted)",  
12    y = "Life expectancy at birth (years)",  
13    color = "Continent:",  
14    size = "Population:",  
15    title = "Health & Income of Nations in 2007",  
16    caption = "Source: Gapminder project"  
17 )
```

Health & Income of Nations in 2007



— Export Your Graphics —



Save the Graphic

```
1 ggsave(g, filename = "my_plot.png")
```

```
1 ggsave("my_plot.png")
```

```
1 ggsave("my_plot.png", width = 8, height = 5, dpi = 600)
```

```
1 ggsave("my_plot.pdf", width = 20, height = 12, unit = "cm", device = cairo_pdf)
```





Vector graphic

e.g. svg, pdf, eps



Raster graphic

e.g. png, jpeg, bmp, tiff

Modified from canva.com



How to Work with Aspect Ratios

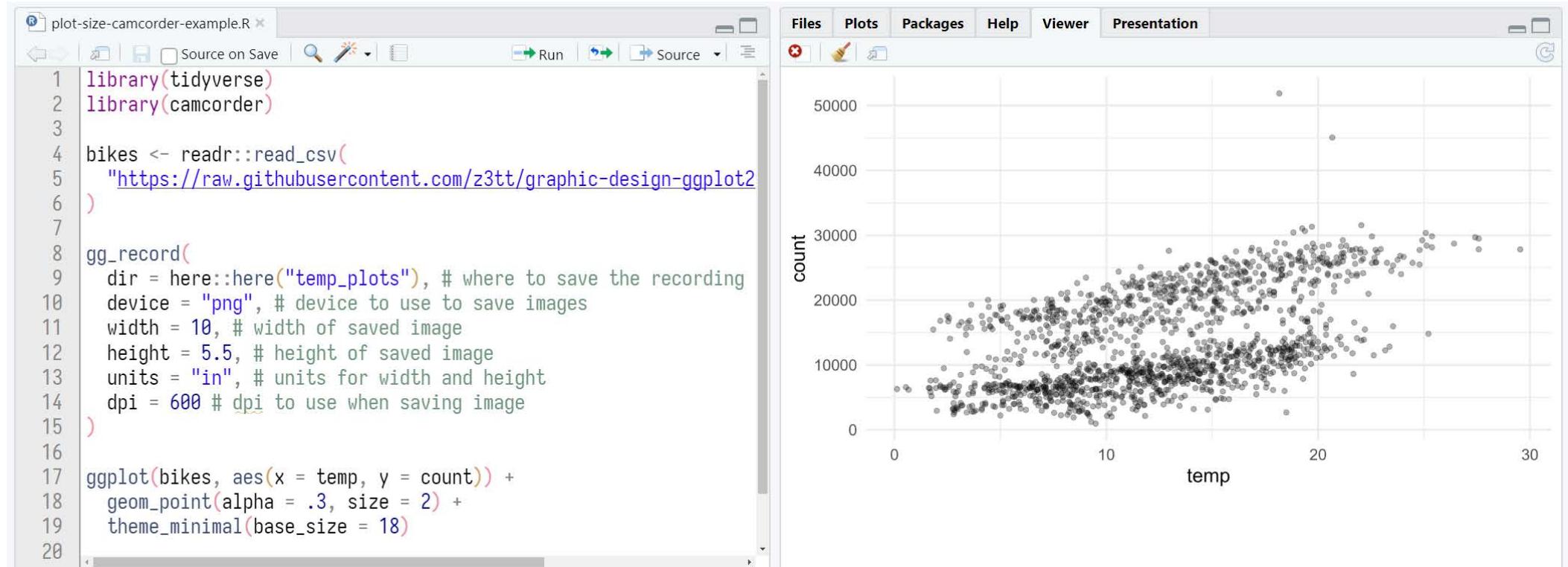
- don't rely on the Rstudio viewer pane!
- once you have a "it's getting close" prototype, settle on a plot size
- **Approach 1:** save the file to disk and inspect it; go back to your IDE
 - tedious and time-consuming...
- **Approach 2:** use a qmd or rmd with inline output and chunk settings
 - set `fig.width` and `fig.height` per chunk or globally
- **Approach 3:** use our `{camcorder}` package
 - saves output from all `ggplot()` calls and displays it in the viewer pane



Setting Plot Sizes in Rmd's



Setting Plot Sizes via {camcorder}



Exercise



Your Turn: A Basic ggplot

- Create a time series (line chart) of GDP per capita for Oceanian countries
 - What is the difference between `geom_line()` and `geom_path()`?
 - Map the color of the lines to `country`.
 - Add points for each observation.
 - Encode the points with the same colors.
 - Turn the points into diamonds.
 - Add meaningful titles to your plot.
 - **Bonus:** Use different point shapes for each country.



Your Turn: Data Preparation

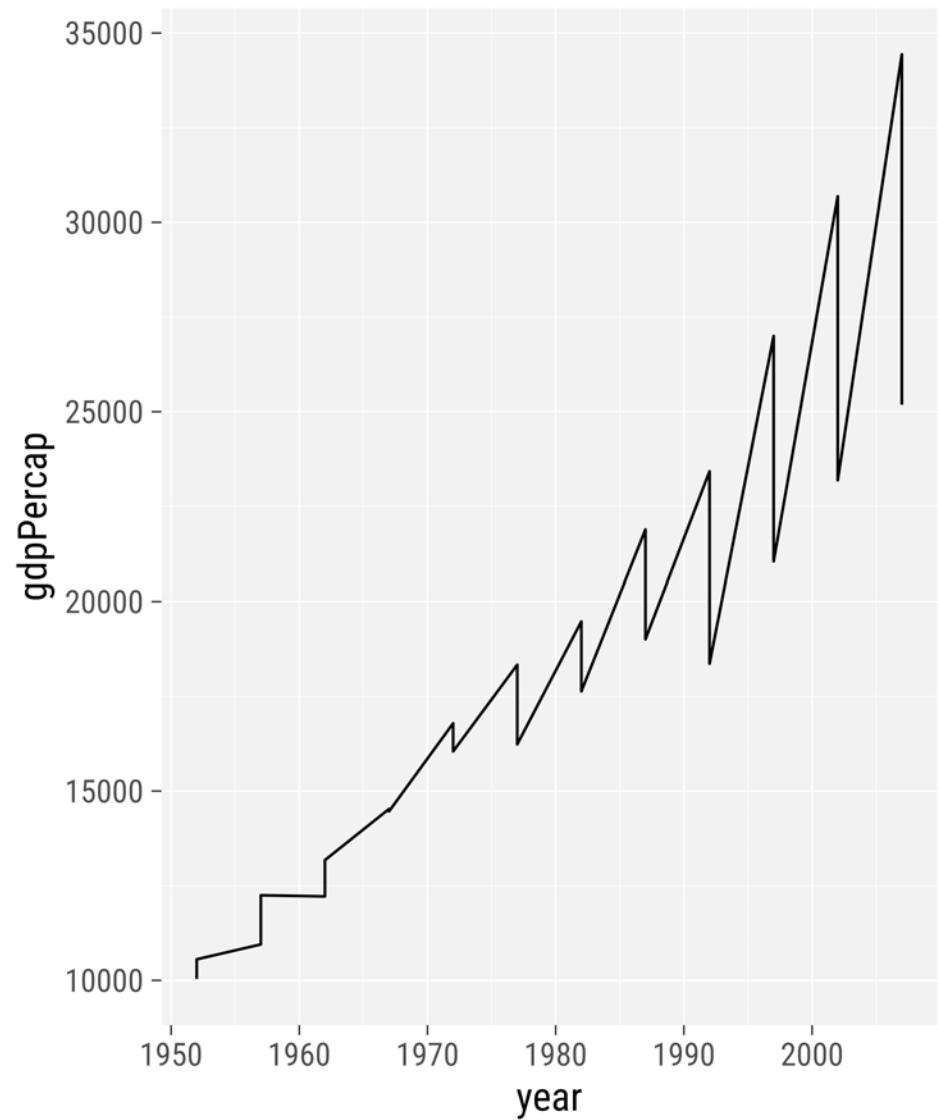
```
1 gm_oceania <- readr::read_csv("./data/gapminder_oceania.csv")
```

```
1 gm_oceania <- filter(gapminder, continent == "Oceania")
```

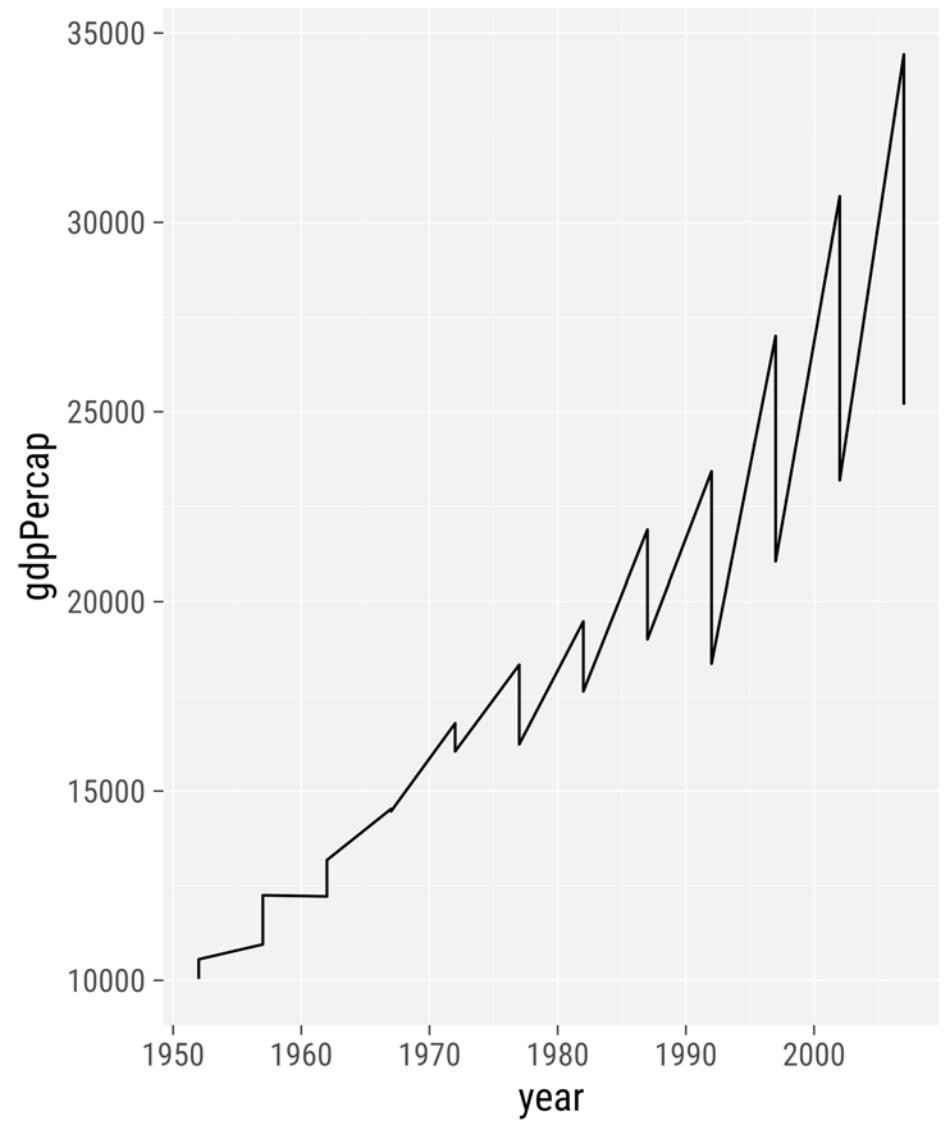


Your Turn: Geometrical Layers

```
1 ggplot(  
2   data = gm_oceania,  
3   aes(x = year, y = gdpPercap)  
4 ) +  
5 geom_line()
```

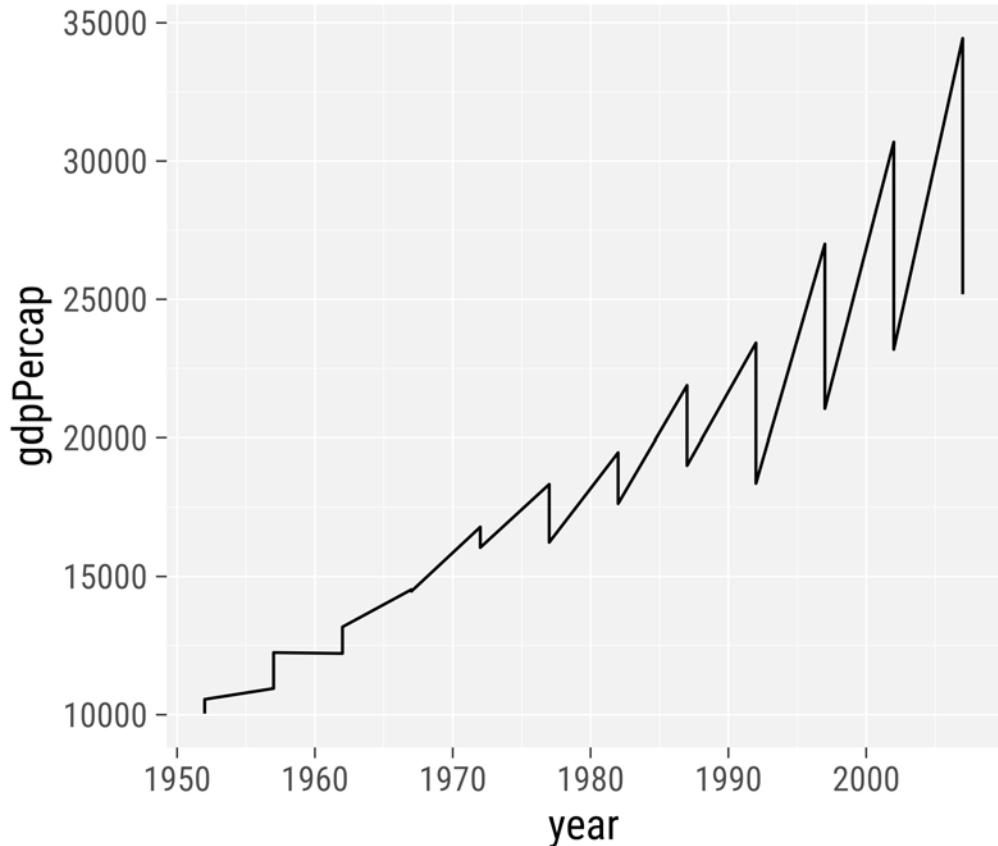


```
1 ggplot(  
2   data = filter(  
3     gapminder, continent == "Oceania"  
4   ),  
5   aes(x = year, y = gdpPercap)  
6 ) +  
7 geom_line()
```

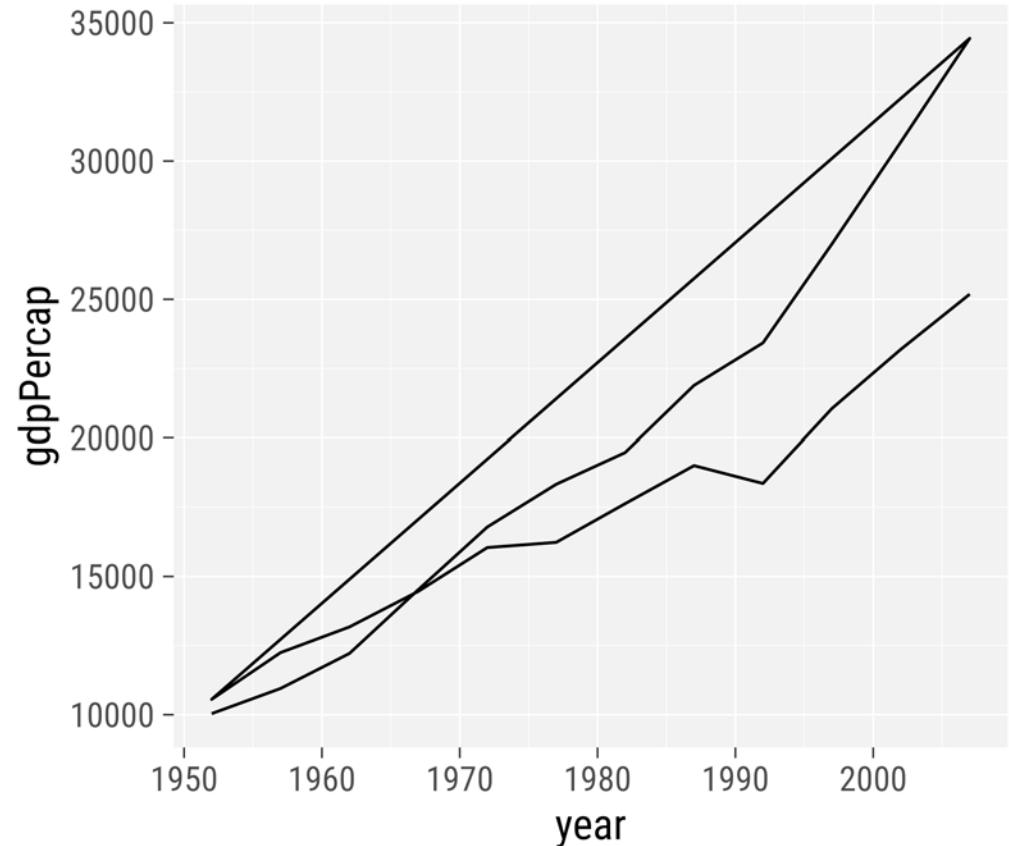


Your Turn: Geometrical Layers

```
1 ggplot(  
2   data = gm_oceania,  
3   aes(x = year, y = gdpPercap)  
4 ) +  
5 geom_line()
```

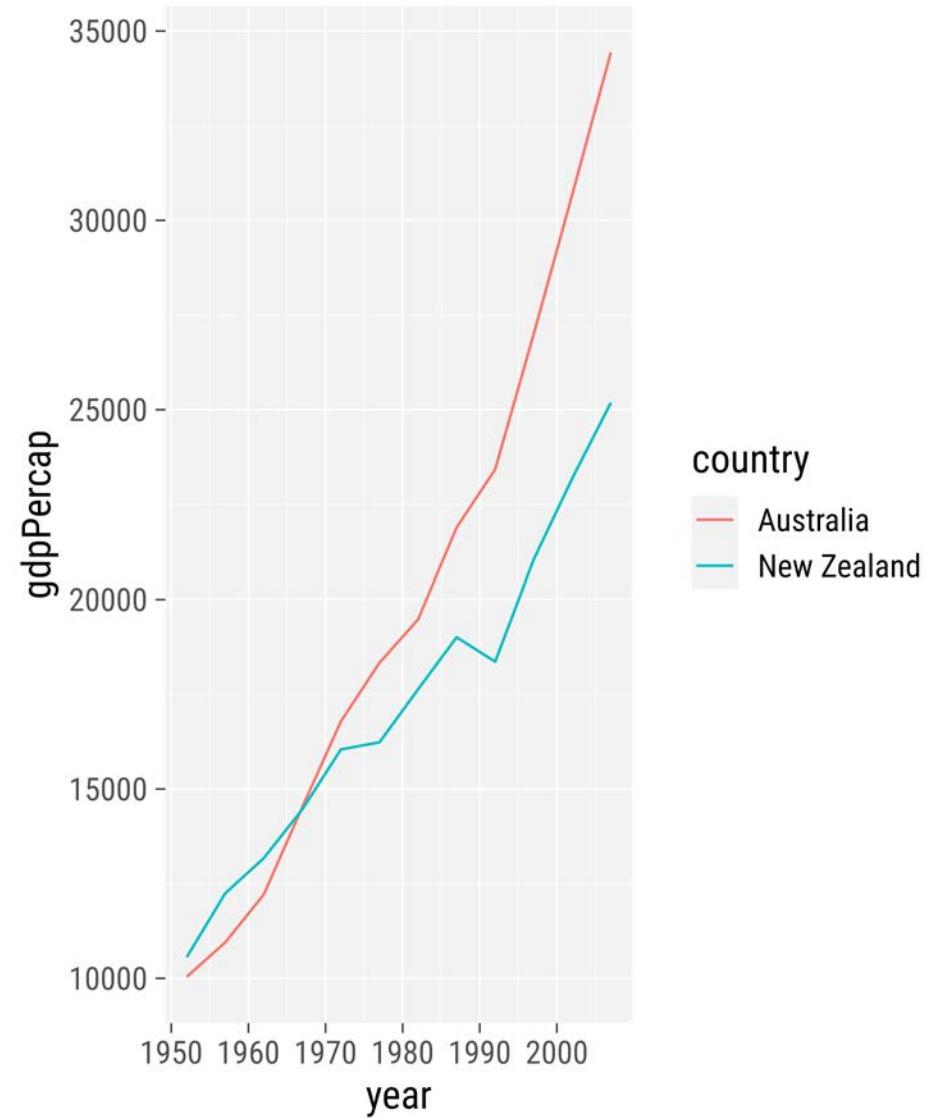


```
1 ggplot(  
2   data = gm_oceania,  
3   aes(x = year, y = gdpPercap)  
4 ) +  
5 geom_path()
```



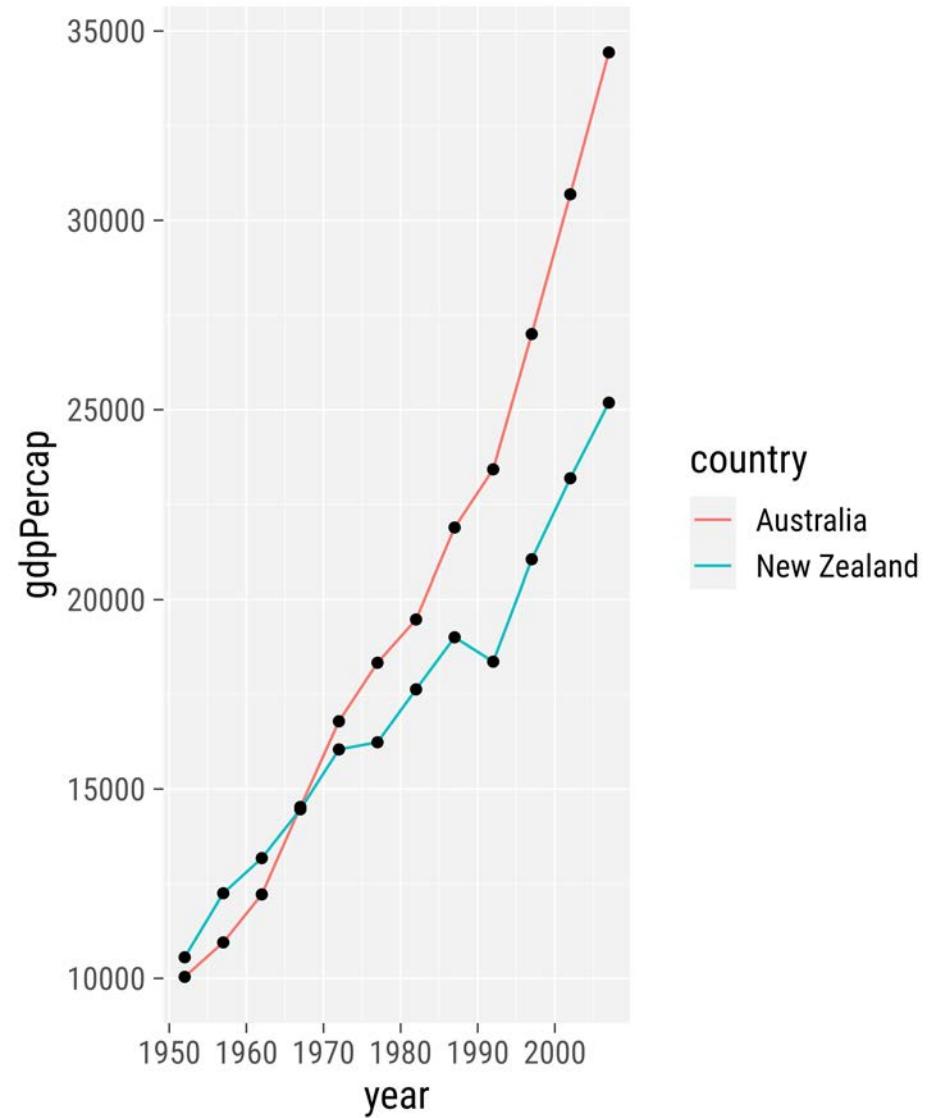
Your Turn: Aesthetics

```
1 ggplot(  
2   data = gm_oceania,  
3   aes(x = year, y = gdpPercap)  
4 ) +  
5 geom_line(  
6   aes(color = country)  
7 )
```



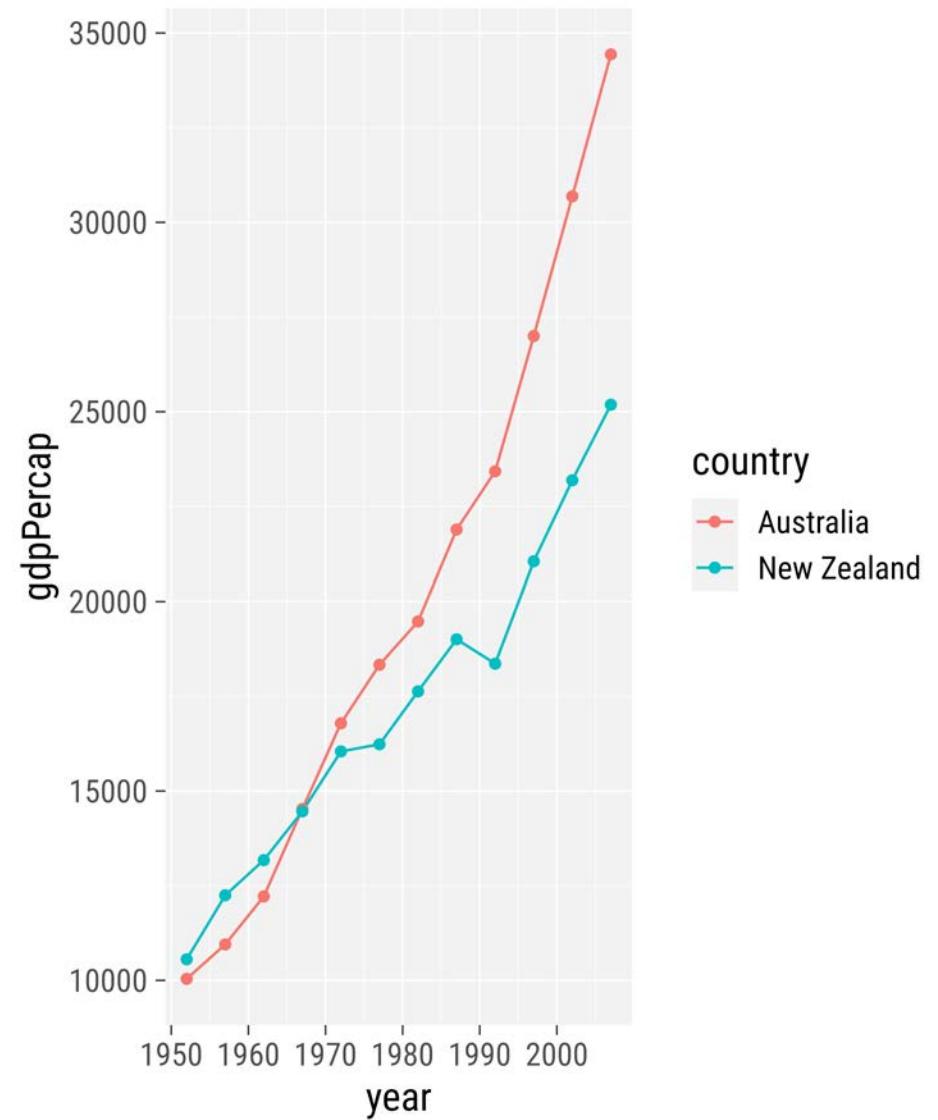
Your Turn: Combine Geometries

```
1 ggplot(  
2   data = gm_oceania,  
3   aes(x = year, y = gdpPercap)  
4 ) +  
5   geom_line(  
6     aes(color = country)  
7 ) +  
8   geom_point()
```



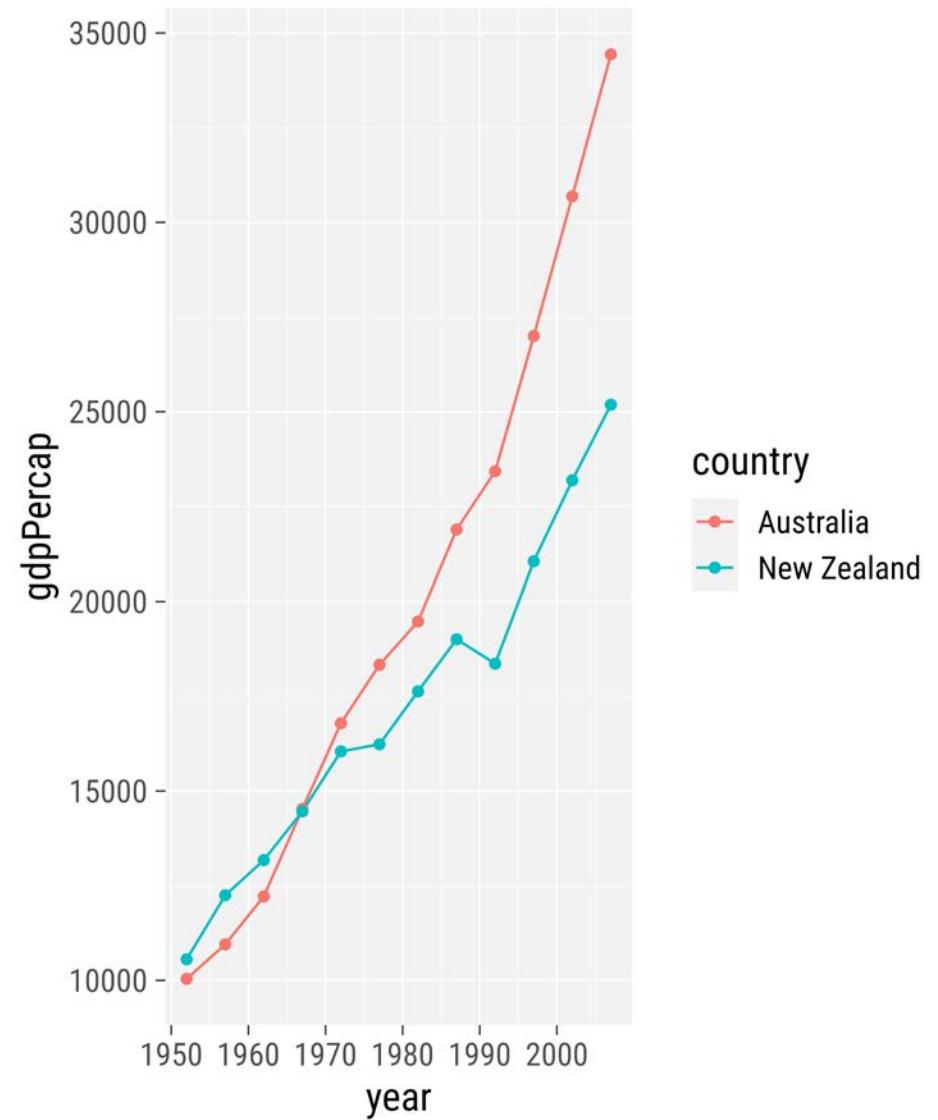
Your Turn: Local vs Global Aesthetics

```
1 ggplot(  
2   data = gm_oceania,  
3   aes(x = year, y = gdpPercap)  
4 ) +  
5   geom_line(  
6     aes(color = country)  
7 ) +  
8   geom_point(  
9     aes(color = country)  
10 )
```



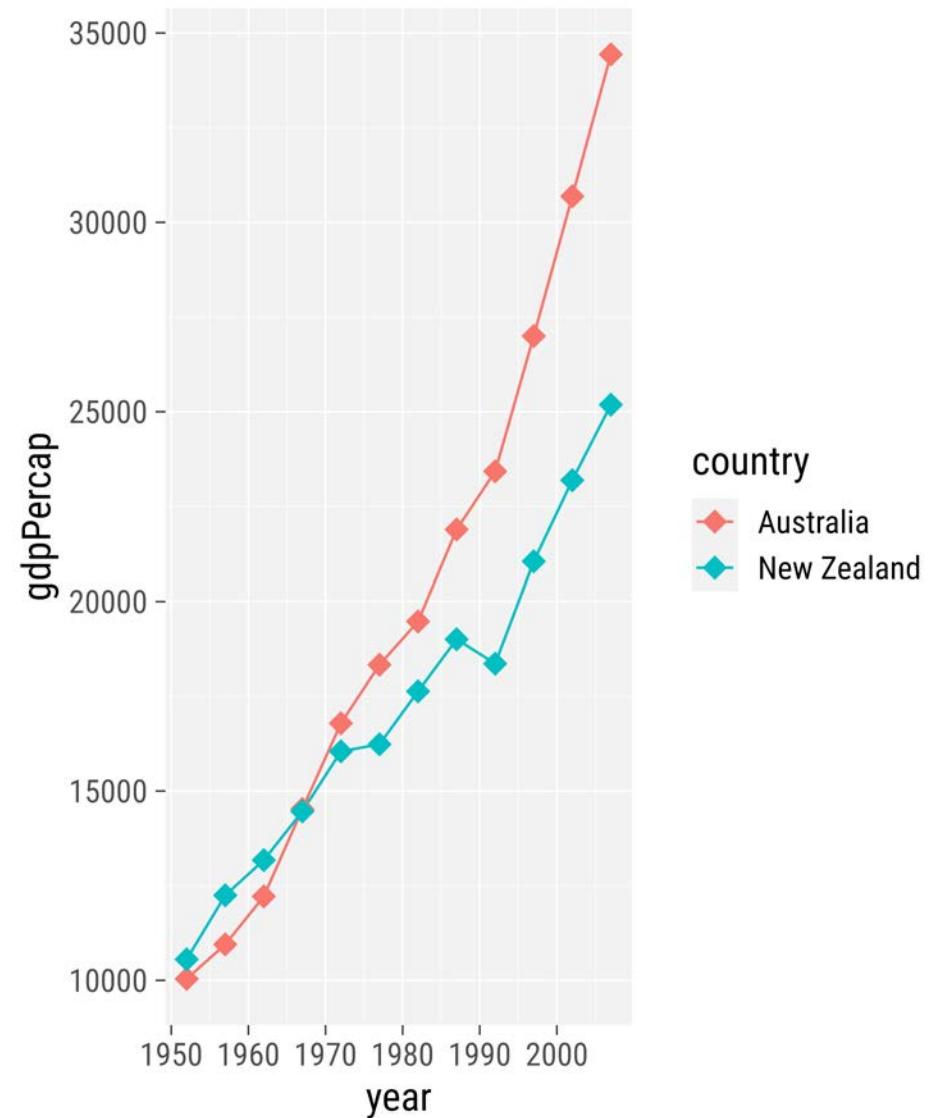
Your Turn: Local vs Global Aesthetics

