Getting Familiar with `ggplot2`

Rei Sanchez-Arias, Ph.D.

Introducing the `ggplot2` package

Grammar of Graphics with `ggplot2`

Data Visualization

We will learn how to visualize data using the `ggplot2` package: one of the most elegant and most versatile systems for creating graphs.

`ggplot2` implements the **grammar of graphics**, a coherent system for describing and building graphs.

> - Do cars with big engines use more fuel than cars with small engines?
> - What does the relationship between engine size and fuel efficiency look like?
> - Is it positive? Negative? Linear? Nonlinear?

`mpg` dataset

A **data frame** is a rectangular collection of variables (in the columns) and observations (in the rows). `mpg` contains observations collected by the US Environment Protection Agency on 38 models of car.

```
library(tidyverse)
```

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
1	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
2	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
3	audi	a4	2	2008	4	manual(m6)	f	20	31	p	compact
4	audi	a4	2	2008	4	auto(av)	f	21	30	p	compact

Previous 1 2 3 4 5 ... 59 Next

Relationship of `displ` and `hwy`

Among the variables in `mpg` are:

- `displ`, a car's engine size, in liters.
- `hwy`, a car's fuel efficiency on the highway, in miles per gallon (mpg).

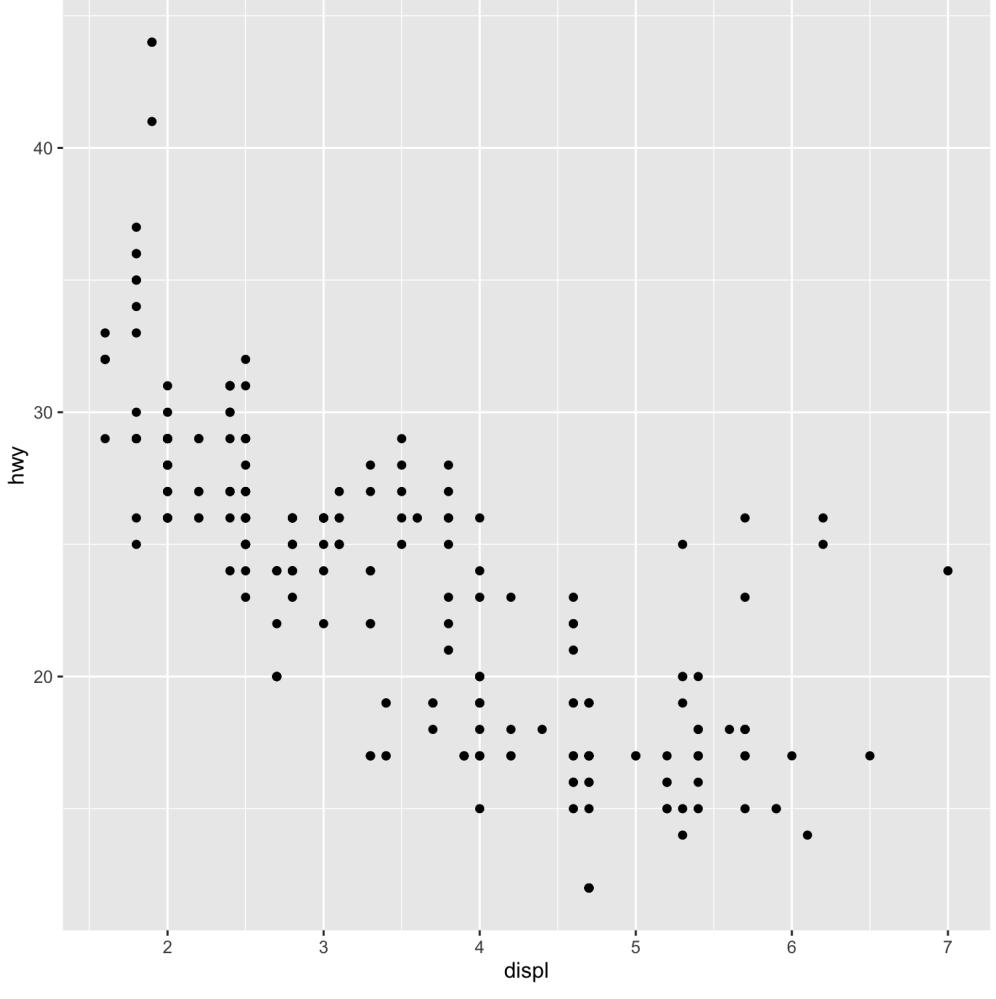
A car with a low fuel efficiency consumes more fuel than a car with a high fuel efficiency when they travel the same distance.

