

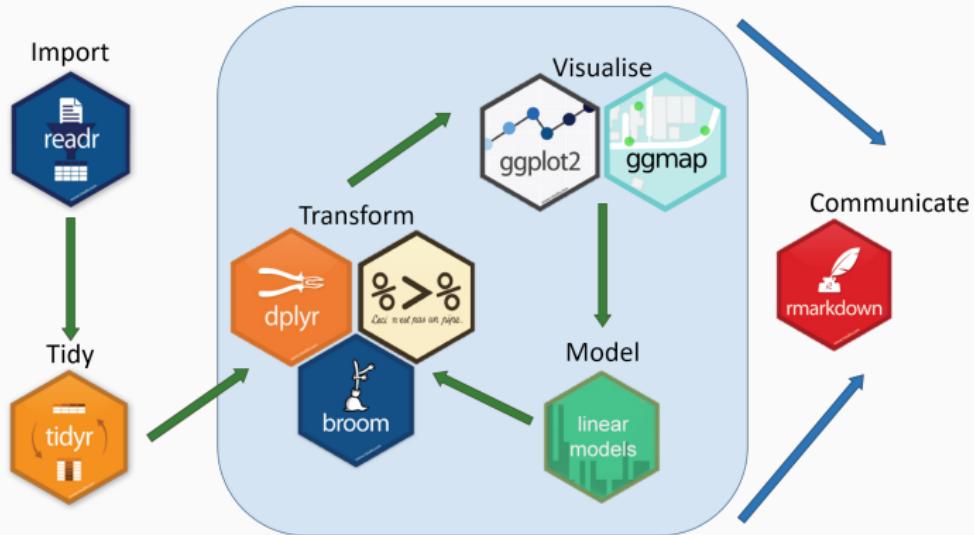
第五章：数据可视化

王敏杰

2020 年 7 月 27 日

