2.1 Data Visualisation

Actuarial Data Science Applications (ACTL4305/5305)

Fei Huang, UNSW

Table of contents

Reading List

- The Art of Data Science, Chapter 4.1, 4.2.
- R for Data Science Online Book, Chapters 2, 3

Learning Objectives of Exploratory Data Analysis

- Understand how to do exploratory data analysis with the tidyverse package in R
- Explore features of data using data visualisation.
- Explain common features of data, such as categorical variables, missing values, unreliable/non-validated data, outliers and high cardinality features, that may lead to problems.
- Apply appropriate methods to deal with common data problems.
- Apply a range of techniques to assess data quality.
- Use R to manipulate (eg filter, merge, sort, group by, summarise, etc.)
- Use R to import, export and tidy data.
- Apply the process to do exploratory data analysis with practical datasets
- Apply R Markdown for communication and reproducible analysis

Exploratory Data Analsis: an Introduction

Exploratory Data Analysis (EDA)

- A set of procedures to produce descriptive and graphical summaries of the data
- Explore the data as they are without making assumptions
- To examine your data and understand relationship among variables
- To determine if there are any problems with your dataset
- To determine whether the question you are asking can be answered by the data that you have
- To develop a sketch of the answer to your question

The process of exploratory data analysis

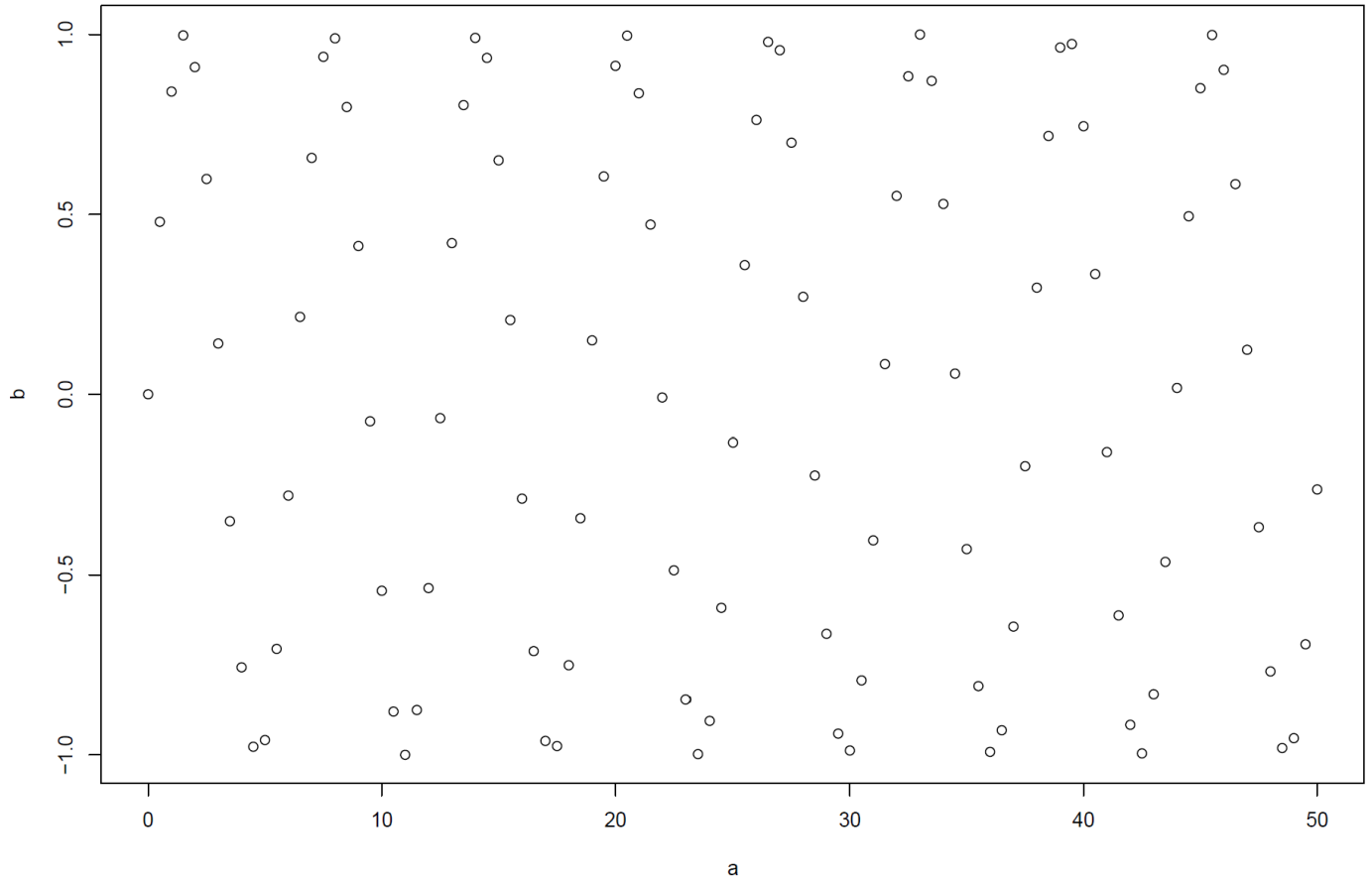
The EDA is an iterative cycle. You:

1. Formulate your question
2. Search for answers by
    1. Collect and Import data
    2. Check data quality and Cleansing data
    3. Manipulate and Transform data
    4. Visualise data
3. Use what you learn to refine your questions and/or generate new questions

- Data Visualization is arguably the most important tool for EDA.

Example

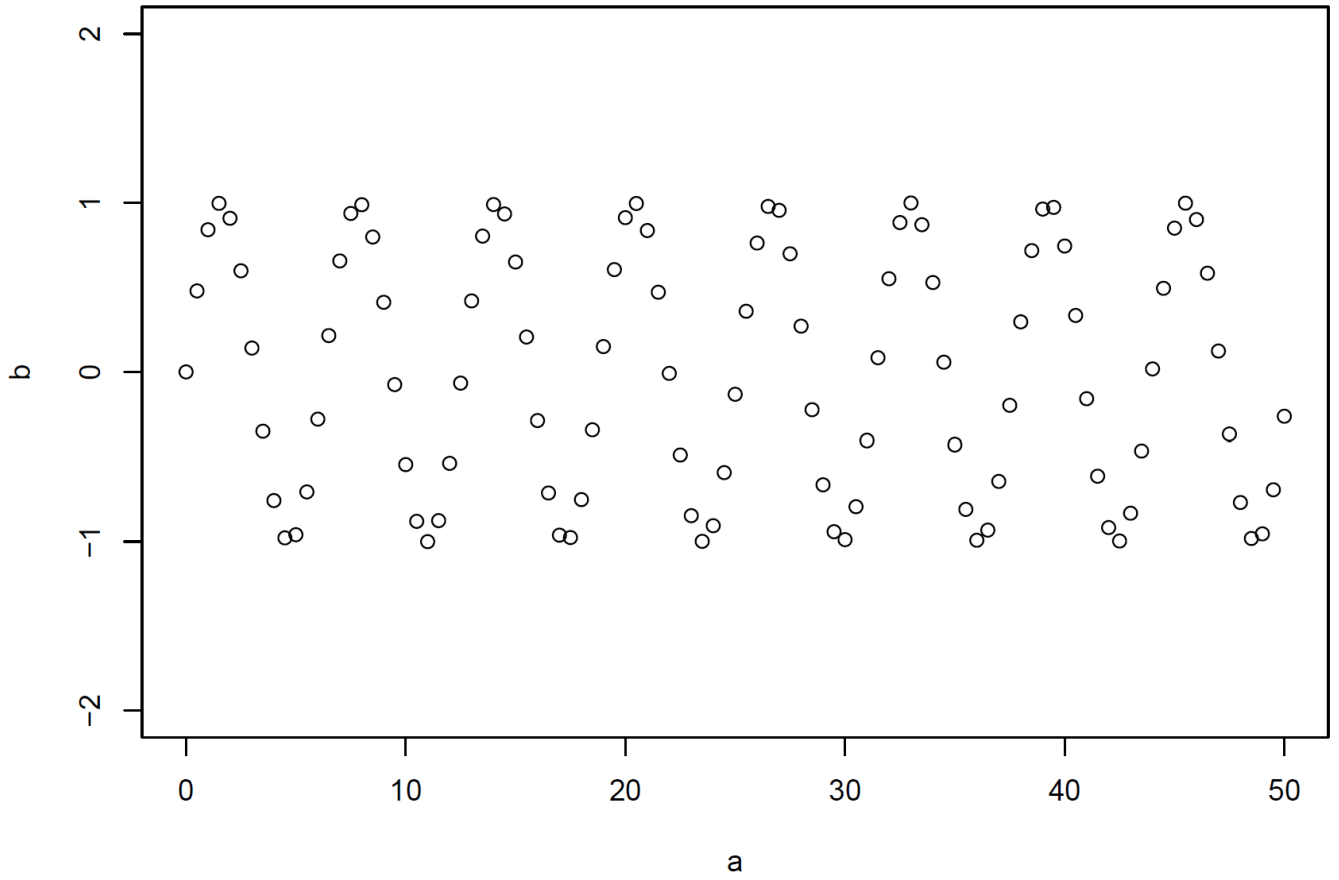
- What pattern can you see from this plot?



Plot 1

Example

- What pattern can you see from this plot?