```
1 ggplot(  
2   data = gm_oceania,  
3   aes(x = year, y = gdpPercap,  
4       color = country)  
5 ) +  
6 geom_line() +  
7 geom_point()
```



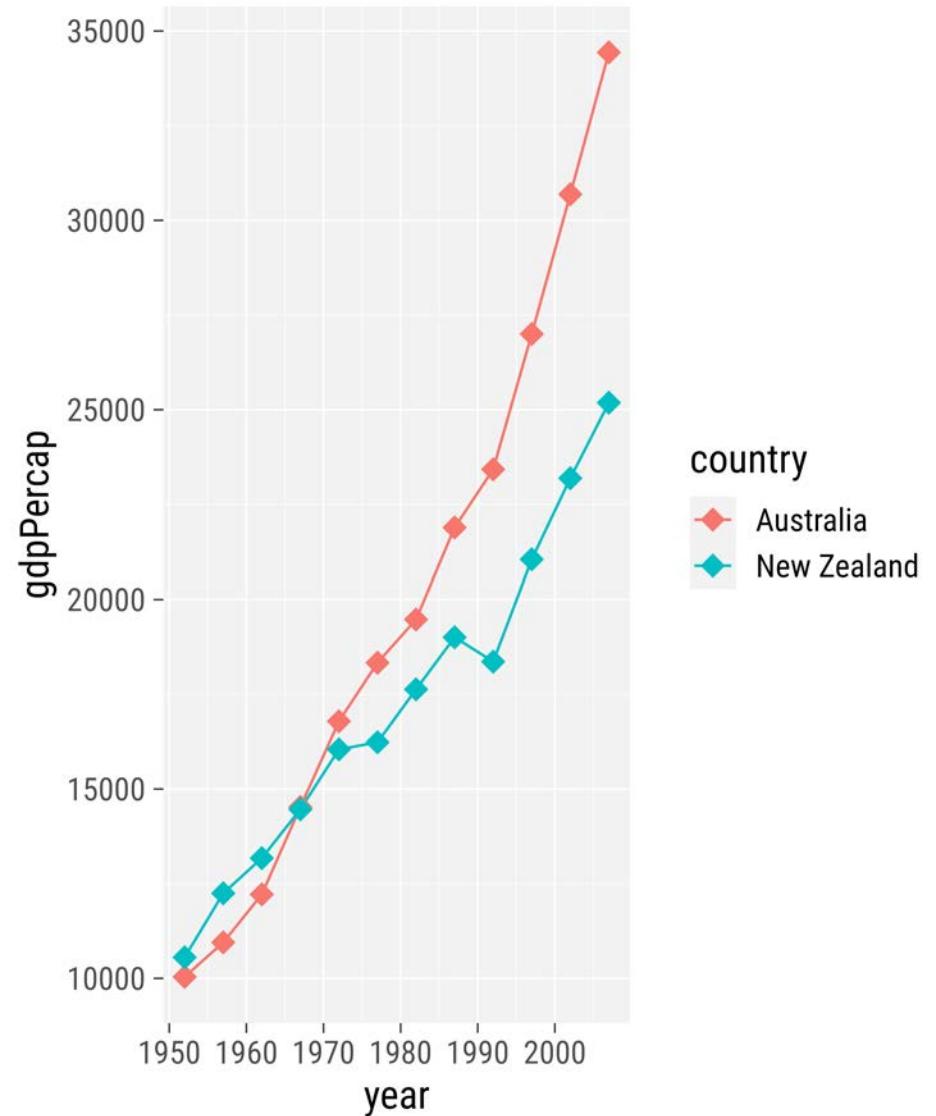
Your Turn: Properties of Layers

```
1 ggplot(  
2   data = gm_oceania,  
3   aes(x = year, y = gdpPercap,  
4       color = country)  
5 ) +  
6 geom_line() +  
7 geom_point(  
8   shape = "diamond",  
9   size = 4  
10 )
```



Your Turn: Properties of Layers

```
1 ggplot(  
2   data = gm_oceania,  
3   aes(x = year, y = gdpPercap,  
4       color = country)  
5 ) +  
6 geom_line() +  
7 geom_point(  
8   shape = 18,  
9   size = 4  
10 )
```



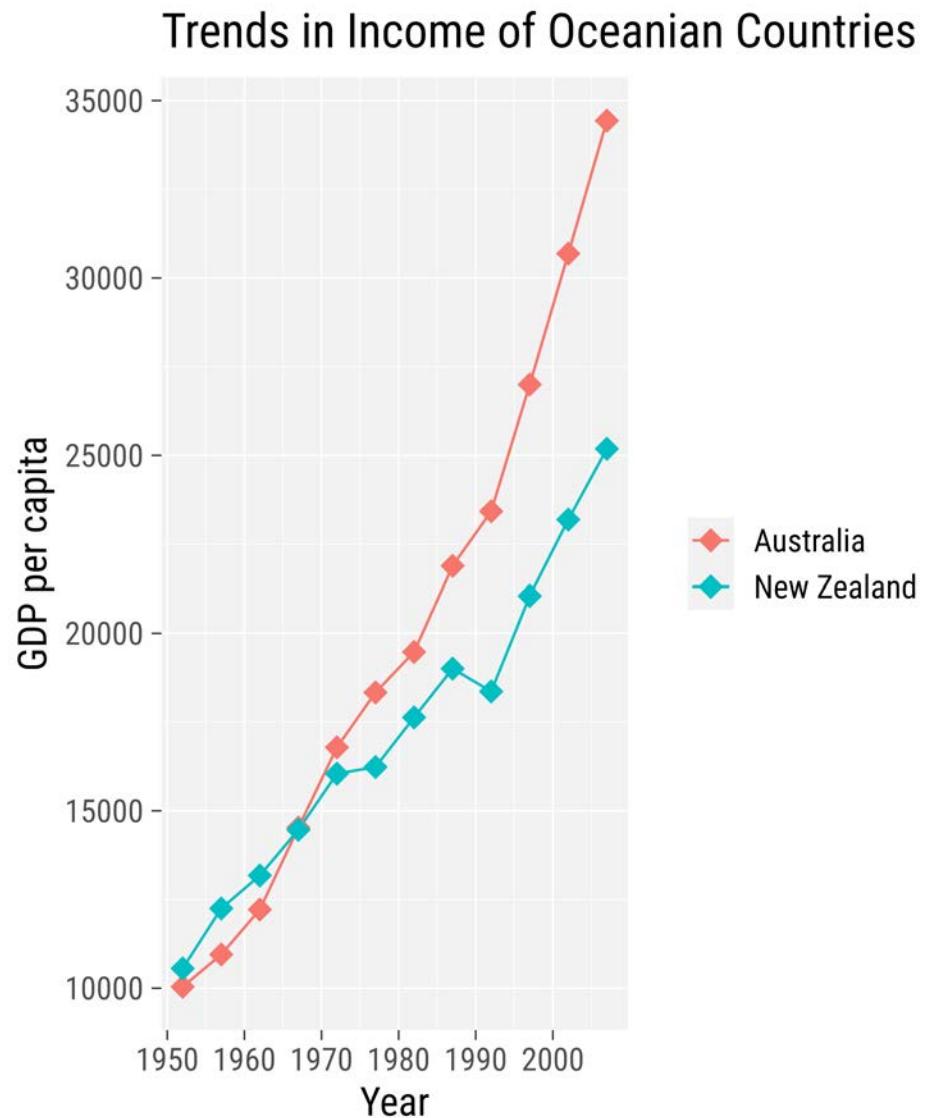
Circles	1 	10 	13 	16 	19 	20 	21
Triangles	2 	6 	17 	24 	25 		
Diamonds	5 	9 	18 	23 			
Squares	0 	7 	12 	14 	15 	22 	
Other	3 	4 	8 	11 			

Source: Albert's Blog



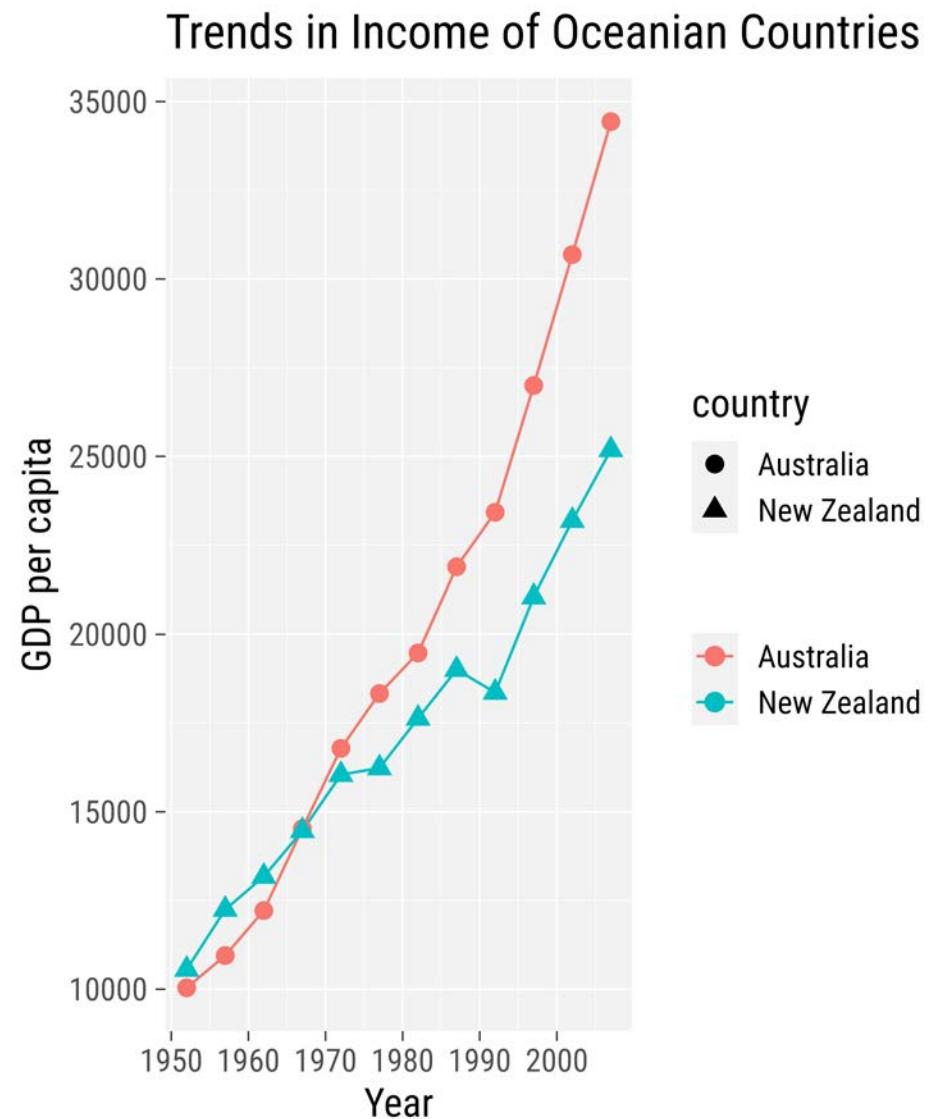
Your Turn: Labels

```
1 ggplot(  
2   data = gm_oceania,  
3   aes(x = year, y = gdpPercap,  
4       color = country)  
5 ) +  
6   geom_line() +  
7   geom_point(  
8     shape = 18,  
9     size = 4  
10 ) +  
11 labs(  
12   x = "Year",  
13   y = "GDP per capita",  
14   color = NULL,  
15   title = "Trends in Income of Oceanian Cou  
16 )
```



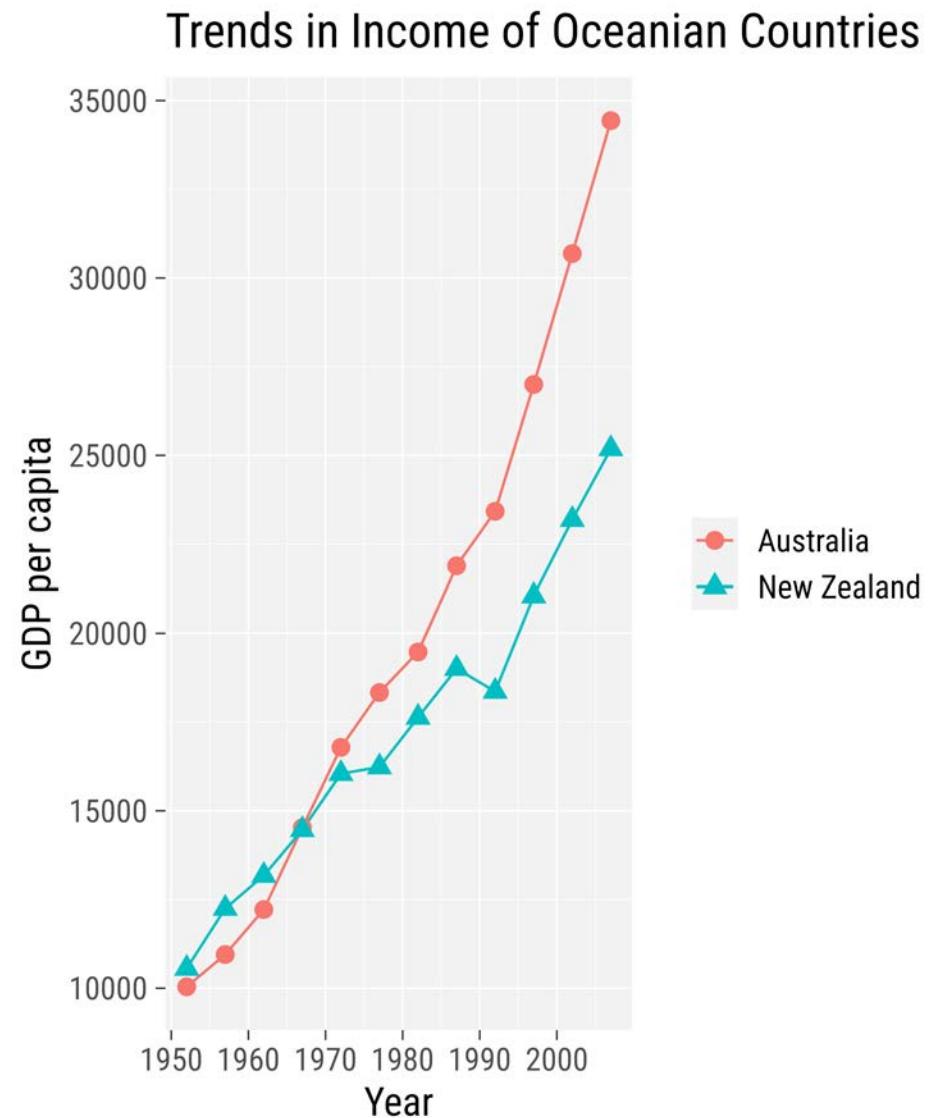
Your Turn: Aesthetics (Bonus)

```
1 ggplot(  
2   data = gm_oceania,  
3   aes(x = year, y = gdpPercap,  
4       color = country)  
5 ) +  
6   geom_line() +  
7   geom_point(  
8     aes(shape = country),  
9     size = 3  
10 ) +  
11 labs(  
12   x = "Year",  
13   y = "GDP per capita",  
14   color = NULL,  
15   title = "Trends in Income of Oceanian Cou  
16 )
```



Your Turn: Aesthetics (Bonus)

```
1 ggplot(  
2   data = gm_oceania,  
3   aes(x = year, y = gdpPercap,  
4       color = country)  
5 ) +  
6   geom_line() +  
7   geom_point(  
8     aes(shape = country),  
9     size = 3  
10 ) +  
11 labs(  
12   x = "Year",  
13   y = "GDP per capita",  
14   color = NULL,  
15   shape = NULL,  
16   title = "Trends in Income of Oceanian Cou  
17 )
```



Break
— 10 minutes —



A Polished *ggplot*



Store a ggplot as Object

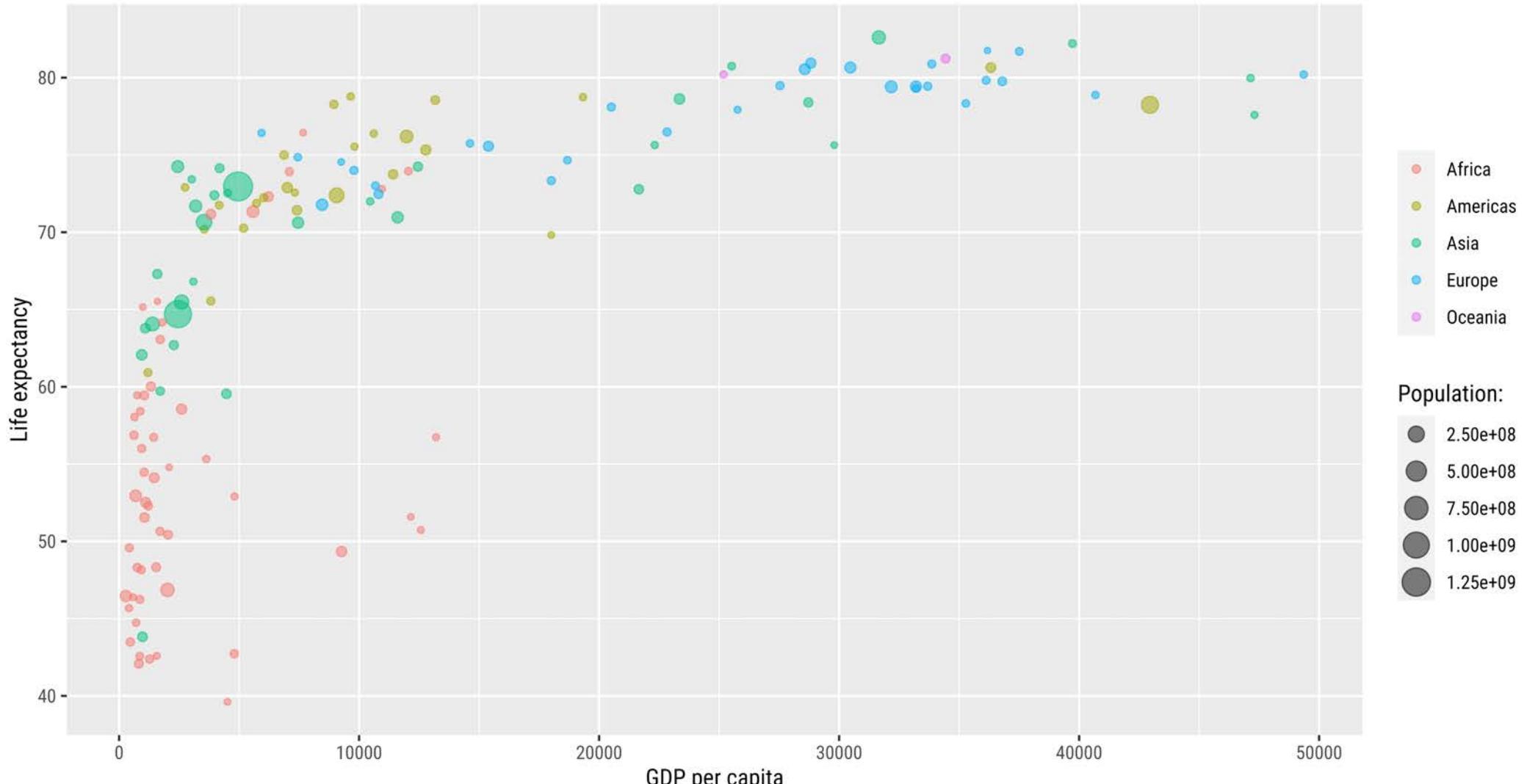
```
1 g <-  
2   ggplot(  
3     data = gm2007,  
4     aes(x = gdpPercap, y = lifeExp,  
5           color = continent,  
6           size = pop)  
7   ) +  
8   geom_point(  
9     alpha = .5  
10  ) +  
11  labs(  
12    x = "GDP per capita",  
13    y = "Life expectancy",  
14    color = NULL,  
15    size = "Population:",  
16    title = "Health & Income of Nations in 2007"  
17  )  
18  
19 class(g)  
[1] "gg"     "ggplot"
```



Print a ggplot Object

```
1 g
```

Health & Income of Nations in 2007



— Theming —



Theming

= **stylistic changes of non-data elements**

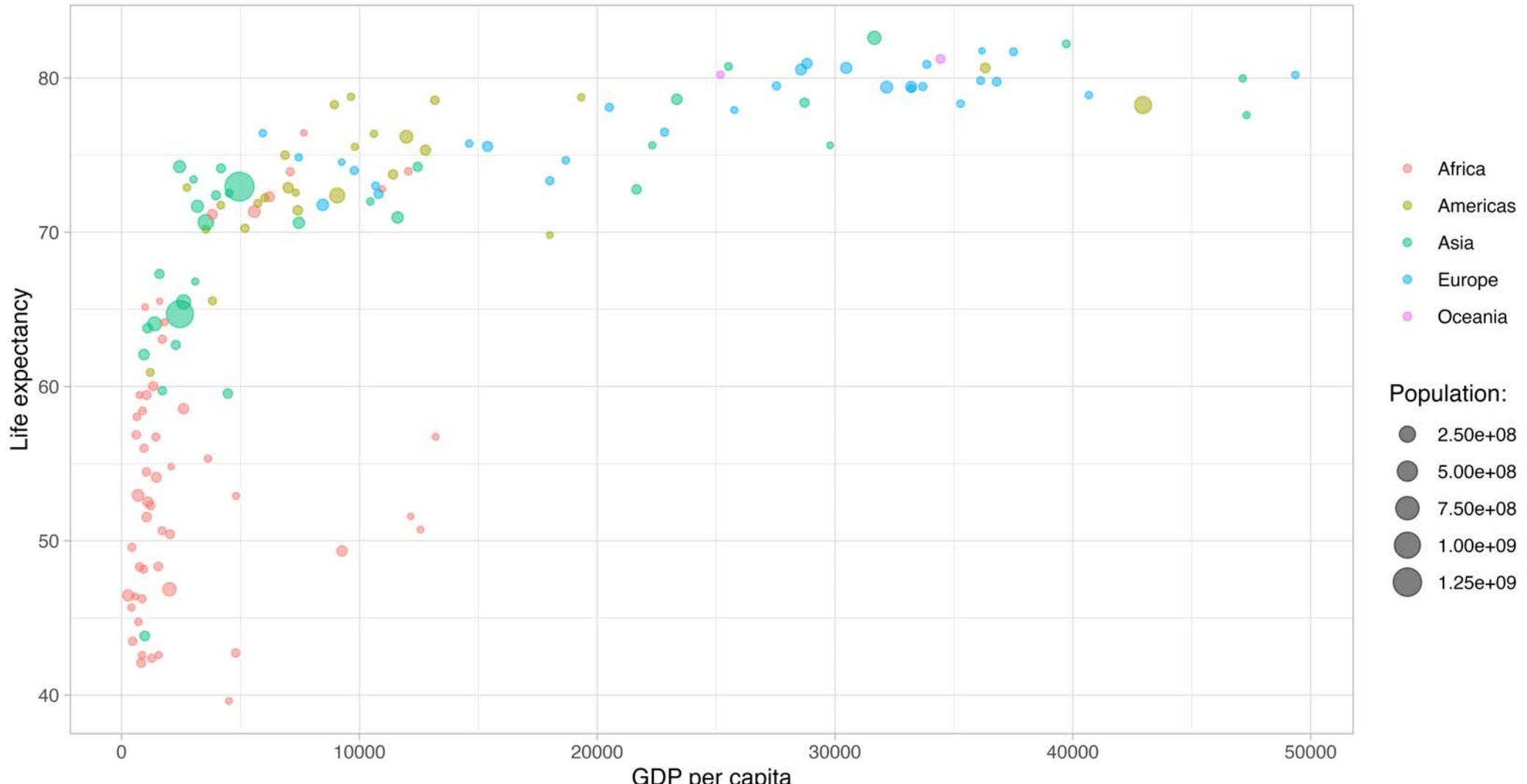
- complete themes plus custom theme adjustments
- either add complete themes to a plot or set a theme globally via `theme_*`()
- afterwards, theme settings can be overwritten via `theme()`



Add a Complete Theme

```
1 g + theme_light()
```

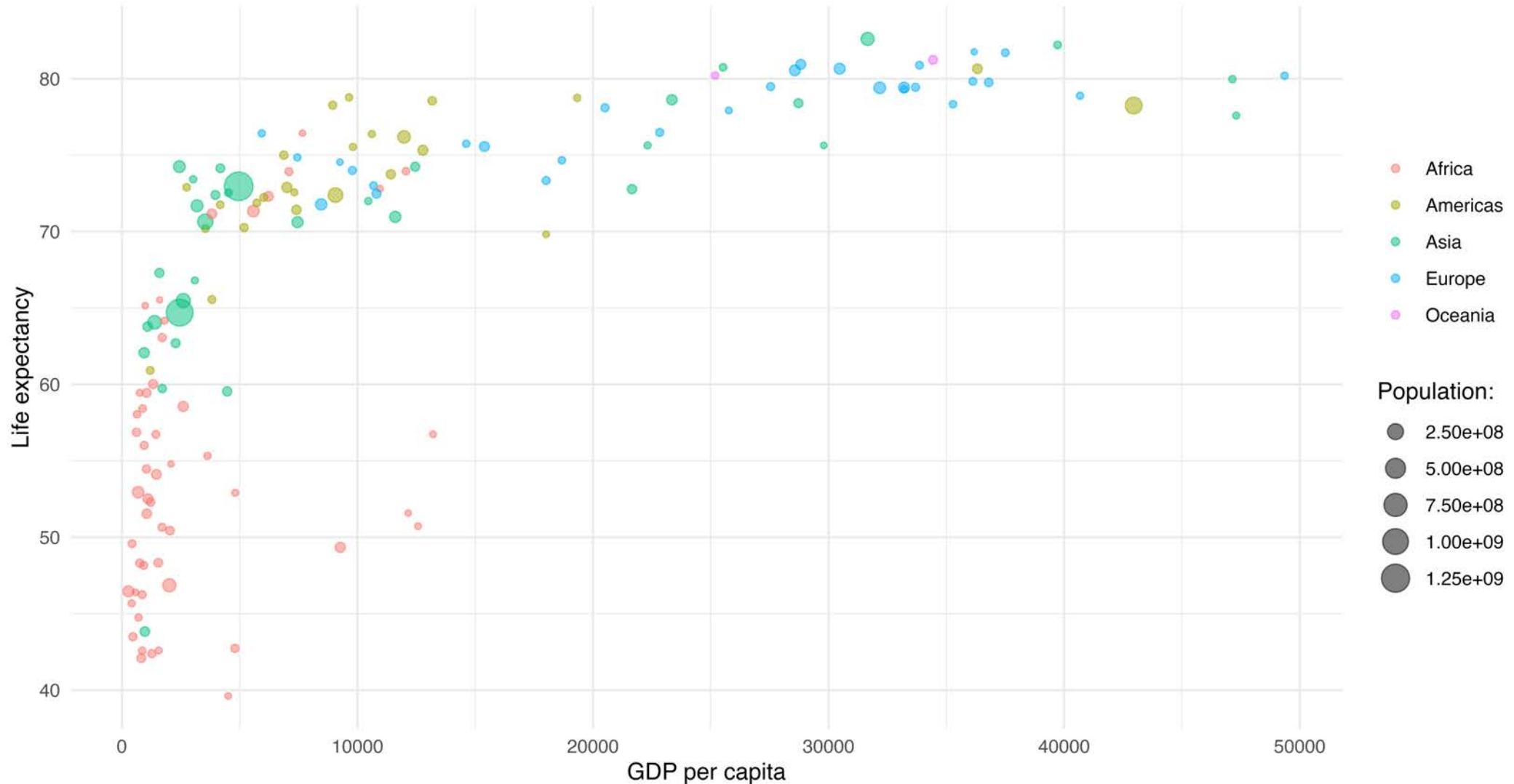
Health & Income of Nations in 2007



Add a Complete Theme

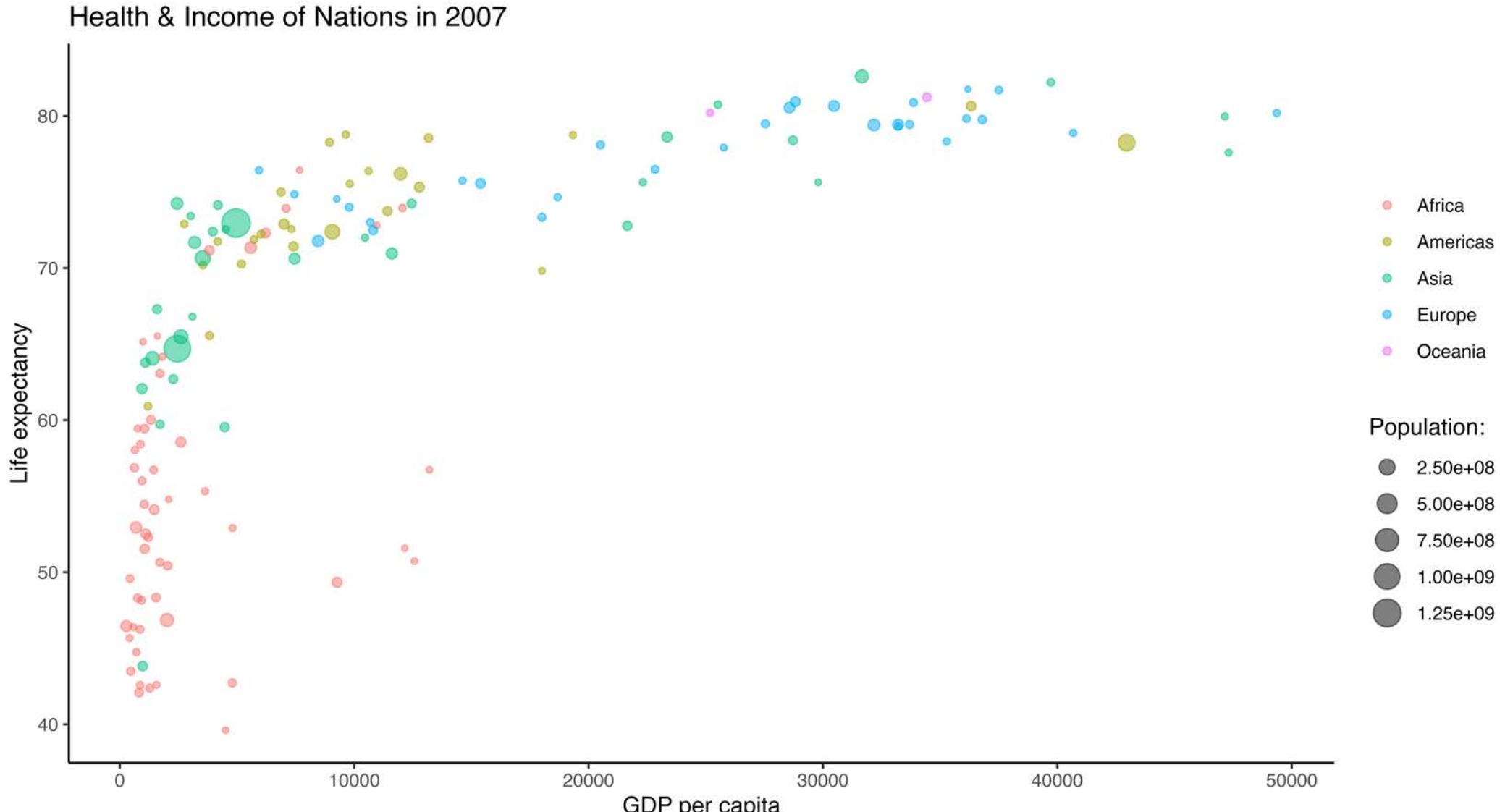
```
1 g + theme_minimal()
```

Health & Income of Nations in 2007



Add a Complete Theme

```
1 g + theme_classic()
```



Add a Complete Theme

```
1 g + theme_void()
```

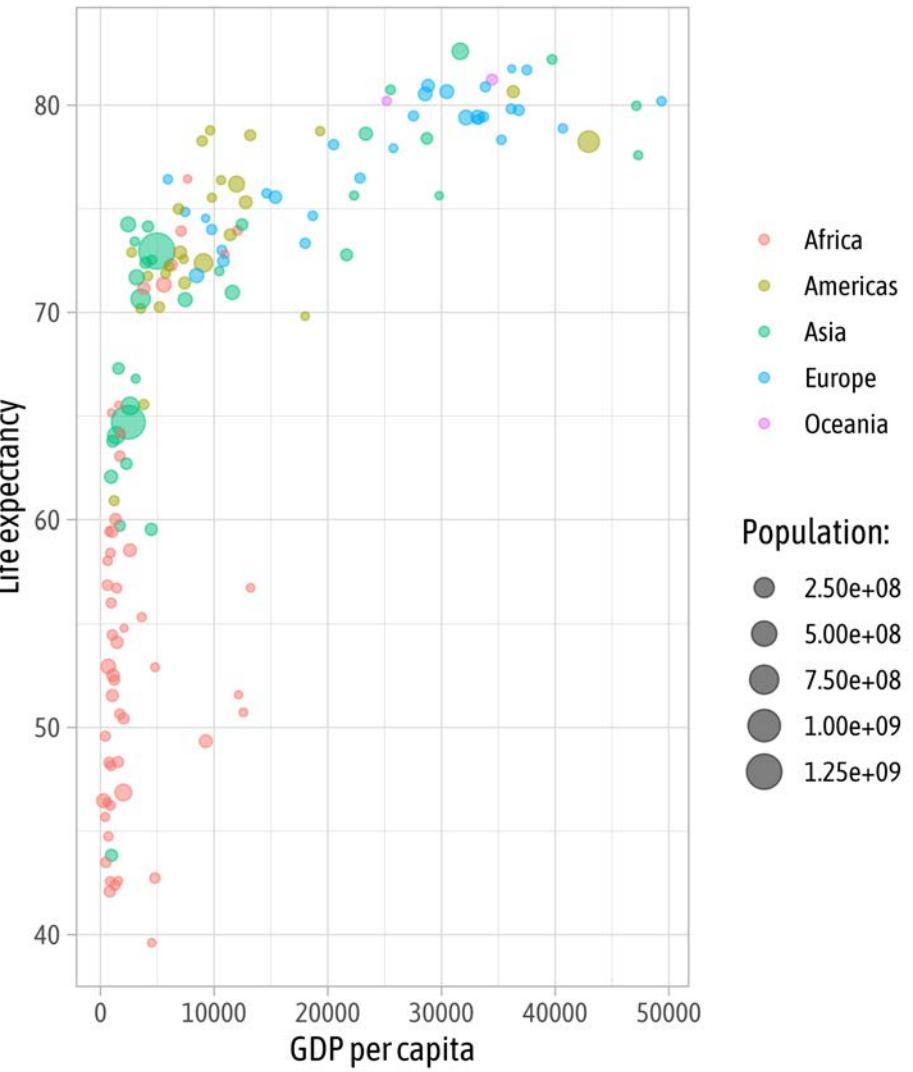
Health & Income of Nations in 2007



Adjust Theme Base Settings