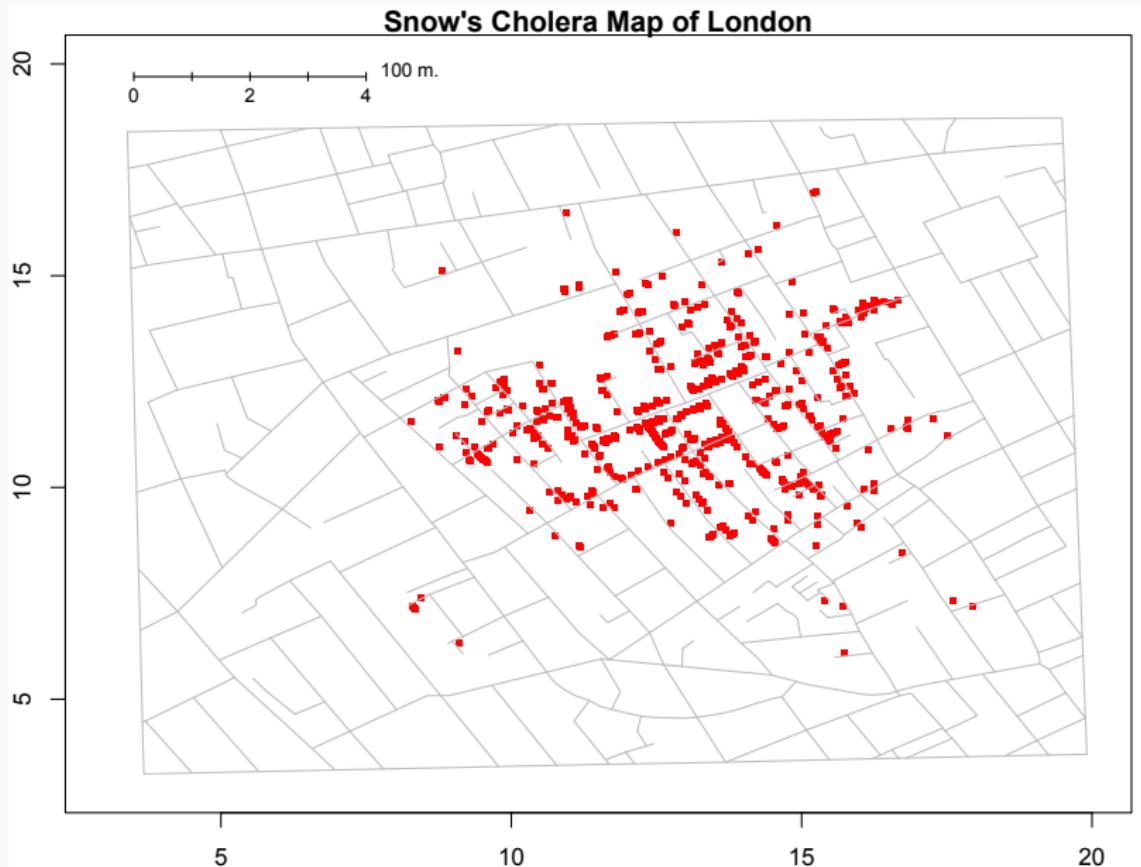
四川师范大学

tidyverse 家族

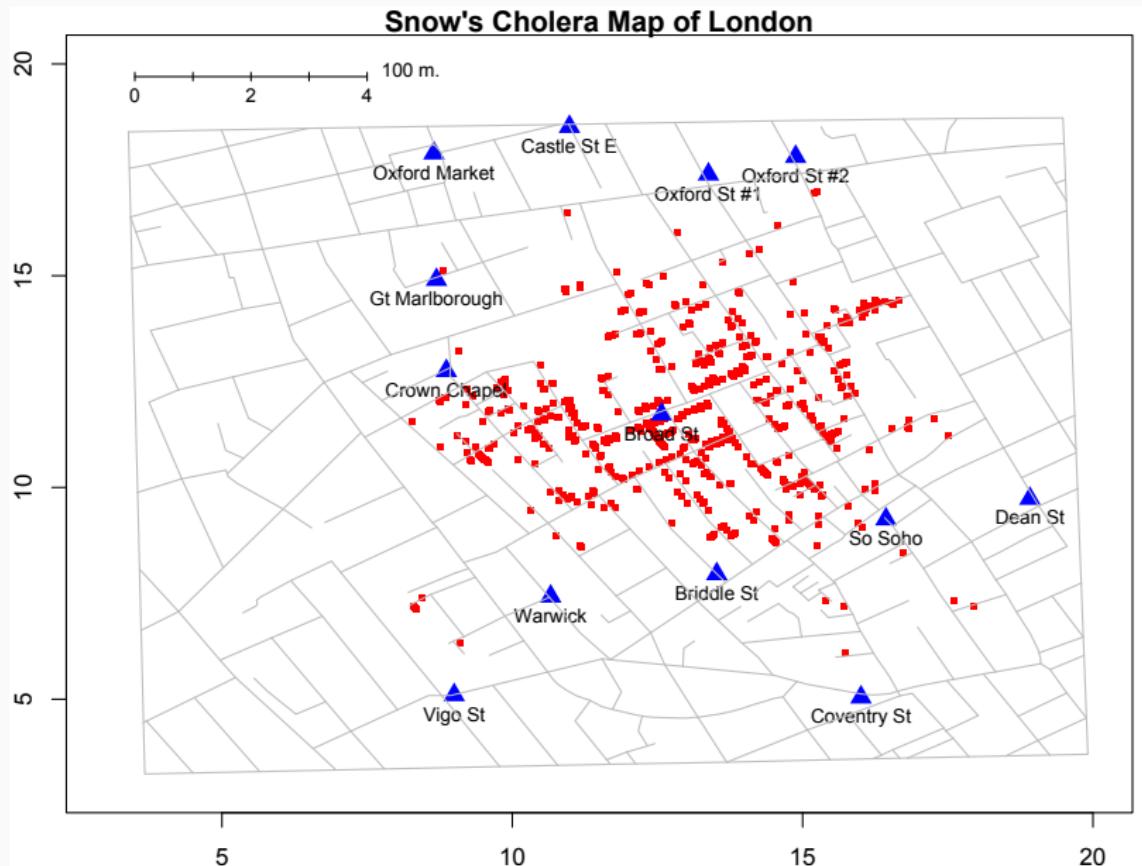