Plot 2

Data Visualization

> ‘Visually attractive graphics also gather their power from content and interpretations beyond the immediate display of some numbers. The best graphics are about the useful and important, about life and death, about the universe. Beautiful graphics do not traffic with the trivial.’ — Edward Tufte

UNSW SYDNEY

Data Visulization using `ggplot2`

Explore features of data using data visualisation

- A *statistical graphic* maps variables of
  1. a *dataset* to
  2. *aesthetic properties* of
  3. *geometric objects*.
- `ggplot2` is part of `tidyverse`
- Using `ggplot2` to visualise your data
- A ggplot2 grammar guide

UNSW SYDNEY

ggplot2

```
#install.packages("tidyverse")
library(tidyverse)
```

- reload the package everytime you start a new session.
- `package::function()`
  - `ggplot2::ggplot()`

UNSW SYDNEY

First steps

- Question: Do cars with big engines use more fuel than cars with small engines?
- Data: The `mpg` data frame in `ggplot2` (`ggplot2::mpg`)
  - `mpg` contains observations collected by the US Environmental Protection Agency on 38 models of car, 1999-2008.
  - A data frame with 234 rows and 11 variables:
- Variables:
  - `manufacturer`: manufacturer name
  - `model`: model name
  - `displ`: engine displacement, in litres
  - `year`: year of manufacture
  - `cyl`: number of cylinders
  - `trans`: type of transmission
  - `drv`: the type of drive train, where f = front-wheel drive, r = rear wheel drive, 4 = 4wd

The `mpg` data

```
1 mpg
```

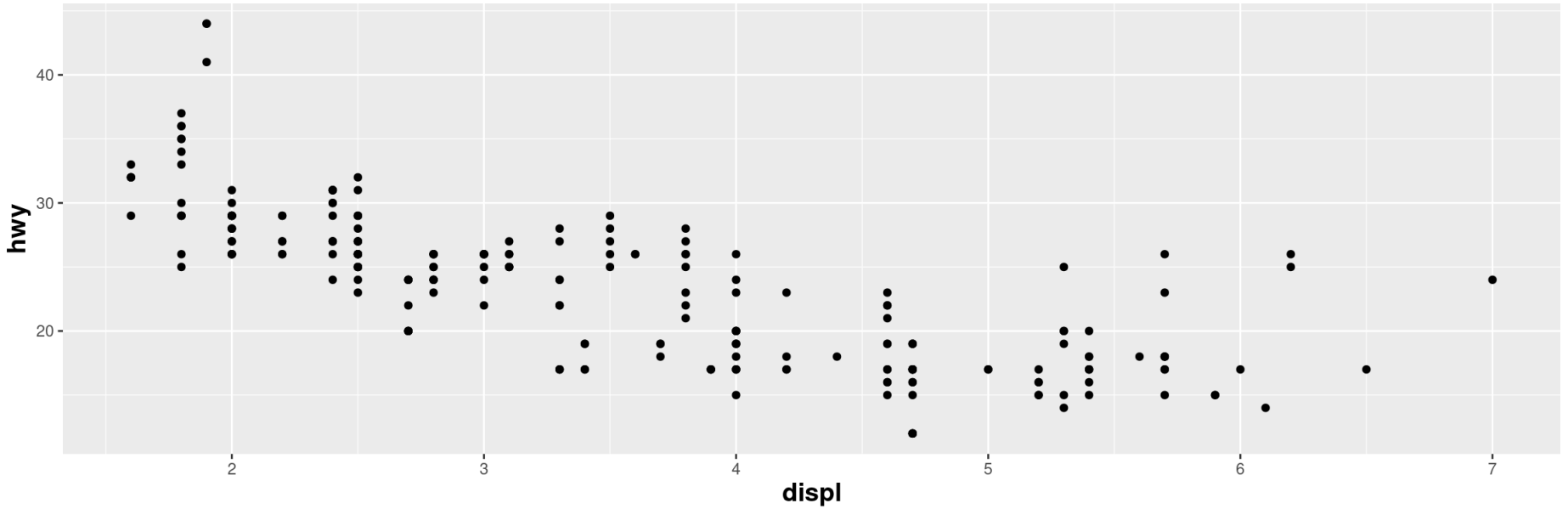
manufacturer <chr>	model <chr>	displ <dbl>	year <int>	cyl <int>
audi	a4	1.8	1999	4
audi	a4	1.8	1999	4
audi	a4	2.0	2008	4
audi	a4	2.0	2008	4
audi	a4	2.8	1999	6
audi	a4	2.8	1999	6
audi	a4	3.1	2008	6
audi	a4 quattro	1.8	1999	4
audi	a4 quattro	1.8	1999	4
audi	a4 quattro	2.0	2008	4

1-10 of 234 rows | 1-5 of 11 columns

Previous 1 2 3 4 5 6 ... 24 Next

Creating a plot

```
ggplot(data = mpg) + # the dataset
  aes(x = displ) + # the x position
  aes(y = hwy) +
  # the y position
  geom_point() + # the point geometric shape
  # Adjust axis titles' front size
  theme(axis.title=element_text(size=14,face="bold"))
```



Declaring Data

Declaring data: method 1

```
1 ggplot(data = mpg)
```

Declaring data: method 2

- Pipe data into `ggplot()` using the pipe operator: `%>%`

```
mpg %>% # data piped into
  ggplot() # initiating plot
```

Exercises

1. Run `ggplot(data = mpg)`. What do you see?
2. How many rows are in `mpg`? How many columns?
3. What does the `drv` variable describe? Read the help for `?mpg` to find out.
4. Make a scatterplot of `hwy` vs `cyl`.
5. What happens if you make a scatterplot of `class` vs `drv`? Why is the plot not useful?

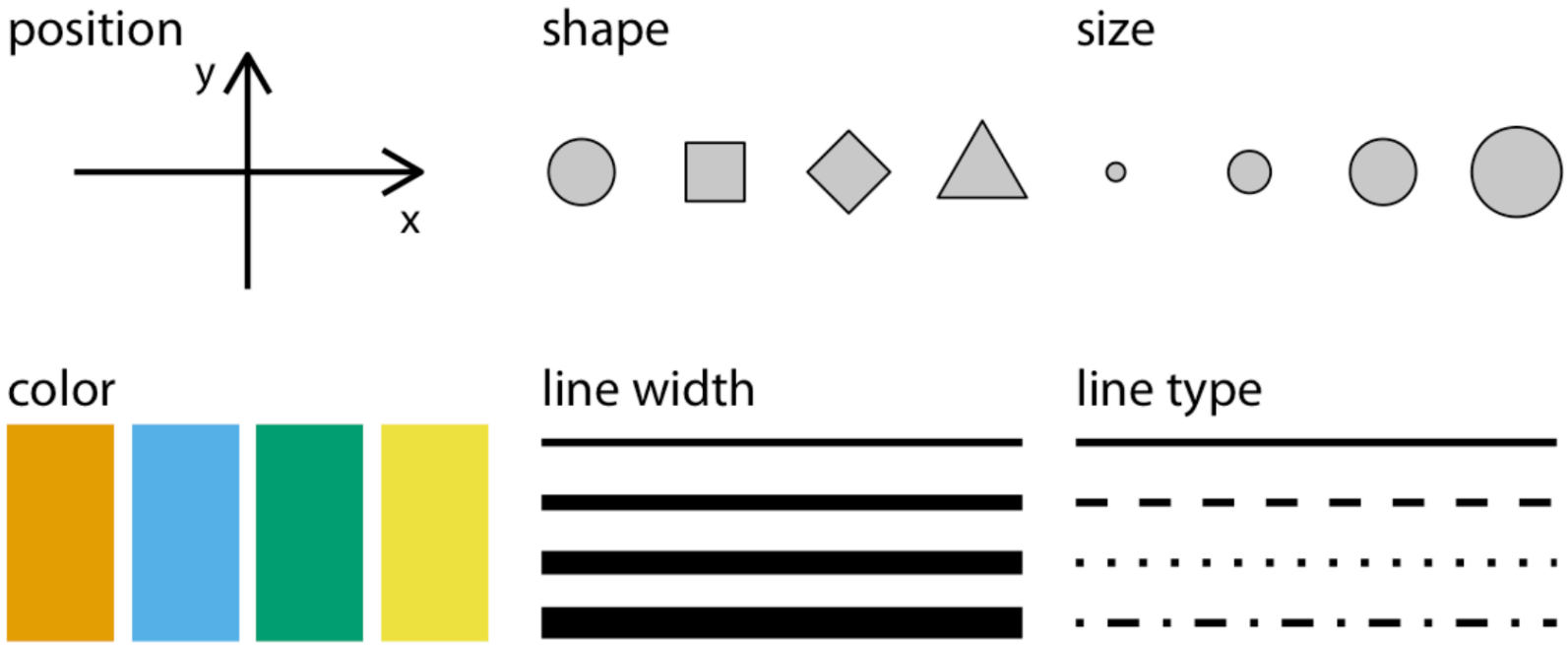
UNSW SYDNEY

Aesthetic Mapping

Aesthetic mappings

- An aesthetic is a visual property of the objects in your plot.
- Aesthetics include things like the *position*, *size*, the *shape*, or the *color* of your points.
- Mapping: variables are ‘mapped’ to (represented by) aesthetics.

A main pool of Aesthetics



A main pool of aesthetics

Note: This figure is from Wilke's *Fundamentals of data Visualization*.

