Statistical Computing with R: Masters in Data Science 503 (S17) Fourth Batch, SMS, TU, 2025

Shital Bhandary

Associate Professor

Statistics/Bio-statistics, Demography and Public Health Informatics

Patan Academy of Health Sciences, Lalitpur, Nepal

Faculty, Masters in Medical Research, NHRC/Kathmandu University

Faculty, FAIMER Fellowship in Health Professions Education, India/USA

R for Data Science: Course Book

- Chapter 3: Data Visualization
- With “ggplot2” package
- https://r4ds.had.co.nz/data-visualisation.html
- **Last class of Unit 3**

Aesthetic mapping

- **“The greatest value of a picture is when it forces us to notice what we never expected to see.”** — John Tukey (Tukey’s boxplot!)

```
library(ggplot2)
```

- What will happen?

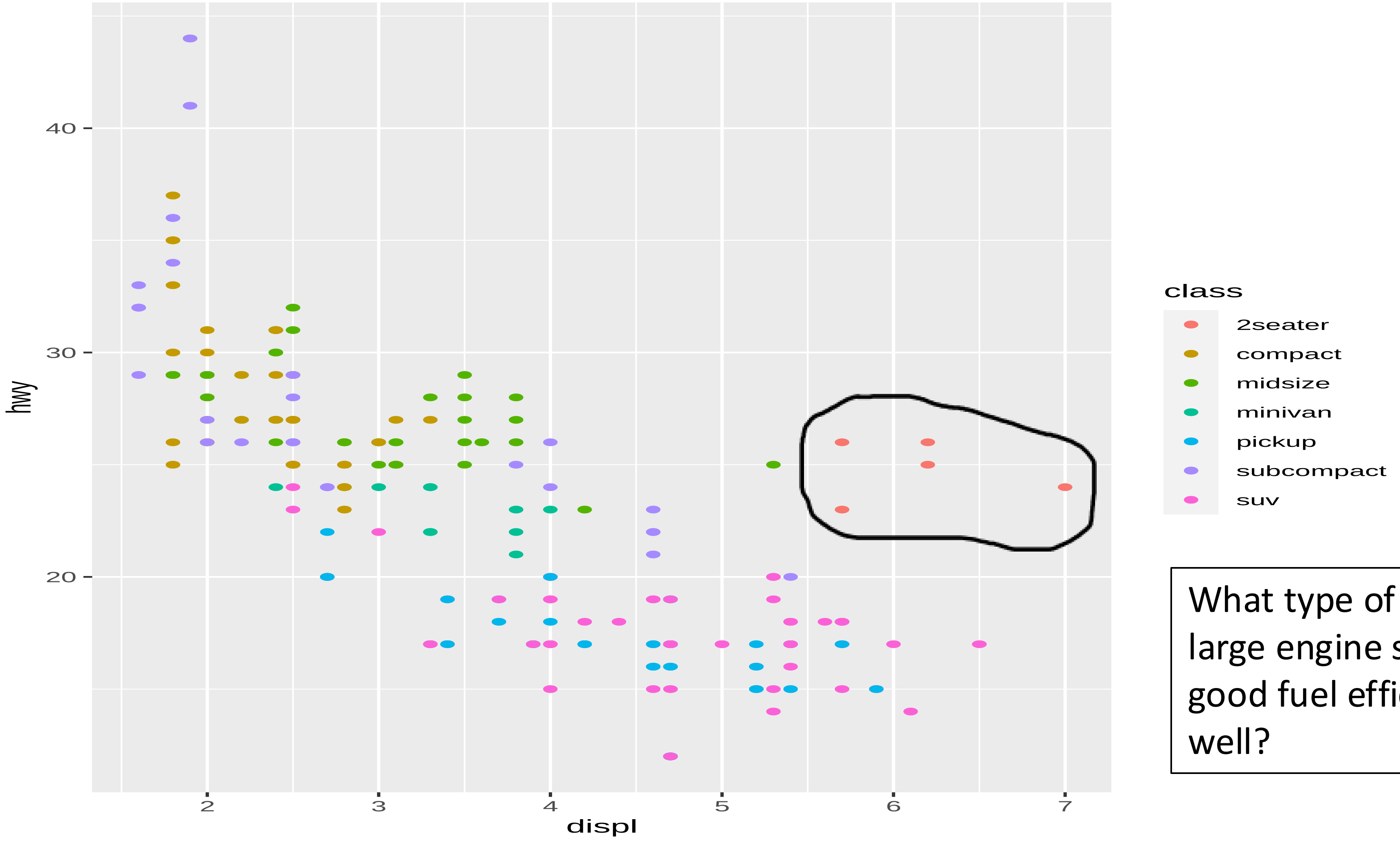
```
ggplot(data = mpg) +
geom_point(mapping = aes(x = displ, y = hwy, color = class))
```

What will happen?

```
ggplot(mpg, aes(displ, hwy, color = class)) +
geom_point()
```

Graph: Is hypothesis still holds true?