A first plot

A car with a low fuel efficiency consumes more fuel than a car with a high fuel efficiency when they travel the same distance.

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ,
                           y = hwy))
```

The plot shows a *negative relationship* between engine size (`displ`) and fuel efficiency (`hwy`). In other words, cars with big engines use more fuel.



Deconstructing the plot

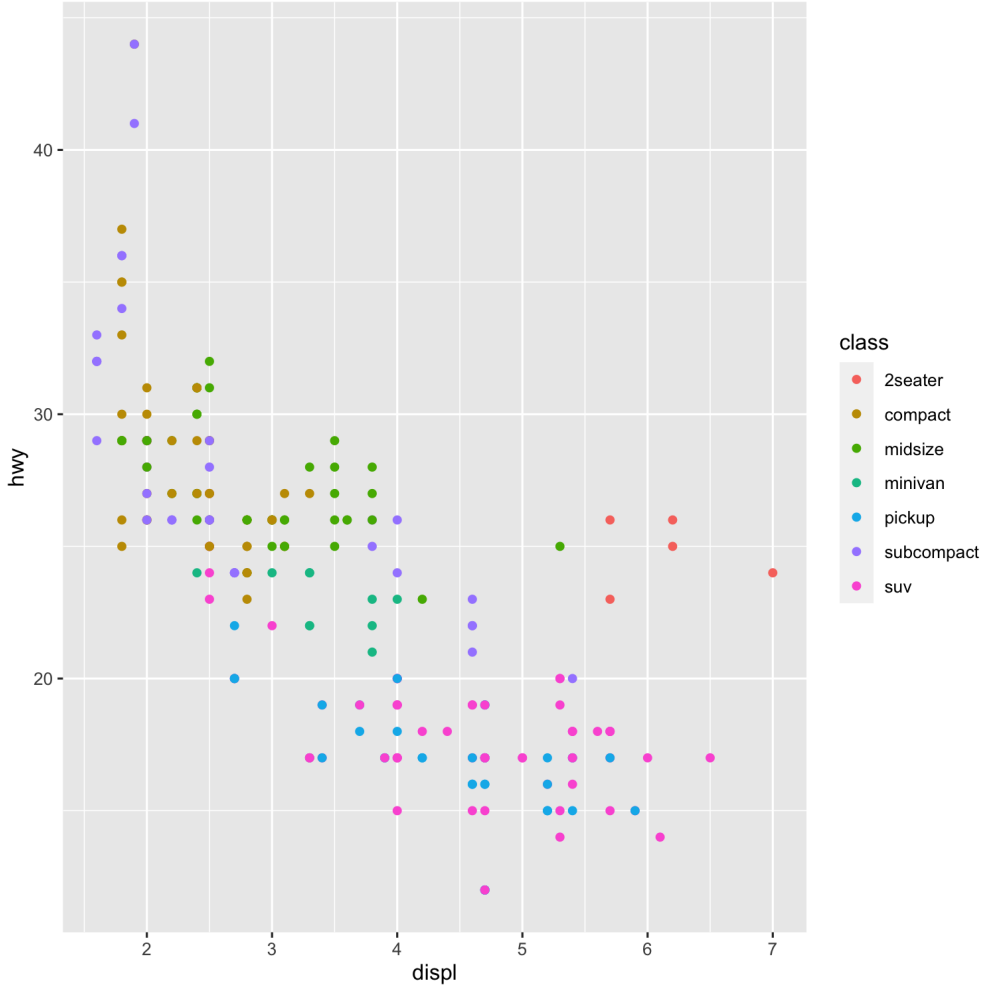
With `ggplot2`, you begin a plot with the function `ggplot()`.

- `ggplot()` creates a *coordinate system* that you can add *layers* to.
- The first argument of `ggplot()` is the *dataset* to use in the graph. So `ggplot(data = mpg)` creates an empty graph.