`aes()` means 'Ask'

- `aes()`: What variables are we asking the aesthetic (color, position, shape, etc.) to represent?
- `aes(color=gender)`: 'Please represent the variable `gender` for me using different colors.'

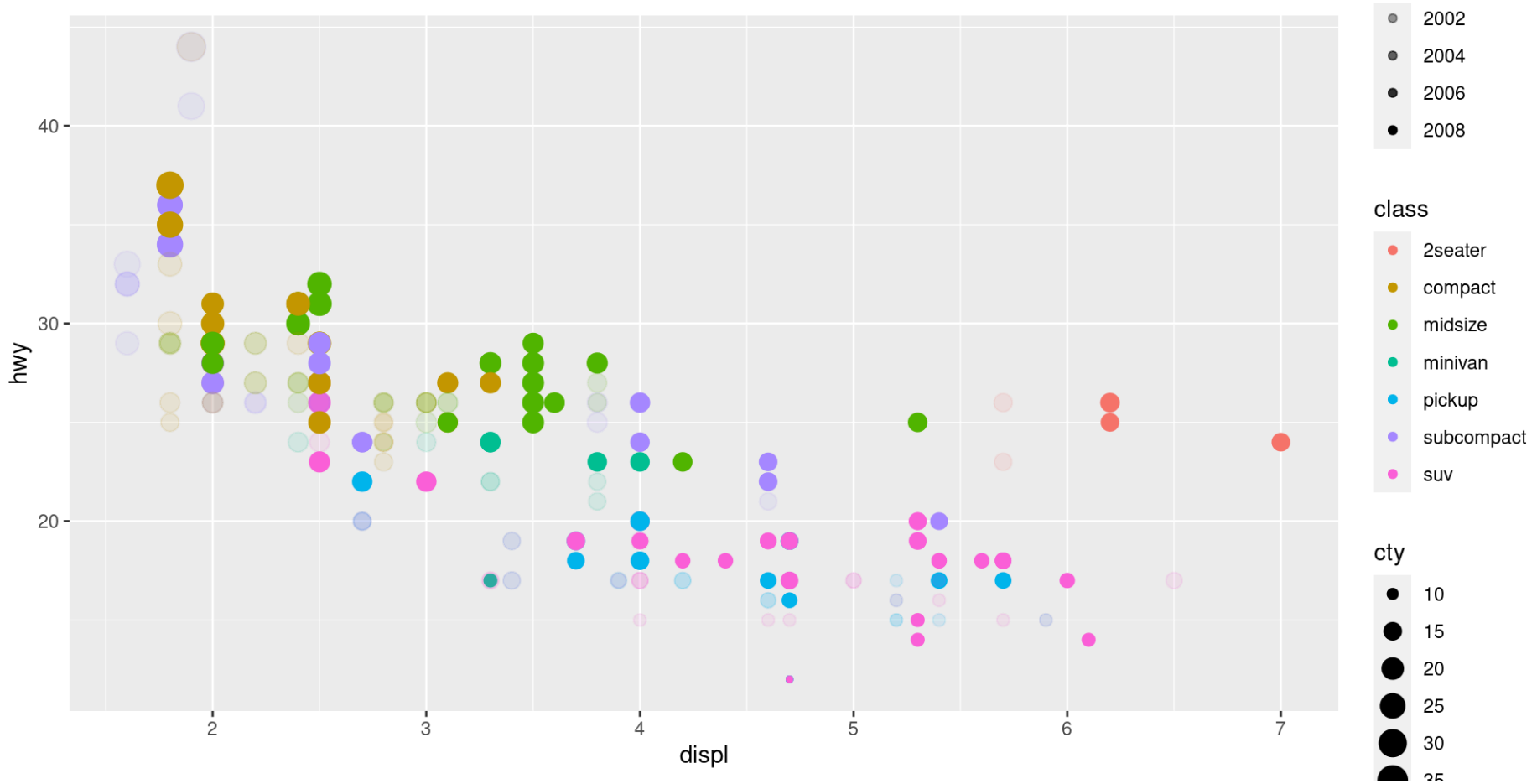
UNSW SYDNEY

More `aes` mappings

```
mpg_plot= ggplot(data = mpg) + # the dataset
  aes(x = displ) + # the x position
  aes(y = hwy) + # the y position
  geom_point() +
  #the point geometric shape, the above aes are requied and
  #the below are optional
  #theme(axis.title=element_text(size=14,face="bold"))+
  aes(color = class) + # Color for type of car
  #aes(shape = class) +
  #ggplot2 will only use six shapes at a time. By default,
  #additional groups will go unplotted when using 'shape'.
  aes(size = cty) +  # Size for city miles per gallon
  aes(alpha = year) # transparency for year of manufacture
```

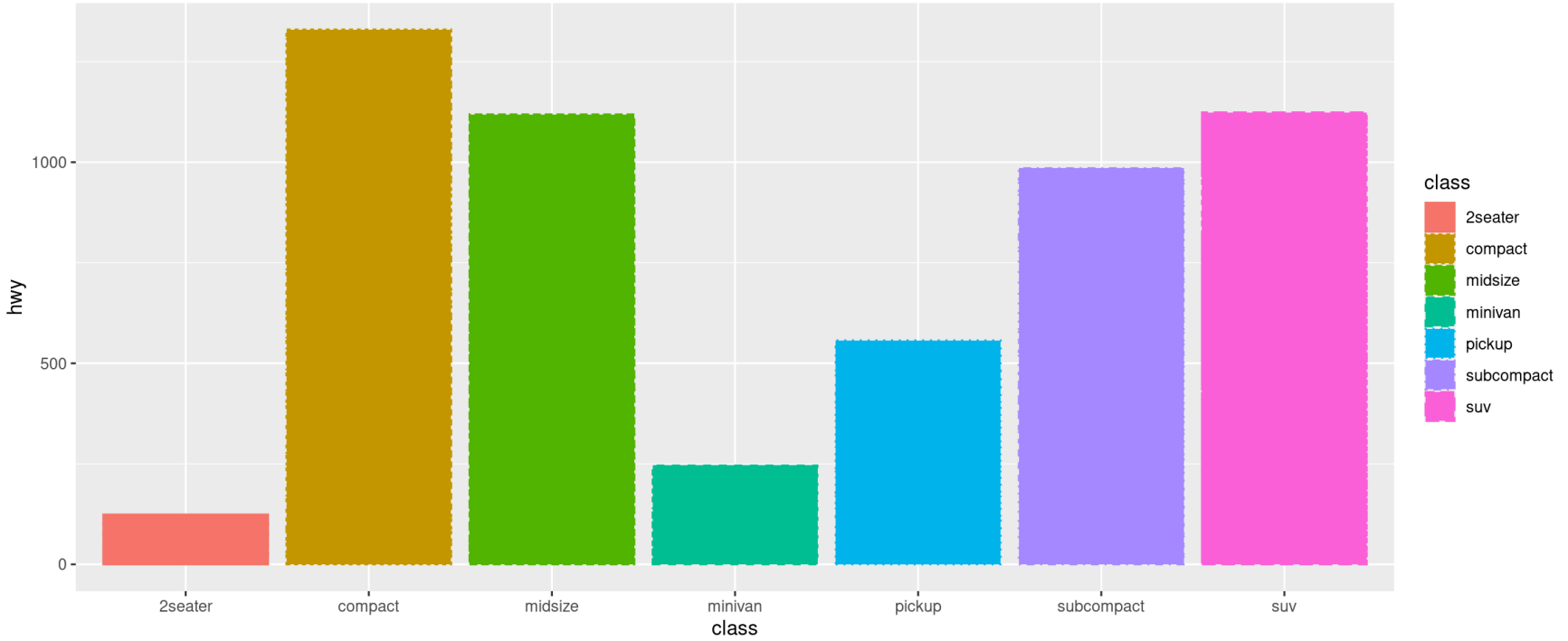
Plot

```
1 print(mpg_plot)
```



Other `aes` for other geometric objects

```
mpg %>% # data piped into
  ggplot() + # initiating plot
  aes(x = class) +  #categorical variable
  aes(y = hwy) +
  geom_col() + #Use `geom_col` to creat a column geometry
  aes(color = class) +
  aes(fill = class) + # new aes 'fill'
  aes(linetype = class) #new aes 'linetype'
```

