PS5841 Data Science in Finance & Insurance
Autumn 2022 (Yubo Wang)

Grammar of Graphics
A Very Brief Introduction

1 – Grammar of Graphics

The Grammar of Graphics describes fundamental features that underlie all statistical graphics.

The combination of independent components makes up a graphic.

- maps the data to the aesthetic attributes of geometric objects
- may include statistical transformations of data and coordinate system information
- may display subsets of data as multiples

Limitations

- does not suggest which graphics to use
- does not describe interactive graphics

2 – Graphic Components

2.1 – Key Components

- **Data**
  the information to be visualized
- **Layer**
  a collection of geometric elements and statistical transformations
  - **geom**
    a geometric element to render observations, e.g. a point
  - **stat**
    summarizes data, e.g. binning in a histogram
- **Scale**
  maps variables to visual properties (aesthetics), e.g. color, shape, size, legend and axis
- **coord**
  maps data coordinates to the graphic, e.g. Cartesian
- **facet**
  specifies how to break up and display subsets of data as multiples
- **theme**
  controls the overall appearance of a graphic, e.g. fonts, background color

2.2 – Components Illustrated

2.2.1 – data, aesthetic mapping, layer

```
library(ggplot2)
options(repr.plot.width = 6, repr.plot.height = 4)
ggplot(data = mpg, aes(x = displ, y = hwy, color = class)) +
  geom_point()
```

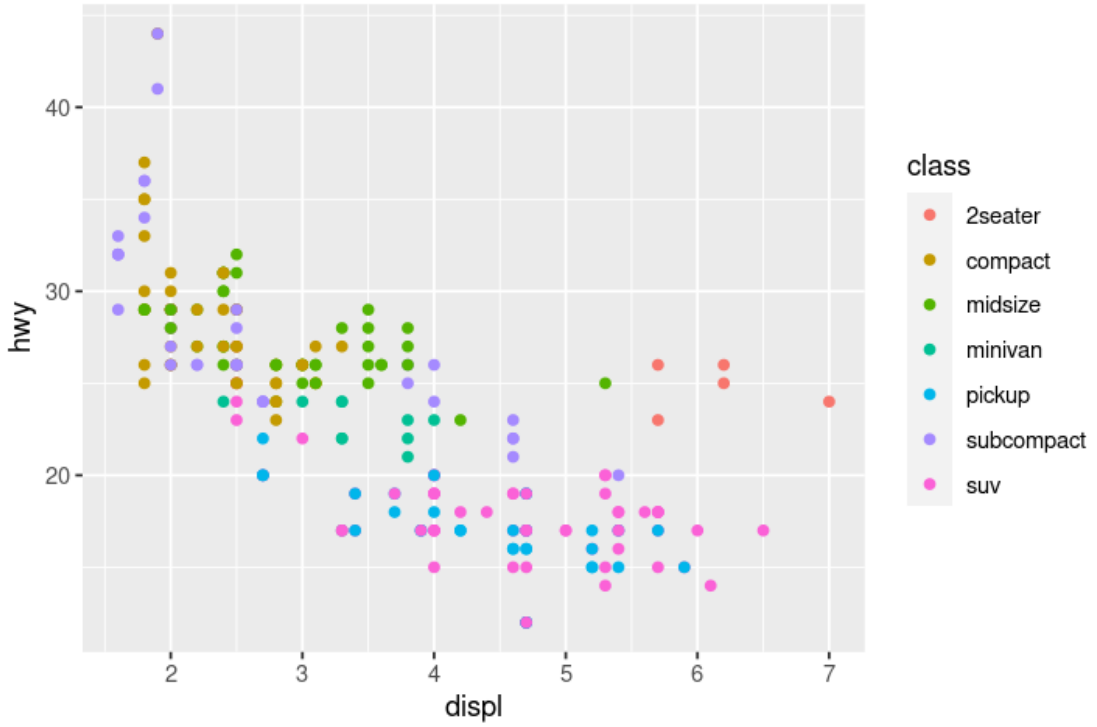

Figure 1: data, aesthetics, layer

2.2.2 – … + facet, theme

```
library(ggplot2)
options(repr.plot.width = 6, repr.plot.height = 4)
ggplot(data = mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  facet_wrap(~ class) +
  theme_bw()
```