- You complete your graph by adding one or more *layers* to `ggplot()`. The function `geom_point()` adds a layer of points to your plot, which creates a **scatterplot**.

Mapping data to aesthetics

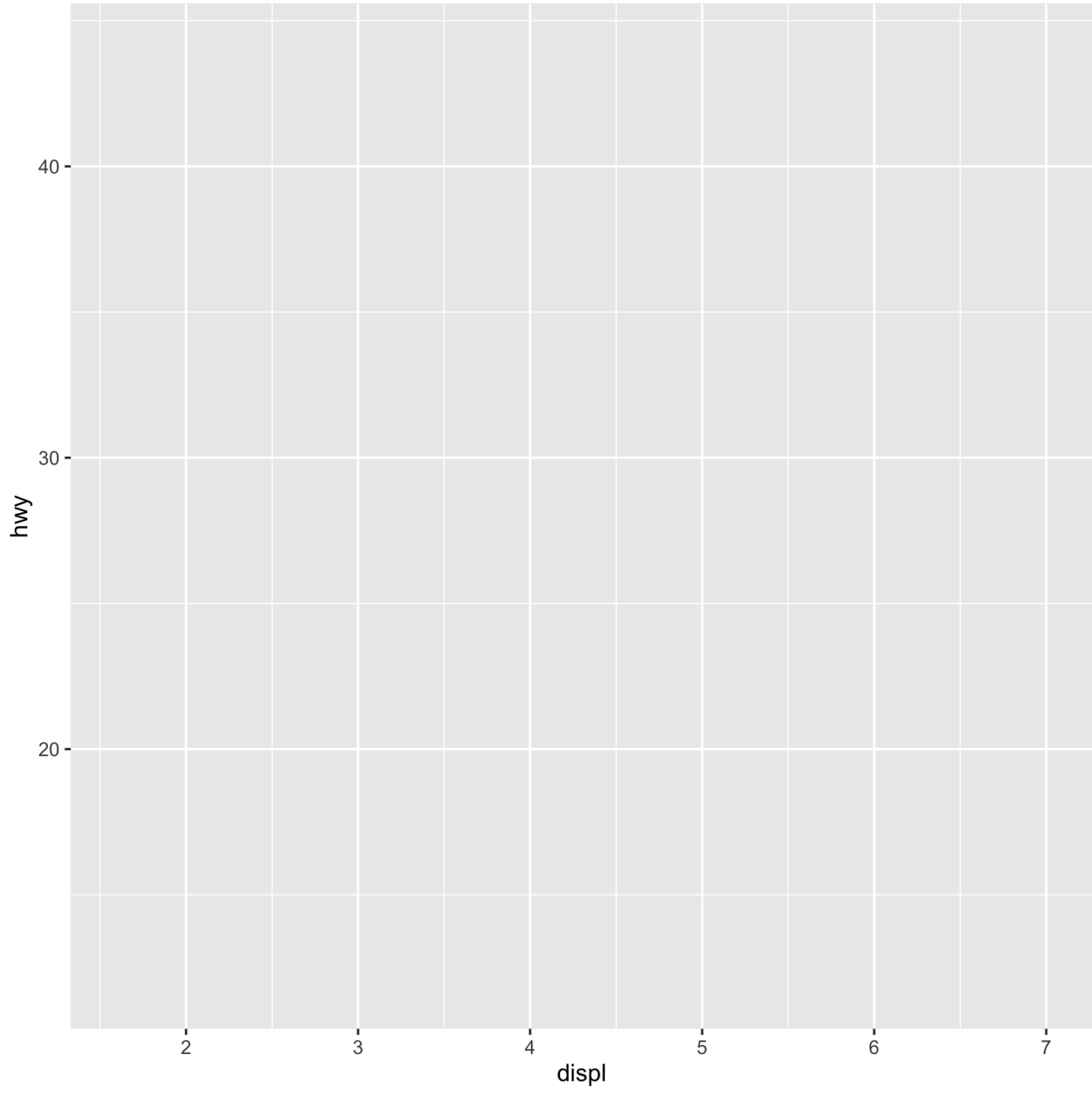
> An **aesthetic** is a visual property of the objects in your plot. Aesthetics include things like the size, the shape, or the color of your points. You can display a point in different ways by changing the values of its aesthetic properties.

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ,
                           y = hwy,
                           color = class))
```