```
1 g + theme_light(  
2   base_size = 13,  
3   base_family = "Asap Condensed"  
4 )
```

Health & Income of Nations in 2007



Set a Complete Theme Globally

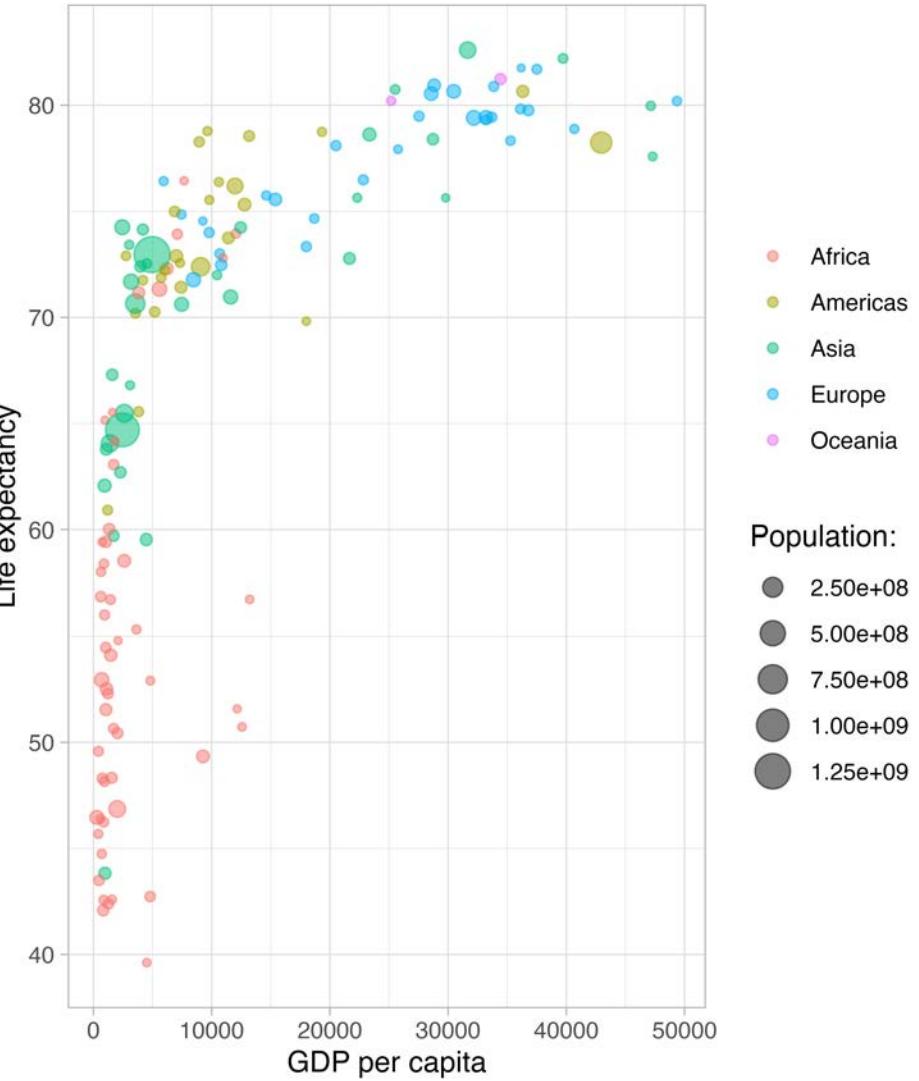
```
1 theme_set(theme_light())
```



Set a Complete Theme Globally

```
1 theme_set(theme_light())  
2  
3 g
```

Health & Income of Nations in 2007



Adjust Theme Base Settings

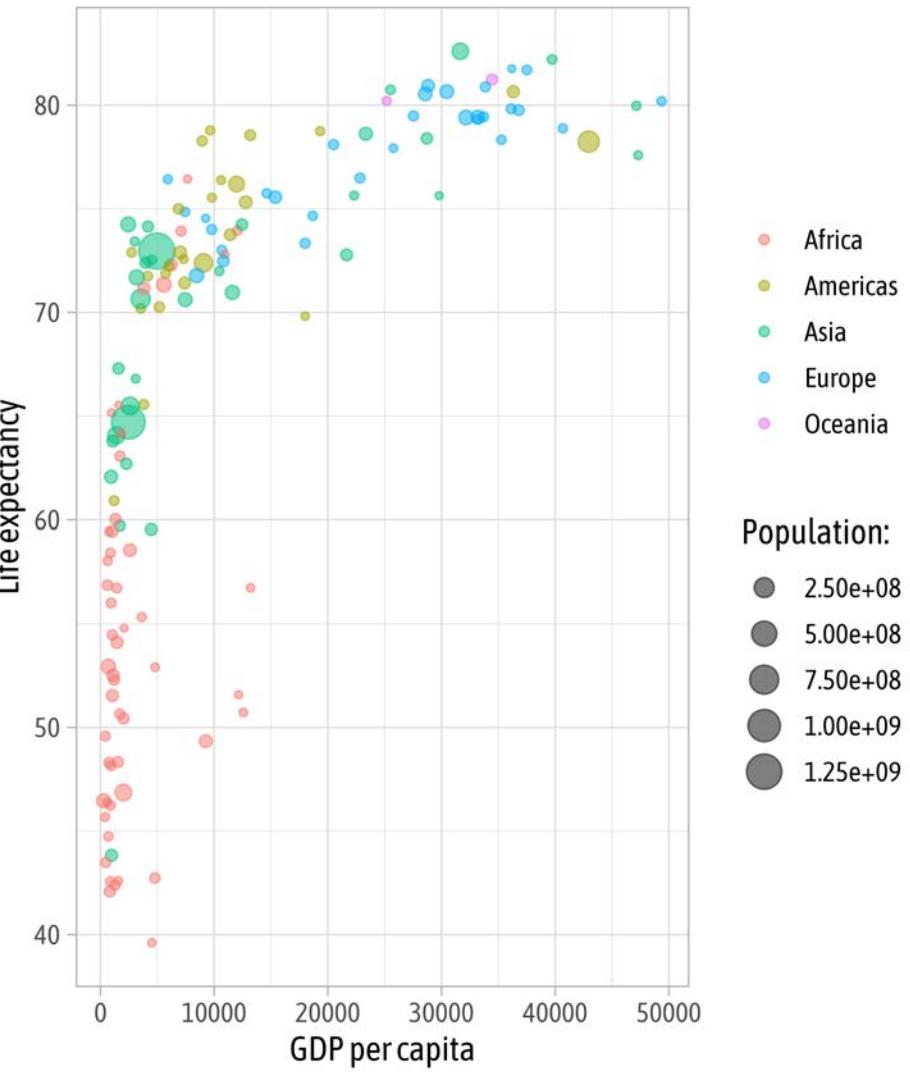
```
1 theme_set(theme_light(  
2   base_size = 13,  
3   base_family = "Asap Condensed"  
4 ))
```



Adjust Theme Base Settings

```
1 theme_set(theme_light(  
2   base_size = 13,  
3   base_family = "Asap Condensed"  
4 ))  
5  
6 g
```

Health & Income of Nations in 2007



{systemfonts}

```
1 library(systemfonts)
2
3 system_fonts() %>%
4   filter(stringr::str_detect(family, "Asap")) %>%
5   pull(family) %>%
6   unique() %>%
7   sort()
```

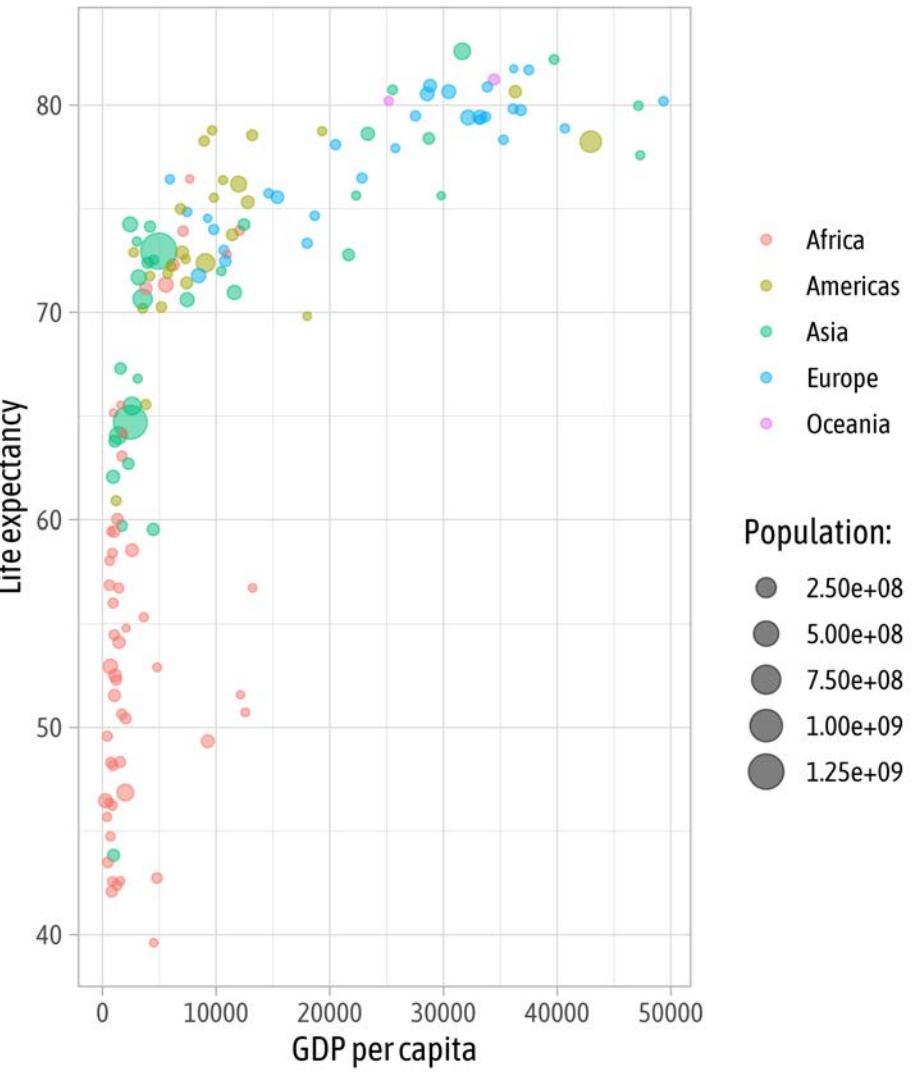
```
[1] "Asap"           "Asap Condensed"
```



Overwrite Theme Settings

```
1 g +  
2 theme()
```

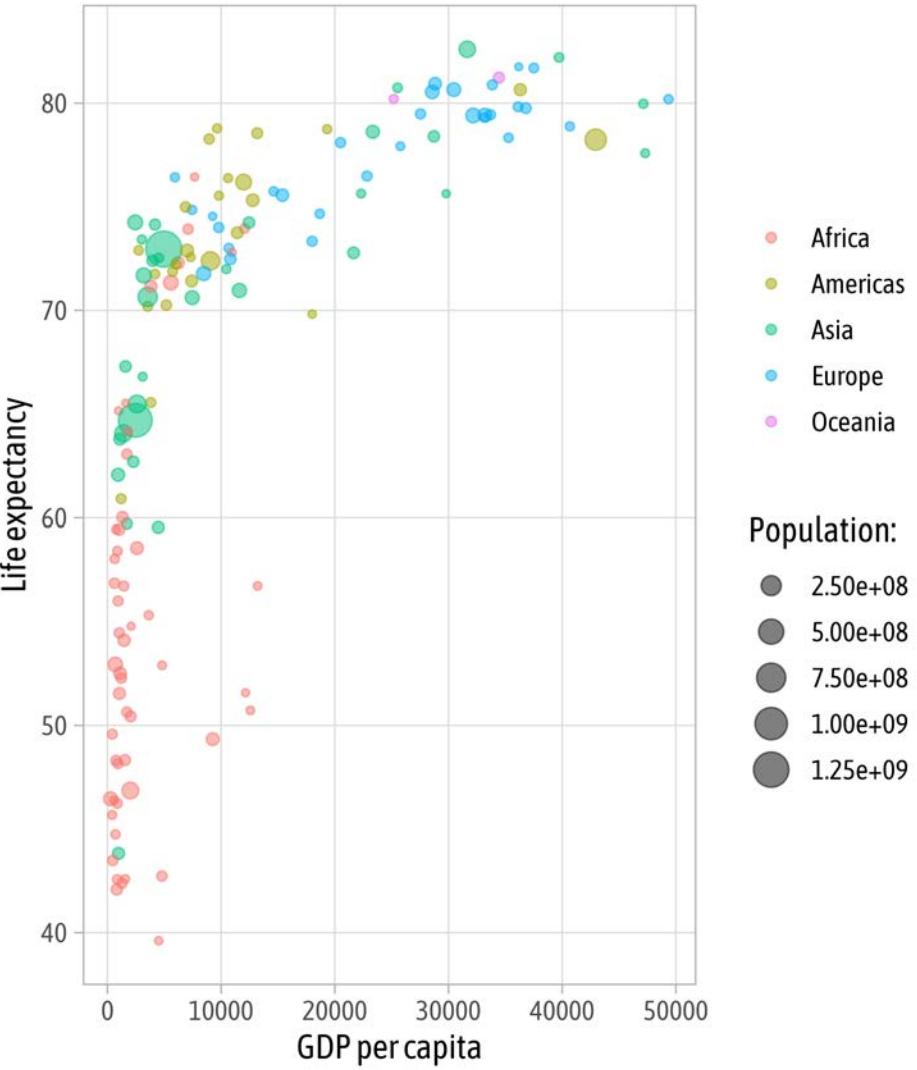
Health & Income of Nations in 2007



Overwrite Theme Settings

```
1 g +  
2 theme(  
3   panel.grid.minor = element_blank()  
4 )
```

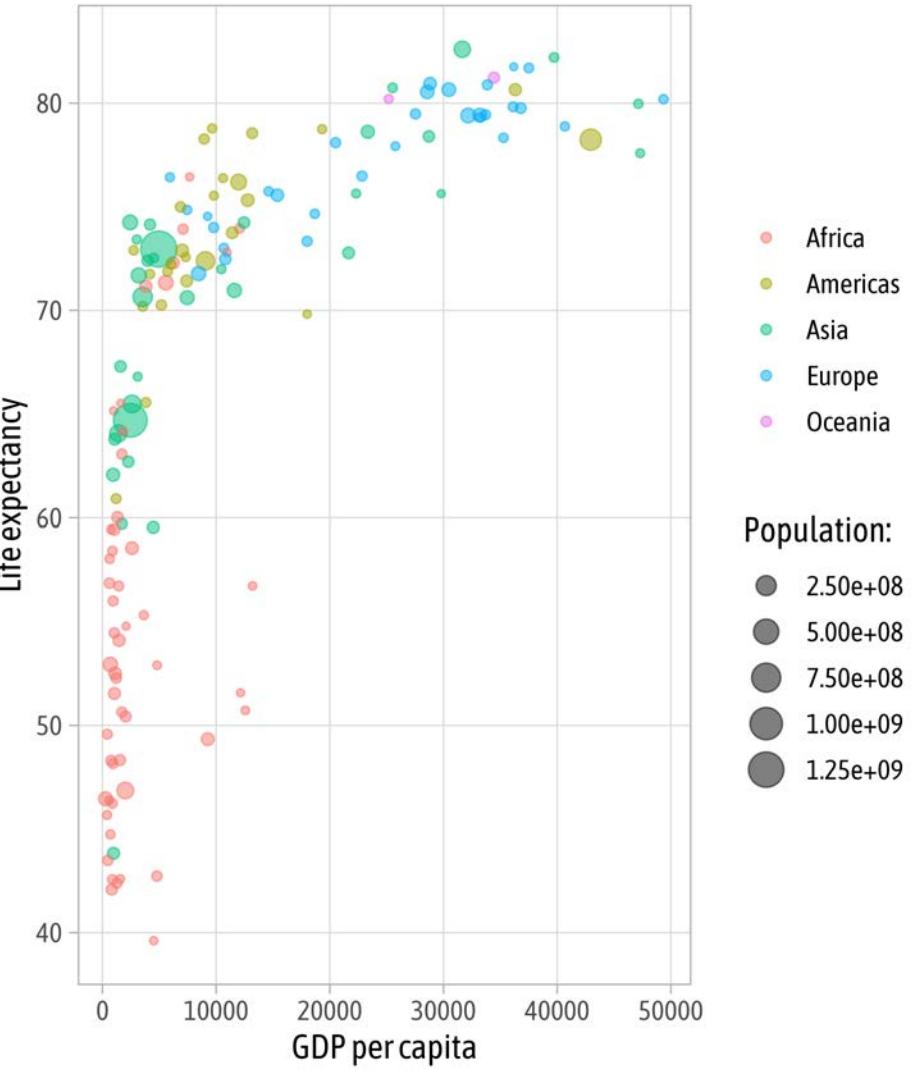
Health & Income of Nations in 2007



Overwrite Theme Settings

```
1 g +  
2 theme(  
3   panel.grid.minor = element_blank(),  
4   plot.title = element_text(face = "bold")  
5 )
```

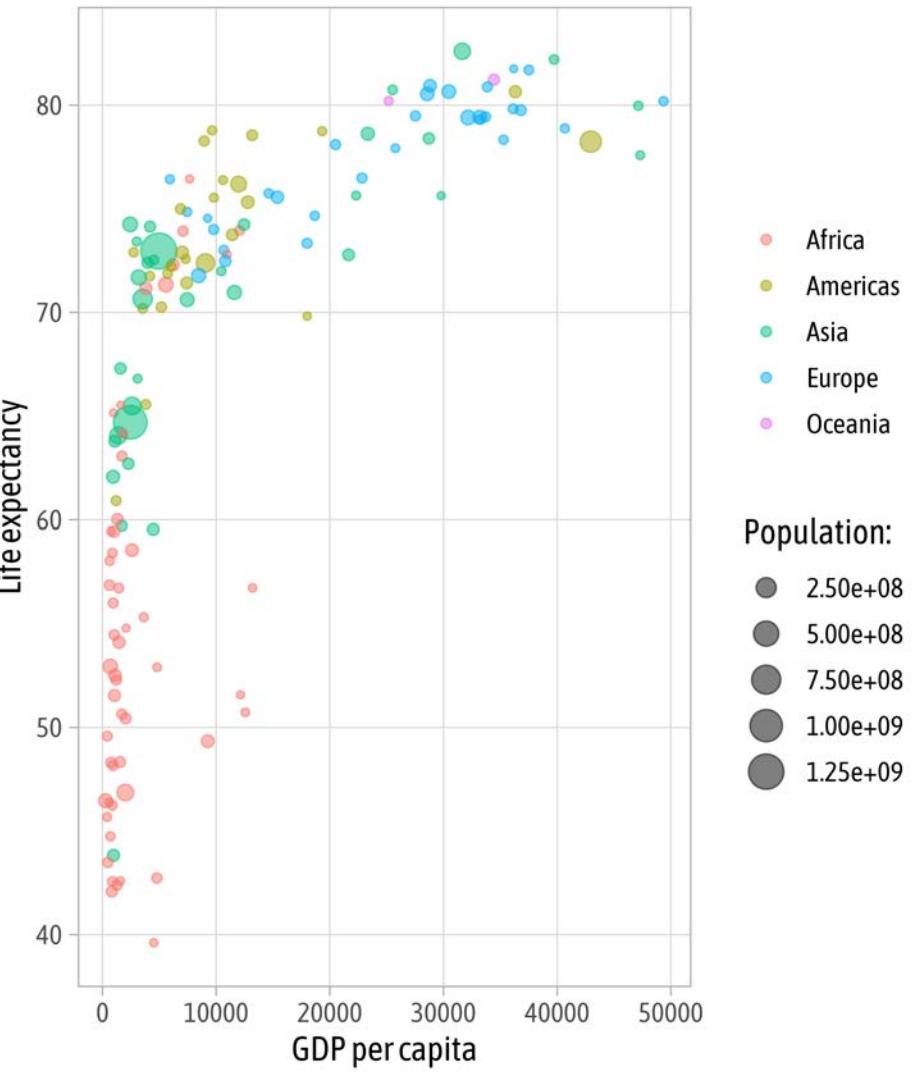
Health & Income of Nations in 2007



Overwrite Theme Settings

```
1 g +  
2 theme(  
3   panel.grid.minor = element_blank(),  
4   plot.title = element_text(face = "bold"),  
5   plot.title.position = "plot"  
6 )
```

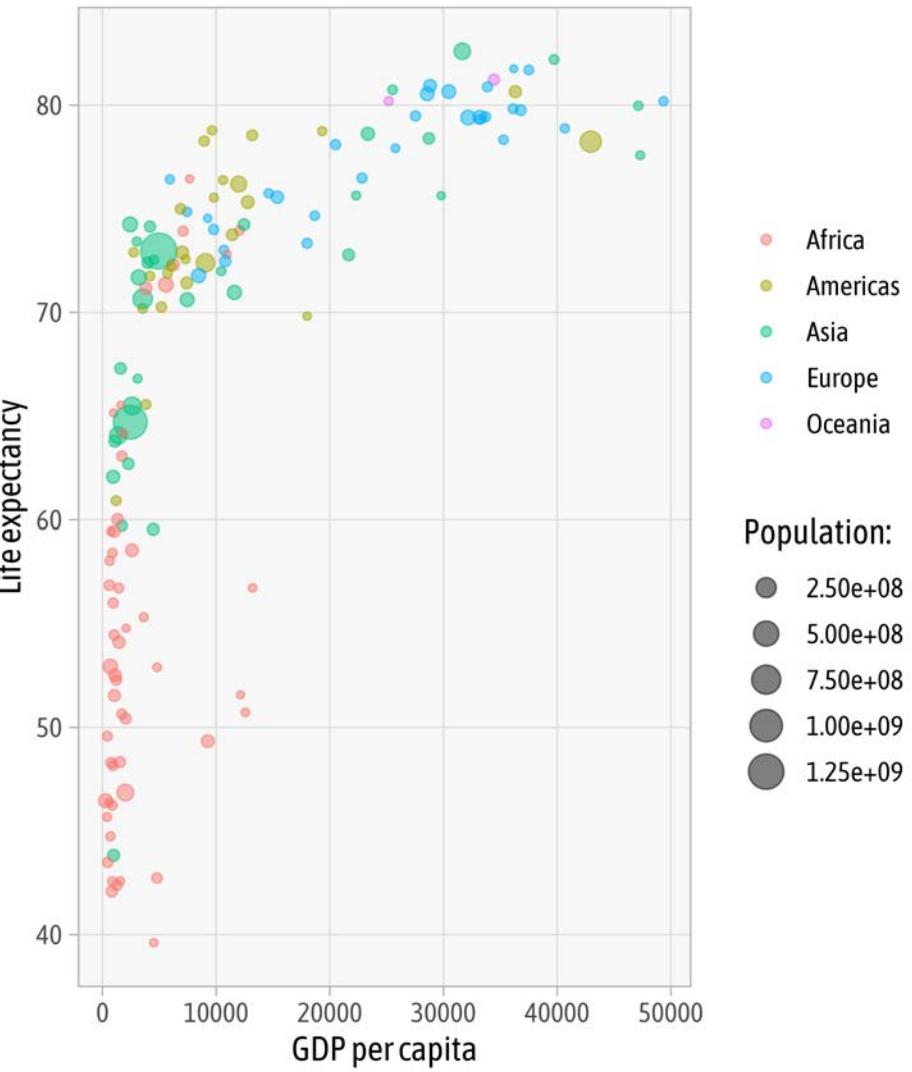
Health & Income of Nations in 2007



Overwrite Theme Settings

```
1 g +  
2   theme(  
3     panel.grid.minor = element_blank(),  
4     plot.title = element_text(face = "bold"),  
5     plot.title.position = "plot",  
6     panel.background = element_rect(  
7       fill = "#f8f8f8", color = NA  
8   )  
9 )
```

Health & Income of Nations in 2007



Overwrite Theme Settings Globally

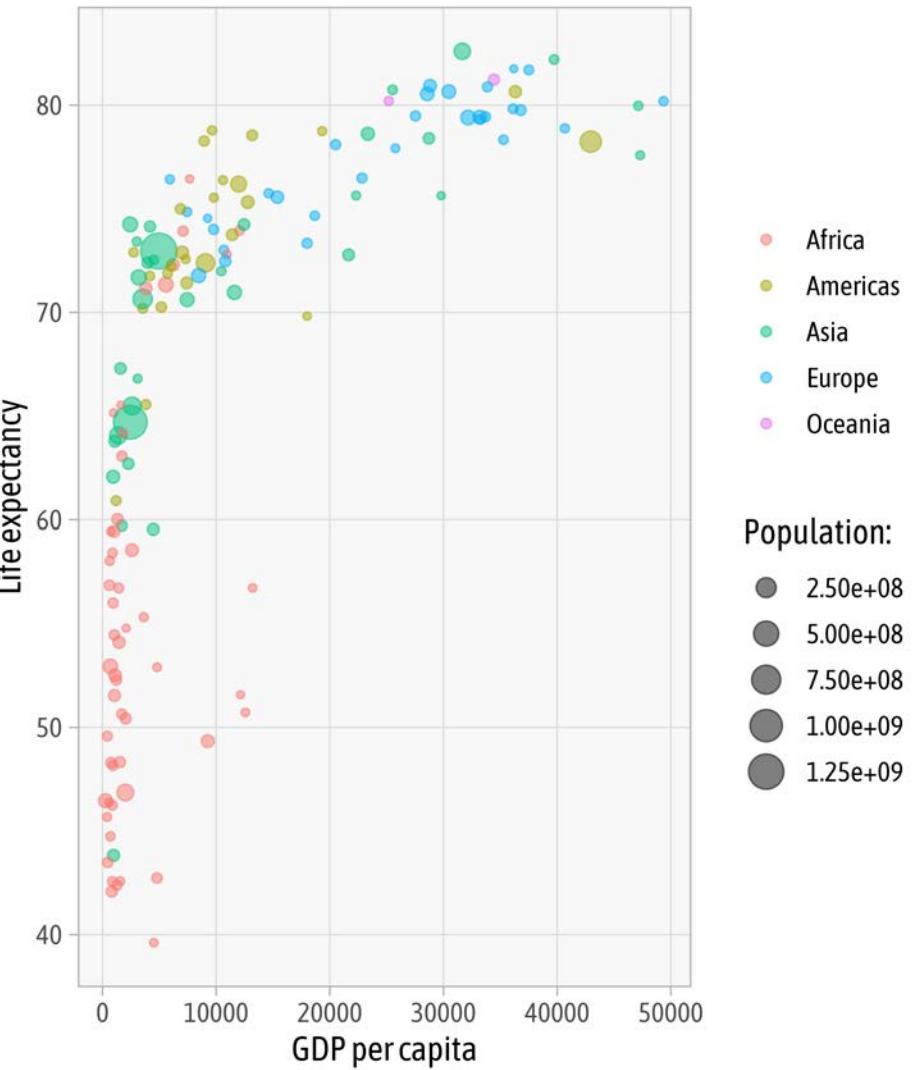
```
1 theme_update(  
2   panel.grid.minor = element_blank(),  
3   plot.title = element_text(face = "bold"),  
4   plot.title.position = "plot",  
5   panel.background = element_rect(  
6     fill = "#f8f8f8", color = NA  
7   )  
8 )
```



Overwrite Theme Settings Globally