为什么要可视化

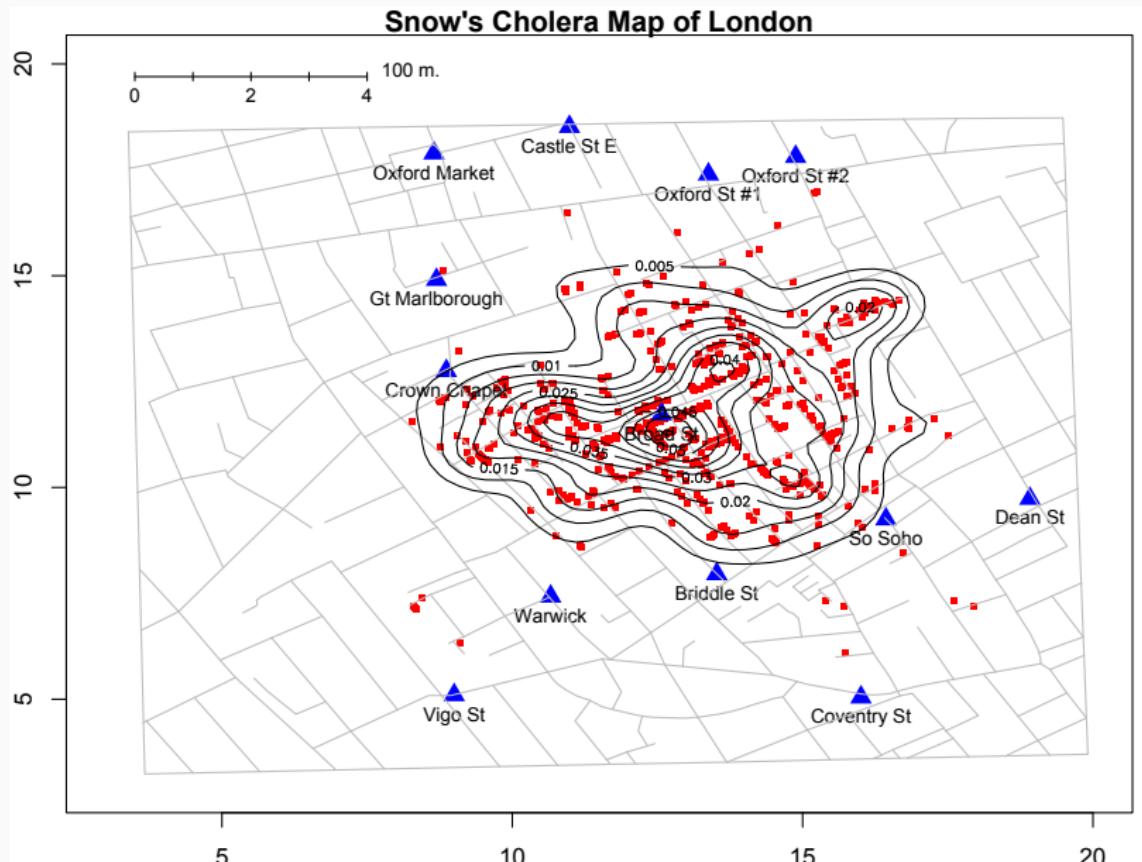
1854 年伦敦霍乱



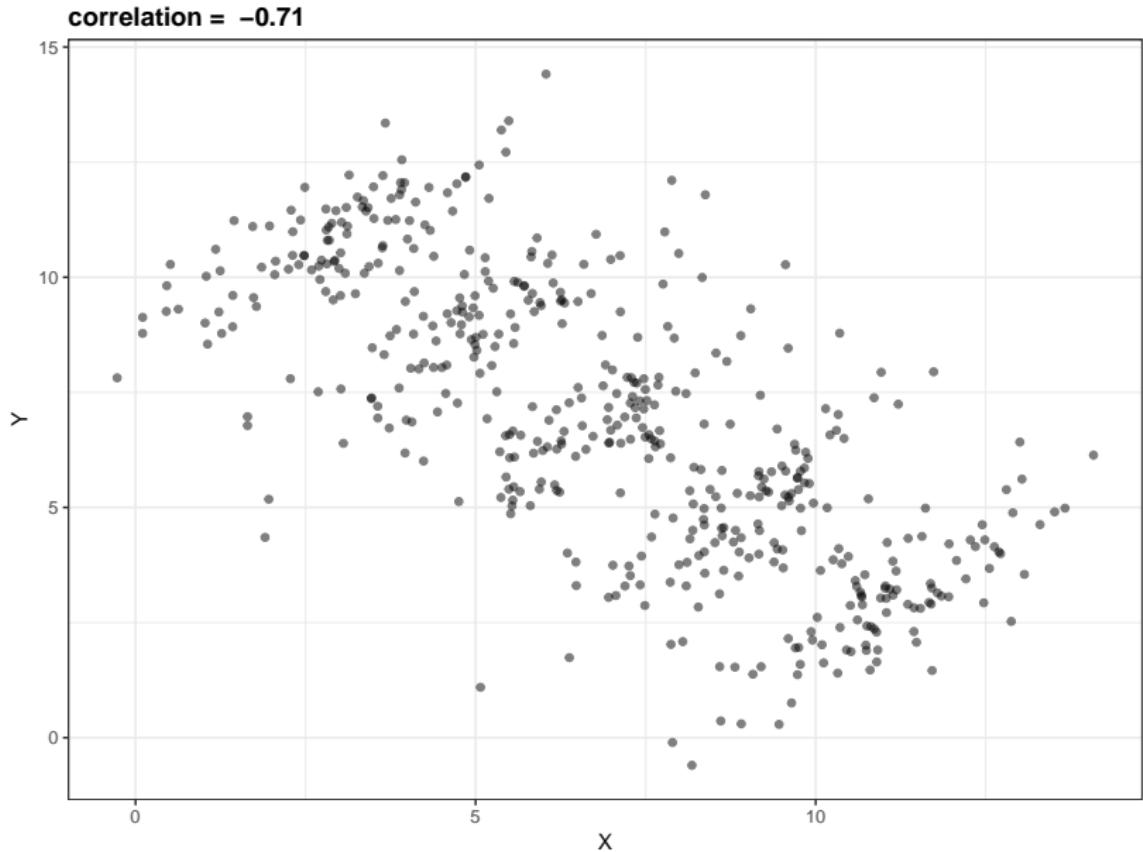