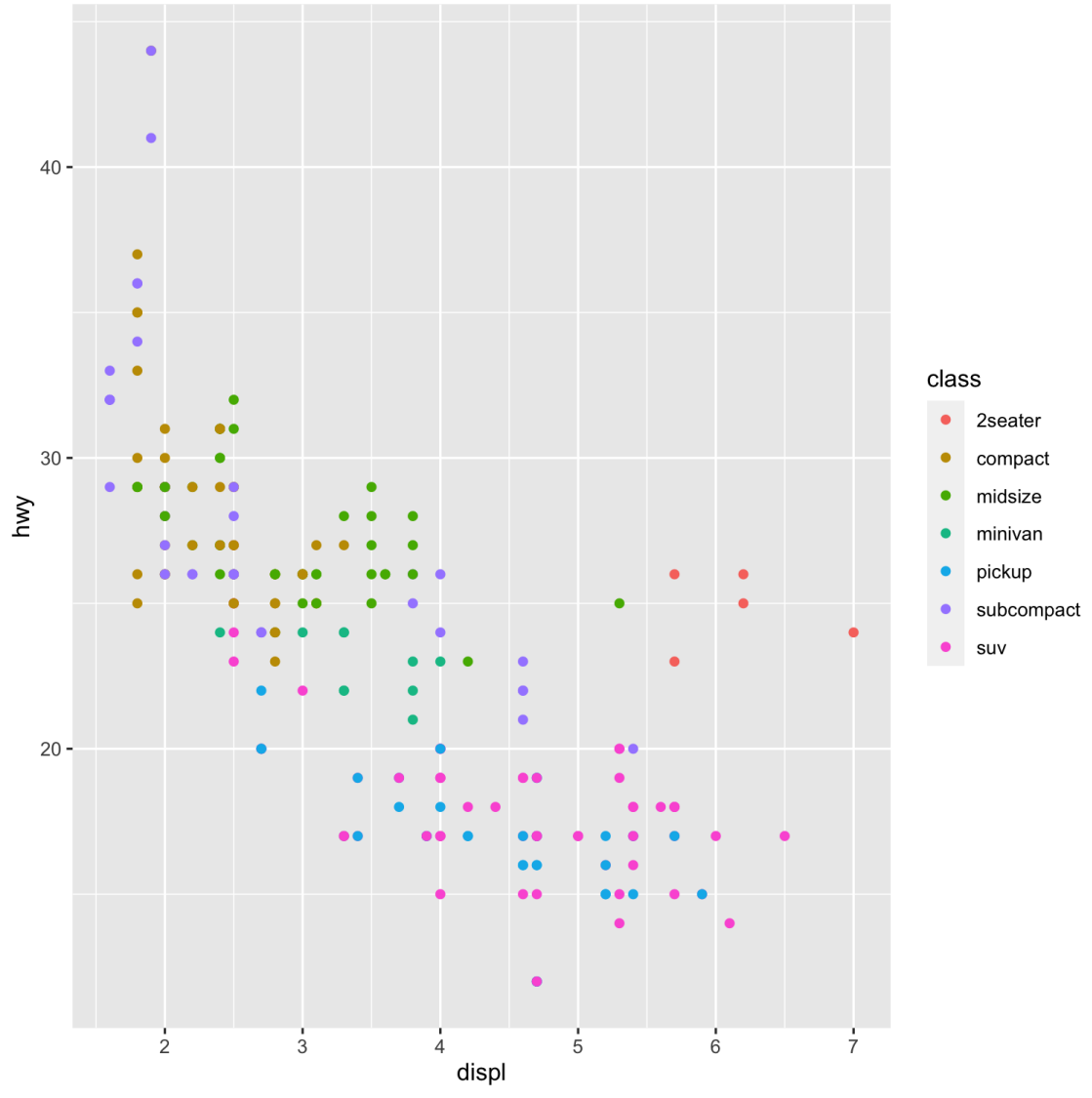
Start with data and aesthetics

```
ggplot(data = mpg,
       mapping = aes(x = displ,
                     y = hwy,
                     color = drv))
```



Add a point geom

```
ggplot(data = mpg,
       mapping = aes(x = displ,
                     y = hwy,
                     color = class)) +
  geom_point()
```



Grammar of Graphics

`ggplot2` implements the **grammar of graphics**, a coherent system for describing and building graphs. The code you write specifies the **connections** between the variables in your data, and the colors, points, and shapes you see on the screen.