HT: Big engine (disp) < highway fuel efficiency (hwy)



What type of cars with large engine size have good fuel efficiency as well?

Interpretation?

- The colors reveal that many of the unusual points are two-seater cars. These cars don’t seem like hybrids, and are, in fact, **sports cars**!
- Sports cars have large engines like SUVs and pickup trucks, but small bodies like midsize and compact cars, which improves their gas mileage.
- In hindsight, these cars were unlikely to be hybrids since they have large engines.

Note:

- In the above example, we mapped class to the color aesthetic, but we could have mapped class to the size aesthetic in the same way. In this case, the exact size of each point would reveal its class affiliation.

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

Note: We get a *warning* here, because mapping an unordered variable (class) to an ordered aesthetic (size) is not a good idea.

What will happen?

Try and check the notes given in red fonts!

```
# First
ggplot(data = mpg) +
geom_point(mapping = aes(x = displ, y = hwy, alpha = class))
#alpha – transparency of the points

# Right
ggplot(data = mpg) +
geom_point(mapping = aes(x = displ, y = hwy, shape = class))
#All the class have shapes?
```

What will happen?

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

Did you get the “blue” point plot?

How to interpret this plot?

Better than the previous one?

What is the problem?

#Why not shown in “blue”?

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

#Why no plot and error?

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

Facets: Face wrap

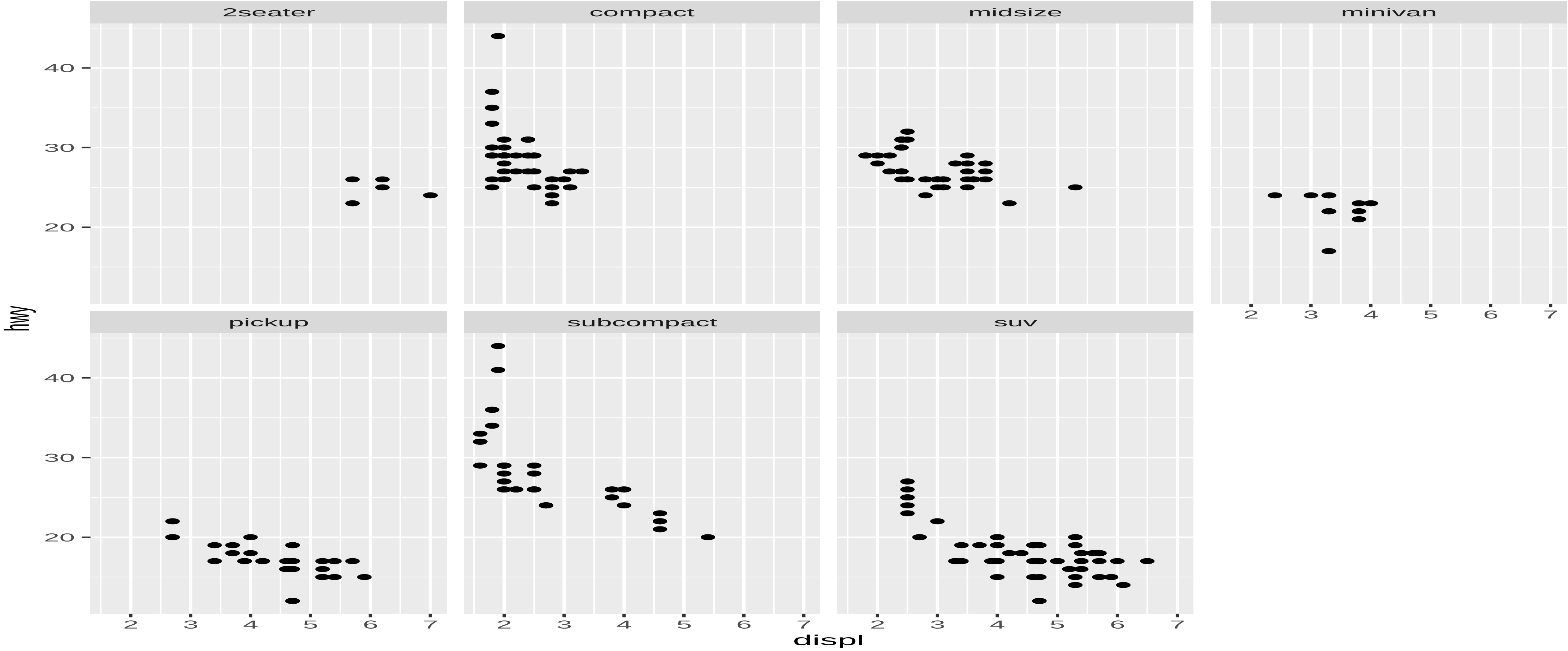
- One way to add additional variables is with aesthetics. Another way, particularly useful for categorical variables, is to split your plot into **facets**, subplots that each display one subset of the data.

