Graphics with ggplot2

David Gerard

2019-01-22

Learning Objectives

- ggplot2
- Basics of data visualization.
- Chapter 3 RDS

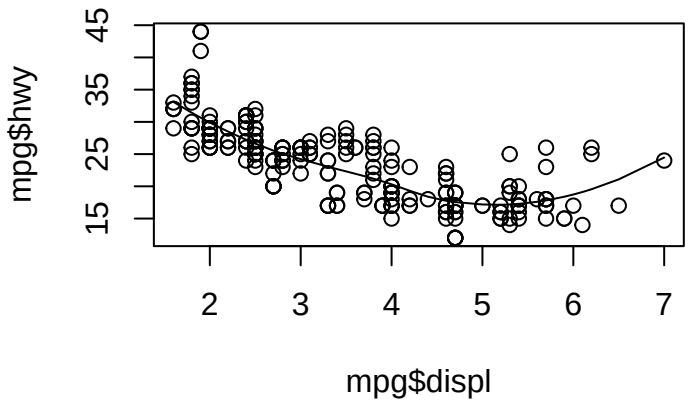
Background

- The three main plotting systems in R are base R, lattice, ggplot2 (from the tidyverse) (with base R-like syntax implemented in the `qplot()` function.
- In base R: start with a blank plot and build on it by adding elements — the "artist's canvas".
- In lattice: One function, specify everything at once.
- ggplot2: A "grammar of graphics". You map from variables to aesthetic attributes (size, color, shape) of geometic objects (lines, bars, points).
- Let's look at the three ways to make a scatterplot with a loess smoother.
  - Load data and packages:

```
library(ggplot2)
library(lattice)
data("mpg")
```

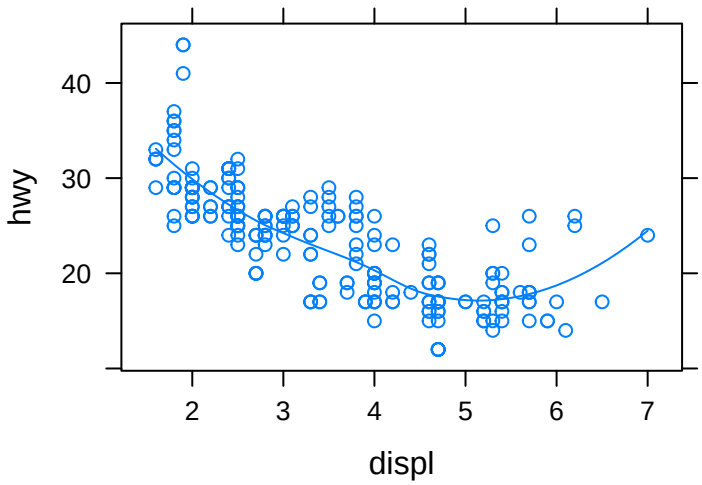
  - base R: need to create loess fit and order points on our own.

```
lout <- loess(hwy ~ displ, data = mpg)
ord_x <- order(lout$x)
plot(x = mpg$displ, y = mpg$hwy)
lines(lout$x[ord_x], lout$fitted[ord_x])
```



– lattice: Everything is done at once within some mega-function:

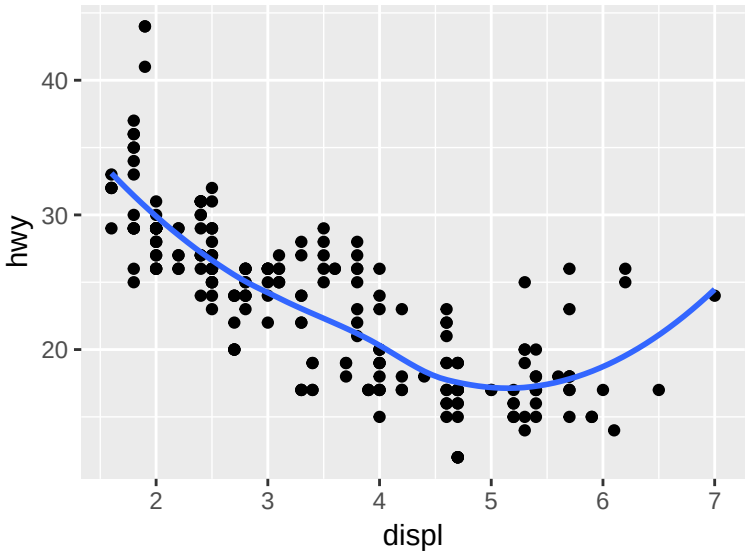
```
xyplot(hwy ~ displ, data = mpg,
       panel = function(x, y) {
         panel.xyplot(x, y)
         panel.loess(x, y, span = 0.75, family = "gaussian", degree = 2)
       })
```



– ggplot2: Sequentially add geometric objects:

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

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



- lattice is the hardest to learn and the least used.
- base R is very flexible, but harder to use for most tasks.

- ggplot2 is very easy to use for most tasks, but very hard to use when you want to do something weird.

ggplot2 basics

- `ggplot()` has two main arguments:
  - `data`: Must be a data frame. Contains the variables that you want to map onto aesthetics.
  - `mapping`: A definition for which variables map onto which aesthetics. You almost always use the `aes()` function to perform this mapping.
- `ggplot()` is almost always followed by a `+` and then a `geom` arguement to specify which geometic arguements receive the aesthetic mapping.

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