- In `ggplot2`, these logical connections between your data and the plot elements are called *aesthetic mappings* or just **aesthetics**.
- You begin every plot by telling the `ggplot()` function what your data is, and then how the variables in this data *logically map* onto the plot's aesthetics.

geoms

Deciding what sort of plot we want, such as a scatterplot, a boxplot, or a bar chart, in `ggplot`, we call the overall type of plot a `geom`. Each `geom` has a function that creates it.

- `geom_point()` makes scatterplots
- `geom_bar()` makes barplots
- `geom_boxplot()` makes boxplots

Combine these two pieces, the `ggplot()` object and the `geom`, by literally *adding them together* in an expression, using the `"+"` symbol.

Example: diamonds dataset

`diamonds` dataset

Let us explore a dataset called `diamonds` included in the `tidyverse`. (More than 50 thousand records are available)

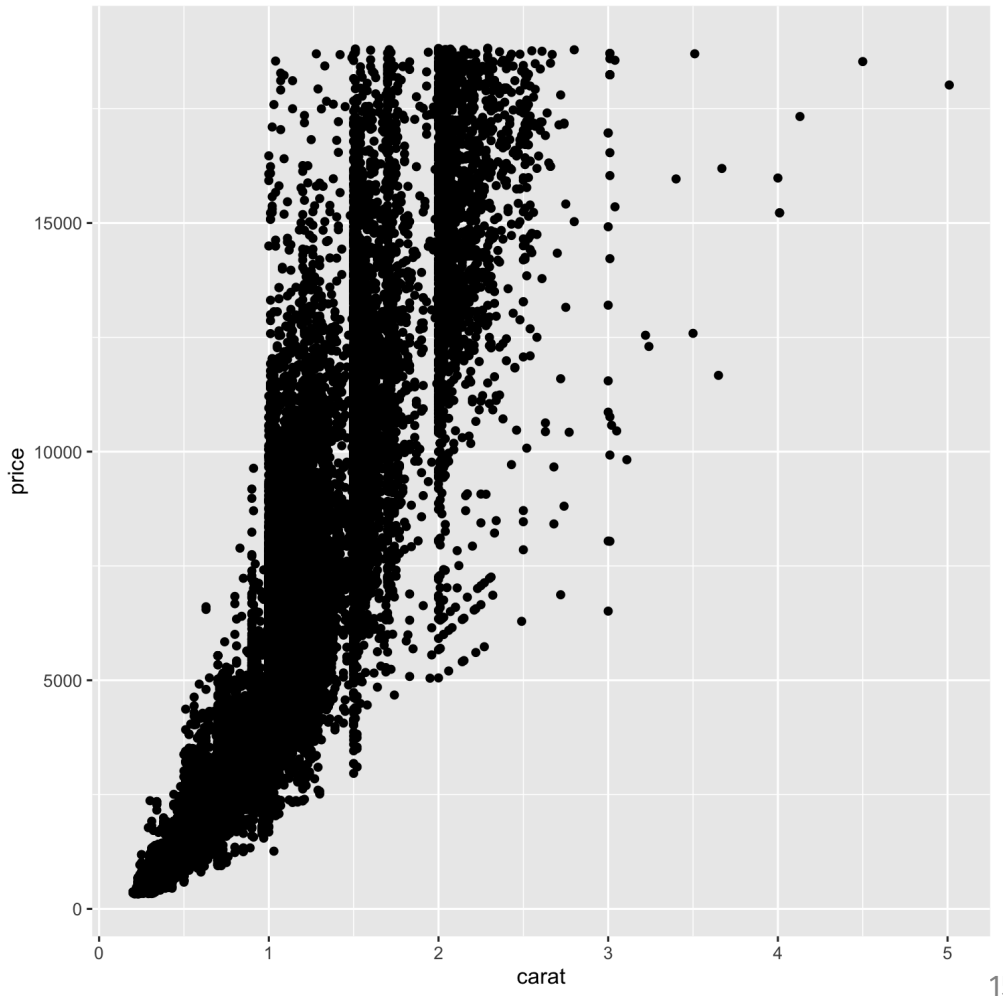
	carat	cut	color	clarity	depth	table	price	x	y	z
1	1.02	Very Good	F	SI1	62.2	56	5222	6.39	6.43	3.99
2	2.53	Premium	H	SI2	61.1	57	15148	8.82	8.76	5.37
3	0.54	Premium	D	SI1	60.2	62	1720	5.31	5.25	3.18
4	0.4	Premium	D	SI1	61.7	58	772	4.7	4.74	2.91
5	0.32	Premium	H	VS1	62.3	58	720	4.39	4.34	2.72
6	0.82	Premium	E	VS2	59.7	61	3697	6.08	6.04	3.62

Previous 1 2 3 4 5 ... 50 Next

Price vs carat relationship

```
ggplot(data = diamonds) +
  geom_point(aes(x = carat, y = price))
```

- `ggplot()` – function that creates the basic `ggplot` graph
- `data` - the data frame that you want to use
- `aes()` – short for *aesthetic*, describes how your variables are graphed
- `geom` – the functions that tell `ggplot` how you want the data presented (scatter plot, histogram, etc.)