```
1 theme_update(  
2   panel.grid.minor = element_blank(),  
3   plot.title = element_text(face = "bold"),  
4   plot.title.position = "plot",  
5   panel.background = element_rect(  
6     fill = "#f8f8f8", color = NA  
7   )  
8 )  
9  
10 g
```

Health & Income of Nations in 2007



— Facets —



Facets

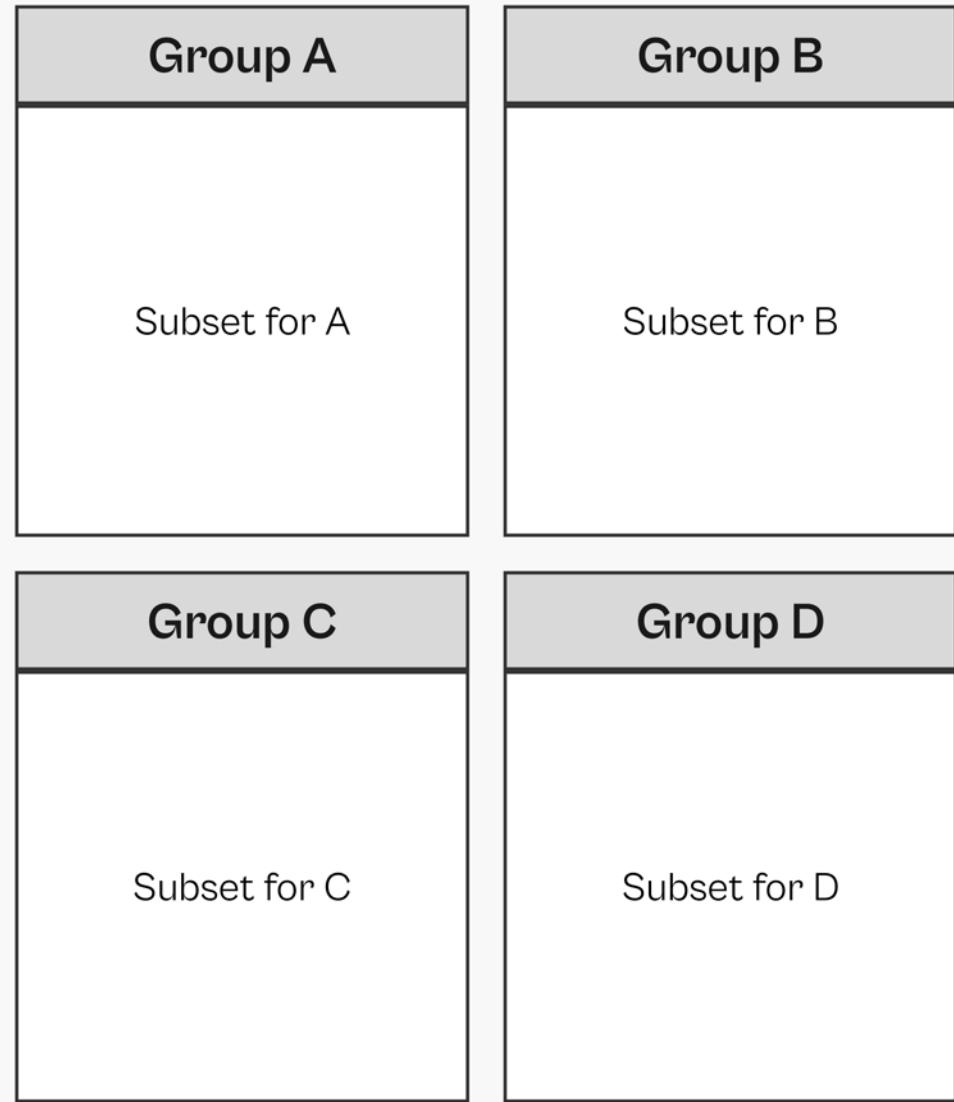
= split variables to multiple panels

Facets are also known as:

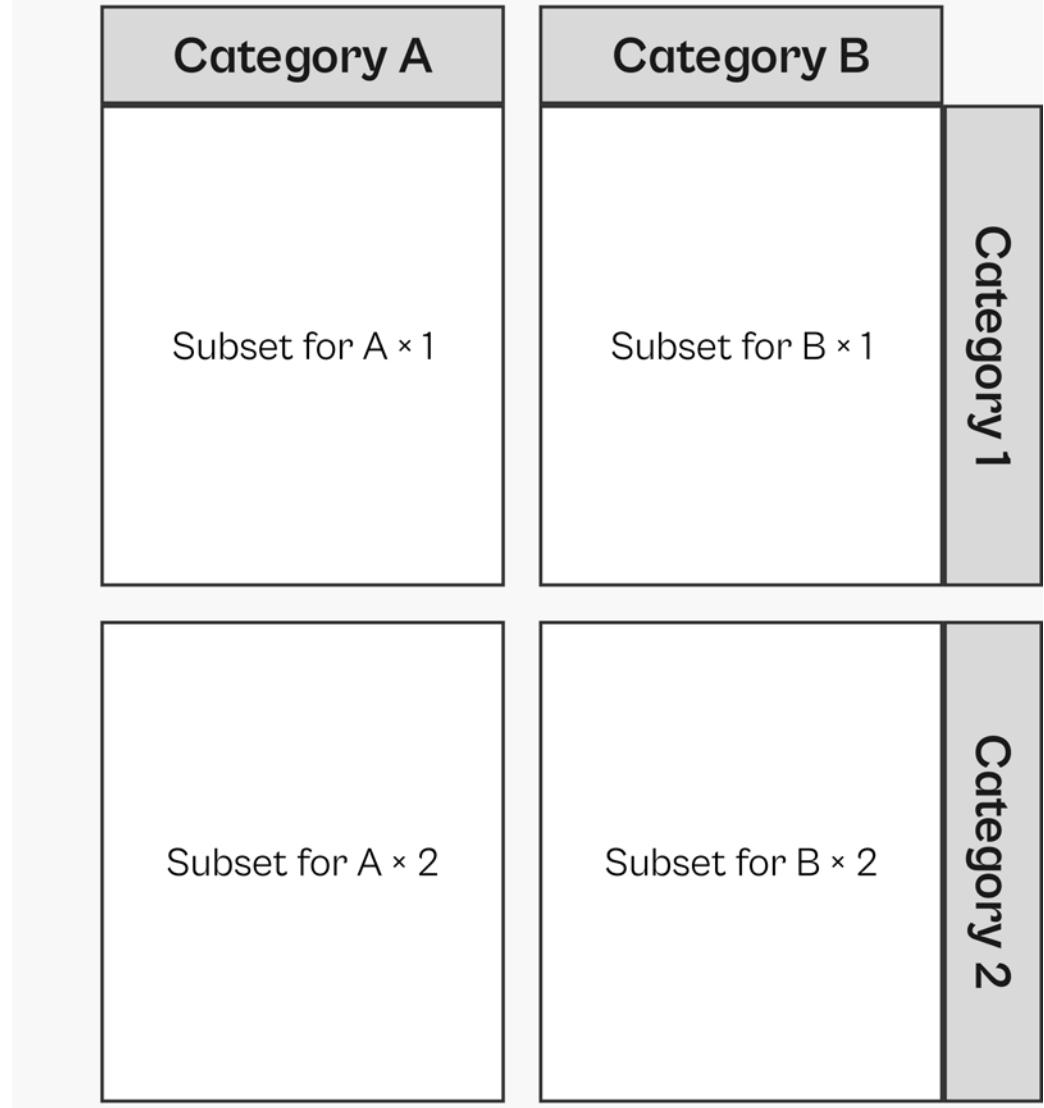
- small multiples
- trellis graphs
- lattice plots
- conditioning



facet_wrap()



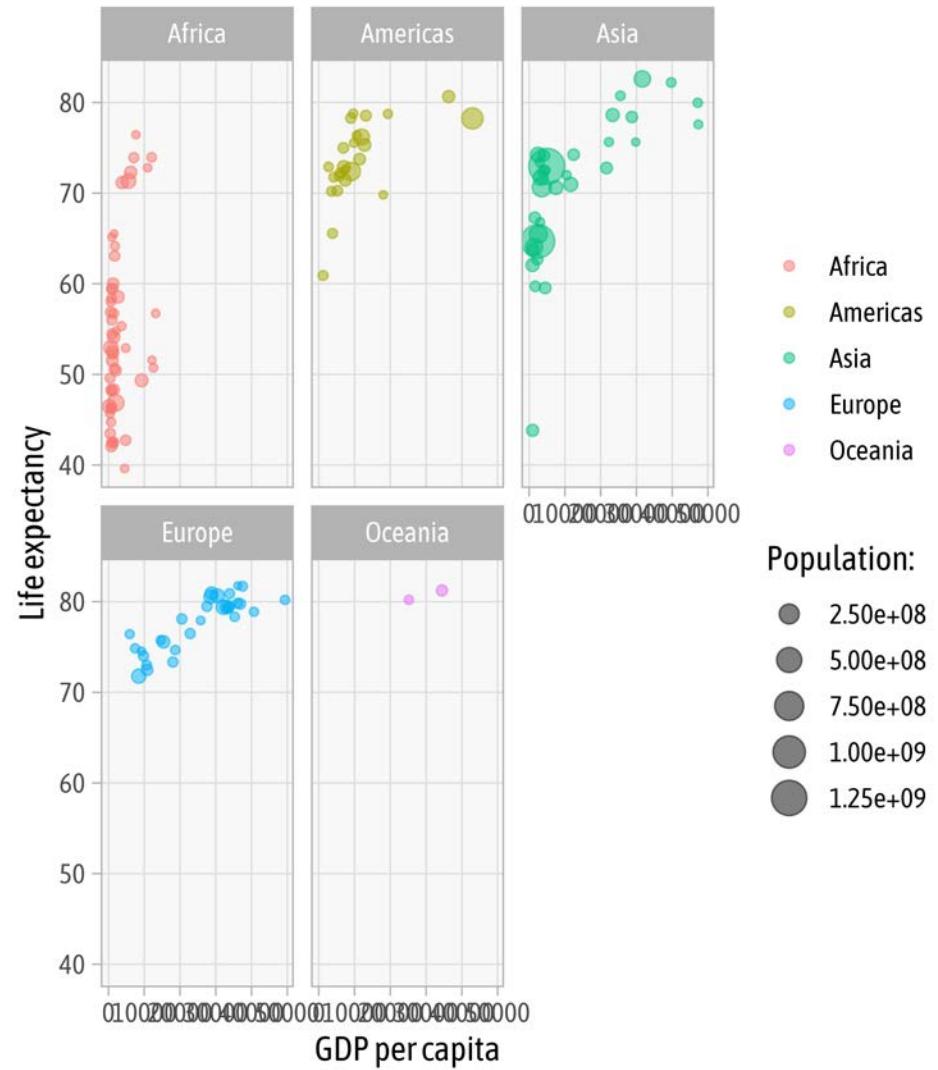
facet_grid()



Wrapped Facets

```
1 g +
2   facet_wrap(
3     vars(continent)
4   )
```

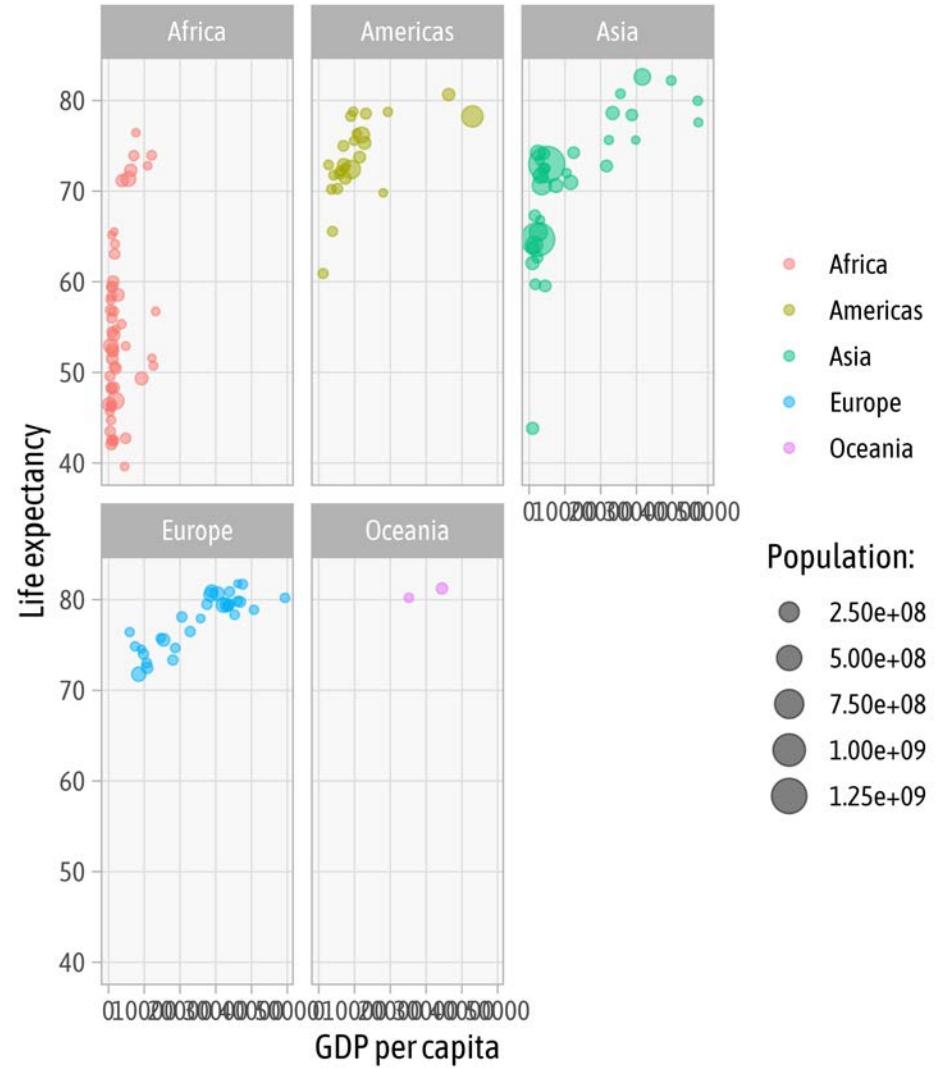
Health & Income of Nations in 2007



Wrapped Facets

```
1 g +  
2   facet_wrap(  
3     ~ continent  
4   )
```

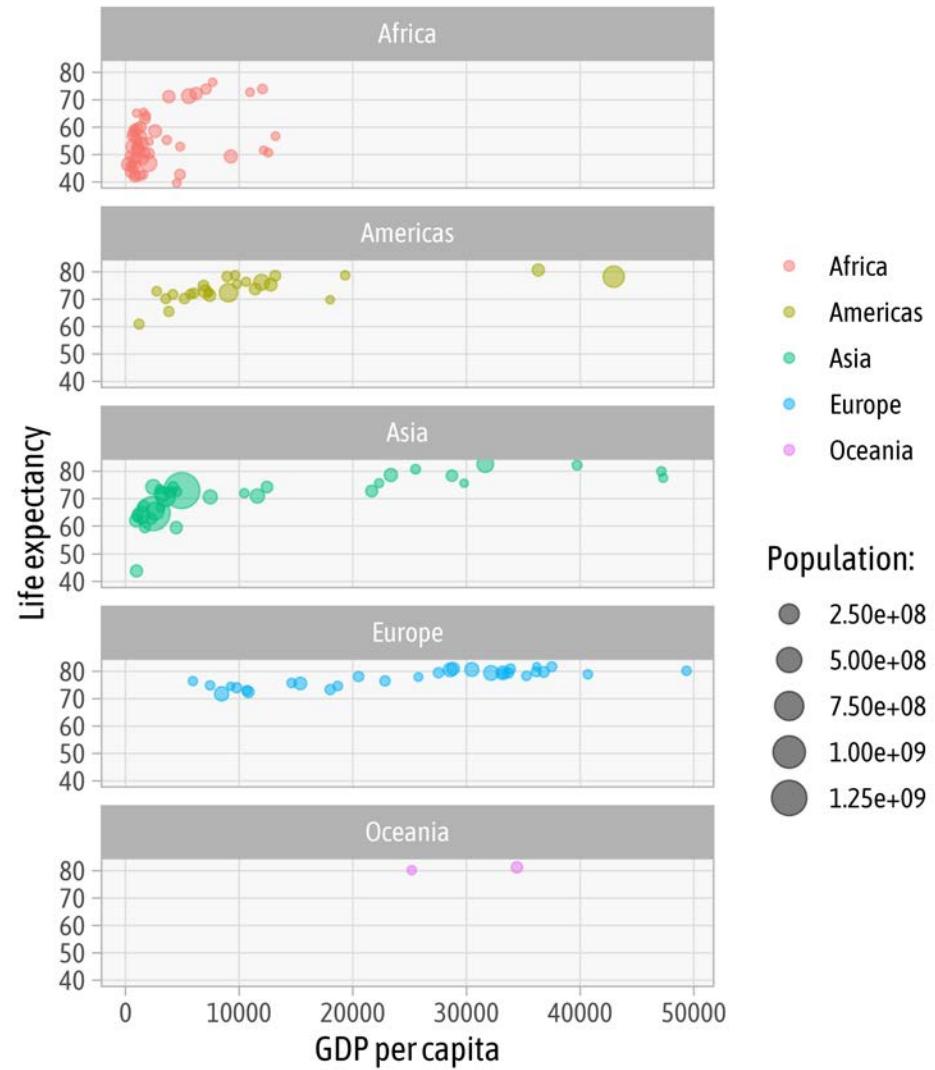
Health & Income of Nations in 2007



Facet Options: Cols + Rows

```
1 g +
2   facet_wrap(
3     ~ continent,
4     ncol = 1
5   )
```

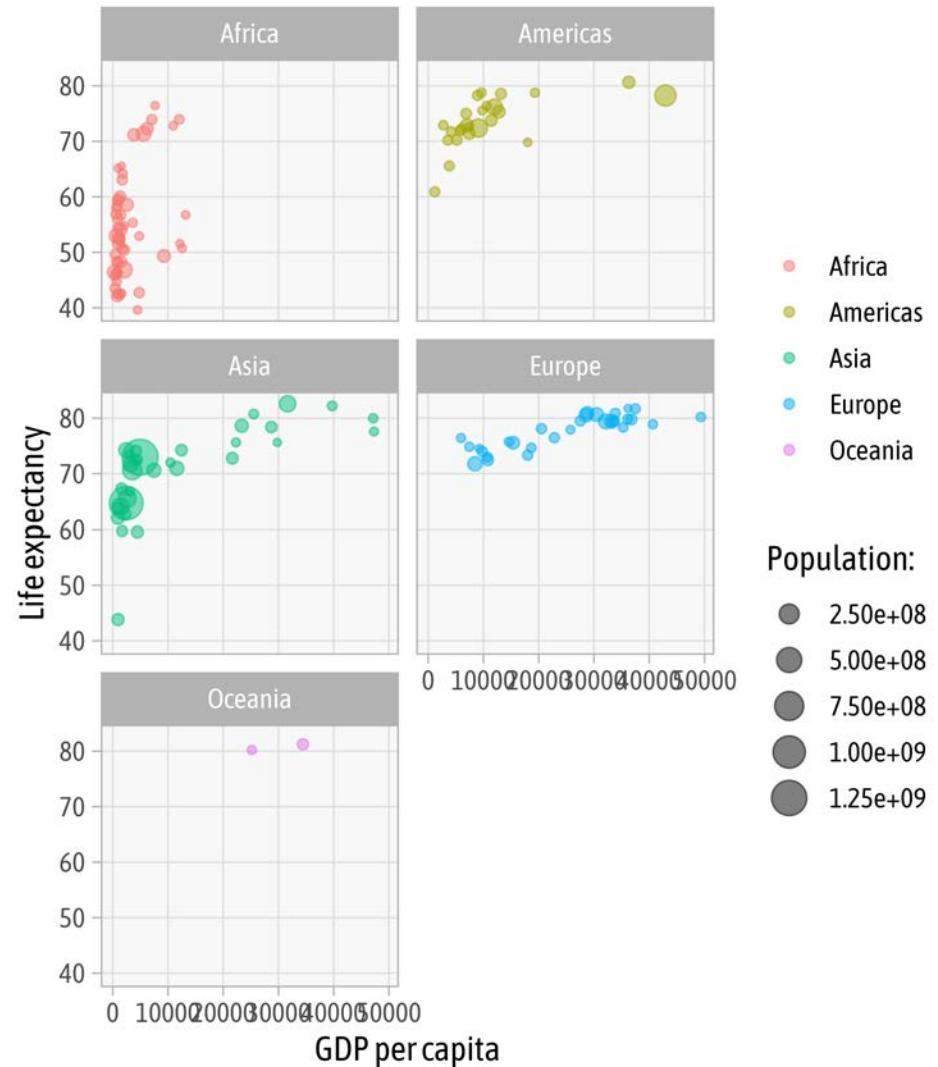
Health & Income of Nations in 2007



Facet Options: Cols + Rows

```
1 g +
2   facet_wrap(
3     ~ continent,
4     ncol = 2
5   )
```

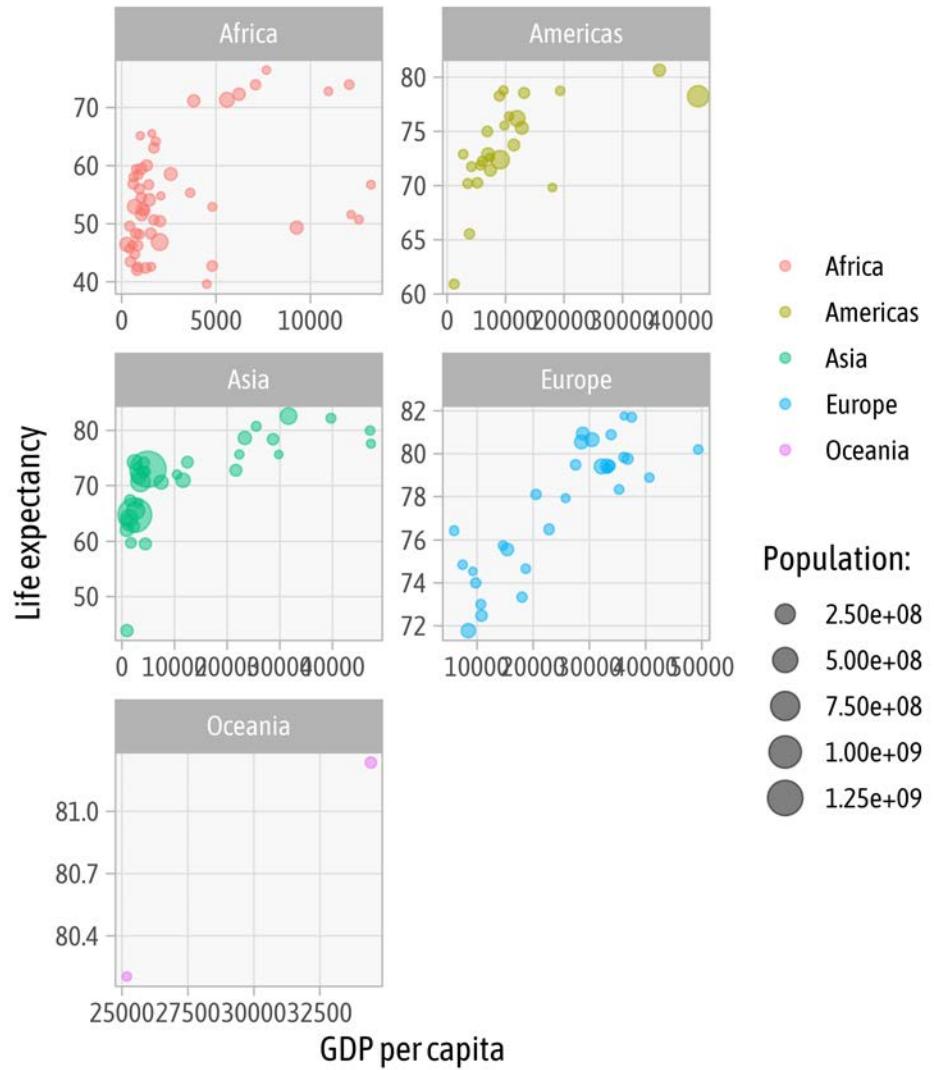
Health & Income of Nations in 2007



Facet Options: Free Scaling

```
1 g +
2   facet_wrap(
3     ~ continent,
4     ncol = 2,
5     scales = "free"
6   )
```

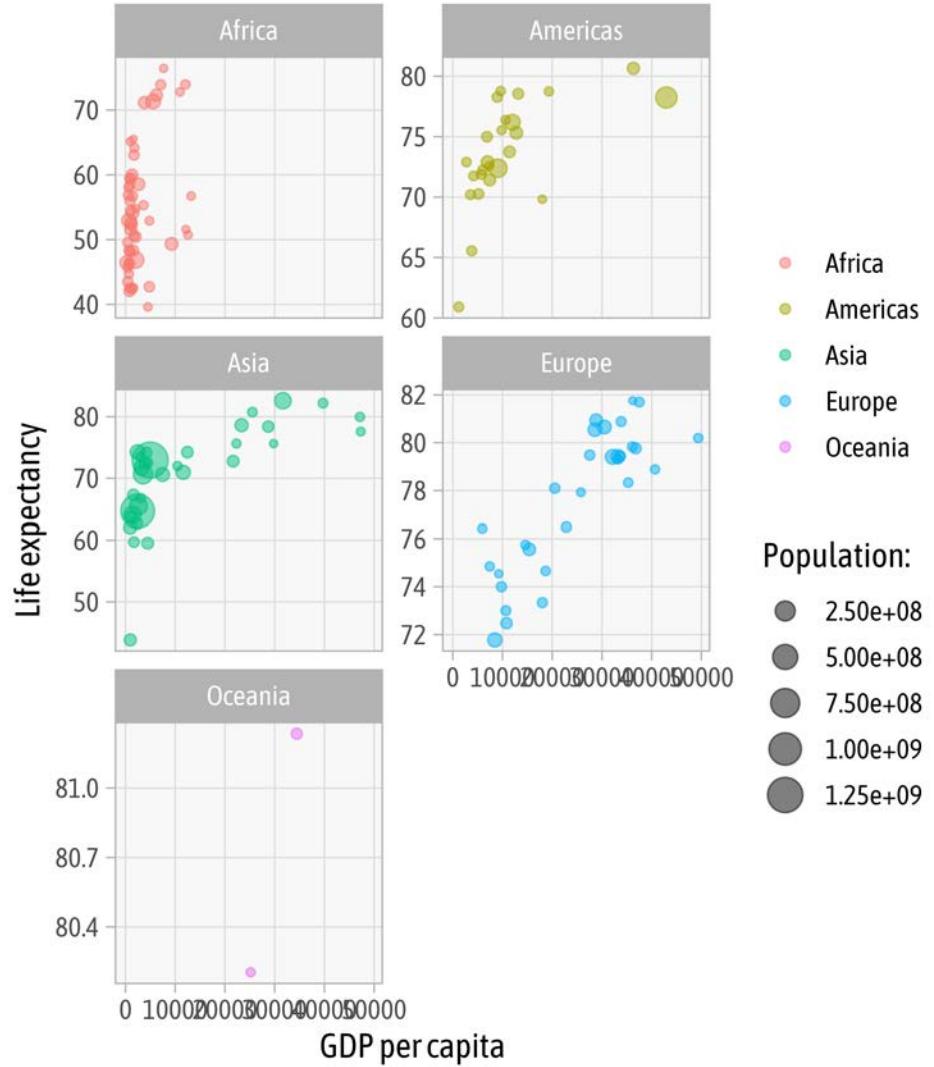
Health & Income of Nations in 2007



Facet Options: Free Scaling

```
1 g +
2   facet_wrap(
3     ~ continent,
4     ncol = 2,
5     scales = "free_y"
6   )
```

Health & Income of Nations in 2007



— Scales —



Scales

= translate between variable and property ranges

- feels-like temperature \rightleftharpoons x
- reported bike shares \rightleftharpoons y
- season \rightleftharpoons color
- year \rightleftharpoons shape
- ...



Scales

The `scale_*`() components control the properties of all the **aesthetic dimensions mapped to the data.**

Consequently, there are `scale_*`() functions for all aesthetics such as:

- **positions** via `scale_x_*`() and `scale_y_*`()
- **colors** via `scale_color_*`() and `scale_fill_*`()
- **sizes** via `scale_size_*`() and `scale_radius_*`()
- **shapes** via `scale_shape_*`() and `scale_linetype_*`()
- **transparency** via `scale_alpha_*`()



Scales

The `scale_*`() components control the properties of all the **aesthetic dimensions mapped to the data.**

The extensions (*) can be filled by e.g.:

- `continuous()`, `discrete()`, `reverse()`, `log10()`, `sqrt()`, `date()` for positions
- `continuous()`, `discrete()`, `manual()`, `gradient()`, `gradient2()`, `brewer()` for colors
- `continuous()`, `discrete()`, `manual()`, `ordinal()`, `area()`, `date()` for sizes
- `continuous()`, `discrete()`, `manual()`, `ordinal()` for shapes
- `continuous()`, `discrete()`, `manual()`, `ordinal()`, `date()` for transparency



Continuous vs Discrete in {ggplot2}

Continuous:

quantitative or numerical data

- height
- weight
- age
- counts

Discrete:

qualitative or categorical data

- species
- sex
- study sites
- age group



Continuous vs Discrete in {ggplot2}

Continuous:

quantitative or numerical data

- height (continuous)
- weight (continuous)
- age (continuous or discrete)
- counts (discrete)

Discrete:

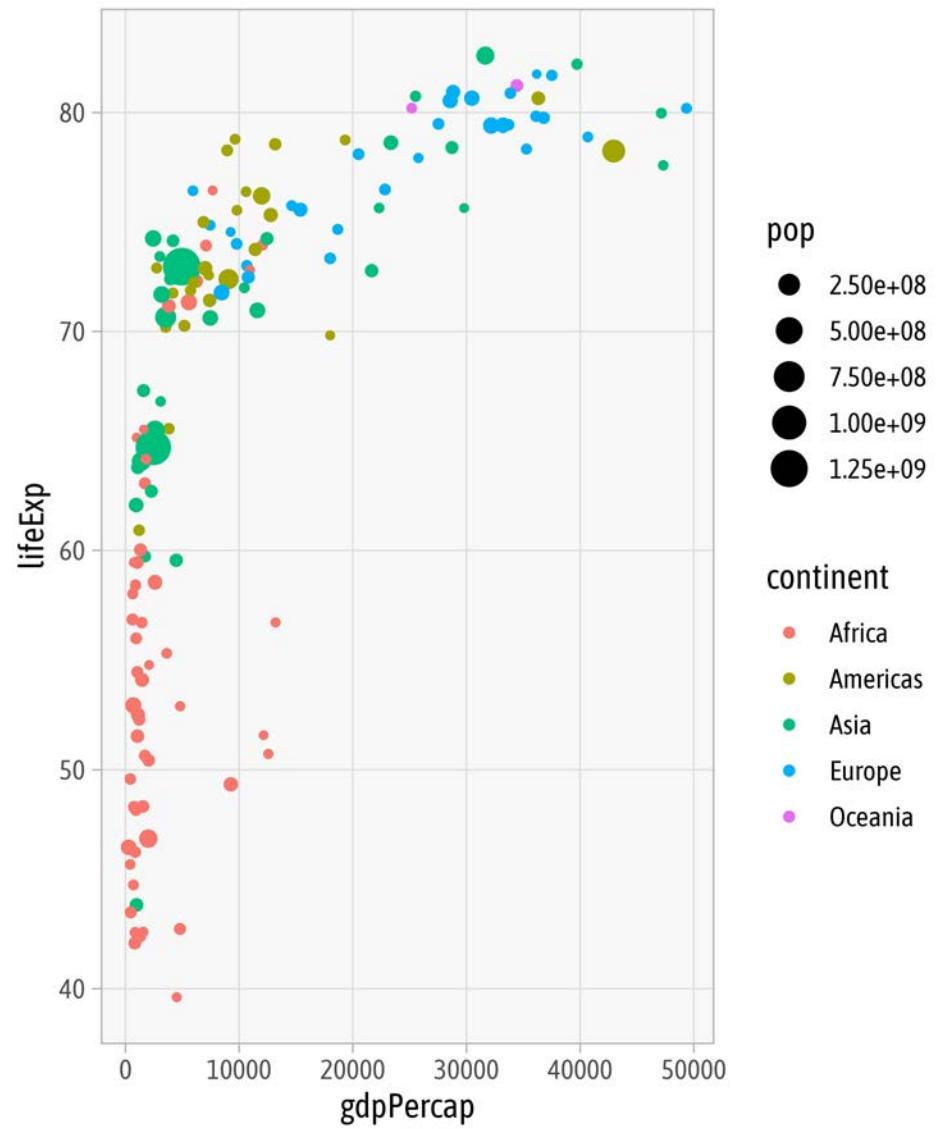
qualitative or categorical data

- species (nominal)
- sex (nominal)
- study site (nominal or ordinal)
- age group (ordinal)



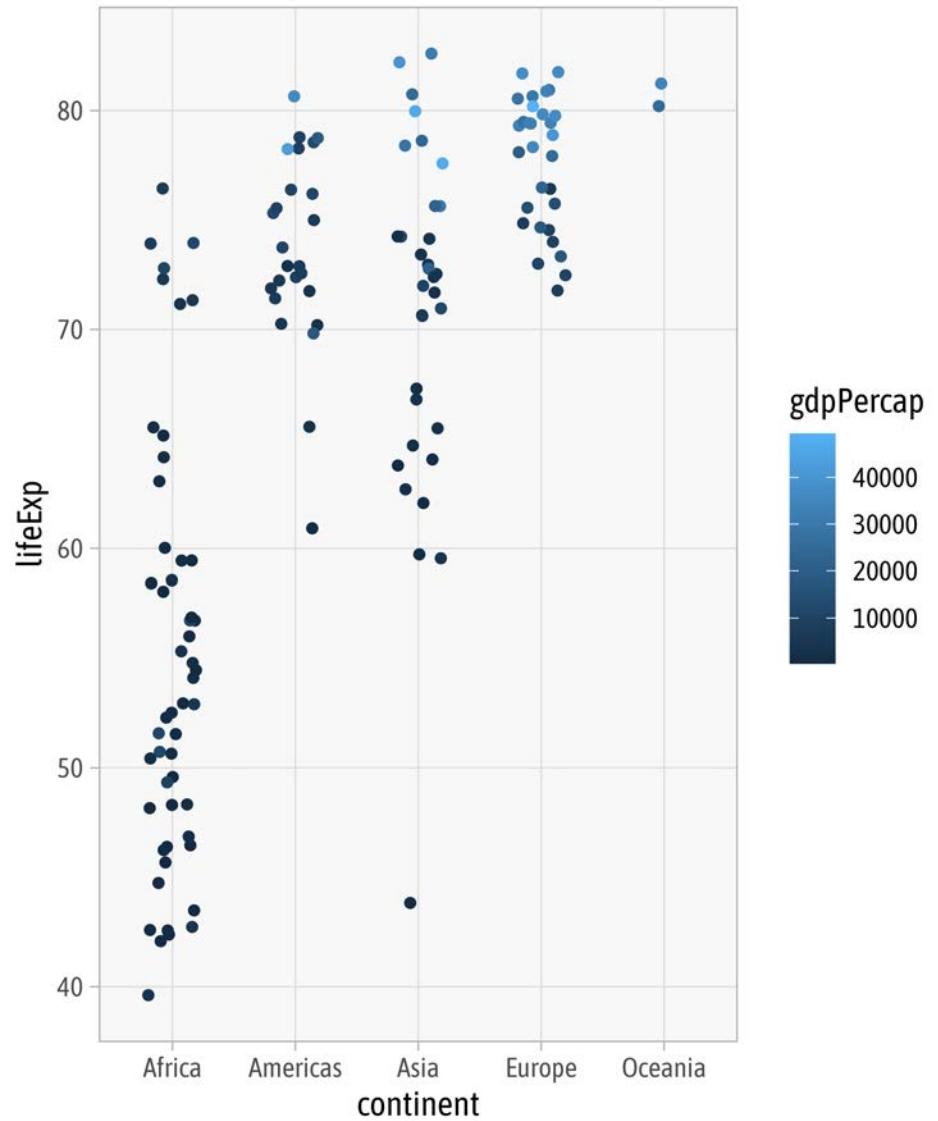
Aesthetics + Scales

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap,  
4       y = lifeExp,  
5       color = continent,  
6       size = pop)  
7 ) +  
8 geom_point() +  
9 scale_x_continuous() +  
10 scale_y_continuous() +  
11 scale_color_discrete() +  
12 scale_size()
```



Aesthetics + Scales

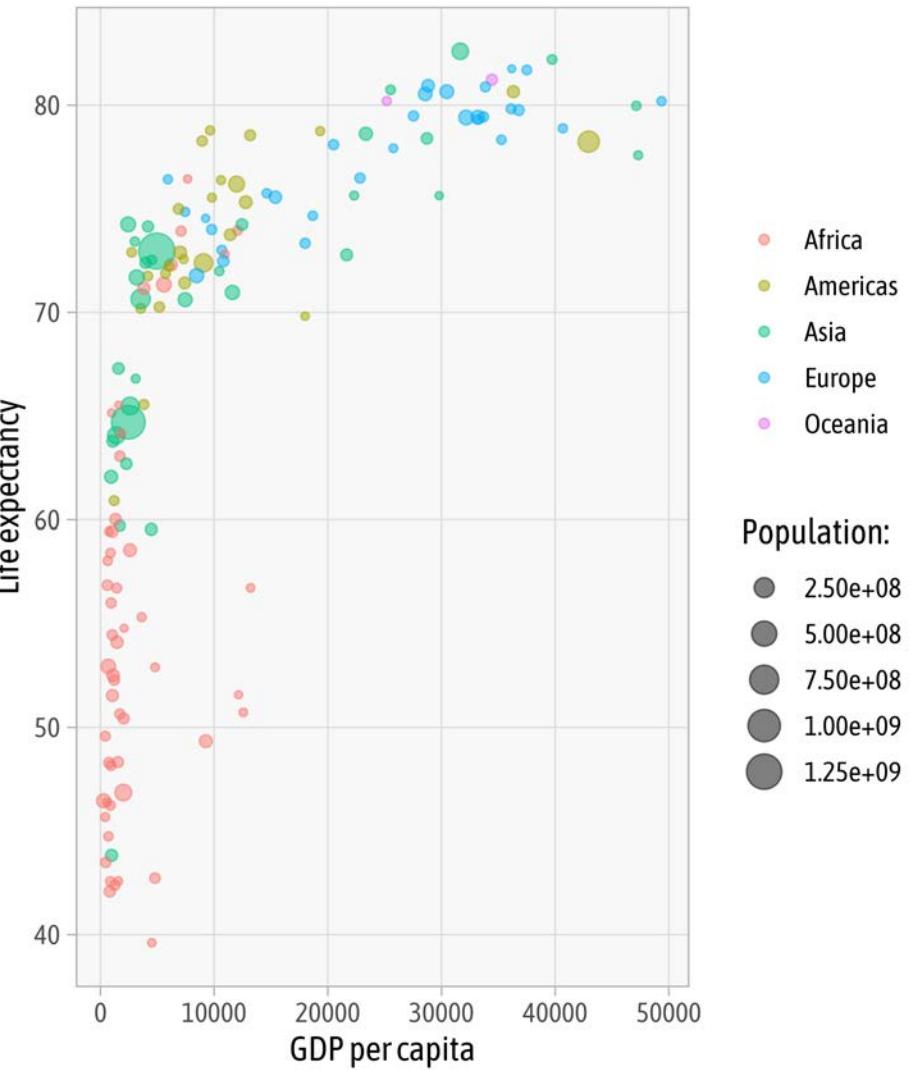
```
1 ggplot(  
2   data = gm2007,  
3   aes(x = continent,  
4        y = lifeExp,  
5        color = gdpPercap)  
6 ) +  
7   geom_jitter(width = .2) +  
8   scale_x_discrete() +  
9   scale_y_continuous() +  
10  scale_color_continuous()
```