- To facet your plot by a single variable, use facet_wrap(). The first argument of facet_wrap() should be a formula, which you create with ~ followed by a variable name (here “formula” is the name of a data structure in R, not a synonym for “equation”). The variable that you pass to facet_wrap() should be discrete.

What will happen?

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

How to interpret?



Facets: facet_grid:

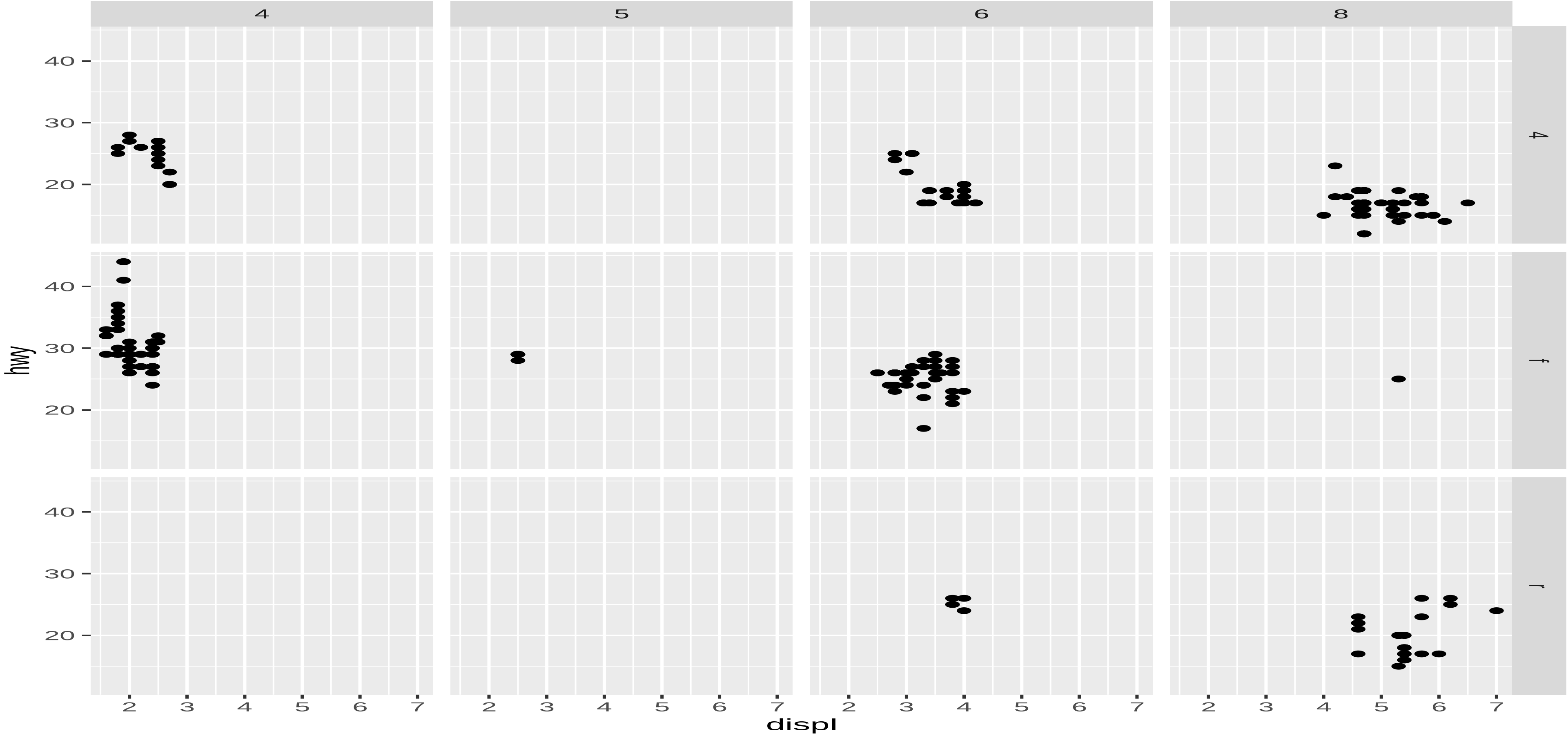
- To facet your plot on the combination of two variables, add facet_grid() to your plot call.