Recap

The graph shows how diamond price changes due to carat.

Let's deconstruct this command:

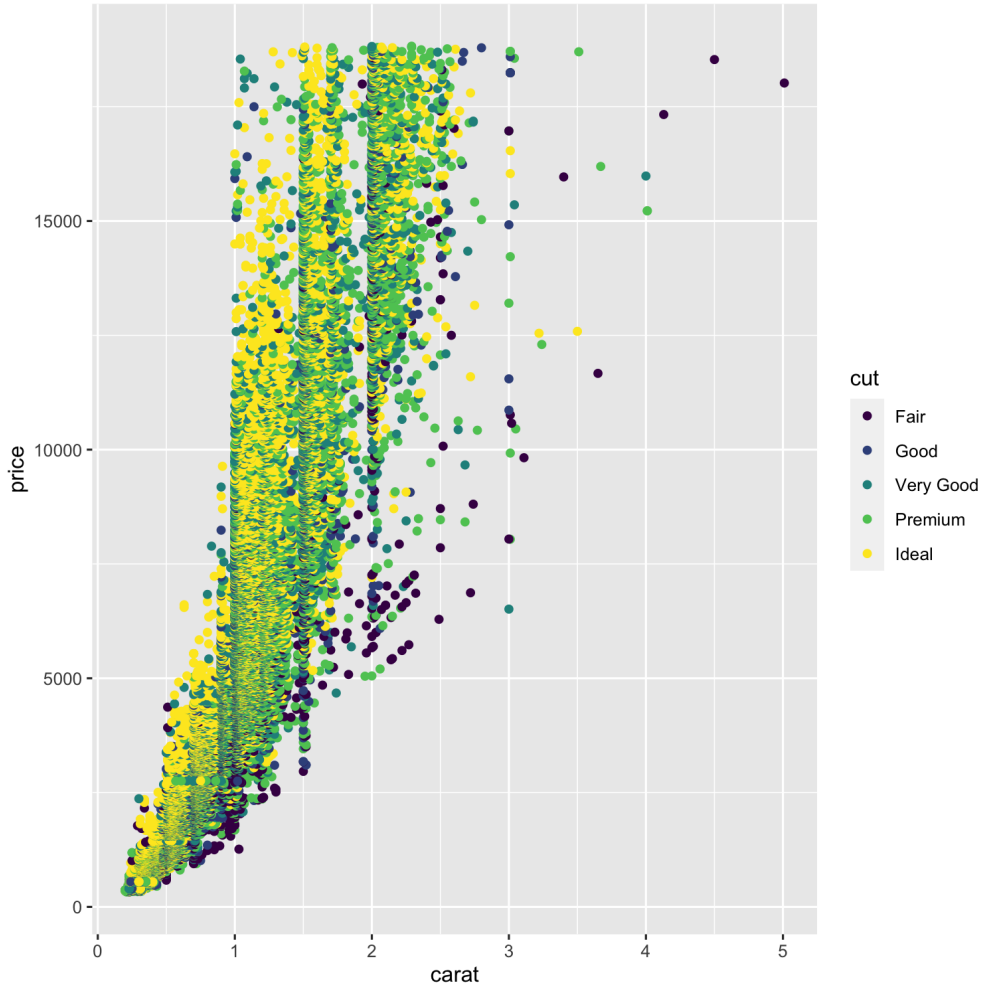
```
ggplot(data = diamonds) +
  geom_point(aes(x = carat, y = price))
```

- `ggplot()` – function that creates the basic `ggplot` graph
- `data` - the data frame that you want to use
- `aes()` – short for *aesthetic*, describes how your variables are graphed
- `geom` – the functions that tell `ggplot` how you want the data presented (scatter plot, histogram, etc.)