Aesthetics + Scales

```
1 g +
2   scale_x_continuous() +
3   scale_y_continuous() +
4   scale_color_discrete() +
5   scale_size()
```

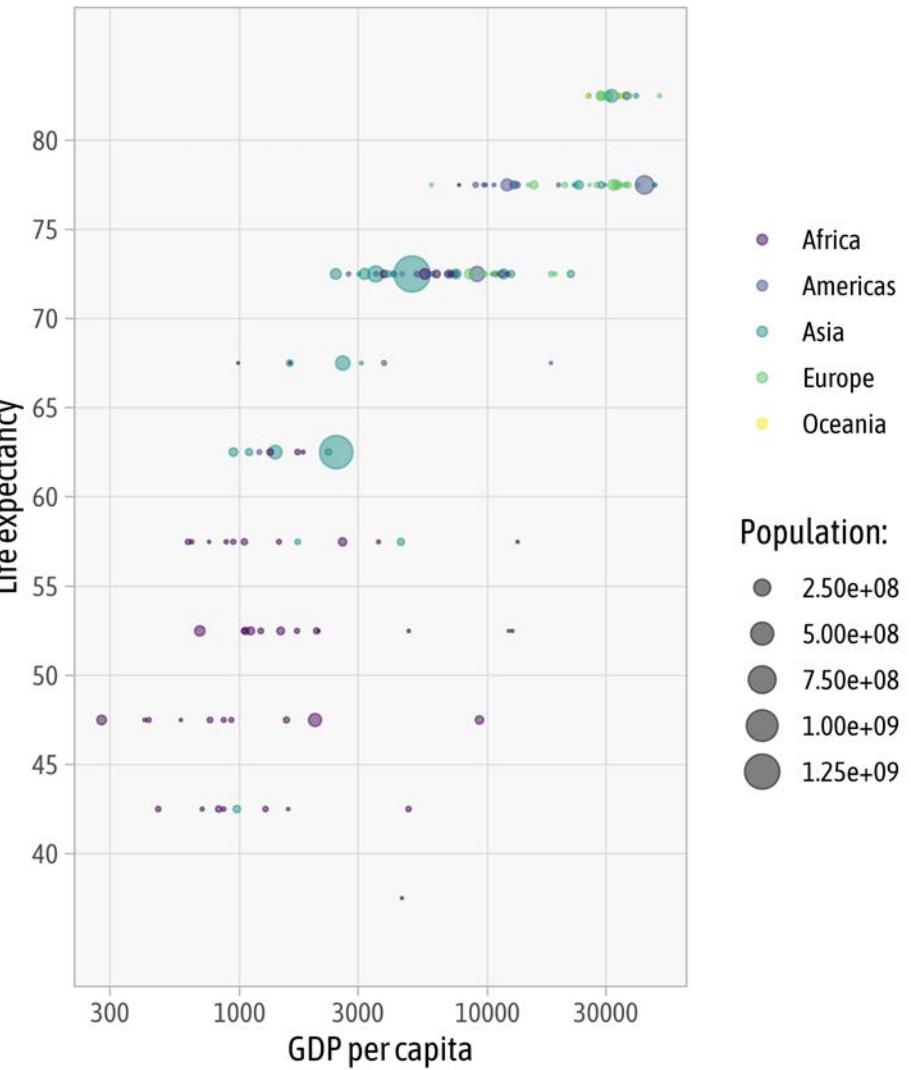
Health & Income of Nations in 2007



Overwrite Scales

```
1 g +  
2   scale_x_log10() +  
3   scale_y_binned() +  
4   scale_color_viridis_d() +  
5   scale_size_area()
```

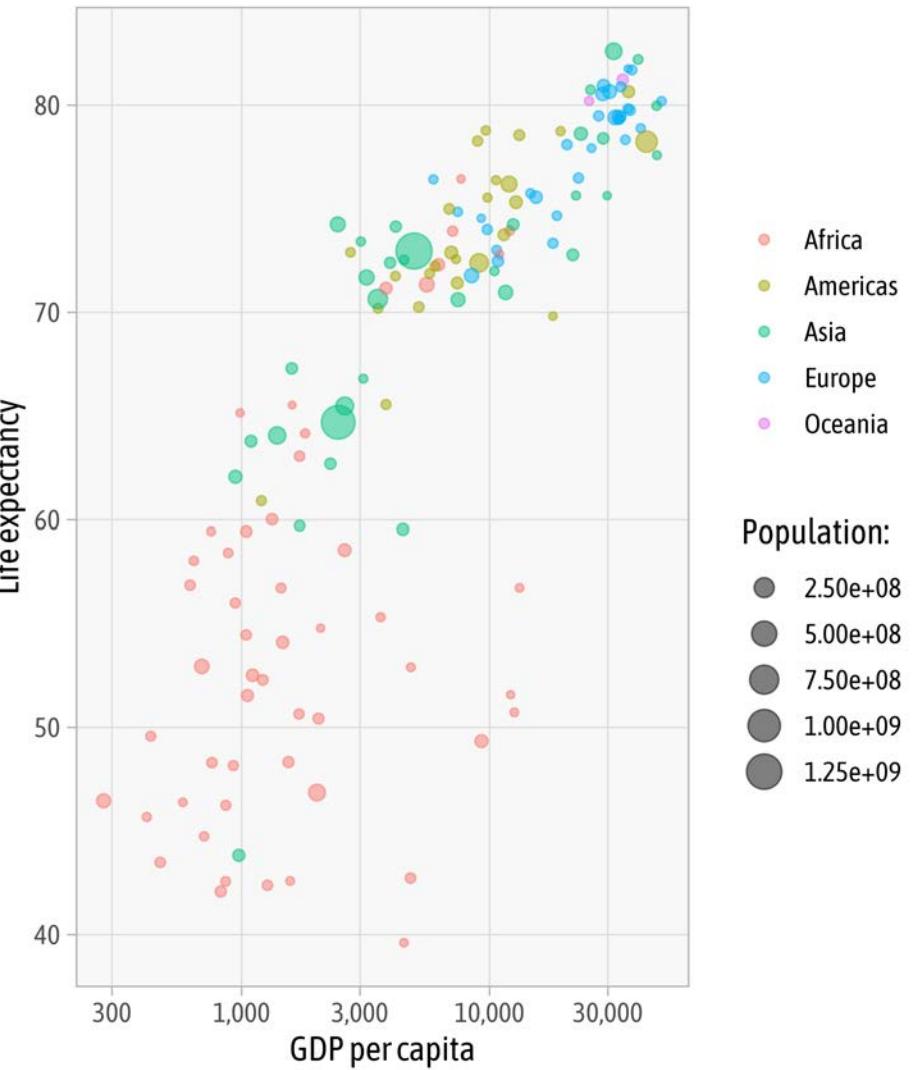
Health & Income of Nations in 2007



Modify Position Scales

```
1 g +
2   scale_x_log10(
3     labels = scales::label_comma()
4   ) +
5   scale_y_continuous() +
6   scale_color_discrete() +
7   scale_size()
```

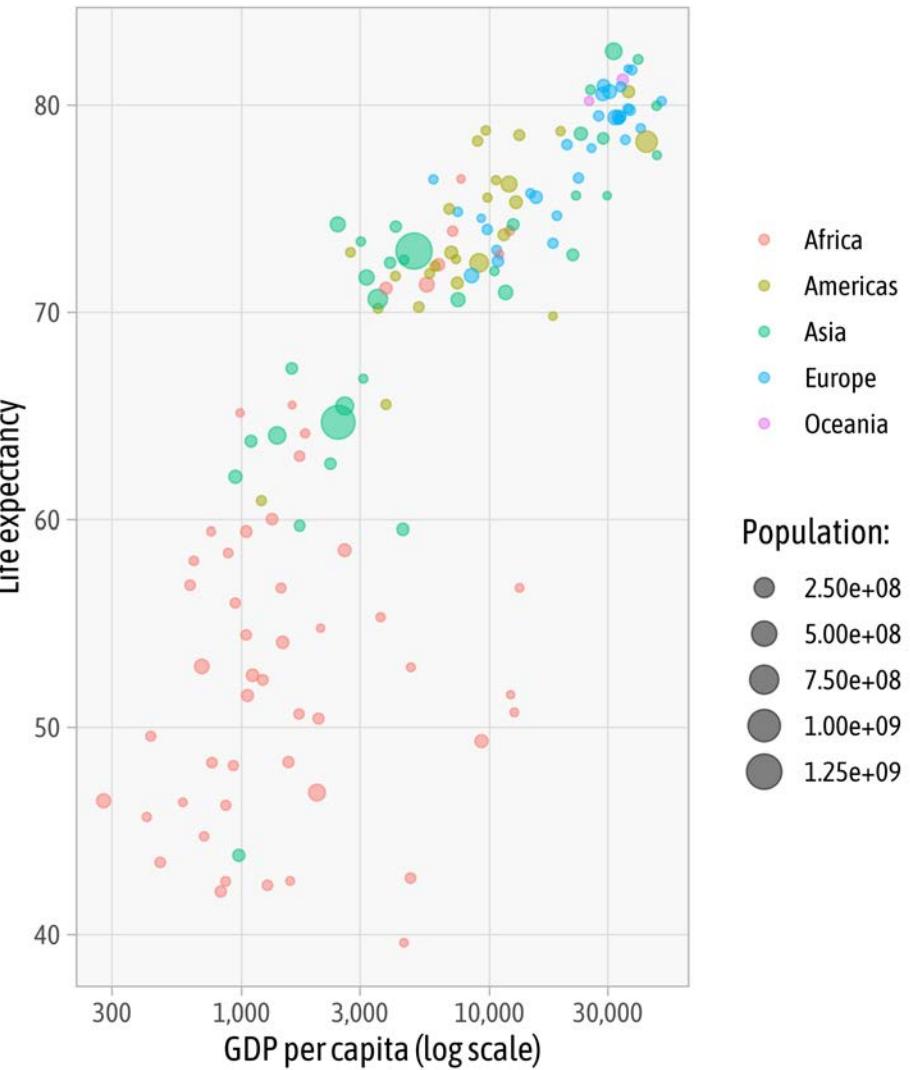
Health & Income of Nations in 2007



Modify Position Scales

```
1 g +
2   scale_x_log10(
3     labels = scales::label_comma(),
4     name = "GDP per capita (log scale)"
5   ) +
6   scale_y_continuous() +
7   scale_color_discrete() +
8   scale_size()
```

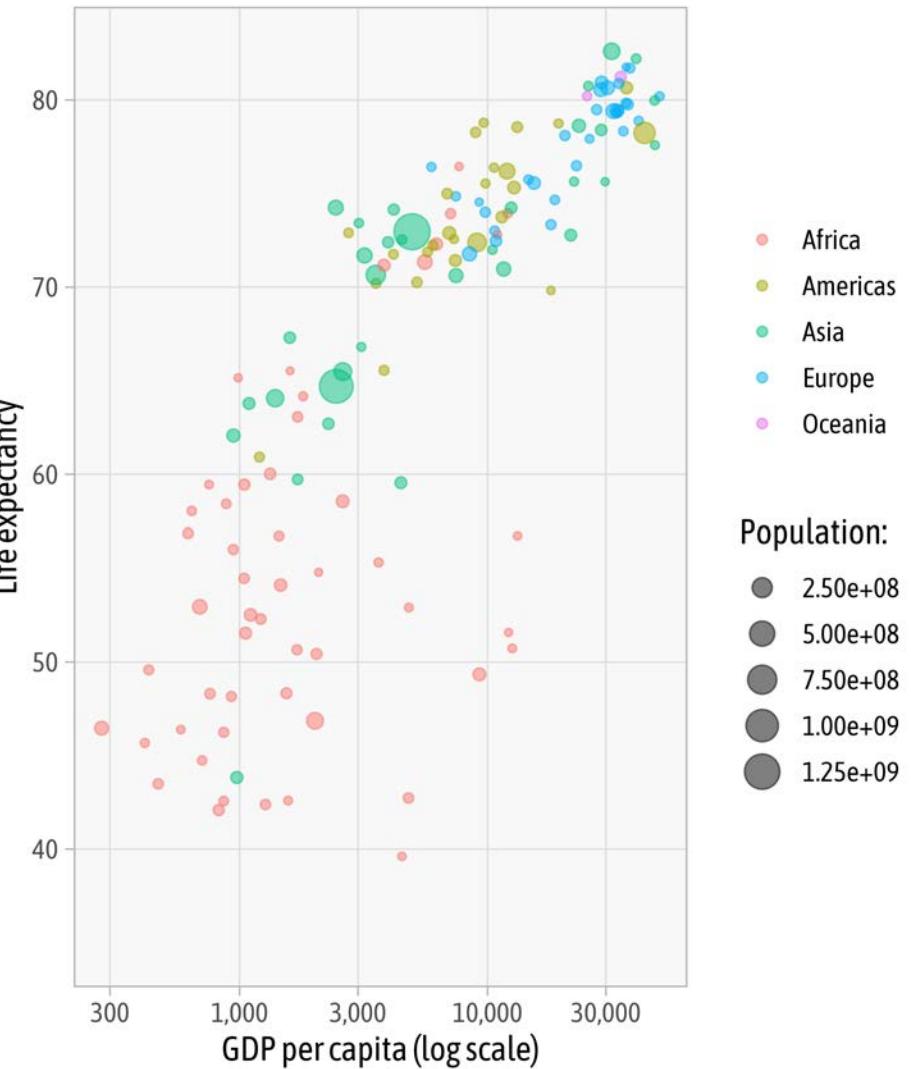
Health & Income of Nations in 2007



Modify Scales

```
1 g +
2   scale_x_log10(
3     labels = scales::label_comma(),
4     name = "GDP per capita (log scale)"
5   ) +
6   scale_y_continuous(
7     limits = c(35, NA),
8   ) +
9   scale_color_discrete() +
10  scale_size()
```

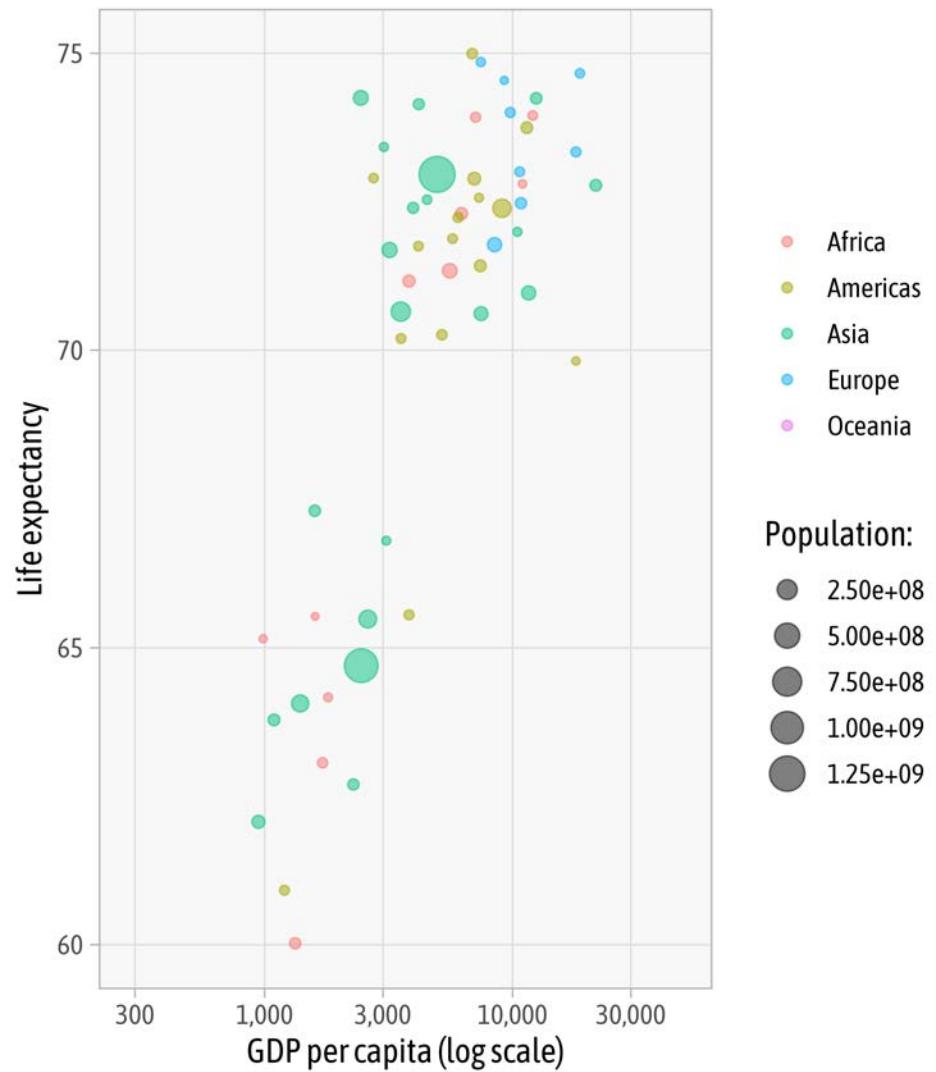
Health & Income of Nations in 2007



Modify Position Scales

```
1 g +
2   scale_x_log10(
3     labels = scales::label_comma(),
4     name = "GDP per capita (log scale)"
5   ) +
6   scale_y_continuous(
7     limits = c(60, 75)
8   ) +
9   scale_color_discrete() +
10  scale_size()
```

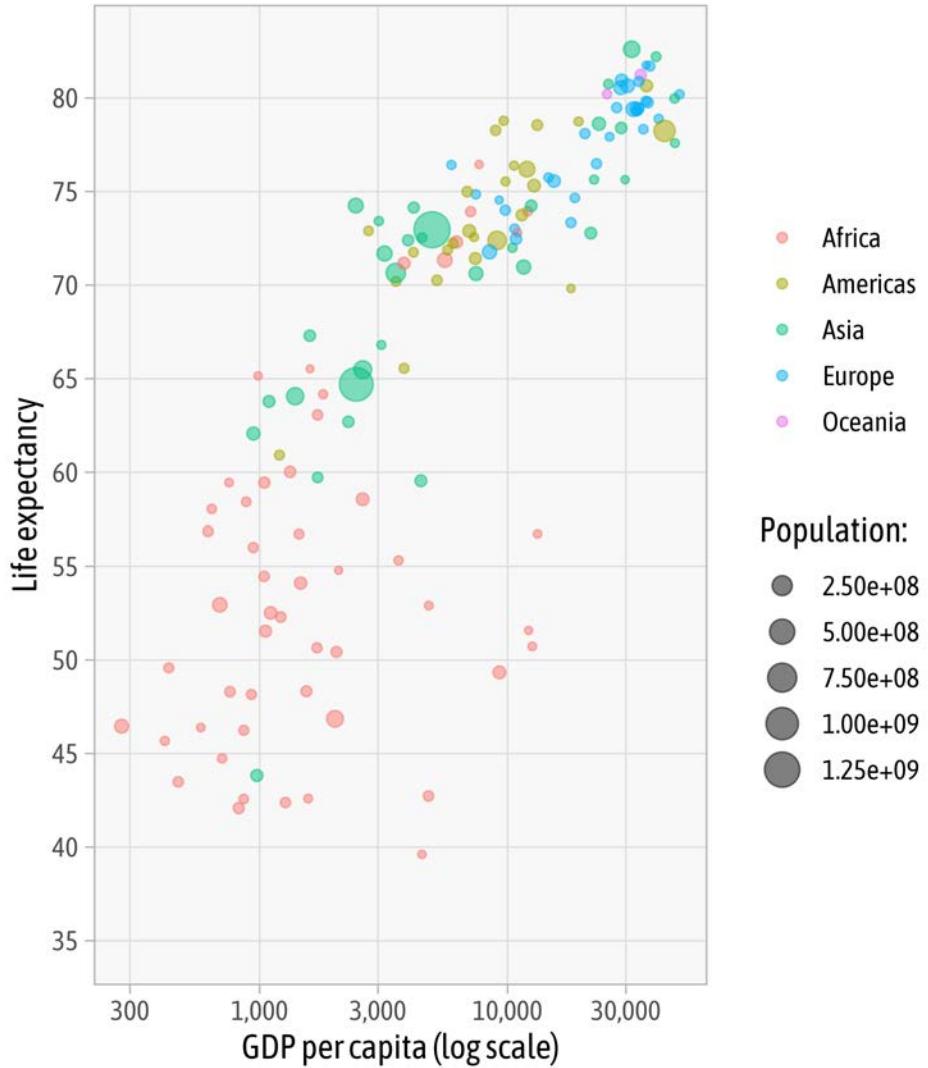
Health & Income of Nations in 2007



Modify Position Scales

```
1 g +
2   scale_x_log10(
3     labels = scales::label_comma(),
4     name = "GDP per capita (log scale)"
5   ) +
6   scale_y_continuous(
7     limits = c(35, NA),
8     breaks = seq(30, 85, by = 5)
9   ) +
10  scale_color_discrete() +
11  scale_size()
```

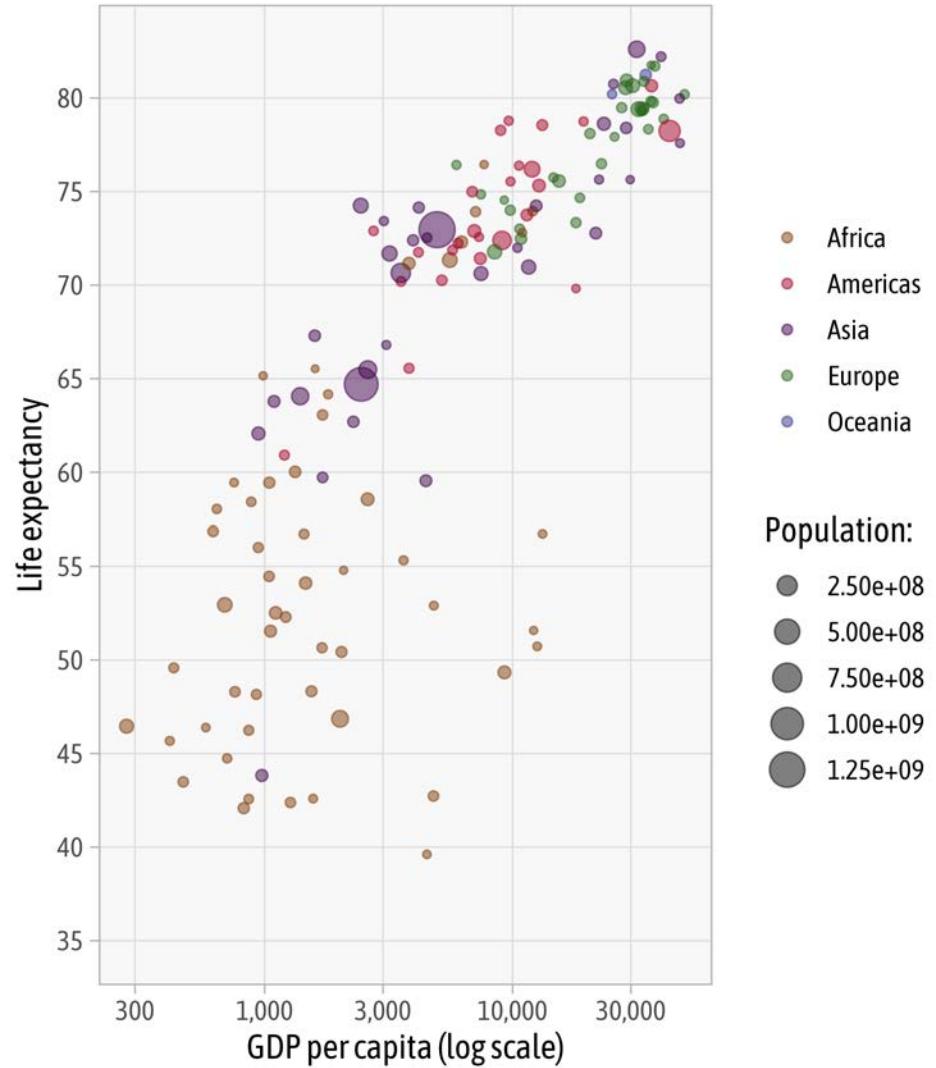
Health & Income of Nations in 2007



Modify Color Scales

```
1 g +
2   scale_x_log10(
3     labels = scales::label_comma(),
4     name = "GDP per capita (log scale)"
5   ) +
6   scale_y_continuous(
7     limits = c(35, NA),
8     breaks = seq(30, 85, by = 5)
9   ) +
10  scale_color_discrete(
11    type = continent_colors
12  ) +
13  scale_size()
```

Health & Income of Nations in 2007



continent_colors

```
1 gapminder:::continent_colors
```

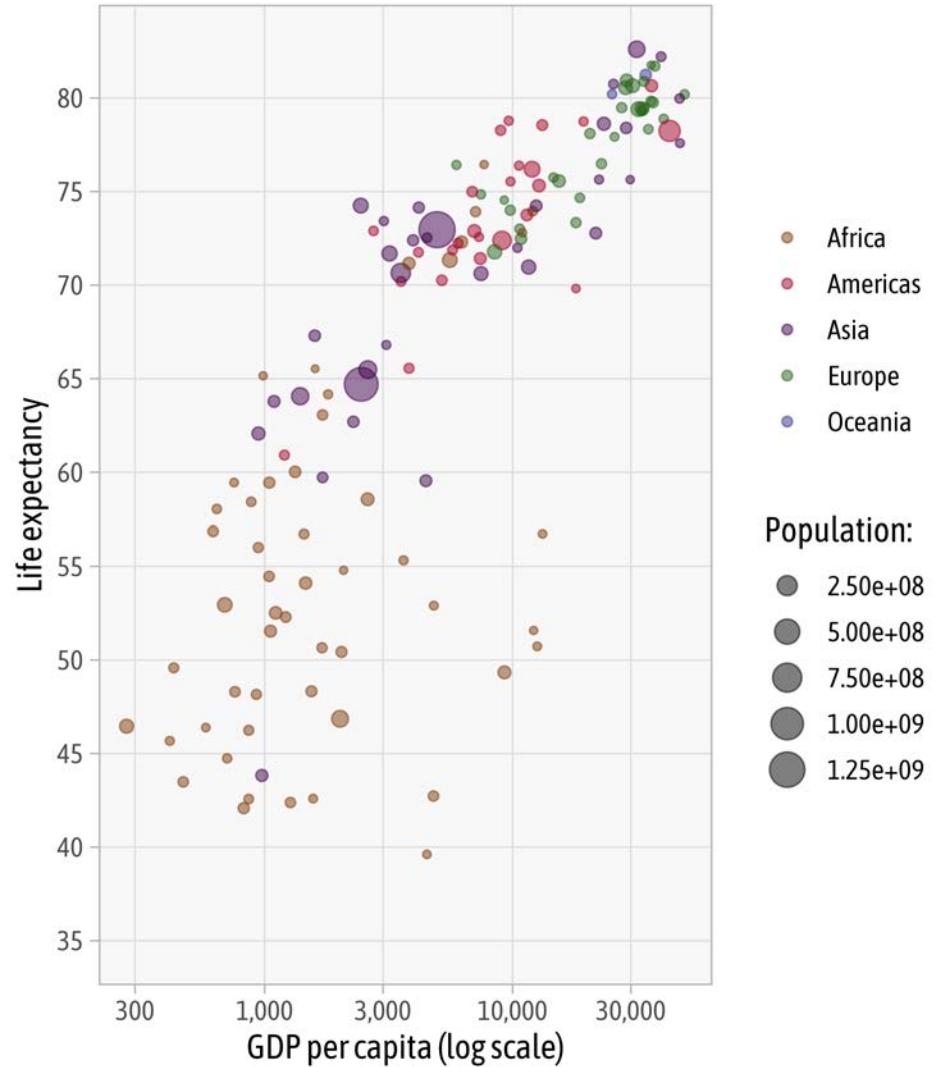
```
Africa   Americas      Asia    Europe   Oceania  
"#7F3B08" "#A50026" "#40004B" "#276419" "#313695"
```



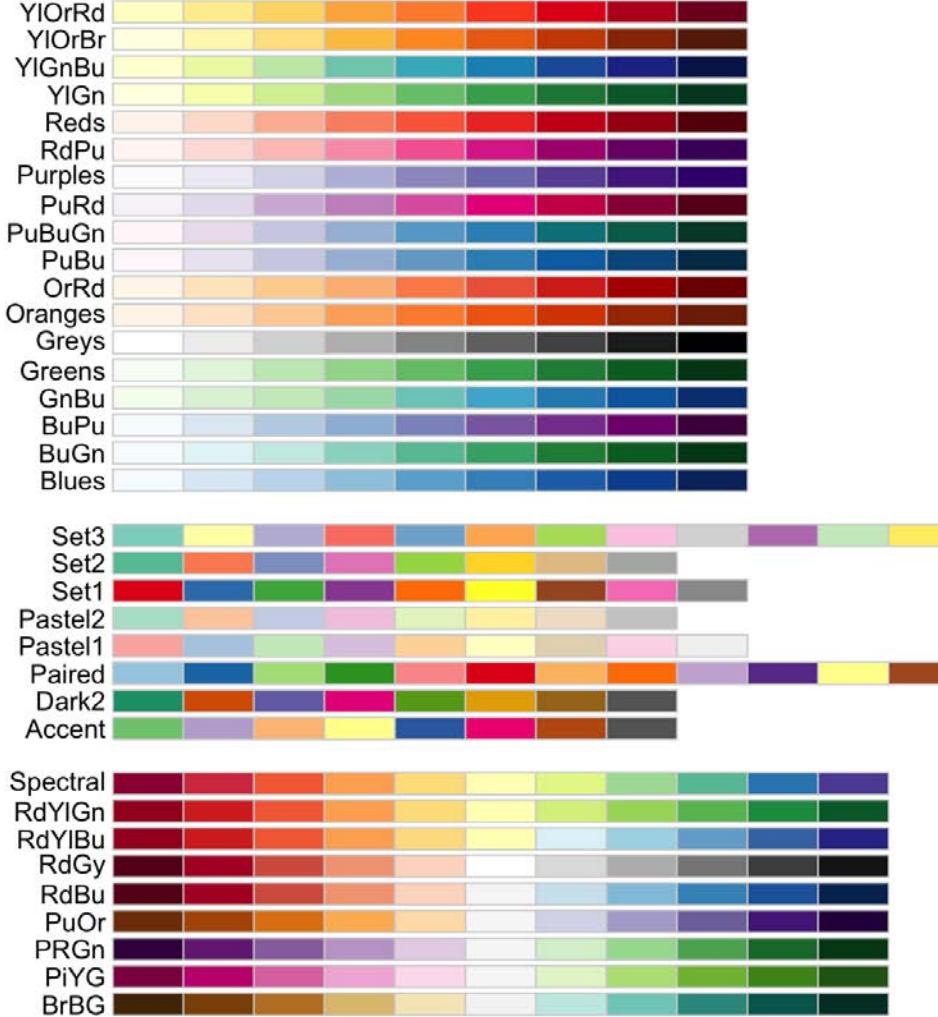
Modify Color Scales

```
1 g +
2   scale_x_log10(
3     labels = scales::label_comma(),
4     name = "GDP per capita (log scale)"
5   ) +
6   scale_y_continuous(
7     limits = c(35, NA),
8     breaks = seq(30, 85, by = 5)
9   ) +
10  scale_color_manual(
11    values = continent_colors
12  ) +
13  scale_size()
```

Health & Income of Nations in 2007



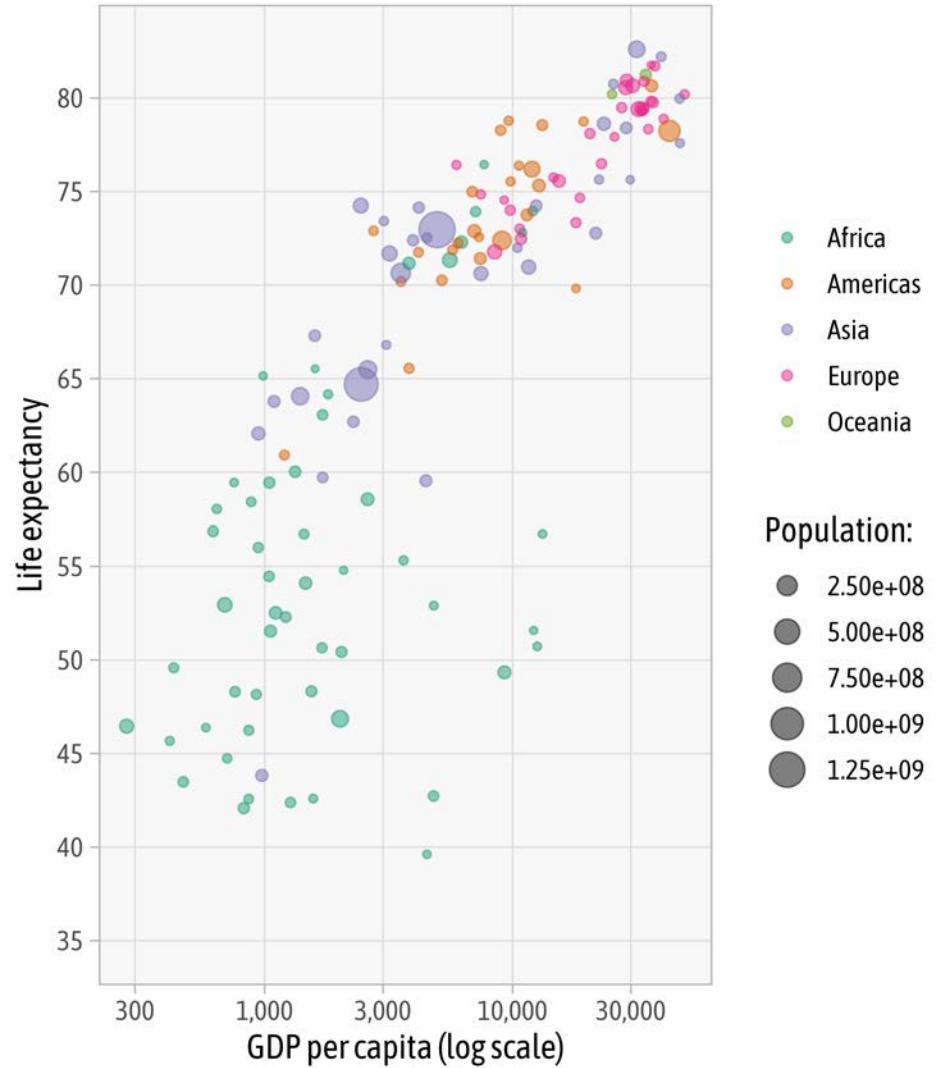
Brewer Palettes



Modify Color Scales

```
1 g +
2   scale_x_log10(
3     labels = scales::label_comma(),
4     name = "GDP per capita (log scale)"
5   ) +
6   scale_y_continuous(
7     limits = c(35, NA),
8     breaks = seq(30, 85, by = 5)
9   ) +
10  scale_color_brewer(
11    palette = "Dark2"
12  ) +
13  scale_size()
```