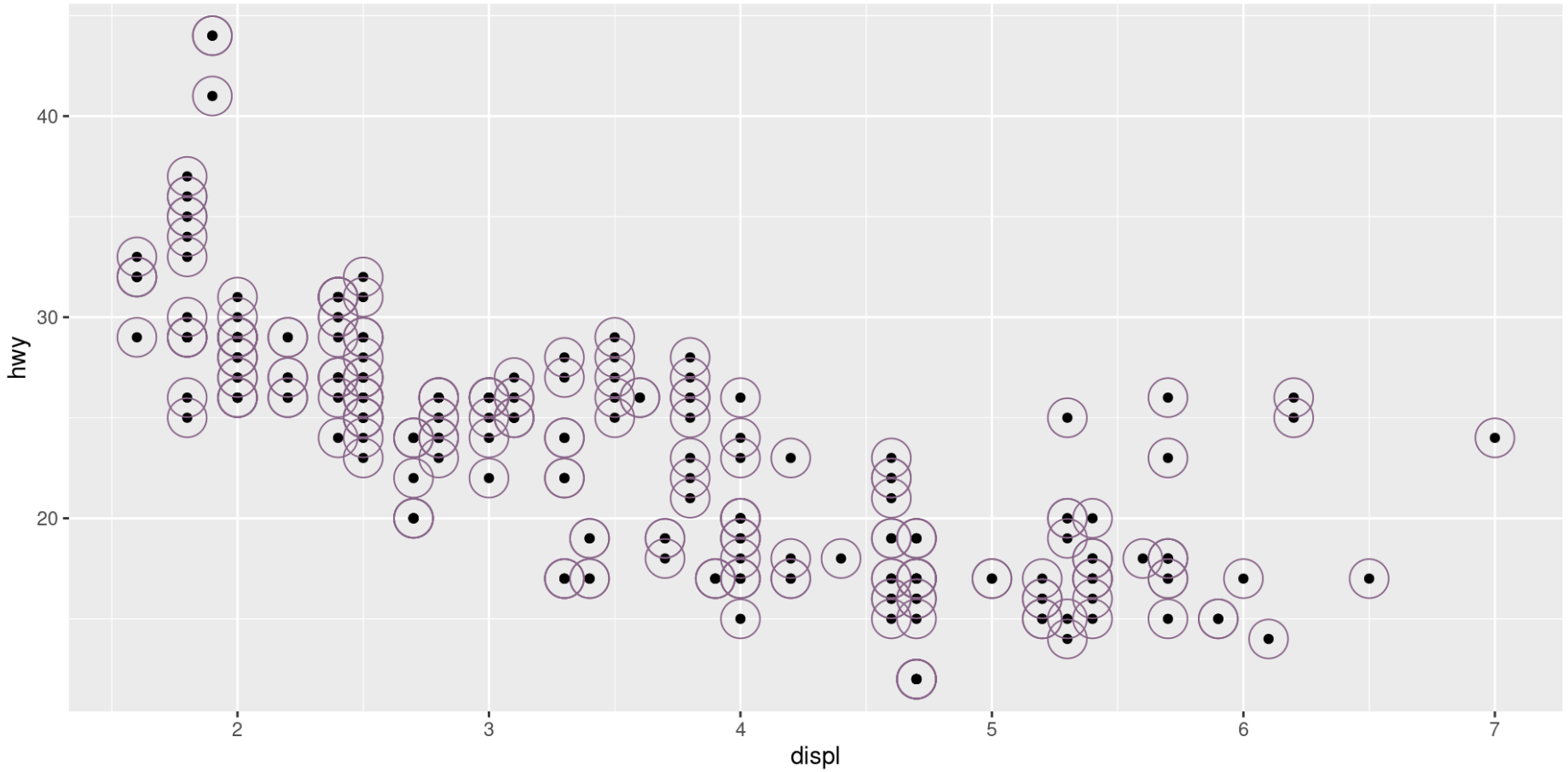


Unmapped aesthetics

```
mpg_plot= ggplot(data = mpg) + # the dataset
  aes(x = displ) + # the x position
  aes(y = hwy) + # the y position
  geom_point() + # the point geometric shape
  #Another geom layer with aesthetics
  #that don't do representation
  geom_point(
    color="plum4",
    size=8,
    shape=21
  )
```

Plot

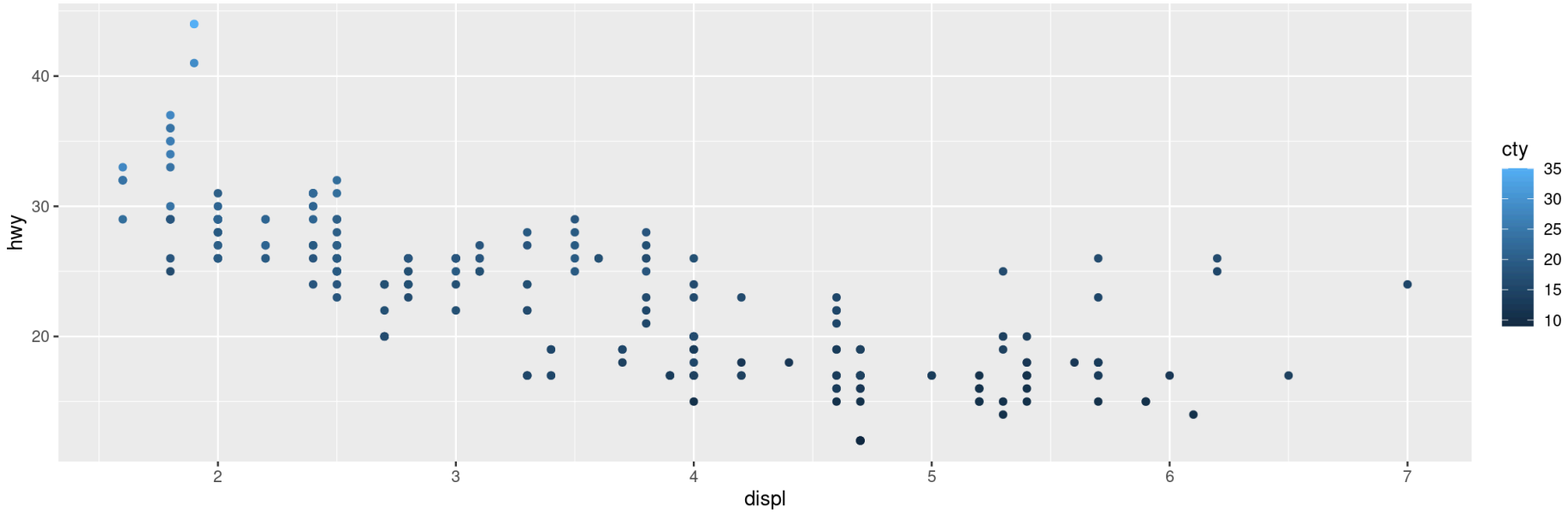
```
1 print(mpg_plot)
```



Exercises:

1. Look at the `help` for `geom_text` (`?geom_text`). What are the required aesthetics?
2. Which variables in `mpg`are categorical? Which variables are continuous? (Hint: type `?mpg` to read the documentation for the dataset). How can you see this information when you run mpg?
3. Map a continuous variable to color, size, and shape. How do these aesthetics behave differently for categorical vs. continuous variables?

```
ggplot(data = mpg) + # the dataset
  aes(x = displ) + # the x position
  aes(y = hwy) +
  aes(color = cty)+ #color, size and shape
  # the y position
  geom_point()  # the point geometric shape
```

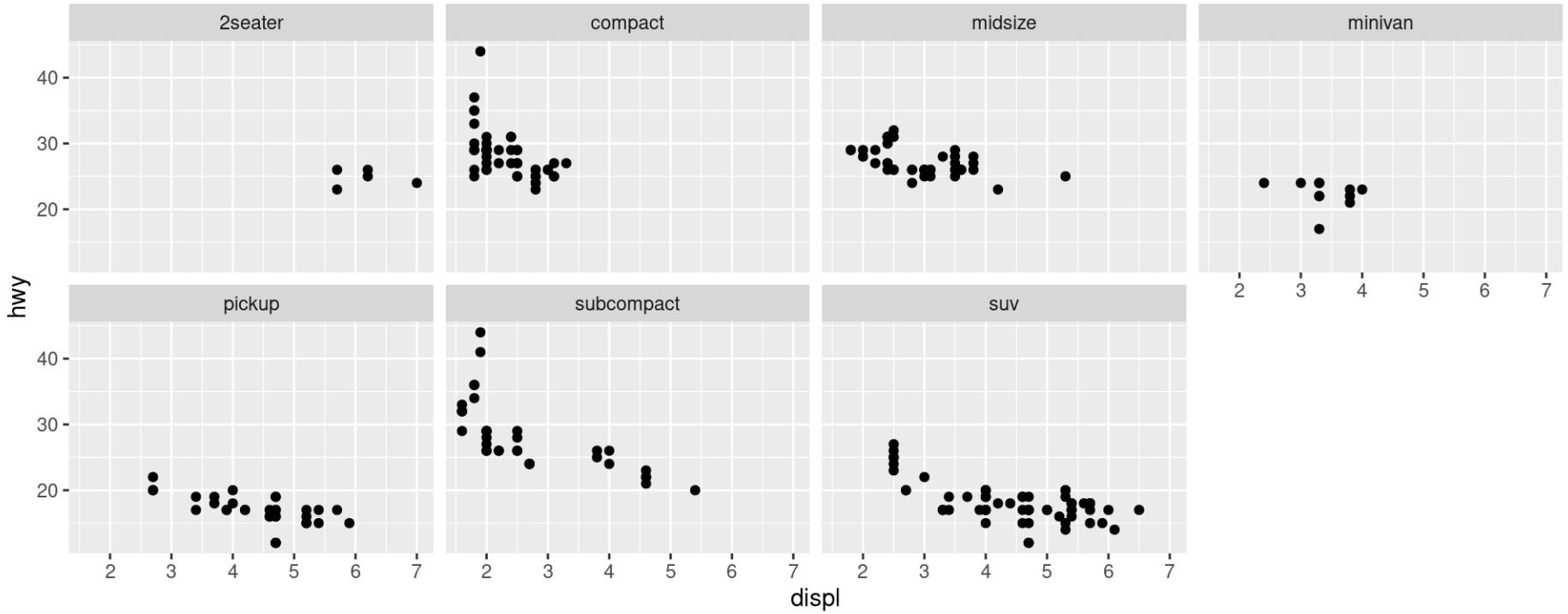


Facets

Facets: facet_wrap

- facet your plot by a single variable

```
1 ggplot(data = mpg) +
2   geom_point(mapping = aes(x = displ, y = hwy)) +
3   # ~ followed by a discrete variable
4   facet_wrap(~ class, nrow = 2)
```