Adding aesthetics

```
ggplot(data = diamonds) +
  geom_point(aes(x = carat,
                 y = price,
                 color = cut))
```

> One *common problem* when creating `ggplot2` graphics is to put the `+` in the *wrong place*: it has to come at the end of the line, *not* the start.



Example: gapminder dataset

`gapminder` dataset

The `gapminder` package includes the `gapminder` dataframe, data on life expectancy, GDP per capita, and population by country.

```
library(tidyverse)
library(gapminder)
```

> Source: http://www.gapminder.org/data/

Quick look

Excerpt of the Gapminder data on life expectancy, GDP per capita, and population by country.

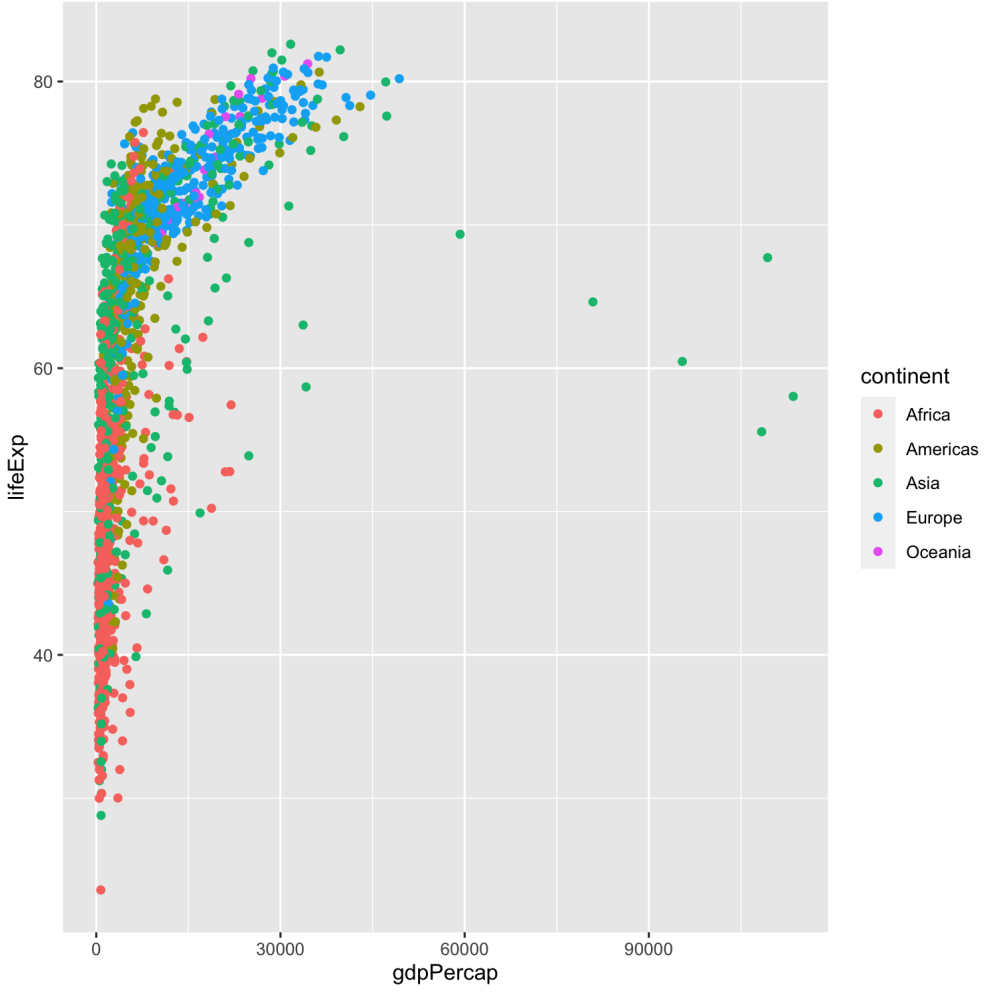
	country	continent	year	lifeExp	pop	gdpPercap
1	Afghanistan	Asia	1952	28.801	8425333	779.4453145
2	Afghanistan	Asia	1957	30.332	9240934	820.8530296
3	Afghanistan	Asia	1962	31.997	10267083	853.10071
4	Afghanistan	Asia	1967	34.02	11537966	836.1971382
5	Afghanistan	Asia	1972	36.088	13079460	739.9811058
6	Afghanistan	Asia	1977	38.438	14880372	786.11336

