Ch 3. Data Visualization

3.1 Basic Stuff

ggplot2 → R's best way of making graphs

library(tidyverse) : must reload every new session

3.2 Making Graphs with ggplot2

example. mpg data frame, 38 models of cars

engine size (engine displacement) and fuel efficiency (highway mpg) relationship graph
plotting the dataset:

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

`ggplot(data = mpg)` : just empty graph spawn
`geom_point()` : adds a layer of points to your plot, which creates a scatterplot.
`aes()` x and y arguments : specify x and y axes variables.

Making graphs with ggplot2:

```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

Complete the template below to build a graph.

```
ggplot (data = <DATA>) +                                   ] required
<GEOM_FUNCTION>(mapping = aes(<MAPPINGS>),
 stat = <STAT>, position = <POSITION>) +                   ] Not required, sensible defaults supplied
<COORDINATE_FUNCTION> +
<FACET_FUNCTION> +
<SCALE_FUNCTION> +
<THEME_FUNCTION>
```

ggplot(data = mpg, **aes(**x = cty, y = hwy**))** Begins a plot that you finish by adding layers to. Add one geom function per layer.

3.3 Aesthetic

can add a "third variable" by mapping it to an *aesthetic*

aesthetic: visual properties of my graph

can help convey separation of the scattered dots into the given category

map the colors of your points to the class variable to reveal the class of each car.

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color = class))
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, size = class))
#> Warning: Using size for a discrete variable is not advised.
```

transparency of the points controlled

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

shape of the points controlled

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

When you want to set an aesthetic manually = put it outside the aes() function and within the geom function!

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

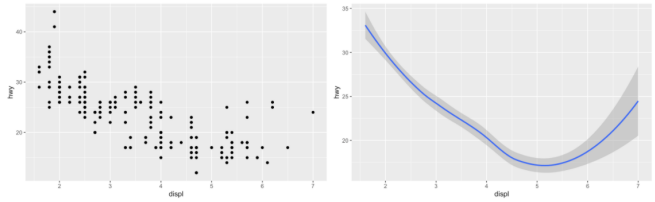
3.5 Facets

facets : subplots that each display one subset of the data.

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_wrap(~ class, nrow = 2)
```

3.6 Geoms

"uses different geoms" = uses different visual object to represent the data



left

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

right

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

3.7

bar chart : geom_bar()

Ch 4. Basic Coding Stuff

create new objects : **object name ← value**
this_is_a_long_name <- 2.5

function_name(arg1 = val1, arg2 = val2, ...)

```
seq(1, 10)
#>  [1]  1  2  3  4  5  6  7  8  9 10

y <- seq(1, 10, length.out = 5)
y
#> [1]  1.00  3.25  5.50  7.75 10.00
```

Ch 5. Data transformation

5 key dplyr functions:

filter( ) : select the specific variable
example) filter flights for specifically January 1st by filtering for day = 1 and month = 1

```
filter(flights, month == 1, day == 1)
```

want to save this data :

```
jan1 <- filter(flights, month == 1, day == 1)
```

Computers use finite precision arithmetic (they obviously can't store an infinite number of digits!) so remember that every number you see is an approximation. Instead of relying on ==, use near():

```
near(sqrt(2) ^ 2,  2)
```

arrange( ) :

select( ) :

mutate( ) :

summarise( ) :