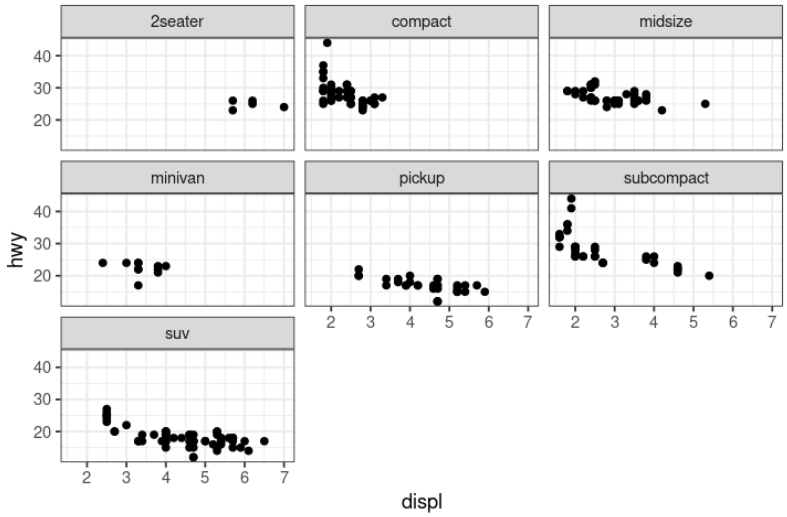

Figure 2: data, aesthetics, layer, facet, theme_bw()

```
library(ggplot2)
options(repr.plot.width = 6, repr.plot.height = 4)
ggplot(data = mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  facet_wrap(~ class) +
  theme_light()
```

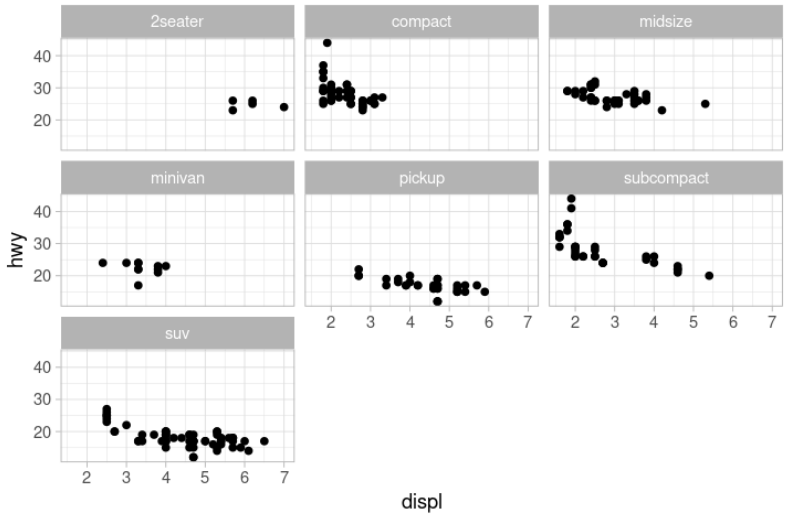

Figure 3: data, aesthetics, geom_layer, facet, theme_light()

2.2.3 – … + coord

```
library(ggplot2)
options(repr.plot.width = 6, repr.plot.height = 4)
r <- seq(0, 3, length.out = 100)
theta <- 2 * pi * r
df <- data.frame(theta = theta, r = r)
ggplot(data = df, mapping = aes(x = theta, y = r)) +
   geom_path() +
   scale_x_continuous(limits = c(0, 2*pi), oob = scales::oob_keep) +
   coord_polar()
```