1854 年伦敦霍乱



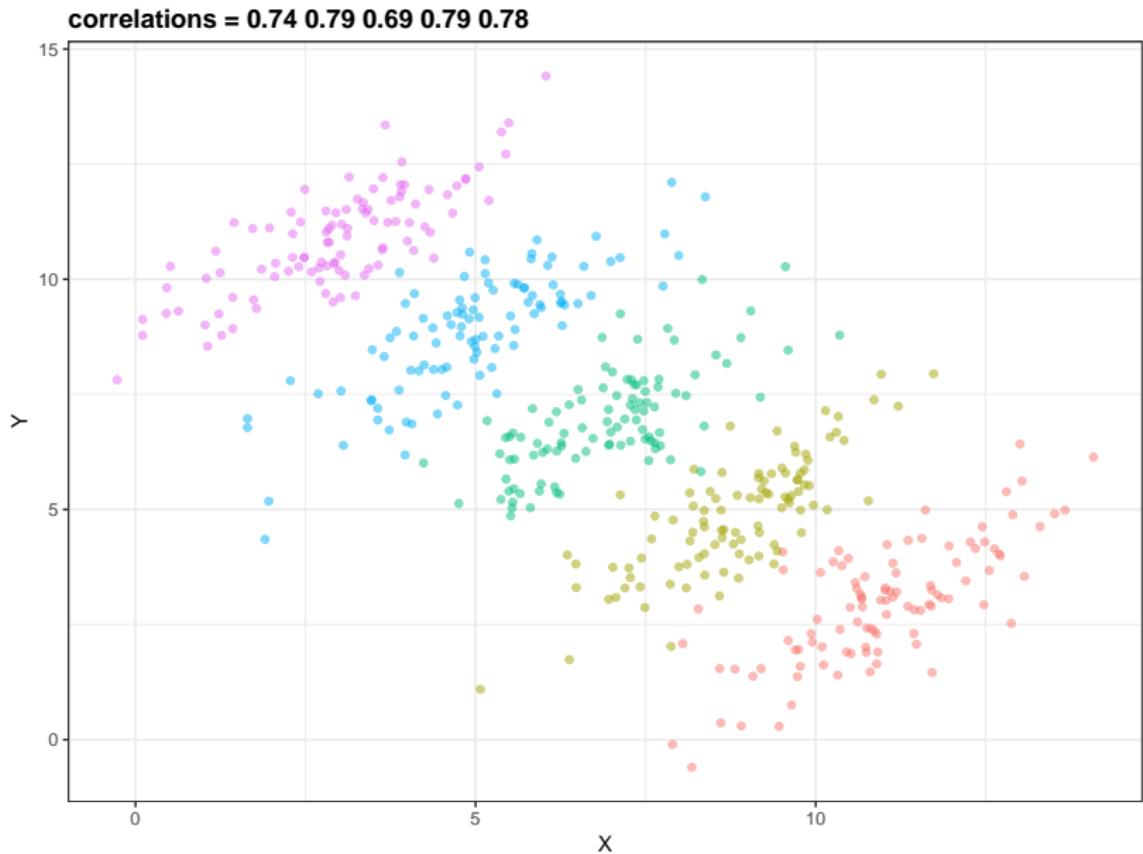
1854 年伦敦霍乱