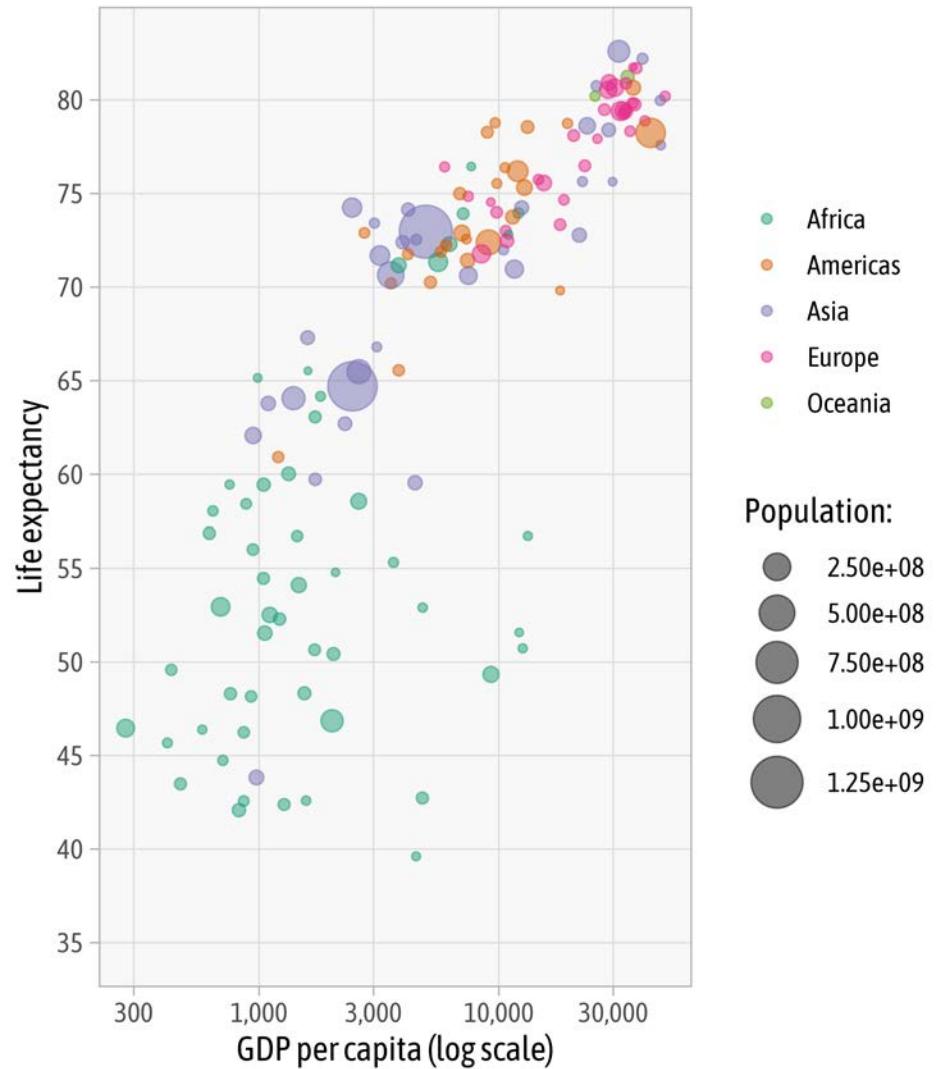
Health & Income of Nations in 2007



Modify Color Scales

```
1 g +
2   scale_x_log10(
3     labels = scales::label_comma(),
4     name = "GDP per capita (log scale)"
5   ) +
6   scale_y_continuous(
7     limits = c(35, NA),
8     breaks = seq(30, 85, by = 5)
9   ) +
10  scale_color_brewer(
11    palette = "Dark2"
12  ) +
13  scale_size(
14    range = c(1, 9)
15 )
```

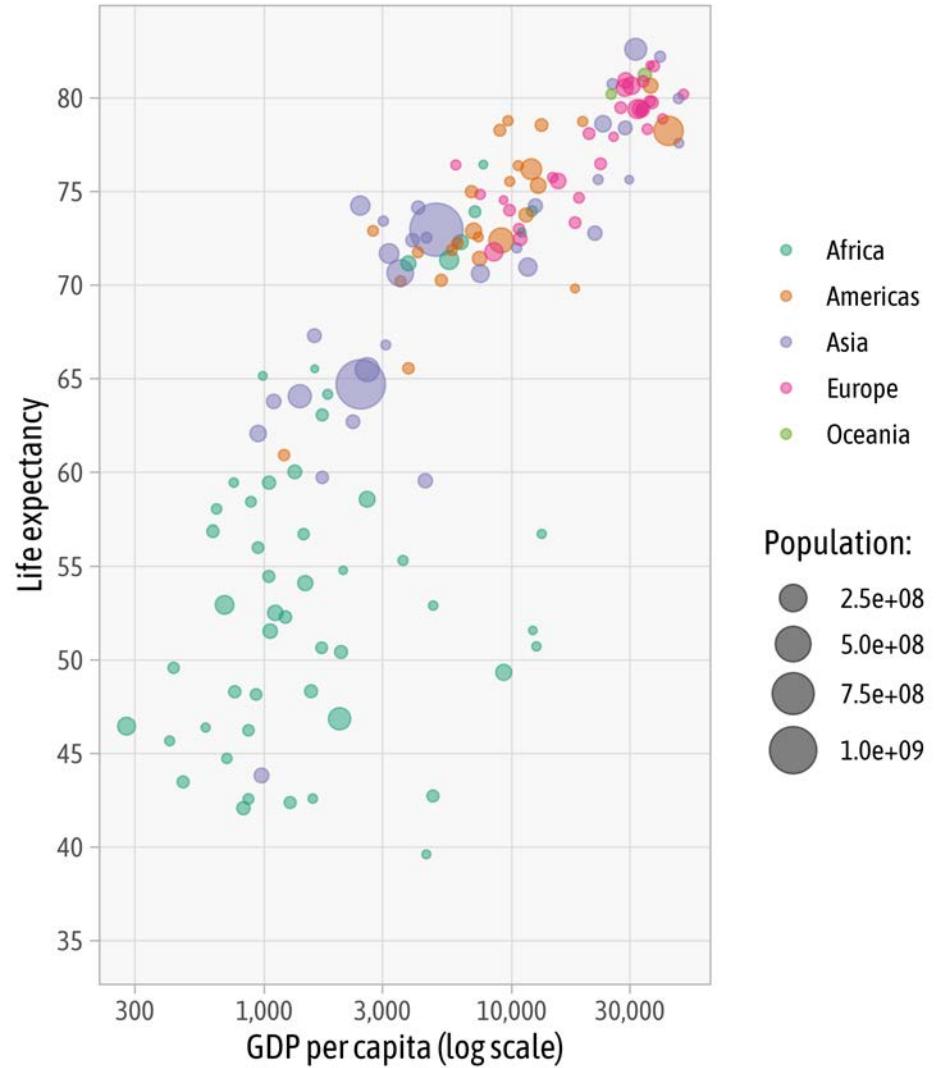
Health & Income of Nations in 2007



Modify Size Scales

```
1 g +
2   scale_x_log10(
3     labels = scales::label_comma(),
4     name = "GDP per capita (log scale)"
5   ) +
6   scale_y_continuous(
7     limits = c(35, NA),
8     breaks = seq(30, 85, by = 5)
9   ) +
10  scale_color_brewer(
11    palette = "Dark2"
12  ) +
13  scale_size(
14    range = c(1, 9),
15    breaks = 1:4*2500000000
16 )
```

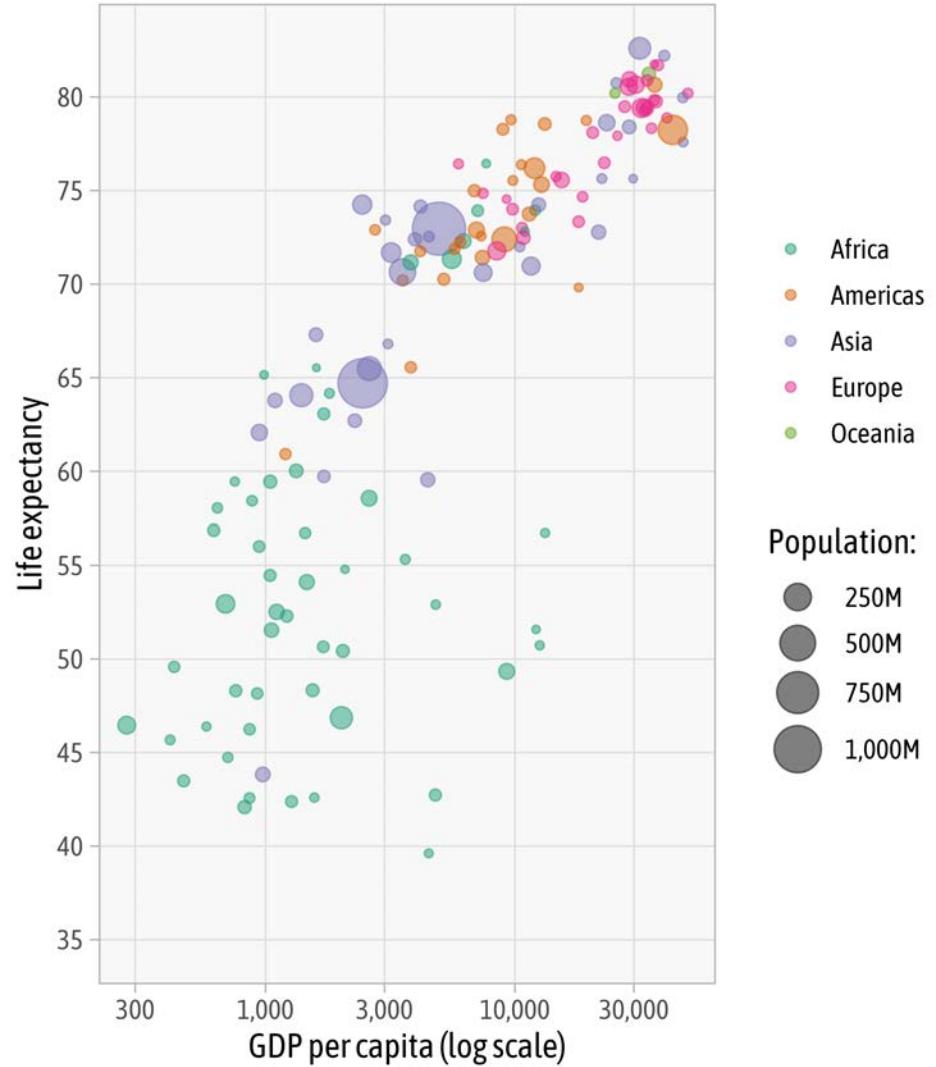
Health & Income of Nations in 2007



Modify Size Scales

```
1 g +
2   scale_x_log10(
3     labels = scales::label_comma(),
4     name = "GDP per capita (log scale)"
5   ) +
6   scale_y_continuous(
7     limits = c(35, NA),
8     breaks = seq(30, 85, by = 5)
9   ) +
10  scale_color_brewer(
11    palette = "Dark2"
12  ) +
13  scale_size(
14    range = c(1, 9),
15    breaks = 1:4*2500000000,
16    labels = scales::label_comma(
17      scale = 1 / 10^6,
18      suffix = "M"
19    )
20  )
```

Health & Income of Nations in 2007



— Coordinate Systems —



Coordinate Systems

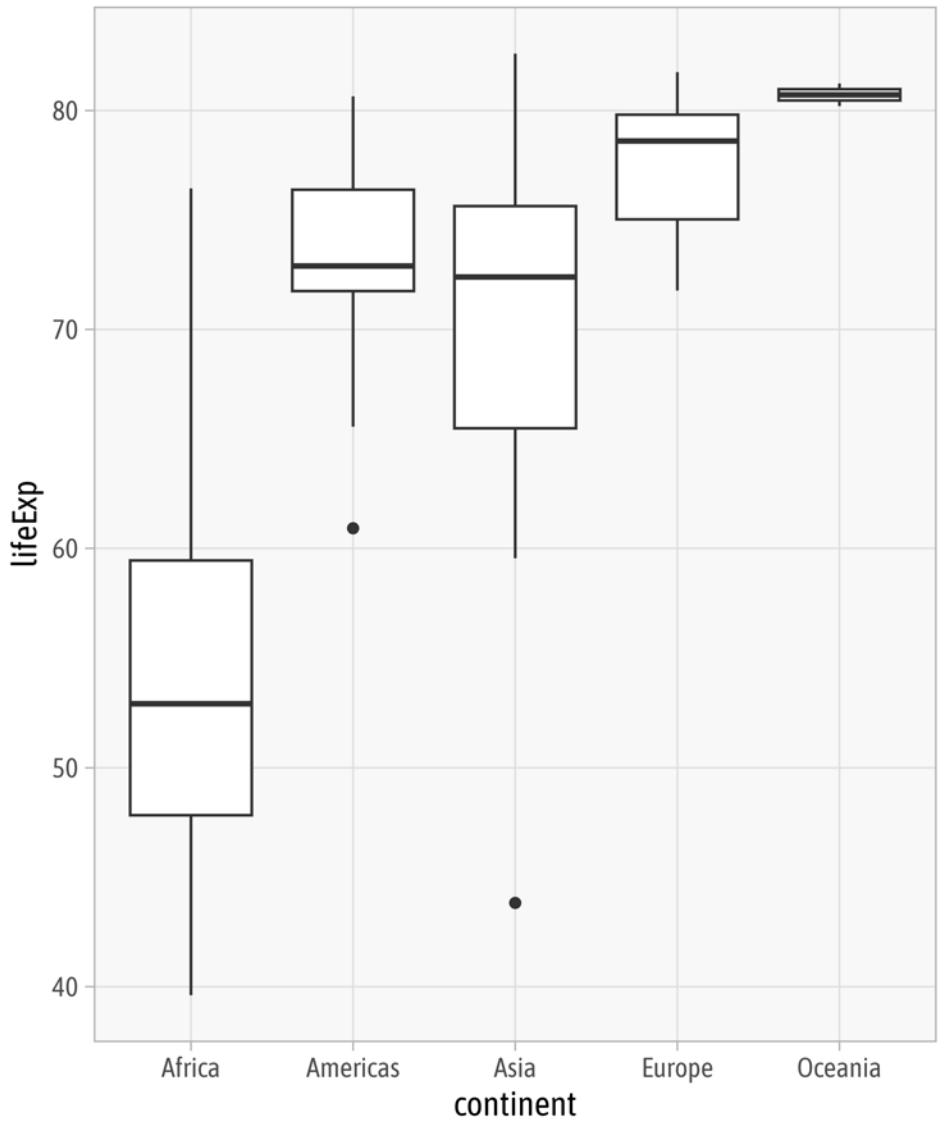
= interpret the position aesthetics

- **linear coordinate systems:** preserve the geometrical shapes
 - `coord_cartesian()`
 - `coord_fixed()`
 - `coord_flip()`
- **non-linear coordinate systems:** likely change the geometrical shapes
 - `coord_polar()`
 - `coord_trans()`
 - `coord_map()` and `coord_sf()`



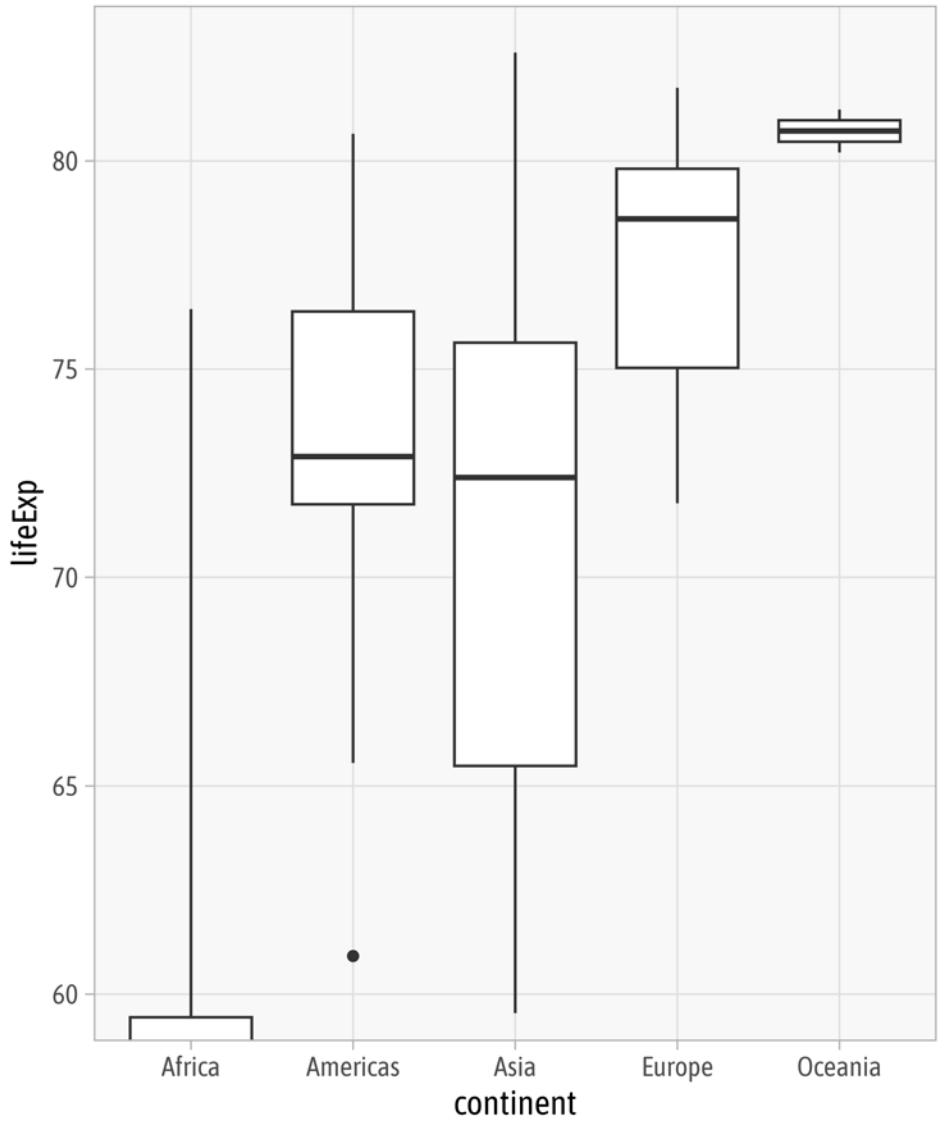
Cartesian Coordinate System

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = continent, y = lifeExp)  
4 ) +  
5 geom_boxplot() +  
6 coord_cartesian()
```



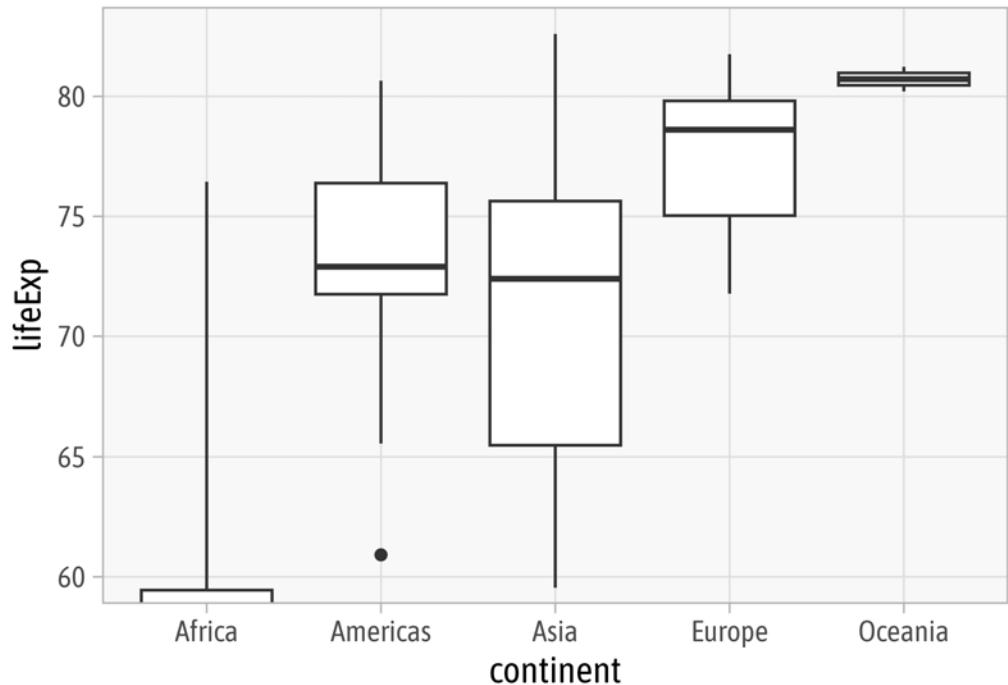
Cartesian Coordinate System

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = continent, y = lifeExp)  
4 ) +  
5 geom_boxplot() +  
6 coord_cartesian(  
7   ylim = c(60, NA)  
8 )
```

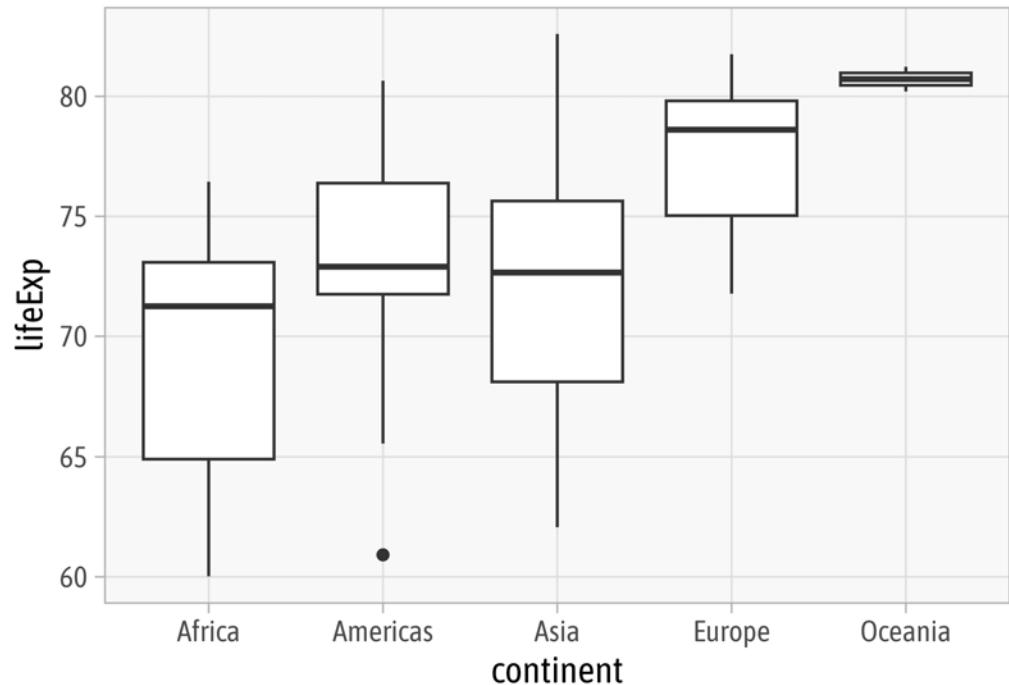


Changing Limits

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = continent, y = lifeExp)  
4 ) +  
5 geom_boxplot() +  
6 coord_cartesian(  
7   ylim = c(60, NA)  
8 )
```

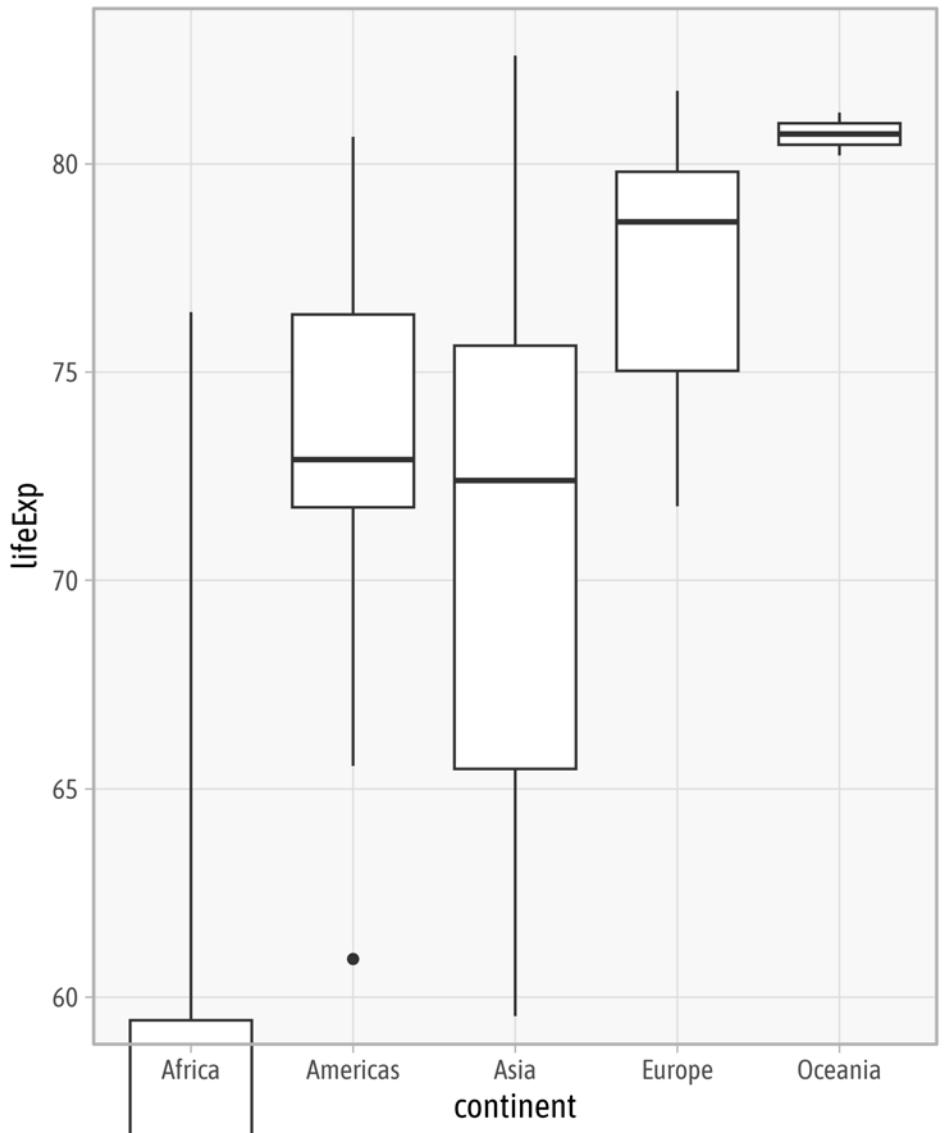


```
1 ggplot(  
2   data = gm2007,  
3   aes(x = continent, y = lifeExp)  
4 ) +  
5 geom_boxplot() +  
6 scale_y_continuous(  
7   limits = c(60, NA)  
8 )
```



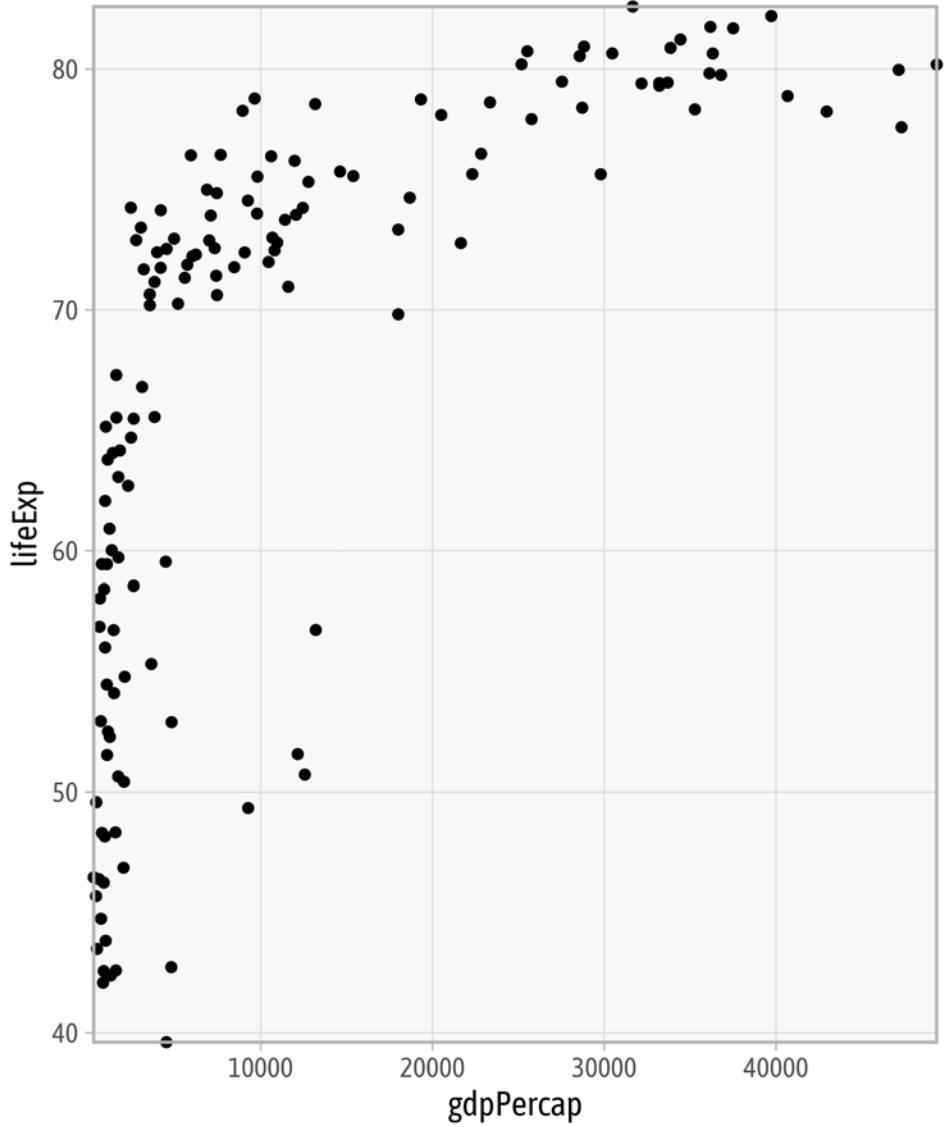
Clipping

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = continent, y = lifeExp)  
4 ) +  
5 geom_boxplot() +  
6 coord_cartesian(  
7   ylim = c(60, NA),  
8   clip = "off"  
9 )
```



Remove All Padding

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp)  
4 ) +  
5 geom_point() +  
6 coord_cartesian(  
7   expand = FALSE,  
8   clip = "off"  
9 )
```

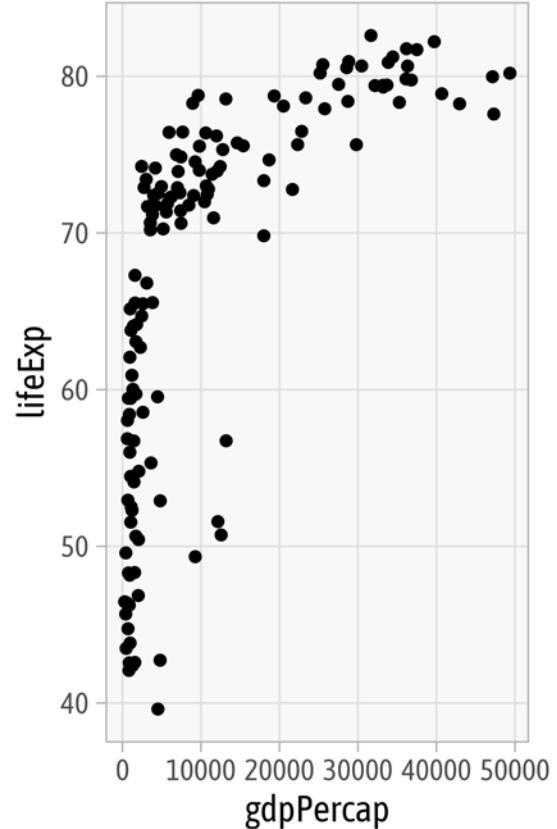


Fixed Coordinate System

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp)  
4 ) +  
5 geom_point() +  
6 coord_fixed()
```