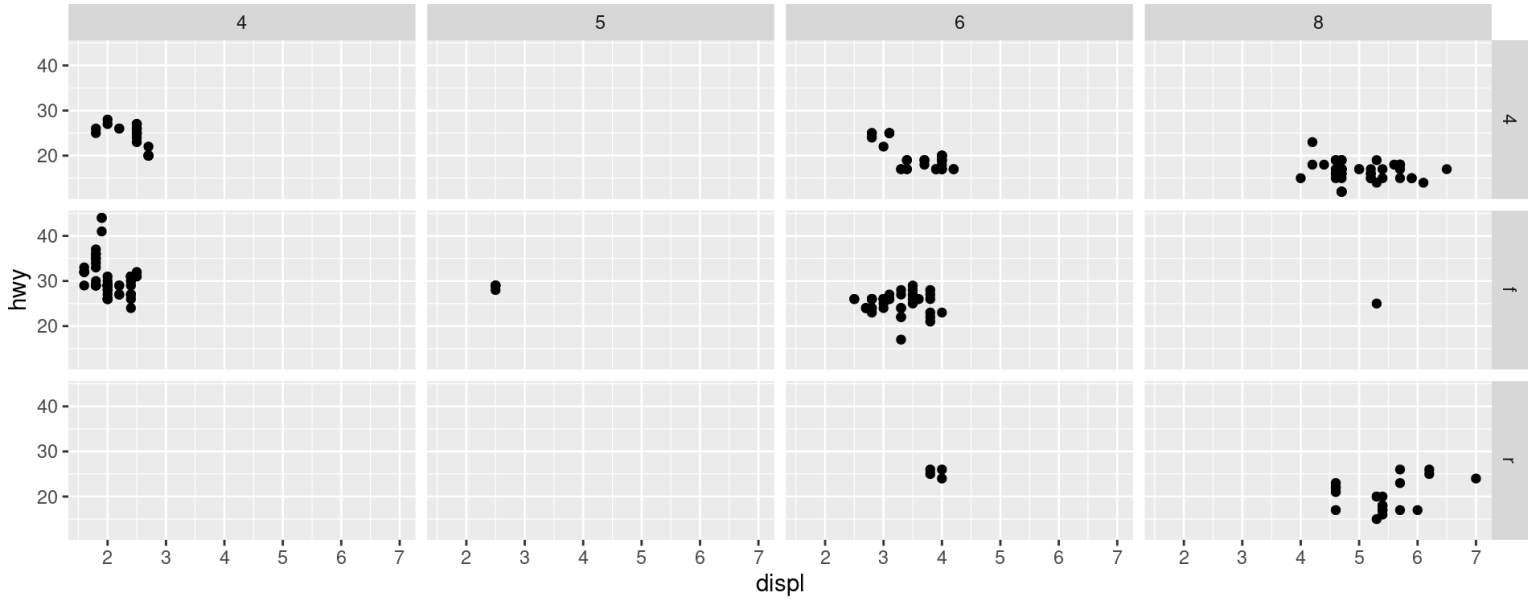


Facets: facet_grid

- facet your plot on the combination of two variables

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  # two variable names separated by a ~
  facet_grid(drv ~ cyl)
```

```
  #facet_grid(. ~ cyl) #not facet in the rows
  #facet_grid(drv~.) #not facet in the rows
```



Geometric Object

A complete sentence of `ggplot`

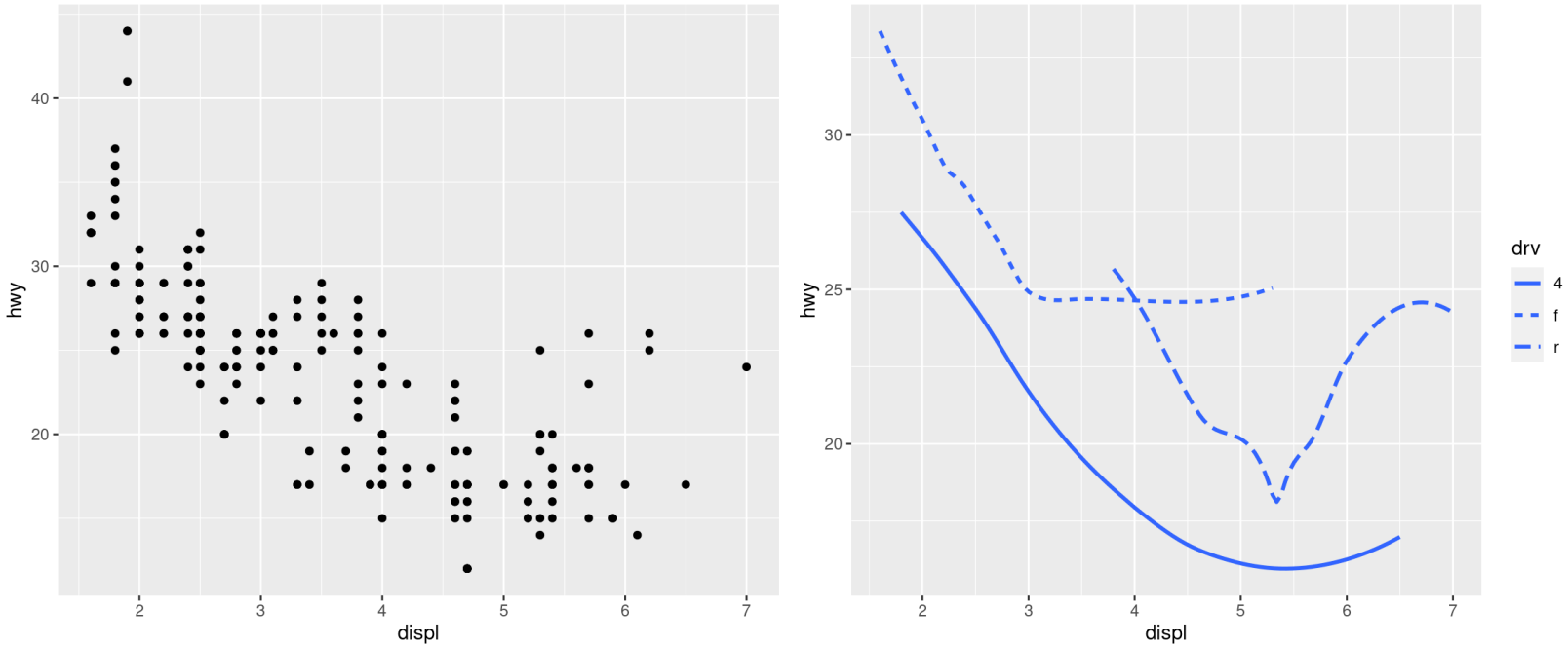
- `data` + `aes` + `geom`
- plots
- Nouns: geometric objects
  - `geom_point()`
  - `geom_col()`
  - `geom_line()`
  - `geom_text()`
  - `geom_segment()`
  - `geom_smooth()`
  - `geom_bar()`
  - etc.
- The conditional mood: `geom` specific data and aesthetic mapping

UNSW SYDNEY

Different Geoms

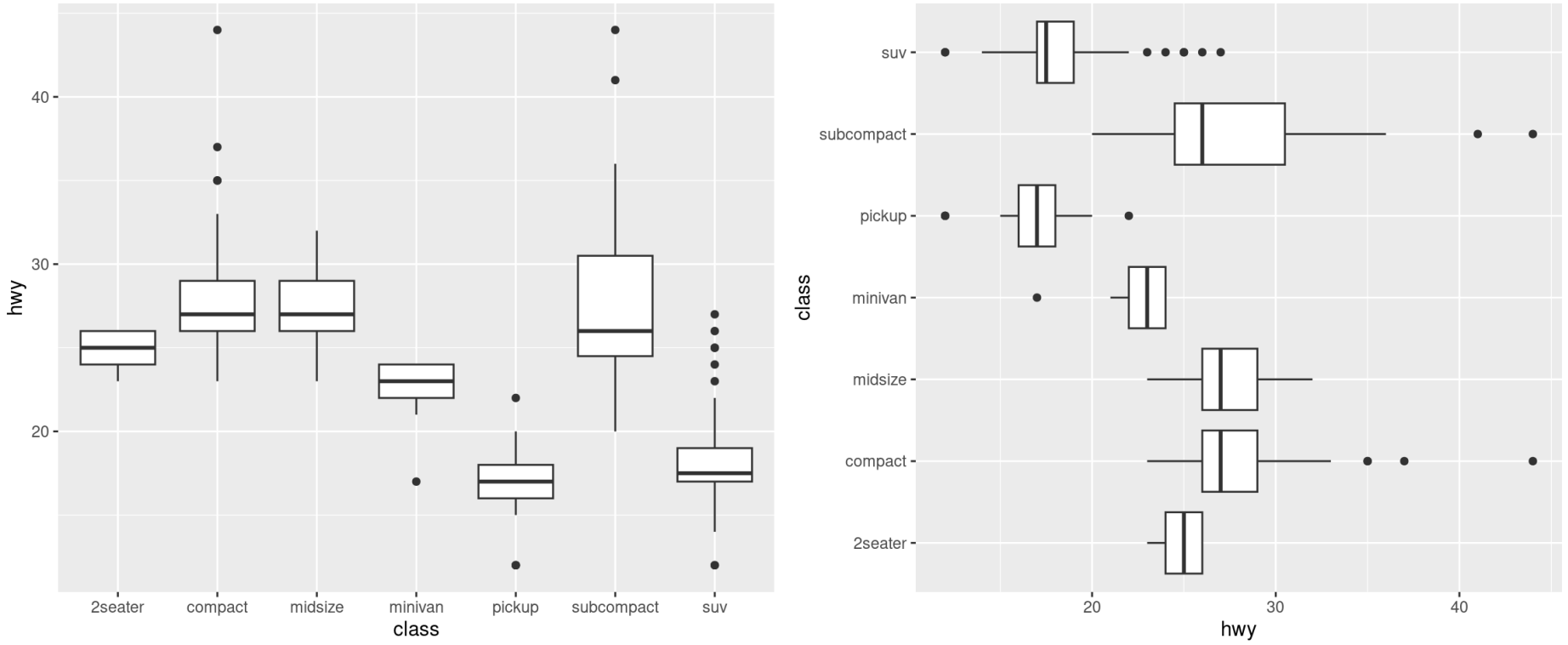
Compare with