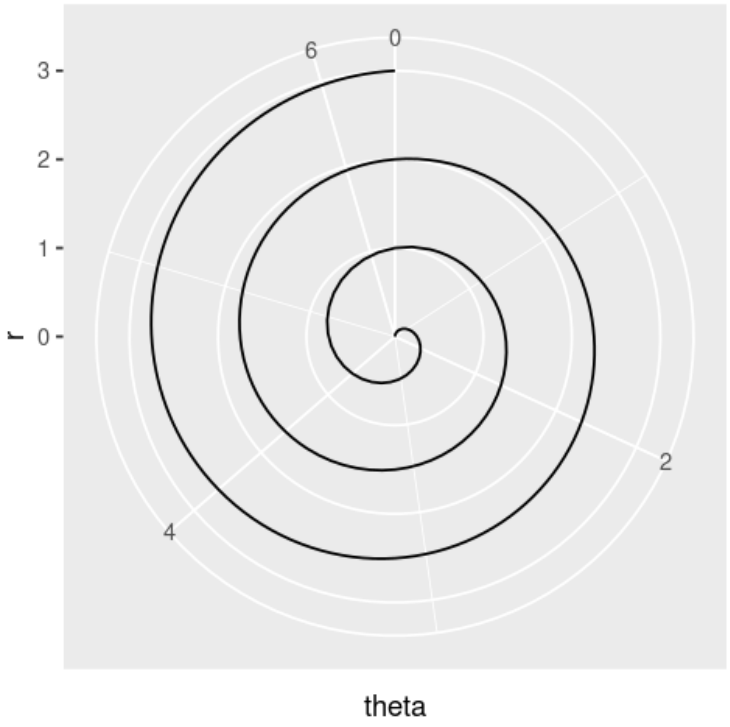

Figure 4: polar coordinates

3 – Implementation in R and Python

3.1 – R with ggplot2

```
library(ggplot2)
ggplot(mpg, aes(displ, hwy)) + geom_point() + geom_smooth(method = lm)
```

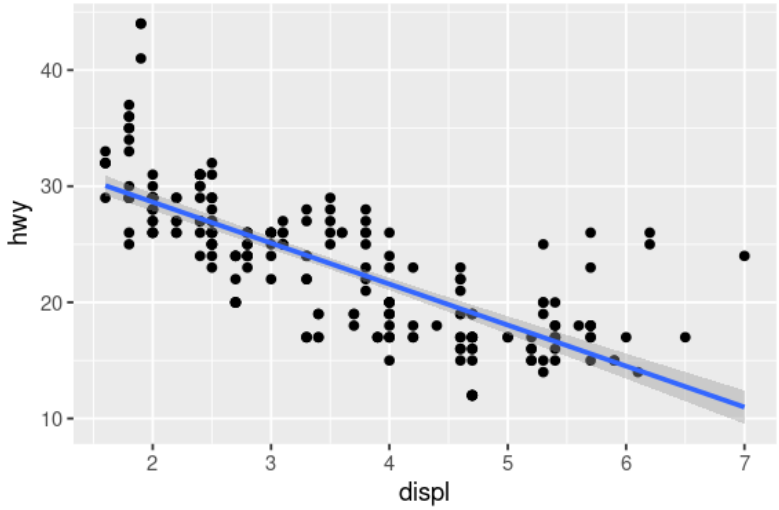

Figure 5: implementation in R

3.2 – Python with plotnine

```
from plotnine import *
from plotnine.data import mpg
p = ggplot(mpg, aes('displ', 'hwy')) +  geom_point() +\
  geom_smooth(method = 'lm', color = 'b')
print(p)
```

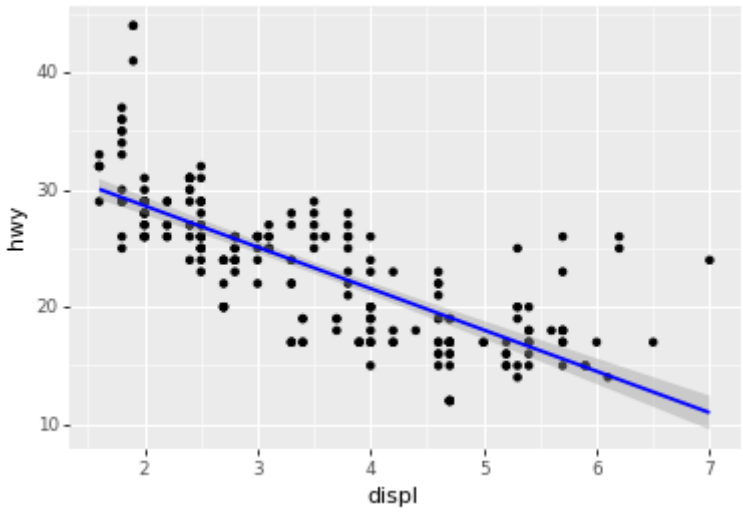

Figure 6: implementation in Python

3.3 – Geoms

- points
  - geom_point()
- lines
  - geom_abline(), geom_hline(), geom_vline()
  - geom_line(), geom_path(), geom_step()
  - geom_function()
- Bar charts
  - geom_bar()
  - geom_col()
- Histogram
  - geom_histogram()
  - geom_freqpoly()
- Box plot
  - geom_boxplot()
- Contour plot
  - geom_contour(), geom_contour_filled()
  - geom_density_2d(), geom_density_2d_filled()
- Kernel density
  - geom_density()
- Shades
  - geom_ribbon()
  - geom_area()
- and more …

4 – Reference Documentation

- ggplot2 function reference
- plotnine api reference