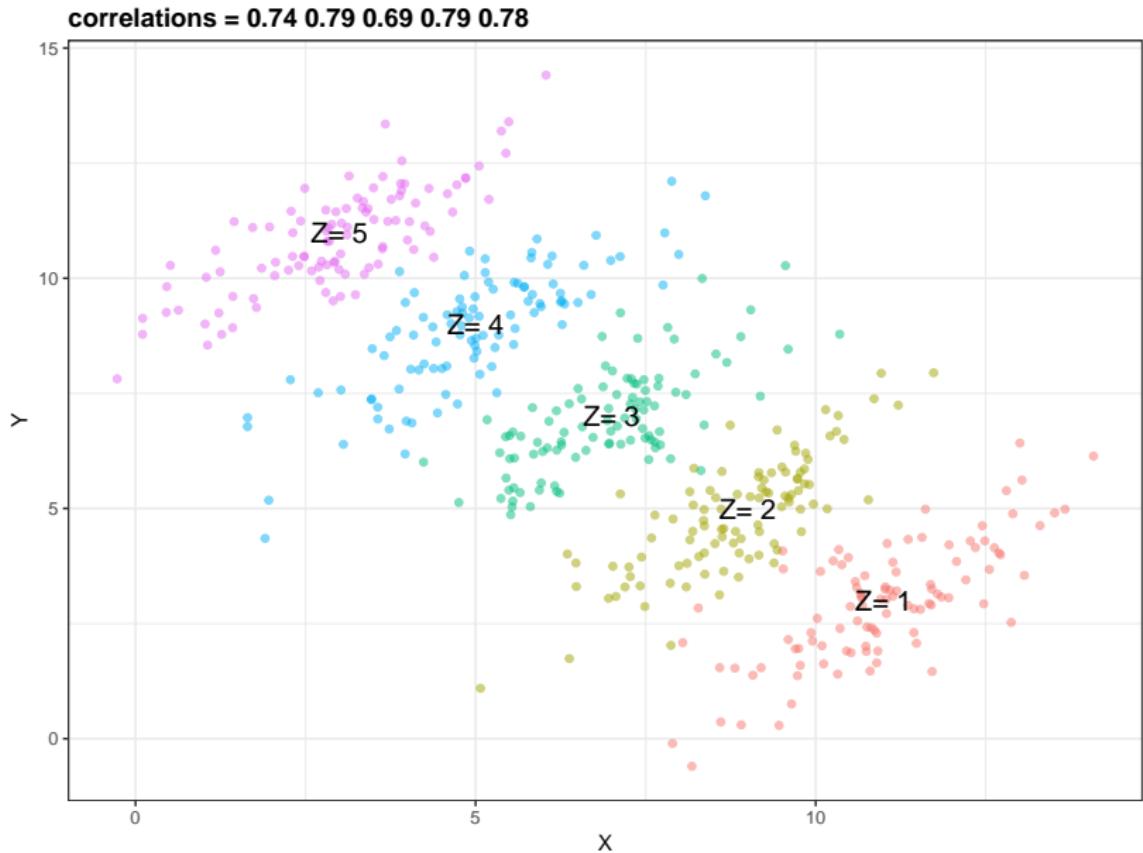
辛普森悖论 (Simpson's Paradox)



辛普森悖论 (Simpson's Paradox)