Previous 1 2 3 4 5 ... 284 Next

Mapping aesthetics vs setting them

Recall that an *aesthetic mapping* specifies that a variable will be expressed by one of the available *visual elements*, such as size, or color, or shape, and so on. As we've seen, we map variables to aesthetics like this:

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

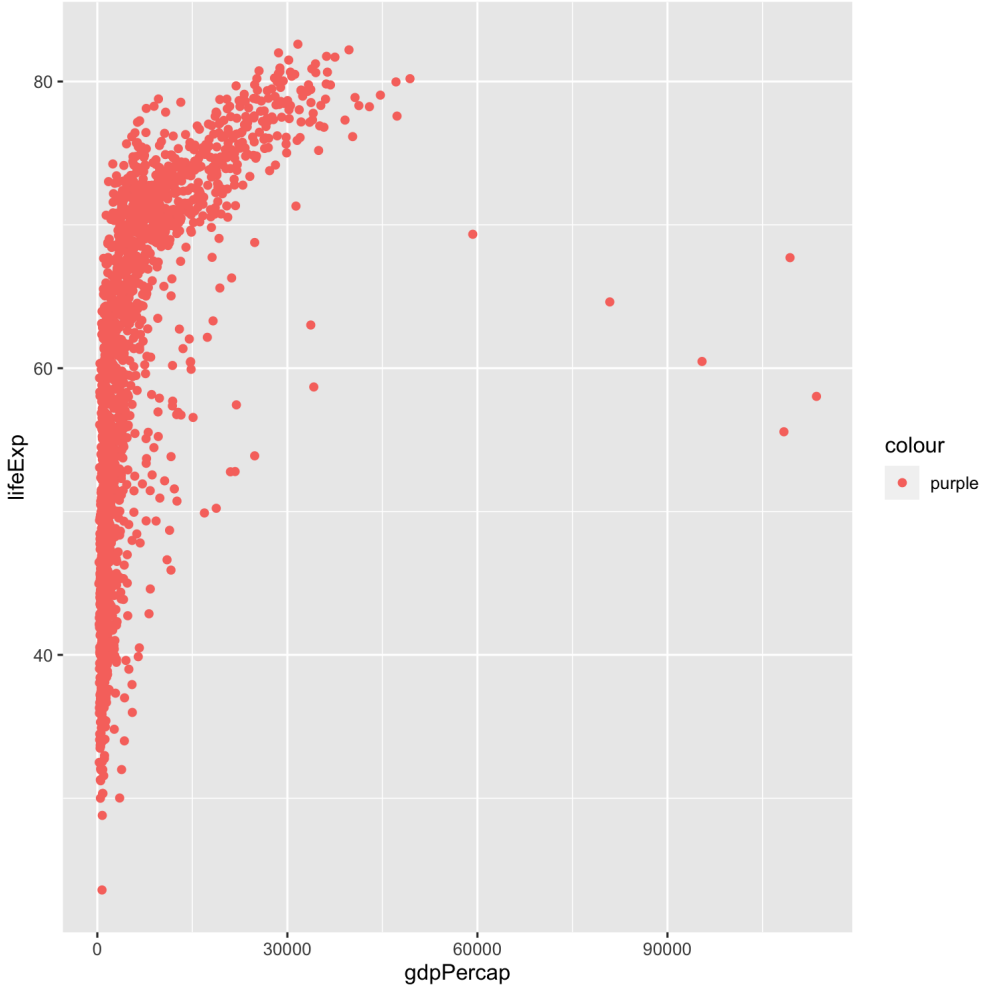


Color all points purple?

The code before *does not* give a direct instruction like "color the points purple". Instead it says, "the property `color` will represent the variable `continent`", or "`color` will map `continent`".

If we want to turn all the points in the figure purple, *we do not do it through the mapping function*. Look at what happens when we try:

```
ggplot(data = gapminder) +
  geom_point(mapping = aes(x = gdpPercap,
                           y = lifeExp,
                           color = "purple"))
```



What went wrong? 😩

What has happened here? Why is there a legend saying "purple"? The `aes()` function is for mappings only.

Do not use it to change properties to a particular value. If we want to *set a property*, we do it in the `geom_` we are using, and outside the `mapping = aes(...)` step.

```
ggplot(data = gapminder) +
  geom_point(mapping = aes(x = gdpPercap, y
             color    = "purple"
            )
```