```
1 library(grid)
2 library(gridExtra)
3 p1=ggplot(data = mpg) +
4   geom_point(mapping = aes(x = displ, y = hwy)) # geom_point
5 p2=ggplot(data = mpg) +
6   geom_smooth(mapping = aes(x = displ, y = hwy, linetype=drv), se=FALSE) # geom_smooth
7 grid.arrange(p1, p2, ncol = 2)
```



Another Example: boxplot

```
p1=ggplot(data = mpg, mapping = aes(x = class, y = hwy)) +
  geom_boxplot()
p2=ggplot(data = mpg, mapping = aes(x = class, y = hwy)) +
  geom_boxplot() +
  coord_flip() #switches the x and y axes
grid.arrange(p1, p2, ncol = 2)
```

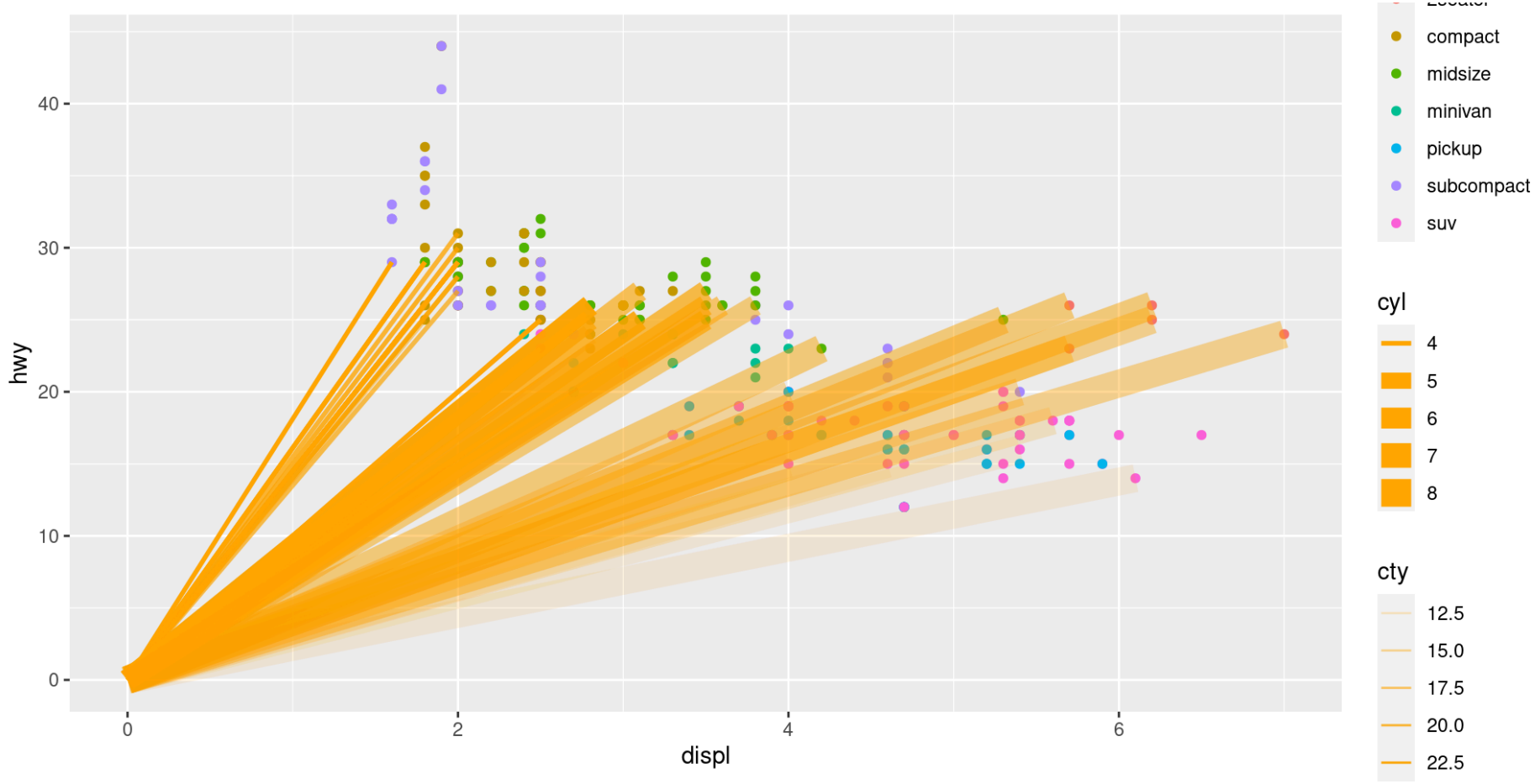


Going local with data and aesthetics

```
mpg_plot= ggplot(data = mpg) + # the dataset
  aes(x = displ, y = hwy) +
  geom_point() +
  aes(color = class) +
  #xend and yend are required for geom_segment
  #like creating a column with a single value
  aes(xend = 0) +  aes(yend = 0) +
  #geom_segment() draws a straight line
  #between points (x, y) and (xend, yend)
  geom_segment(
    #geom specific data, using 'sebset' to select data
    data = subset(mpg, fl=="p"),
    # geom specific (local) aesthetics
    aes(size = cyl, alpha = cty),
    color = "orange"
  )
```

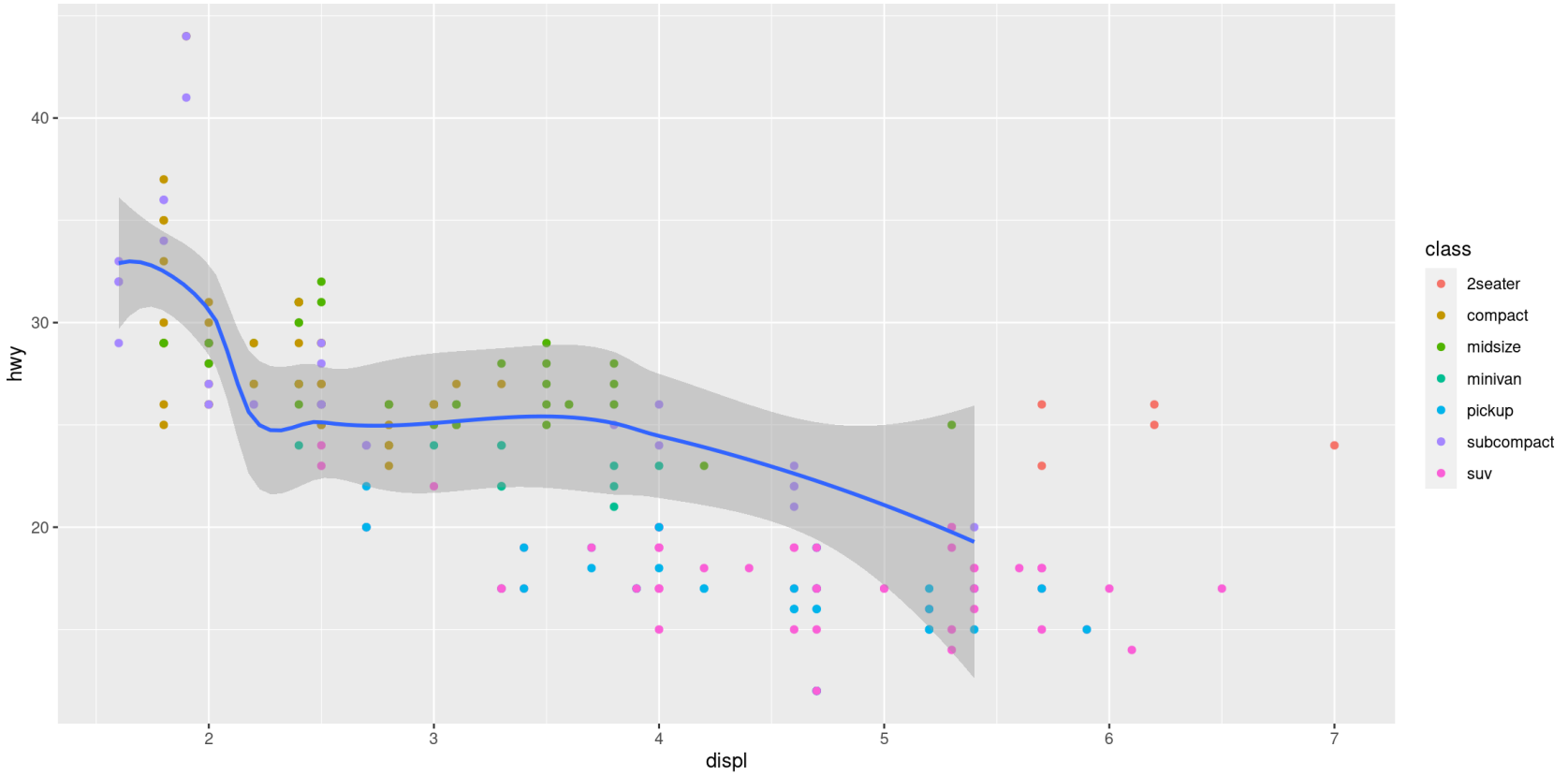
Plot