辛普森悖论 (Simpson's Paradox)



ggplot2 宏包

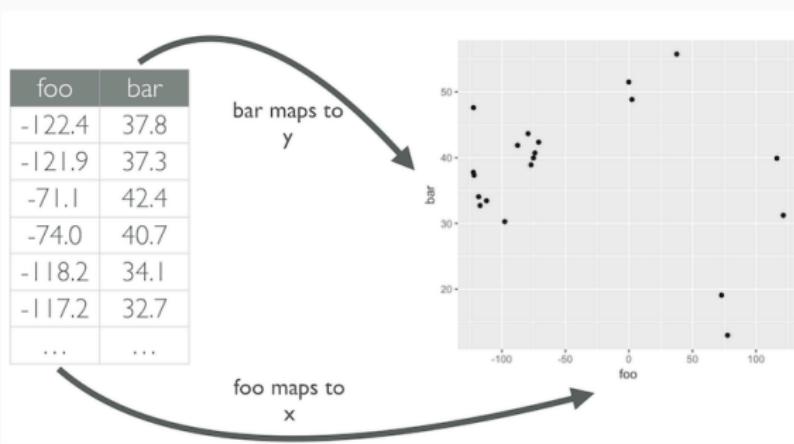
宏包 ggplot2

- ggplot2 是 RStudio 首席科学家 Hadley Wickham 在 2005 年读博士期间的作品。
- 很多人学习 R 语言，就是因为 ggplot2 宏包
- ggplot2 已经发展成为最受欢迎的 R 宏包，没有之一

```
library(ggplot2)    # install.packages("ggplot2")
# or
library(tidyverse) # install.packages("tidyverse")
```

ggplot2 的图形语法

ggplot2 有一套优雅的绘图语法 (grammar of graphics)



Hadley Wickham 将这套语法诠释为：

一张统计图形就是从数据到几何对象 (geometric object, 缩写 geom) 的图形属性 (aesthetic attribute, 缩写 aes) 的一个映射。

ggplot2 的图形语法

ggplot() 函数包括 9 个部件：

- 数据 (data)
- 映射 (mapping)
- 几何对象 (geom)
- 统计变换 (stats)
- 标度 (scale)
- 坐标系 (coord)
- 分面 (facet)
- 主题 (theme)
- 存储和输出 (output)

其中前三个是必需的。

语法模板

```
ggplot(data = <DATA>, mapping = aes(<MAPPINGS>)) +  
  <GEOM_FUNCTION> (  
    stat = <STAT>,  
    position = <POSITION>  
  ) +  
  <COORDINATE_FUNCTION> +  
  <FACET_FUNCTION> +  
  <SCALE_FUNCTION> +  
  <THEME_FUNCTION>
```

案例

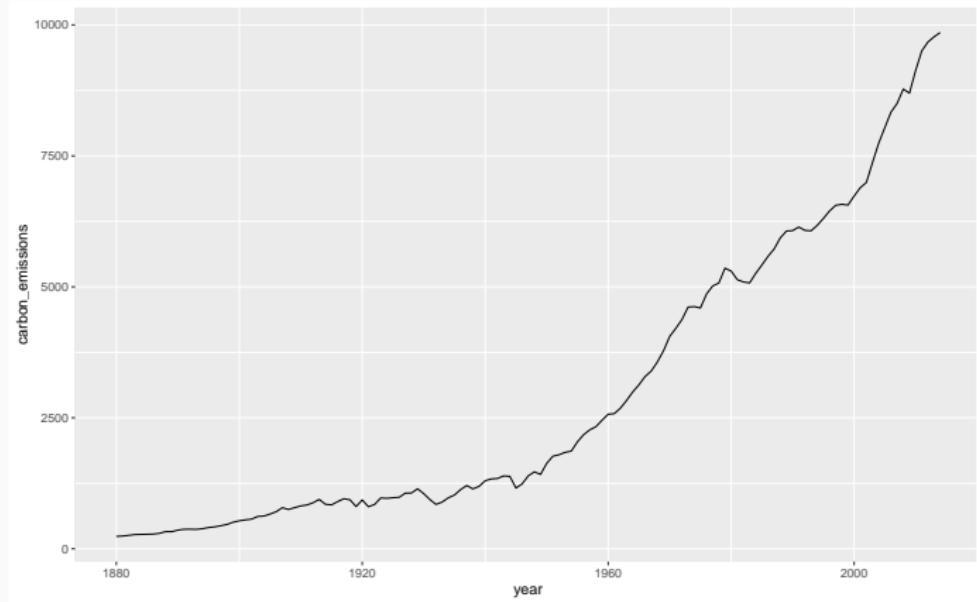
简单的案例（1880-2014 年温度变化和二氧化碳排放量）

```
d <- readr::read_csv("./demo_data/temp_carbon.csv")
```