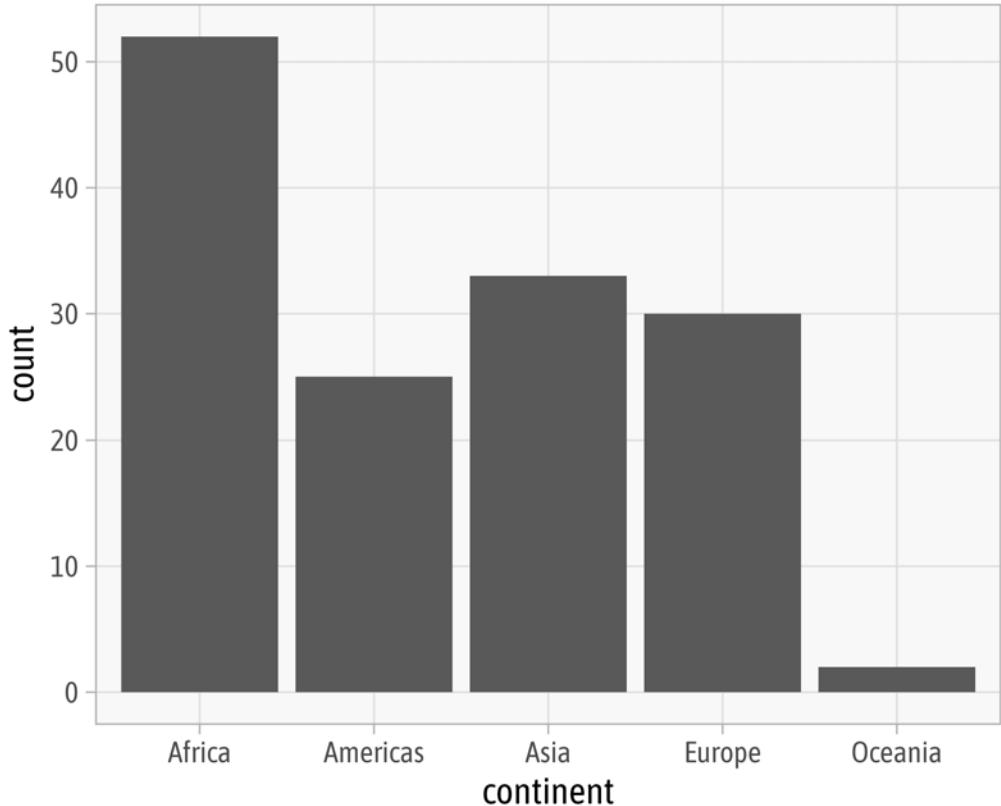


```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp)  
4 ) +  
5 geom_point() +  
6 coord_fixed(ratio = 2000)
```

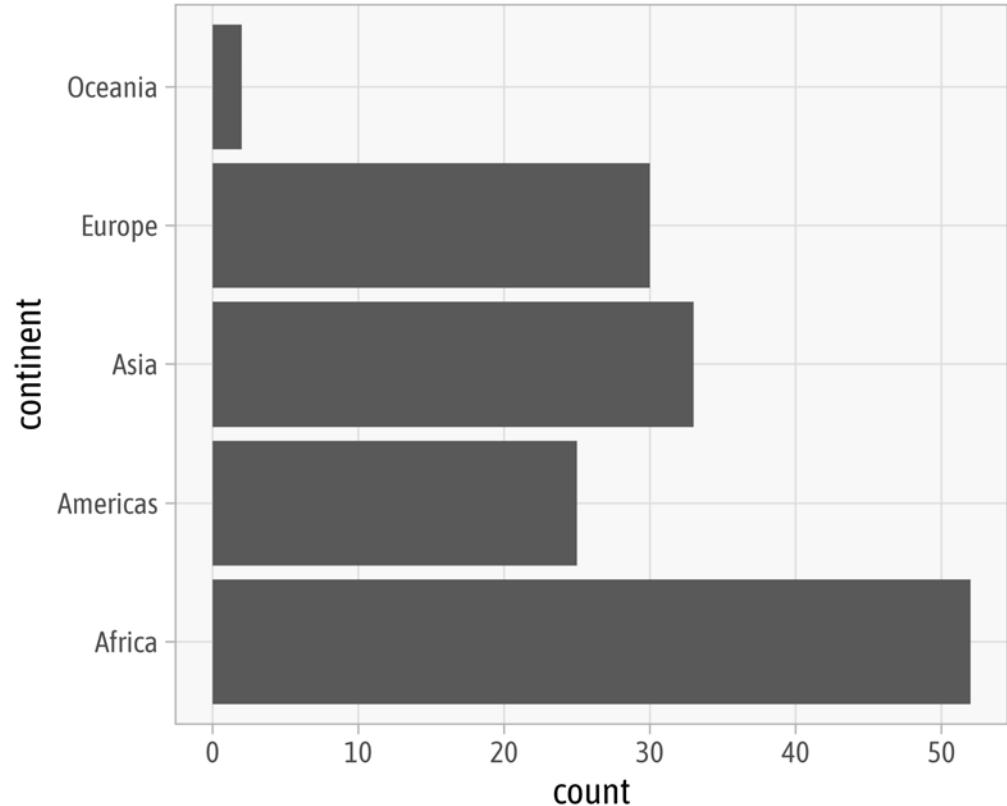


Flipped Coordinate System

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = continent)  
4 ) +  
5 geom_bar() +  
6 coord_cartesian()
```

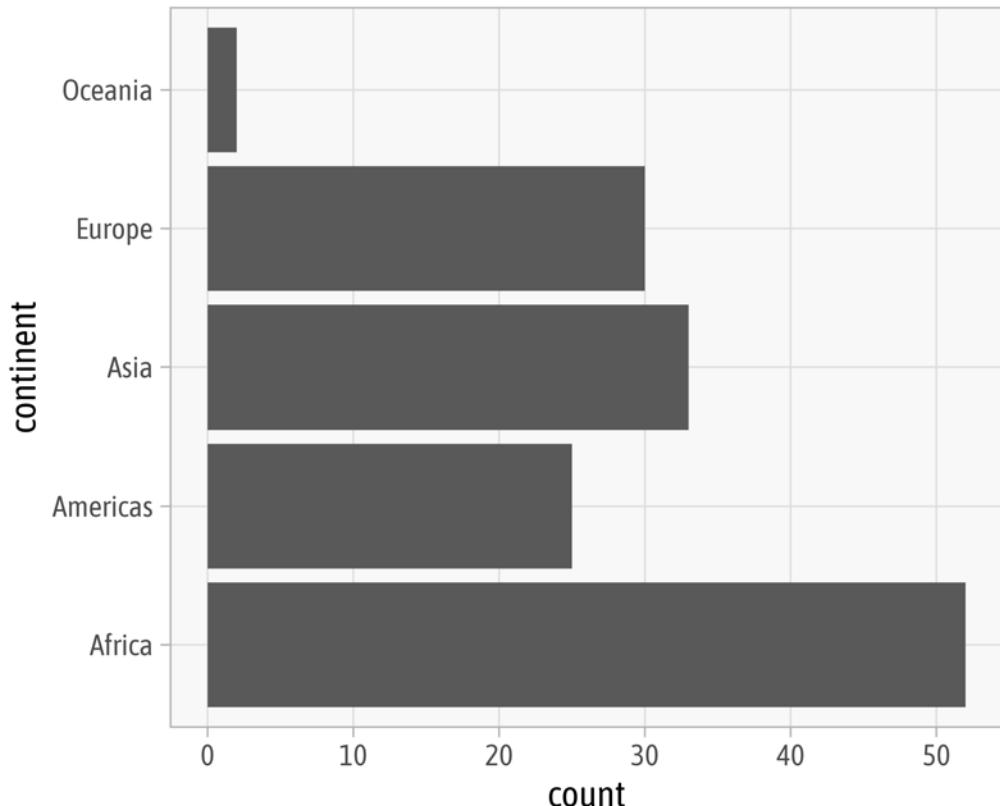


```
1 ggplot(  
2   data = gm2007,  
3   aes(x = continent)  
4 ) +  
5 geom_bar() +  
6 coord_flip()
```

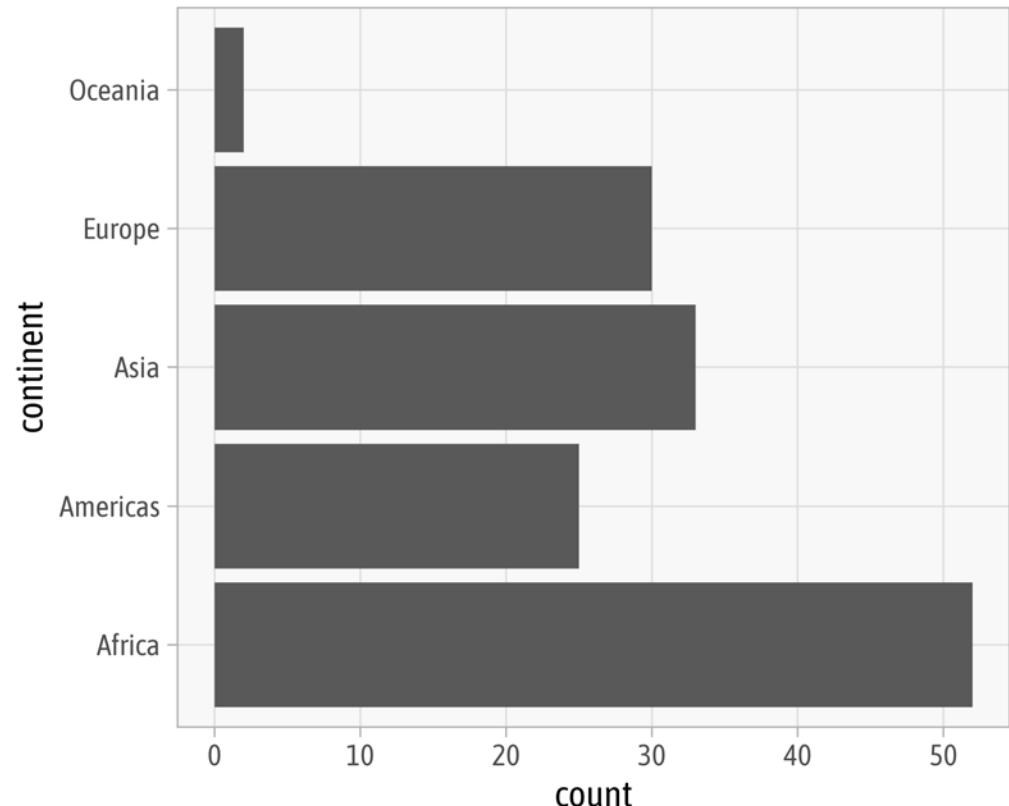


Flipped Coordinate System

```
1 ggplot(  
2   data = gm2007,  
3   aes(y = continent)  
4 ) +  
5 geom_bar() +  
6 coord_cartesian()
```

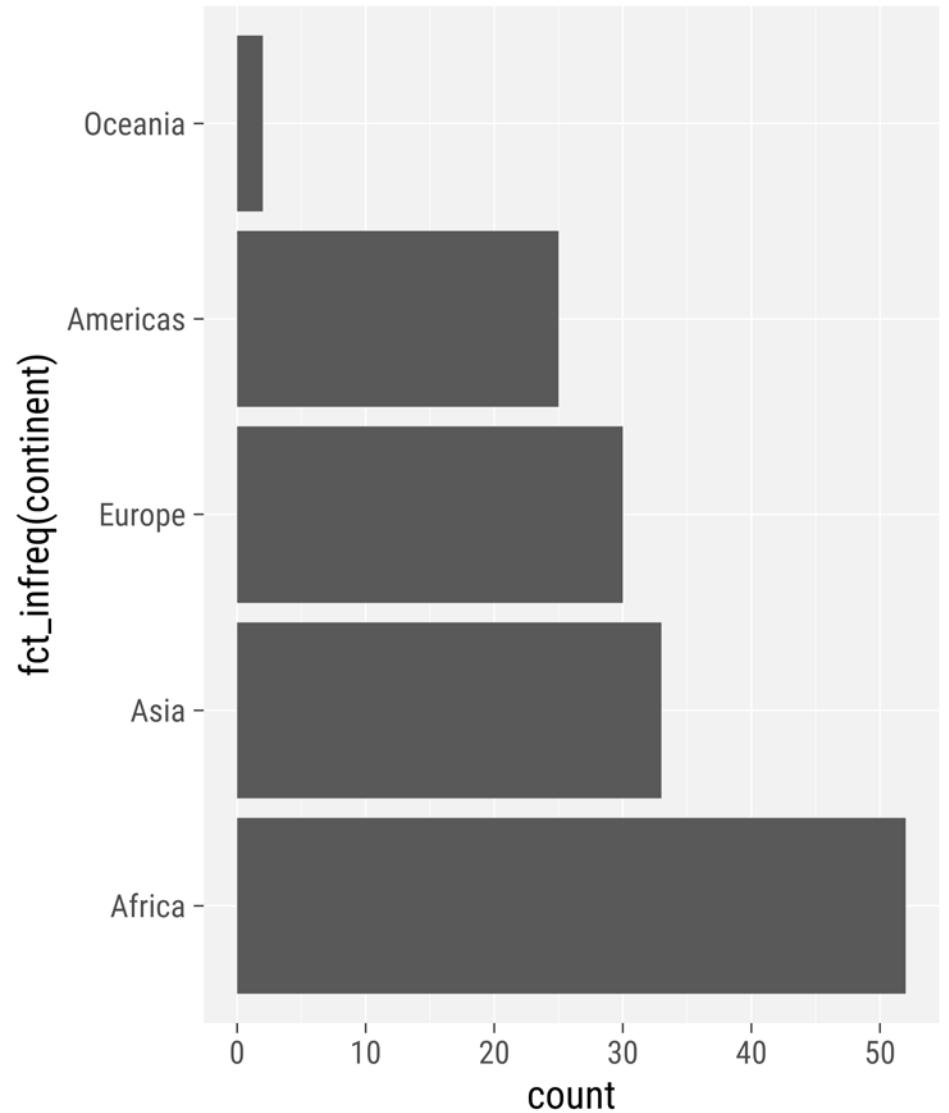


```
1 ggplot(  
2   data = gm2007,  
3   aes(x = continent)  
4 ) +  
5 geom_bar() +  
6 coord_flip()
```



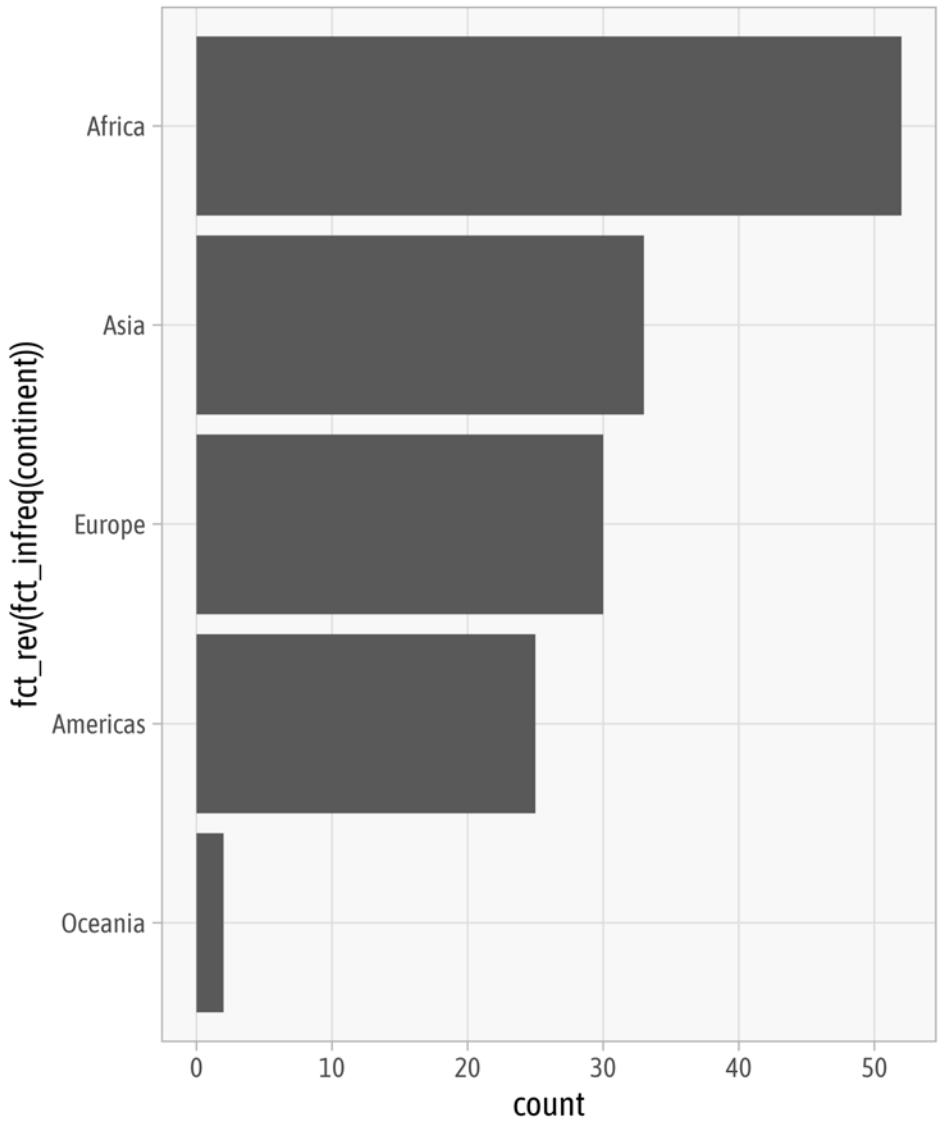
Reminder: Sort Your Bars!

```
1 library(forcats)
2
3 ggplot(
4   data = gm2007,
5   aes(y = fct_infreq(continent)))
6 ) +
7 geom_bar()
```



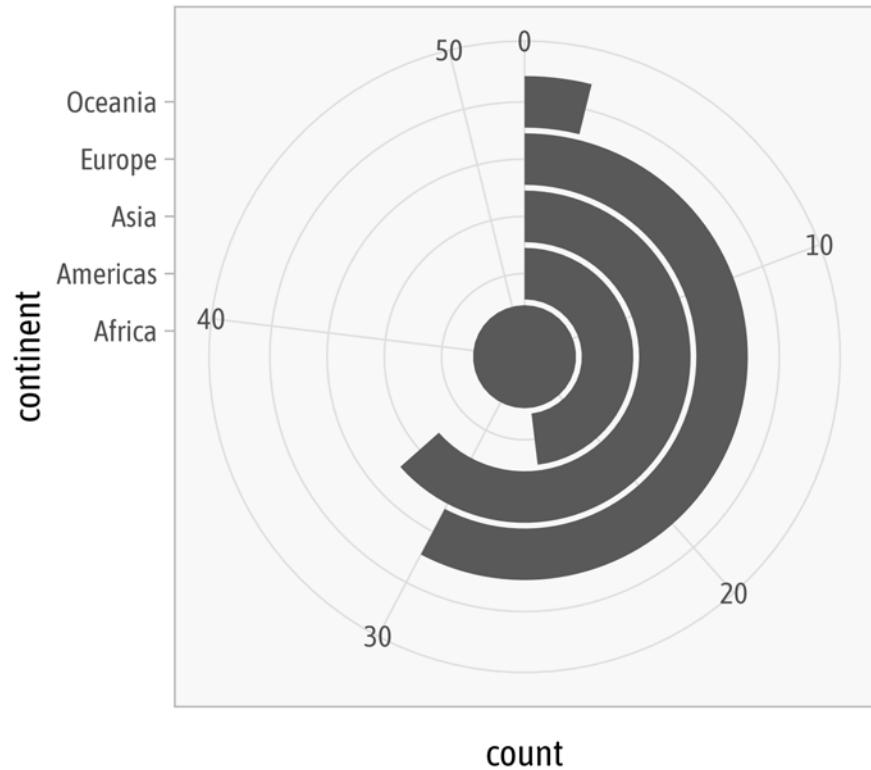
Reminder: Sort Your Bars!

```
1 ggplot(  
2   data = gm2007,  
3   aes(y = fct_rev(  
4     fct_infreq(continent)  
5   ))  
6 ) +  
7 geom_bar()
```

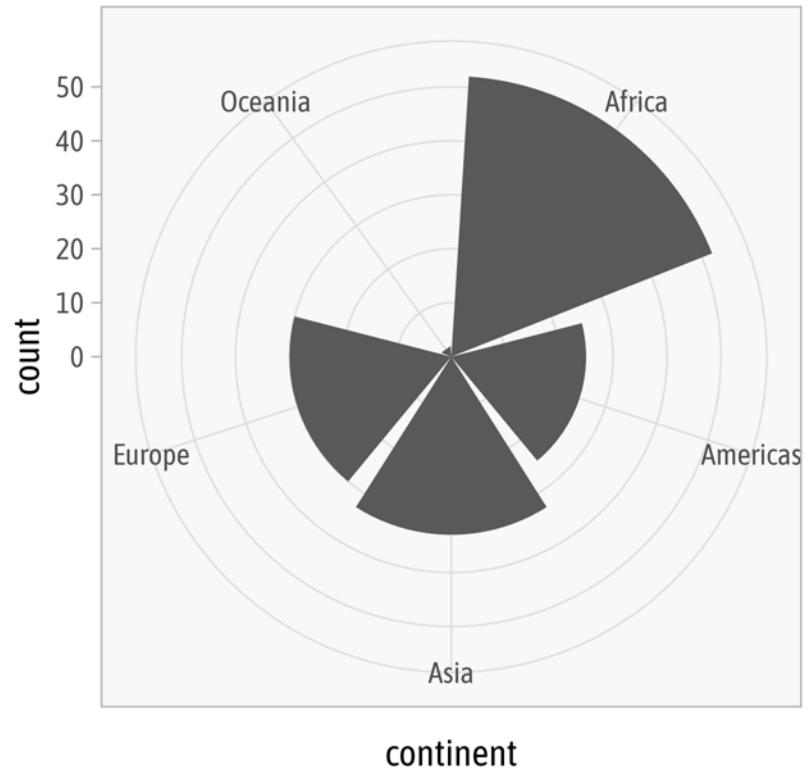


Circular Coordinate System

```
1 ggplot(  
2   data = gm2007,  
3   aes(y = continent))  
4 ) +  
5 geom_bar() +  
6 coord_polar()
```

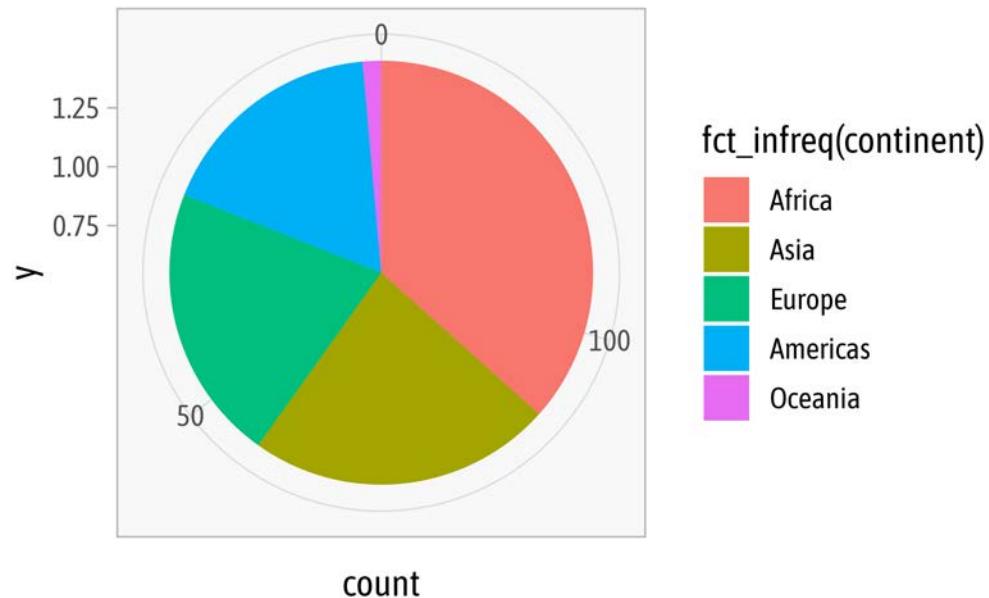


```
1 ggplot(  
2   data = gm2007,  
3   aes(y = continent))  
4 ) +  
5 geom_bar() +  
6 coord_polar(theta = "y")
```

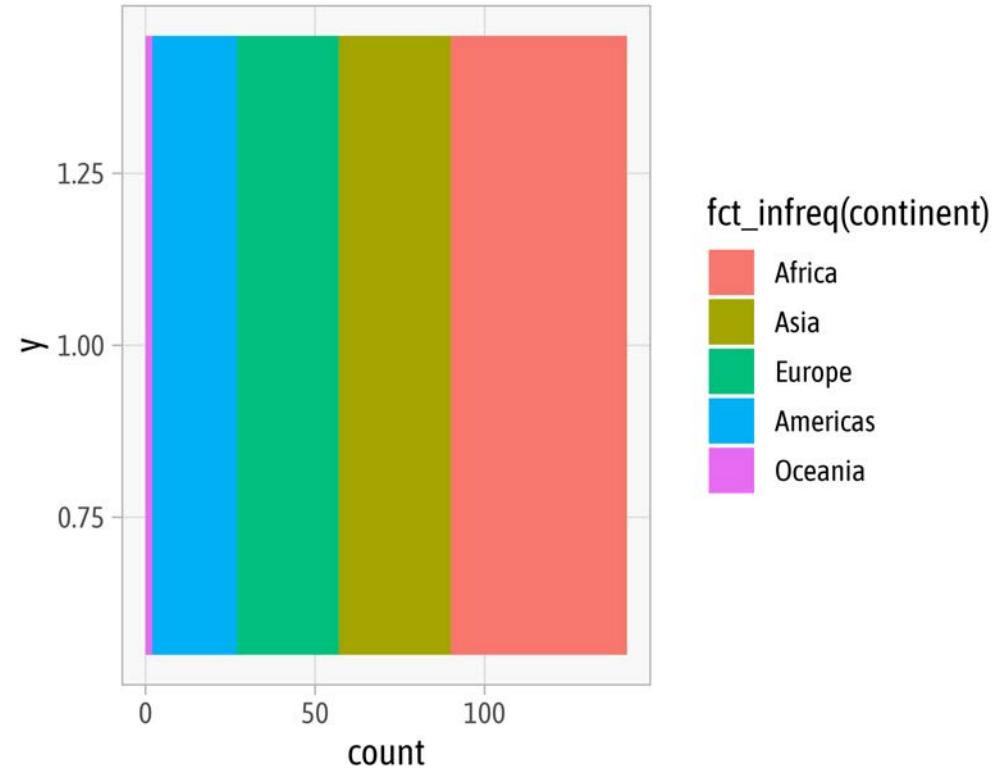


Circular Coordinate System

```
1 ggplot(  
2   data = gm2007,  
3   aes(y = 1,  
4       fill = fct_infreq(continent))  
5 ) +  
6 geom_bar(position = "stack") +  
7 coord_polar(direction = -1)
```

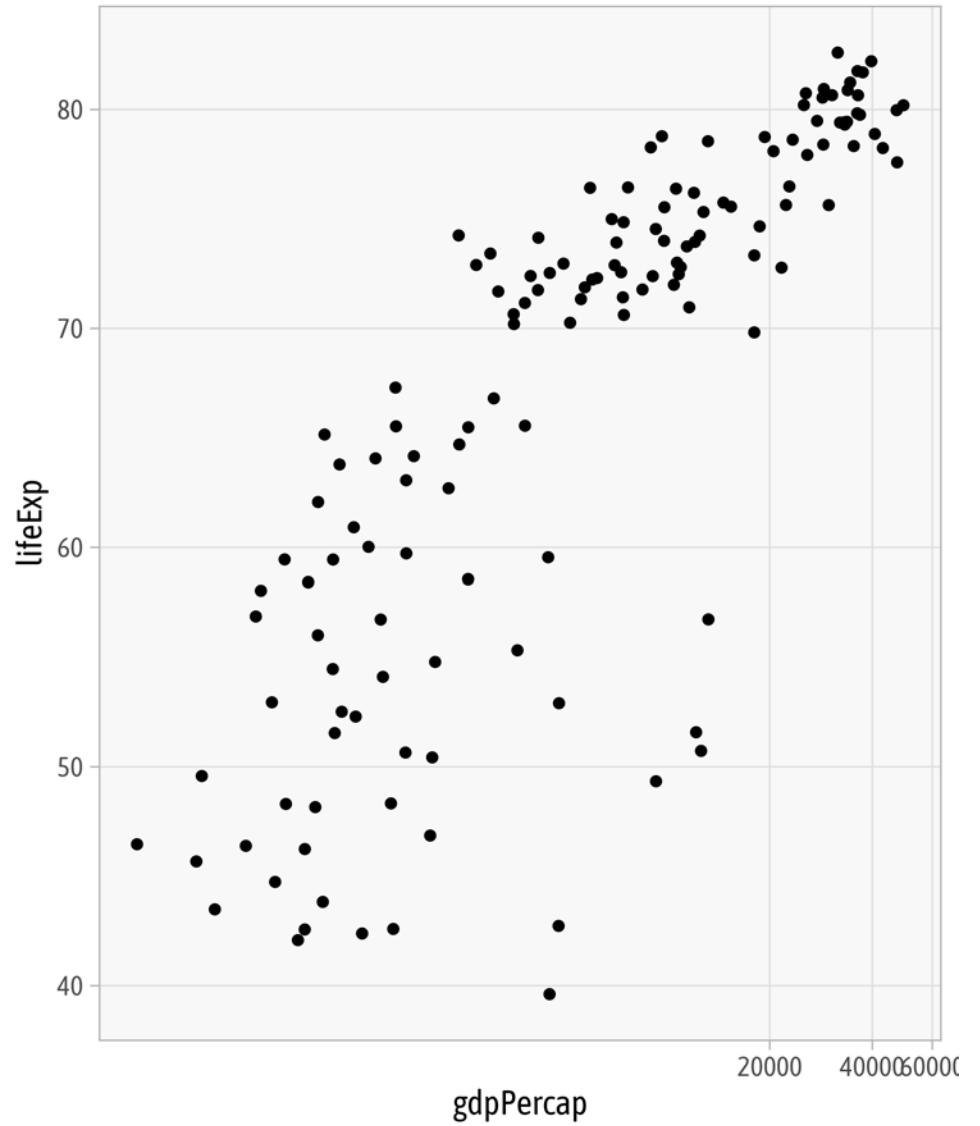


```
1 ggplot(  
2   data = gm2007,  
3   aes(y = 1,  
4       fill = fct_infreq(continent))  
5 ) +  
6 geom_bar(position = "stack") +  
7 coord_cartesian()
```



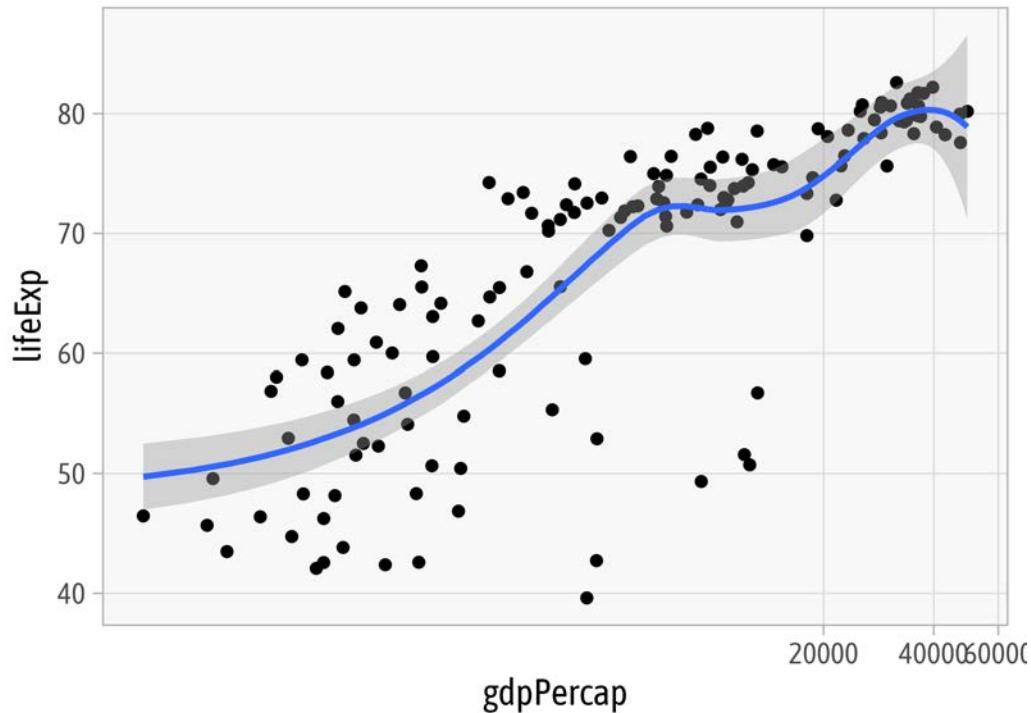
Transform a Coordinate System

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp)  
4 ) +  
5 geom_point() +  
6 coord_trans(x = "log10")
```

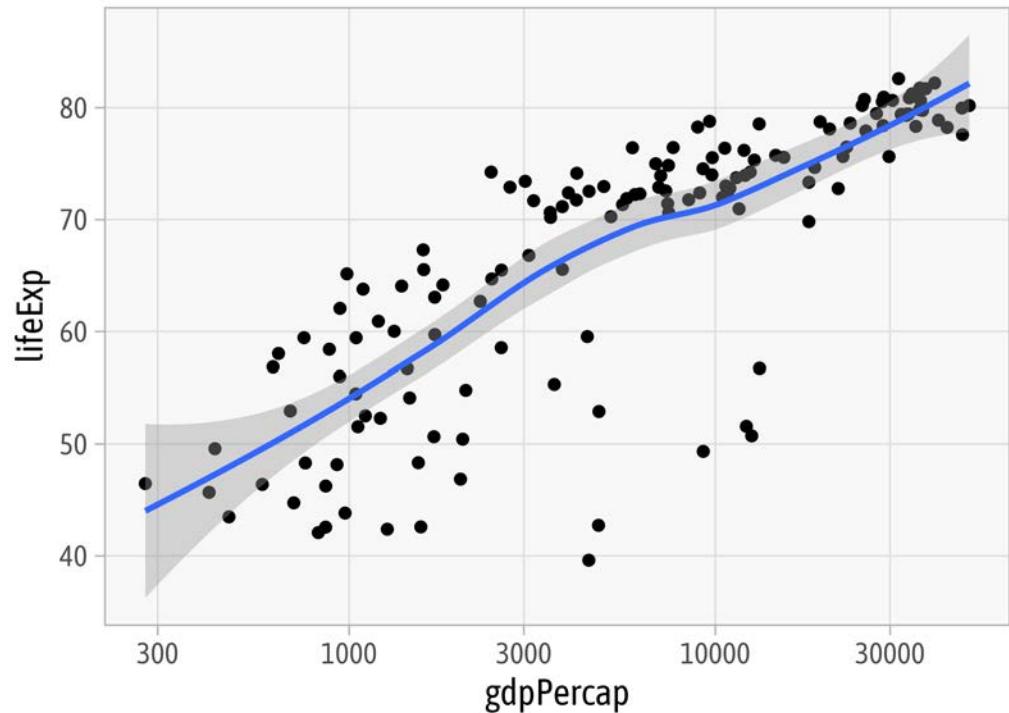


Transform a Coordinate System

```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp)  
4 ) +  
5 geom_point() +  
6 geom_smooth() +  
7 coord_trans(x = "log10")
```



```
1 ggplot(  
2   data = gm2007,  
3   aes(x = gdpPercap, y = lifeExp)  
4 ) +  
5 geom_point() +  
6 geom_smooth() +  
7 scale_x_log10()
```



Exercise

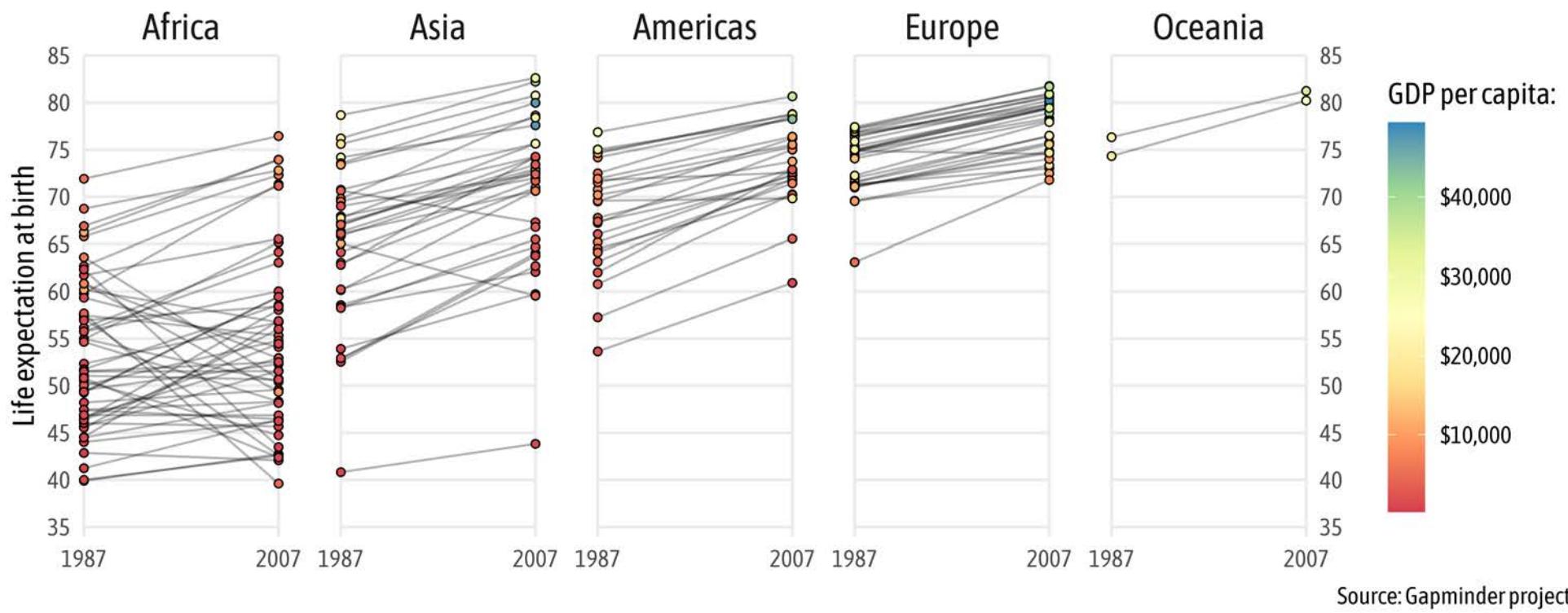


Your Turn: A Polished ggplot

Create the following small multiples of slope charts as close as possible

Life expectancy has risen in most countries within the last 20 years

However, several African countries, especially in the South, such as Zimbabwe, Eswatini, Botswana, Namibia and South Africa have experienced a considerable drop. Away from Africa, only Iraq and DR Korea (Asia), and Montenegro (Europe) are the only countries with a negative trend.