```
1 print(mpg_plot)
```



Another Example

```
1 ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) + #global aes
2   geom_point(mapping = aes(color = class)) +  #local aes
3   #local data and aes
4   geom_smooth(data = filter(mpg, class == "subcompact"),
5               se = TRUE) #se: standard error
```

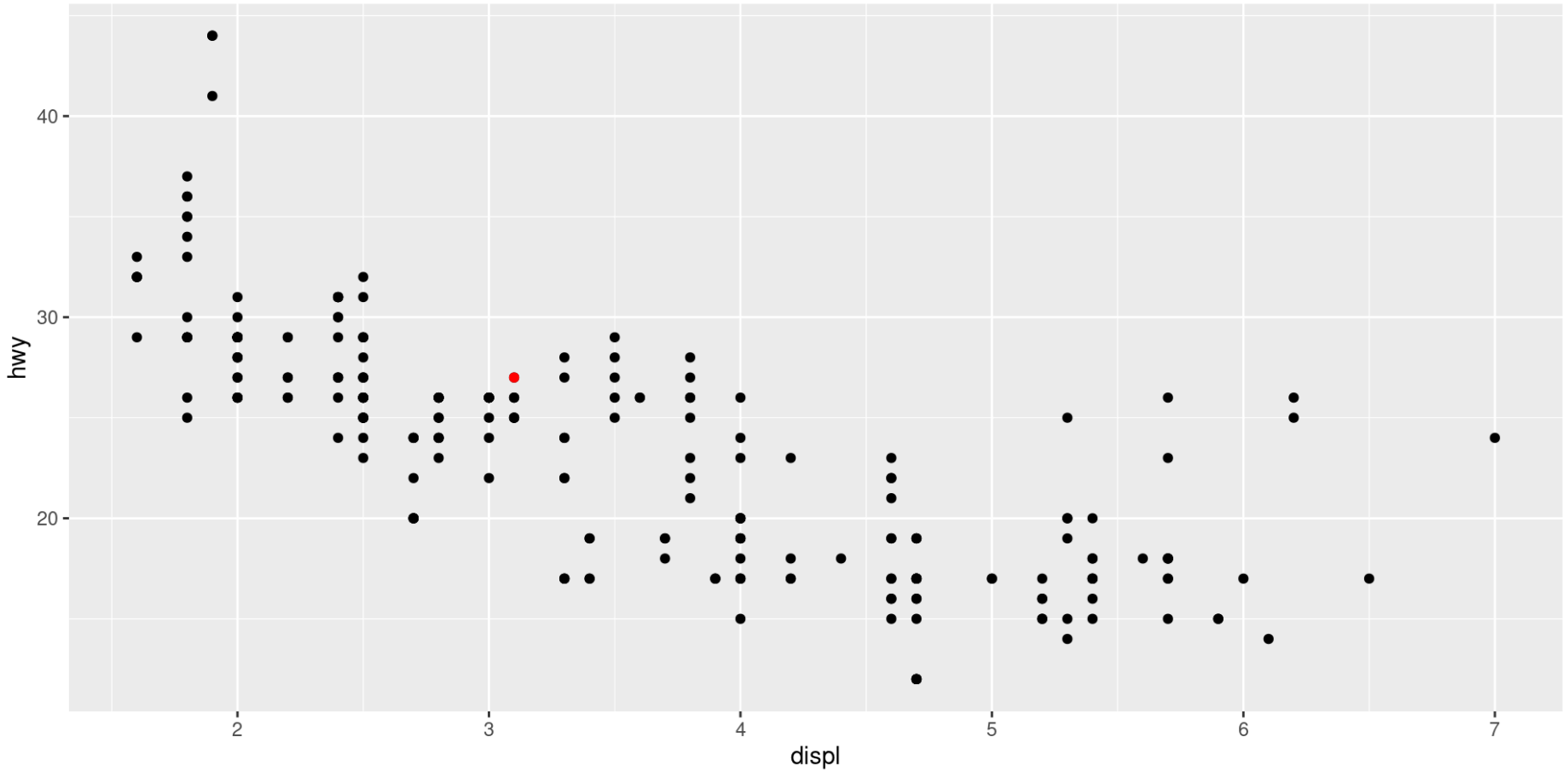


Annotation 1

```
mpg_plot= ggplot(data = mpg) + # the dataset
  aes(x = displ) + # the x position
  aes(y = hwy) + # the y position
  geom_point() + # the point geometric shape
  annotate(geom="point",
           x=3.1,
           y=27,
           color="red")
```

Plot 1

```
1 print(mpg_plot)
```

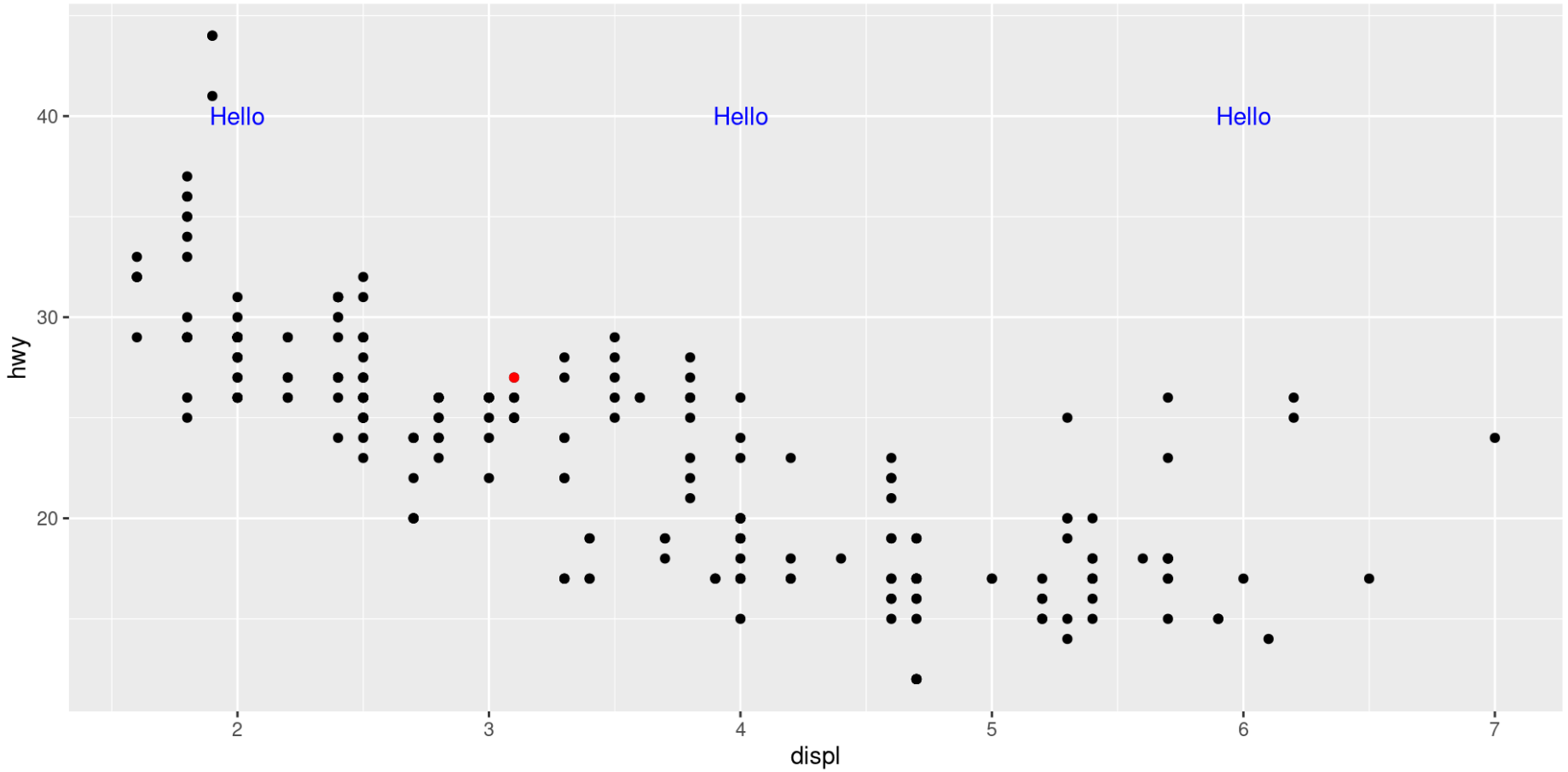


Annotation 2

```
mpg_plot= ggplot(data = mpg) + # the dataset
  aes(x = displ) + # the x position
  aes(y = hwy) + # the y position
  geom_point() + # the point geometric shape
  annotate(geom="point",
           x=3.1,
           y=27,
           color="red")+
  annotate(geom="text",
           x=c(2, 4, 6),
           y=40,
           label="Hello",
           color="blue")
```

Plot 2