- The first argument of facet_grid() is also a formula. This time the formula should contain two variable names separated by a ~.

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

Interpretation? **Use facet_grid wisely!**

3 x 4 = 12 levels of two variables!



Cyl = 4, 5, 6, 8 cylinders

Drv = 4, f, r drive

4 = 4 wheel
f = front wheel
r = rear wheel

How to display different graphs in a single window in ggplot2 package?

Arranging plots:

- https://ggplot2-book.org/arranging-plots.html

1. Laying out plots side by side
   - Taking control of the layout
   - Modifying subplots
   - Adding annotations

2. Arranging plots on top of each other

What will happen?

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

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

Geometric objects

- A **geom** is the geometrical object that a plot uses to represent data. People often describe plots by the type of geom that the plot uses. For example, bar charts use bar geoms, line charts use line geoms, boxplots use boxplot geoms, and so on. Scatterplots break the trend; they use the point geom.

- To change the geom in your plot, change the geom function that you add to ggplot().

- ggplot2 provides over 40 geoms

What will happen?

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

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

#Interpretation?

What will happen?

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

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

#Which type of smoothing done? Why? Interpretation?

What happens here?

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

```
#Single category smoothing?
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(color = class)) +
  geom_smooth(data = filter(mpg, class == "subcompact"), se = FALSE)
```

Statistical transformations

```
ggplot(data = diamonds) +
stat_count(mapping = aes(x = cut))

ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut))
```

#Do these codes give the same plot? Why?

What other statistical transformations can be used with “ggplot2”?

- https://ggplot2-book.org/statistical-summaries.html

What will happen?

```
ggplot(data = diamonds) +
stat_summary(
mapping = aes(x = cut, y = depth),
fun.min = min,
fun.max = max,
fun = median
)
```

#Interpretation?

# Position adjustments

```
ggplot(data = diamonds) +
geom_bar(mapping = aes(x = cut, colour = cut))

ggplot(data = diamonds) +
geom_bar(mapping = aes(x = cut, fill = cut))
```

#Position adjustment happens here! How?

```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, fill = clarity))
```

What will happen?

```
ggplot(data = diamonds, mapping = aes(x = cut, fill = clarity)) +
geom_bar(alpha = 1/5, position = "identity")
```

```
ggplot(data = diamonds, mapping = aes(x = cut, colour = clarity)) +
geom_bar(fill = NA, position = "identity")
```

#position = "identity" will place each object exactly where it falls in the context of the graph.

What will happen?

```
ggplot(data = diamonds) +
geom_bar(mapping = aes(x = cut, fill = clarity), position = "fill")
```

#position = "dodge" places overlapping objects directly *beside* one another. This makes it easier to compare individual values.

```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, fill = clarity), position = "dodge")
```

Coordinate systems

- Coordinate systems are probably the most complicated part of ggplot2.

- The default coordinate system is the Cartesian coordinate system where the x and y positions act independently to determine the location of each point.

- There are a number of other coordinate systems that are occasionally helpful.

What will happen?

```
ggplot(data = mpg, mapping = aes(x = class, y = hwy)) +
geom_boxplot()
```

```
ggplot(data = mpg, mapping = aes(x = class, y = hwy)) +
geom_boxplot() +
coord_flip()
```

What will happen?
maps package required!

```
nz <- map_data("nz")
ggplot(nz, aes(long, lat, group = group)) +
geom_polygon(fill = "white", colour = "black")

ggplot(nz, aes(long, lat, group = group)) +
geom_polygon(fill = "white", colour = "black") +
coord_quickmap()
```

# **coord_quickmap()** sets the aspect ratio correctly for maps. This is very important if you’re plotting spatial data with ggplot2

More on maps:

- https://ggplot2-book.org/maps.html

What will happen?

```
bar <- ggplot(data = diamonds) +
geom_bar(
mapping = aes(x = cut, fill = cut),
show.legend = FALSE,
width = 1
) +
theme(aspect.ratio = 1) +
labs(x = NULL, y = NULL)

bar + coord_flip()        #Flipped bar diagram
bar + coord_polar()       #With polar coordinat
```

Annotations:

- https://ggplot2-book.org/annotations.html
- Plot and Axis titles
- Text labels
- Building custom annotations
- Direct labelling
- Annotation across facets

Question/queries?

Thank you!

@shitalbhandary