Your Turn: Data Preparation

```
1 gapminder %>%
2   filter(year %in% c(1987, 2007))
```

```
# A tibble: 284 × 7
  country    continent year lifeExp      pop gdpPercap income_lvl
  <chr>      <chr>     <dbl>  <dbl>      <dbl>    <dbl>    <chr>
1 Afghanistan Asia      1987    40.8 13867957     852.  1
2 Afghanistan Asia      2007    43.8 31889923     975.  1
3 Albania     Europe    1987     72    3075321    3739.  2
4 Albania     Europe    2007    76.4  3600523    5937.  3
5 Algeria     Africa    1987    65.8 23254956    5681.  3
6 Algeria     Africa    2007    72.3 33333216    6223.  3
7 Angola      Africa    1987    39.9  7874230    2430.  2
8 Angola      Africa    2007    42.7 12420476    4797.  3
9 Argentina   Americas  1987    70.8 31620918    9140.  3
10 Argentina  Americas  2007    75.3 40301927   12779.  4
# i 274 more rows
```

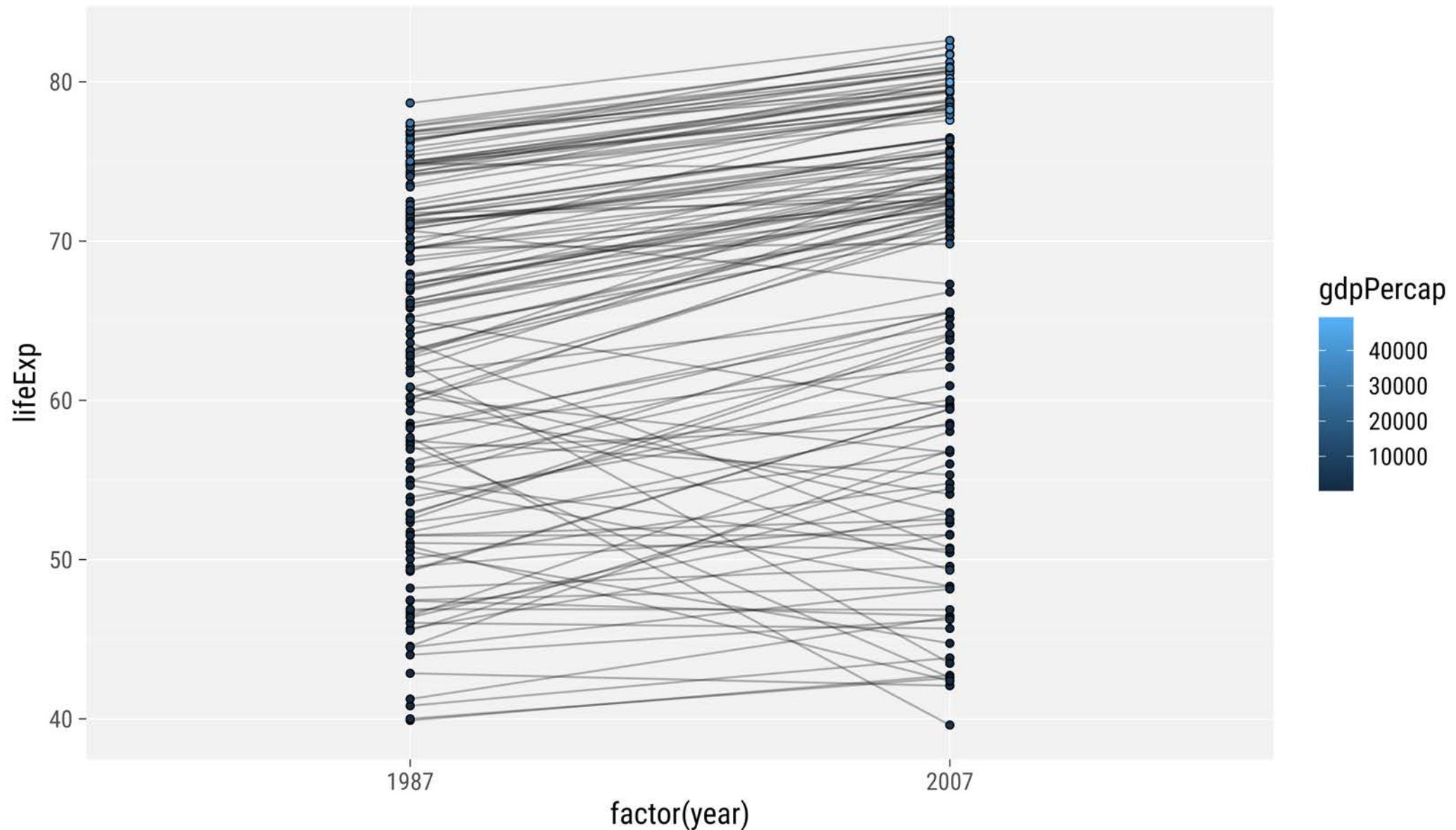


Your Turn: Slope Chart

```
1 gapminder %>%
2   filter(year %in% c(1987, 2007)) %>%
3   ggplot(aes(y = lifeExp, x = factor(year))) +
4   geom_line(aes(group = country), alpha = .3) +
5   geom_point(aes(fill = gdpPercap), shape = 21)
```



Your Turn: Slope Chart

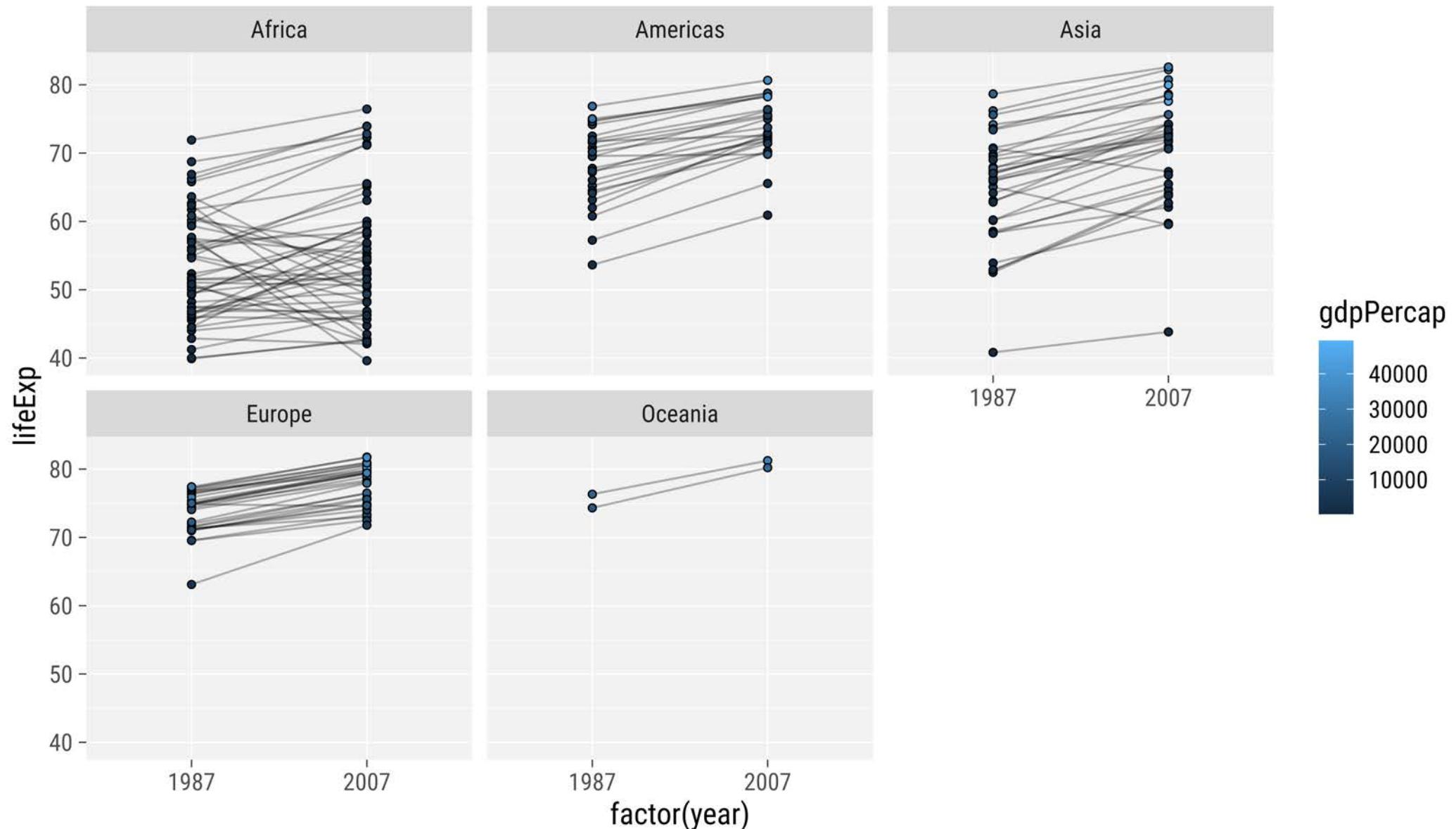


Your Turn: Small Multiples

```
1 gapminder %>%
2   filter(year %in% c(1987, 2007)) %>%
3   ggplot(aes(y = lifeExp, x = factor(year))) +
4   geom_line(aes(group = country), alpha = .3) +
5   geom_point(aes(fill = gdpPercap), shape = 21) +
6   facet_wrap(~continent)
```



Your Turn: Small Multiples

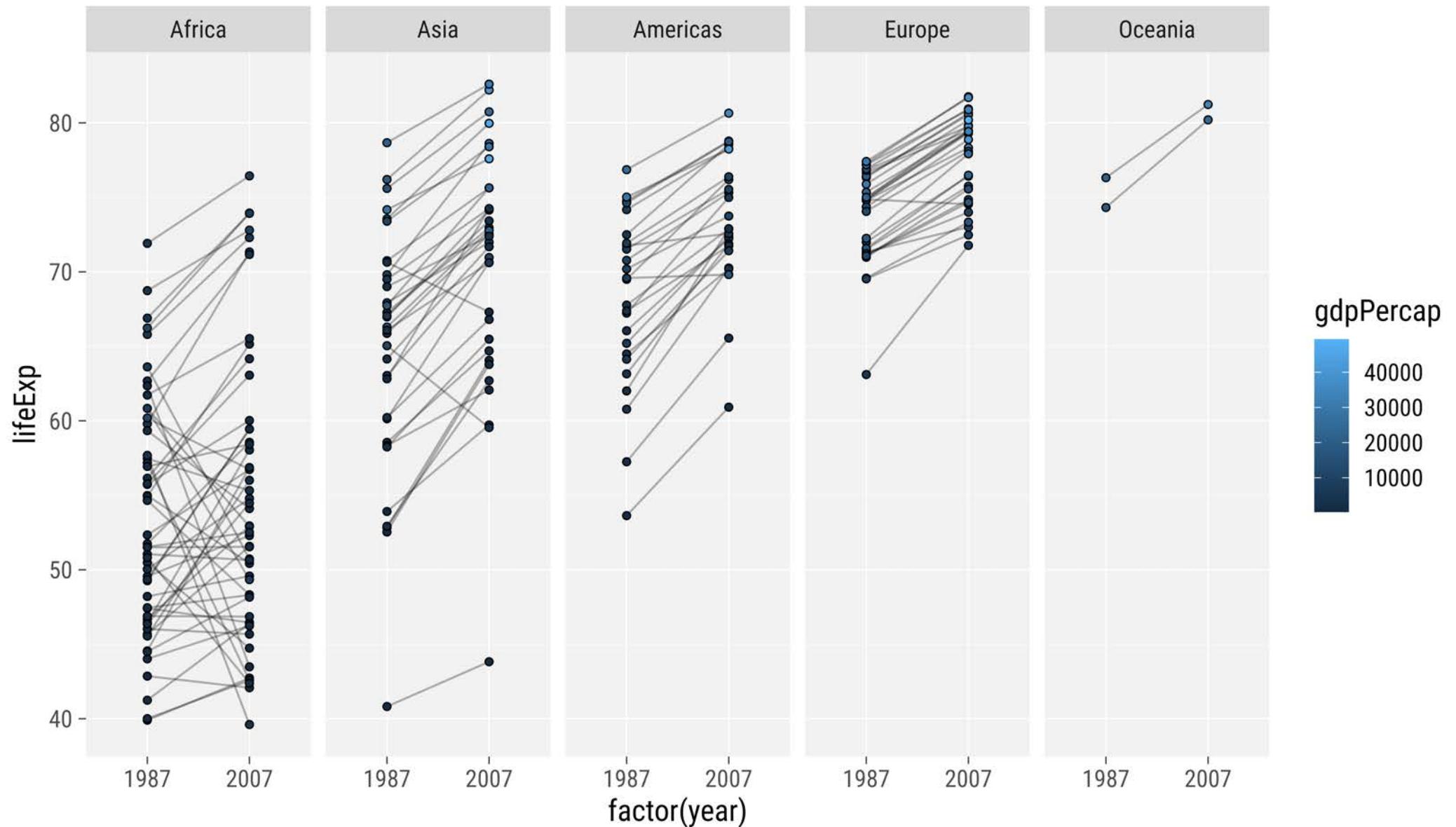


Your Turn: Small Multiples

```
1 gapminder %>%
2   filter(year %in% c(1987, 2007)) %>%
3   ggplot(aes(y = lifeExp, x = factor(year))) +
4   geom_line(aes(group = country), alpha = .3) +
5   geom_point(aes(fill = gdpPercap), shape = 21) +
6   facet_wrap(~fct_reorder(continent, lifeExp), nrow = 1)
```



Your Turn: Small Multiples

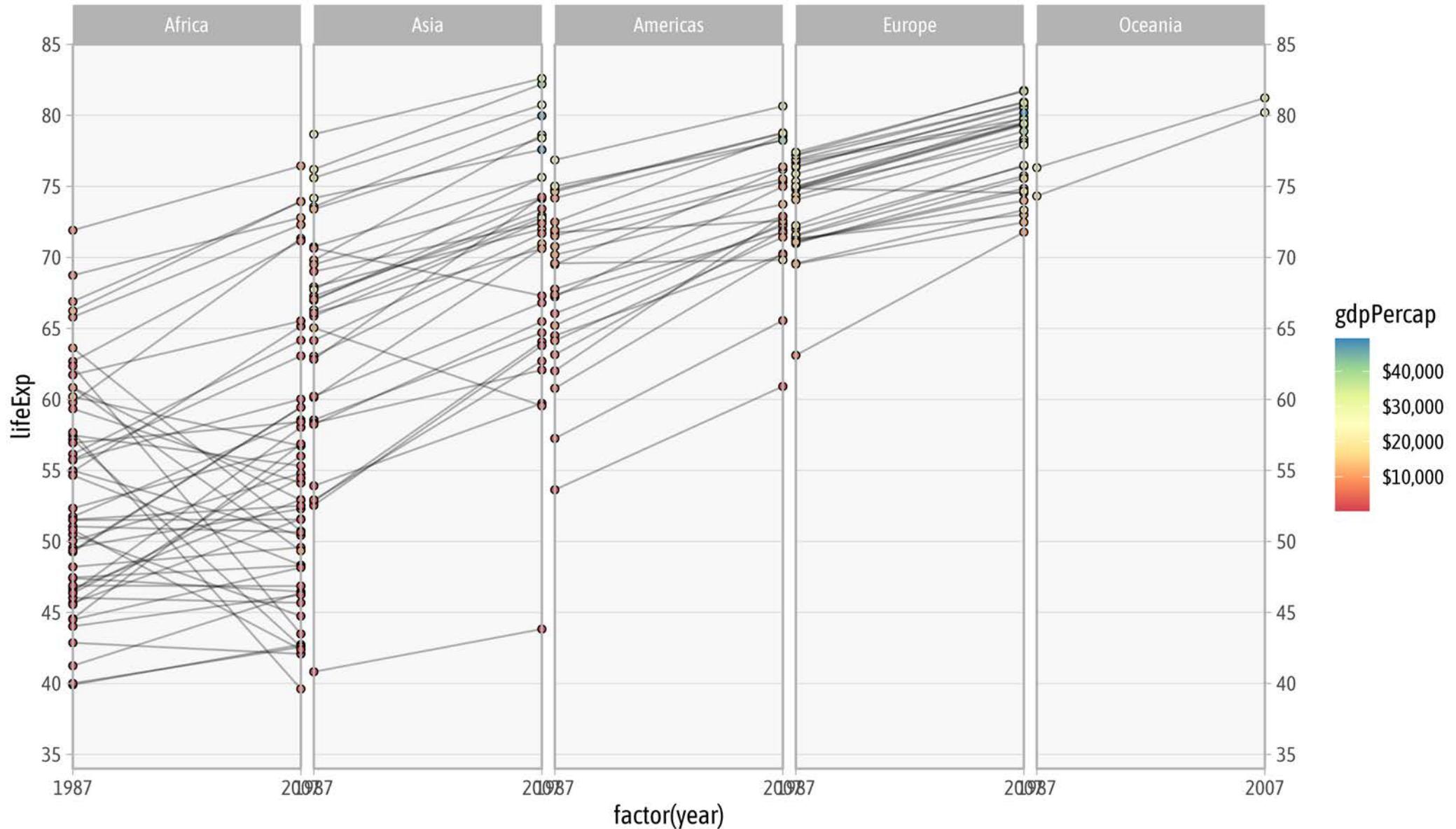


Your Turn: Scales + Coordinate System

```
1 gapminder %>%
2   filter(year %in% c(1987, 2007)) %>%
3   ggplot(aes(y = lifeExp, x = factor(year))) +
4   geom_line(aes(group = country), alpha = .3) +
5   geom_point(aes(fill = gdpPercap), shape = 21) +
6   facet_wrap(~fct_reorder(continent, lifeExp), nrow = 1) +
7   coord_cartesian(expand = FALSE, clip = "off") +
8   scale_y_continuous(
9     breaks = 7:17*5, limits = c(34, 85),
10    sec.axis = dup_axis(name = NULL)
11  ) +
12  scale_fill_distiller(
13    palette = "Spectral", direction = 1,
14    labels = scales::label_dollar()
15  ) -> g_slope
```



Your Turn: Scales + Coordinate System

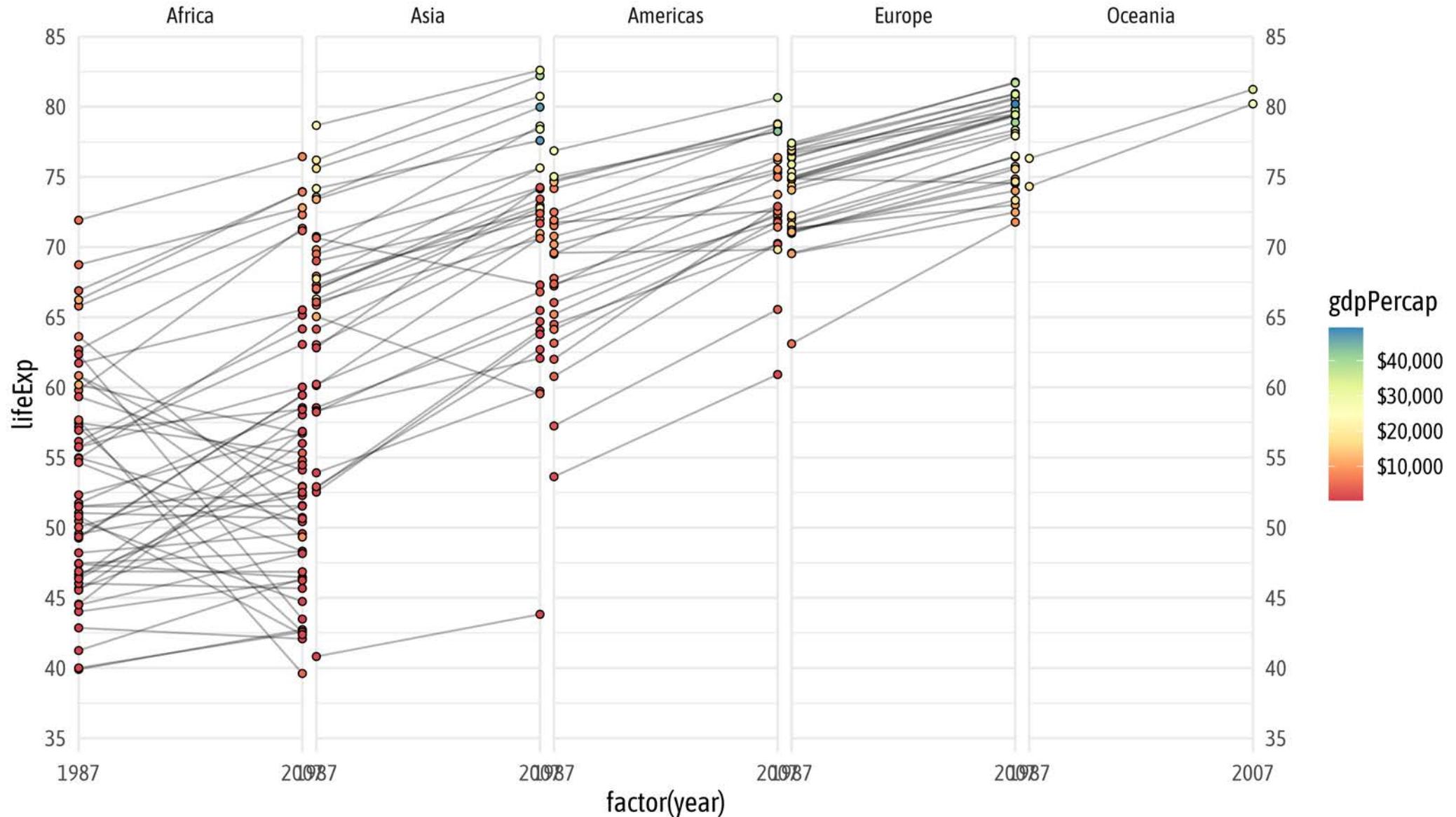


Your Turn: Theming

```
1 g_slope +  
2   theme_minimal(base_family = "Asap Condensed", base_size = 14)
```



Your Turn: Theming



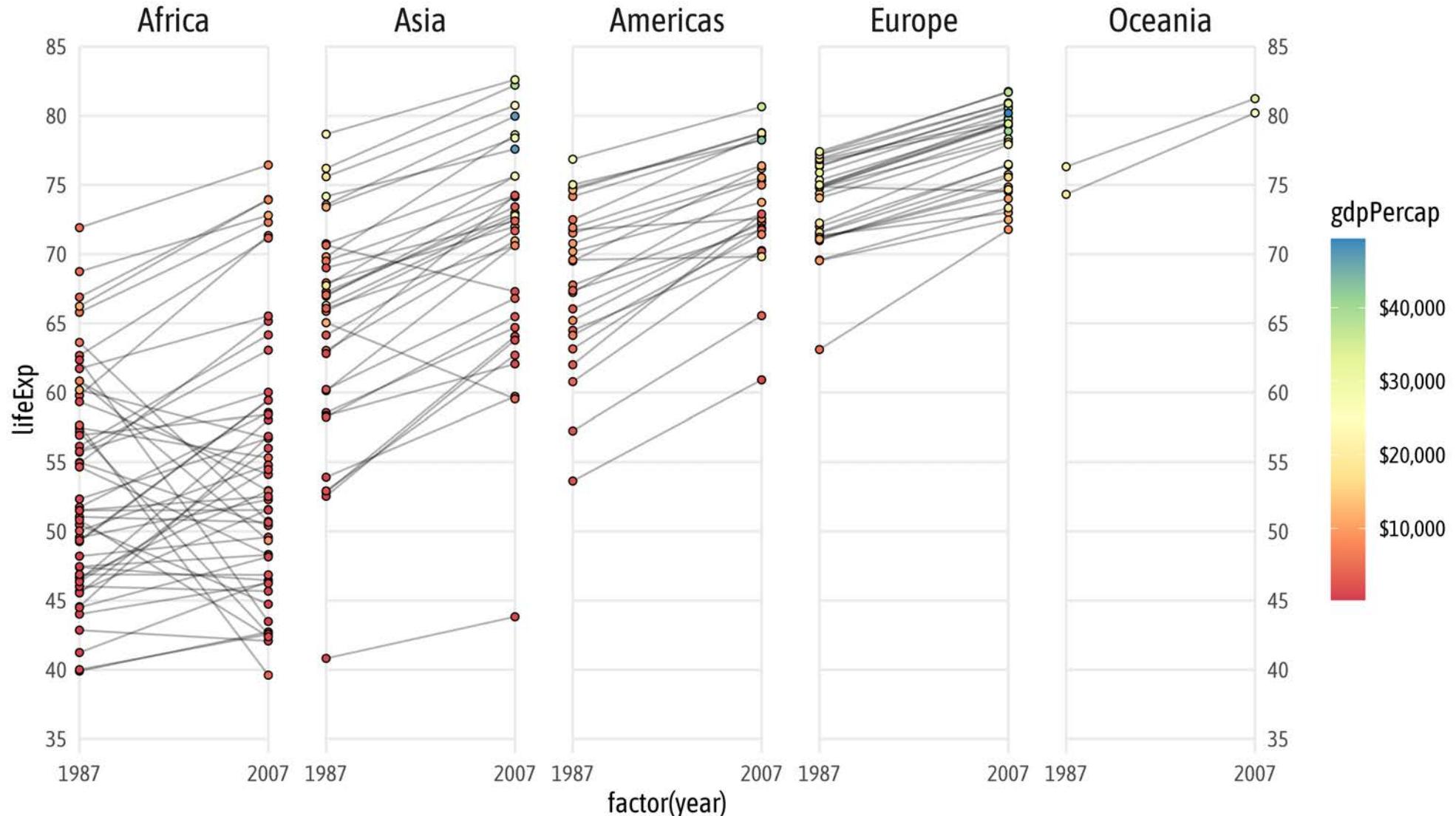
Your Turn: Theming

```
1 g_slope +  
2   theme_minimal(base_family = "Asap Condensed", base_size = 14) +  
3   theme(  
4     panel.grid.minor = element_blank(),  
5     plot.title = element_text(face = "bold", size = 24),  
6     plot.subtitle = element_text(color = "grey35"),  
7     plot.title.position = "plot",  
8     plot.caption.position = "plot",  
9     strip.text = element_text(size = 17),  
10    panel.spacing = unit(2, "lines"),  
11    legend.key.height = unit(2.5, "lines")  
12  ) -> g_slope_themed
```



Your Turn: Theming

1 g_slope_themed

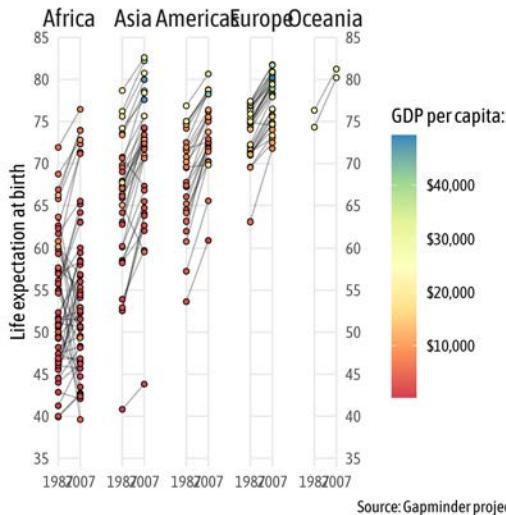


Your Turn: Labels + Titles

```
1 g_slope_themed +
2   labs(
3     x = NULL,
4     y = "Life expectation at birth",
5     fill = "GDP per capita:",
6     title = "Life expectancy has risen in most countries within the last 20 years",
7     subtitle = "However, several African countries, especially in the South, such as Zimbabwe, Eswa
8     caption = "Source: Gapminder project"
9   )
```

Life expectancy has risen in most countries

However, several African countries, especially in the South, such as Zimbabwe, Eswatini, and Malawi, have experienced a considerable drop. Away from Africa, only Iraq and DR Korea (Asia), and a few countries in Central America (Latin America) show a slight decrease.



Your Turn: Labels + Titles



Wrap-up



The Components of {ggplot2}

Component	Function	Explanation
Data	<code>ggplot(data)</code>	<i>The raw data that you want to visualise.</i>
Aesthetics	<code>aes()</code>	<i>Aesthetic mappings between variables and visual properties.</i>
Geometries	<code>geom_*</code> ()	<i>The geometric shapes representing the data.</i>
Statistics	<code>stat_*</code> ()	<i>The statistical transformations applied to the data.</i>
Scales	<code>scale_*</code> ()	<i>Maps between the data and the aesthetic dimensions.</i>
Coordinate System	<code>coord_*</code> ()	<i>Maps data into the plane of the data rectangle.</i>
Facets	<code>facet_*</code> ()	<i>The arrangement of the data into a grid of plots.</i>
Visual Themes	<code>theme()</code> and <code>theme_*</code> ()	<i>The overall visual defaults of a plot.</i>



Resources

- “ggplot2: Elegant Graphics for Data Analysis”
by Hadley Wickham, Danielle Navarro, and Thomas Lin Pedersen.
- “Data Visualization: A Practical Introduction” by Kieran Healy
- “R Graphics Cookbook” by Winston Chang
- ggplot2 Extension Gallery
- R Graph Gallery
- Official ggplot2 reference
- Official ggplot2 cheatsheet
- aesthetics finder
- My “A {ggplot2} Tutorial for Beautiful Plotting in R” tutorial



That's it Folks...
Thank you!

You want more? Join the follow-up course!

“Advanced Data Visualization with {ggplot2}”