```
1 print(mpg_plot)
```

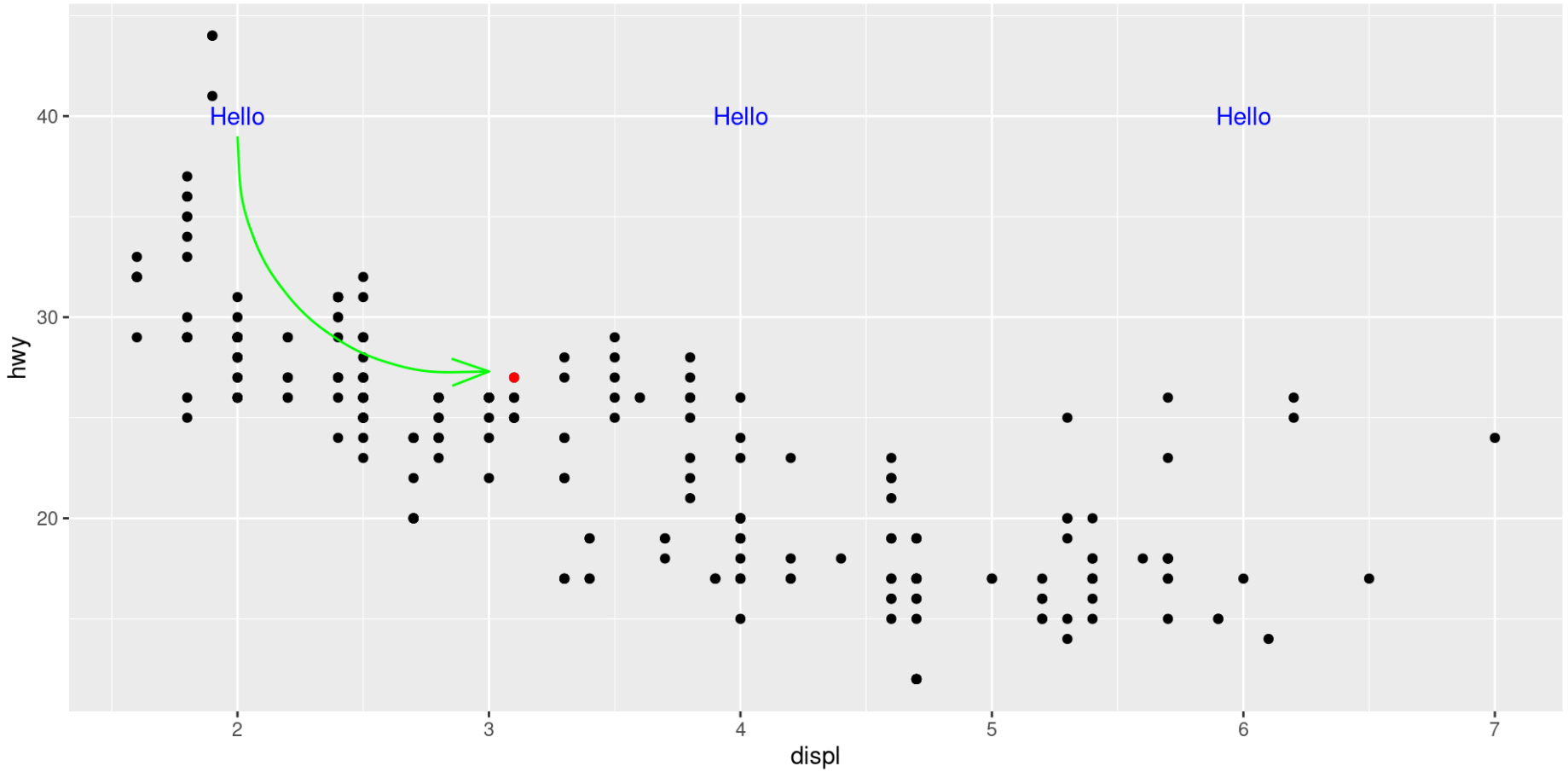


Annotation 3

```
mpg_plot= ggplot(data = mpg) + # the dataset
  aes(x = displ) + # the x position
  aes(y = hwy) + # the y position
  geom_point() + # the point geometric shape
  annotate(geom="point",
           x=3.1,
           y=27,
           color="red")+
  annotate(geom="text",
           x=c(2, 4, 6),
           y=40,
           label="Hello",
           color="blue")+
  annotate(geom="curve",
           x=2,
           y=39,
           xend=3,
           yend=27.3,
           color="green",
           arrow=arrow(angle=20))
```

Plot 3

```
1 print(mpg_plot)
```



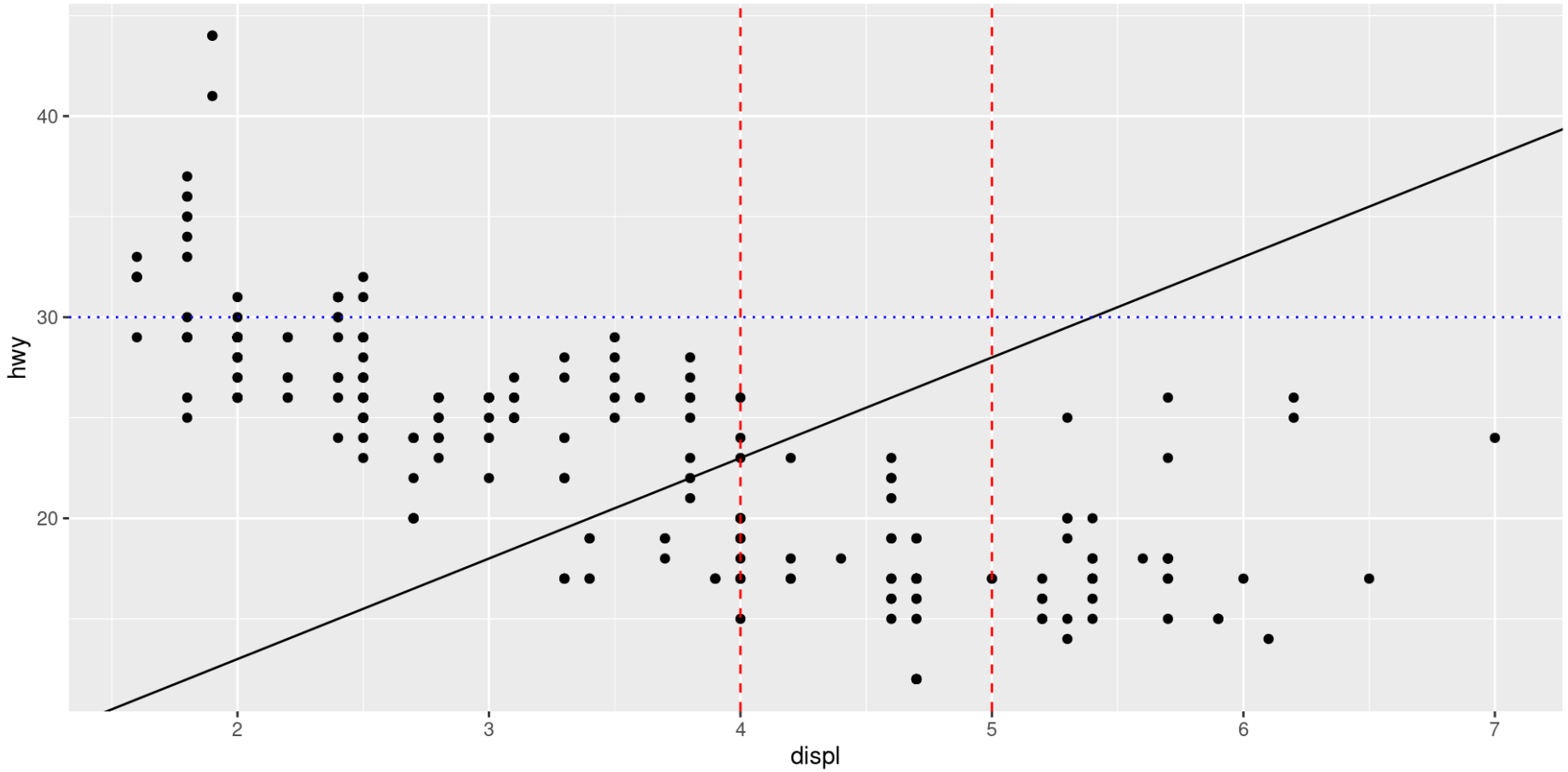
Annotation 4

- use `geom_abline`, `geom_hline`, and `geom_vline`

```
mpg_plot= ggplot(data = mpg) + # the dataset
  aes(x = displ) + # the x position
  aes(y = hwy) + # the y position
  geom_point() + # the point geometric shape
  geom_abline(slope=5, intercept=3) +
  geom_hline(yintercept= 30,
             linetype="dotted", color="blue")+
  geom_vline(xintercept=c(4,5),
             linetype="dashed", color="red")
```

Plot 4

```
1 print(mpg_plot)
```



Exercises

1. What's gone wrong with this code? Why are the points not blue?

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

2. What happens if you map an aesthetic to something other than a variable name, like aes(colour = displ < 5)? Note, you'll also need to specify x and y.

Interactive data visualisation (optional)

- R package: *Shiny*
- can host standalone apps on a webpage
  - Example: Life Expectancy using data from the
- can embed them in R Markdown documents or build dashboards.