year	temp_anomaly	land_anomaly	ocean_anomaly	carbon_emissions
1880	-0.11	-0.48	-0.01	236
1881	-0.08	-0.40	0.01	243
1882	-0.10	-0.48	0.00	256
1883	-0.18	-0.66	-0.04	272
1884	-0.26	-0.69	-0.14	275
1885	-0.25	-0.56	-0.17	277
1886	-0.24	-0.51	-0.17	281
1887	-0.28	-0.47	-0.23	295
1888	-0.13	-0.41	-0.05	327
1889	-0.09	-0.31	-0.02	327

是不是很简单？

```
ggplot(data = d, mapping = aes(x = year, y = carbon_emissions))  
  geom_line()
```



ggplot2 语法详解

演示数据

我们用 ggplot2 宏包内置的燃油经济性数据[mpg](#)演示

序号	变量	含义
1	manufacturer	生产厂家
2	model	类型
3	displ	发动机排量, 升
4	year	生产年份
5	cyl	气缸数量
6	trans	传输类型
7	drv	驱动类型
8	cty	每加仑城市里程
9	hwy	每加仑高速公路英里
10	fl	汽油种类
11	class	类型

排量越大，越耗油吗？

回答这个问题，要用到 mpg 数据集中的三个变量

序号	变量	含义
3	displ	排量
9	hwy	油耗
11	class	汽车类型

```
mpg %>%
  select(displ, hwy, class) %>%
  head(4)

#> # A tibble: 4 x 3
#>   displ    hwy  class
#>   <dbl> <int> <chr>
#> 1 1.8     29  compact
```

映射

为考察发动机排量 (displ) 与每加仑英里数 (hwy) 之间的关联，先绘制这两个变量的散点图，

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

data set

aes()

x variable

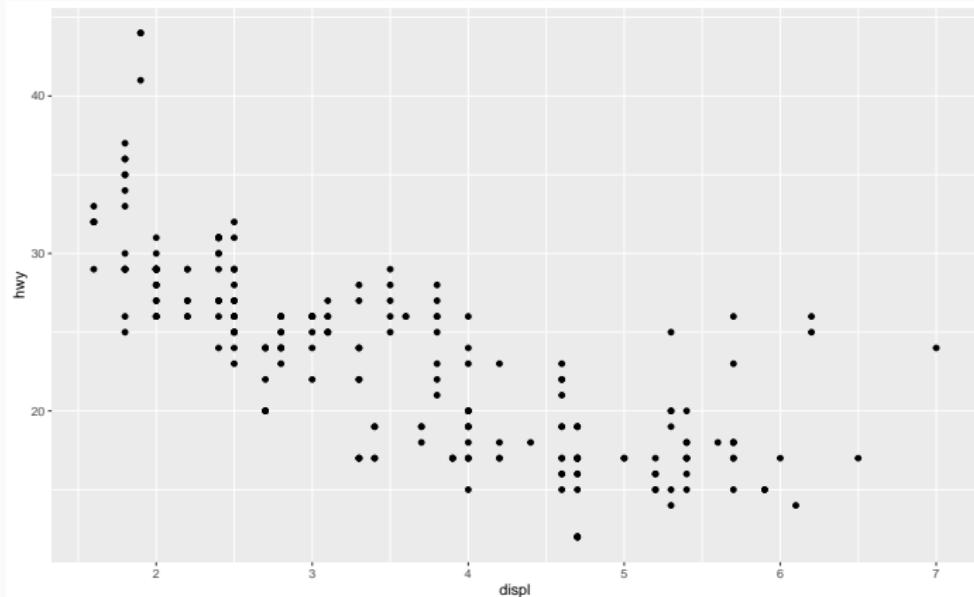
y variable

"type" of layer

运行

运行脚本后生成图片：

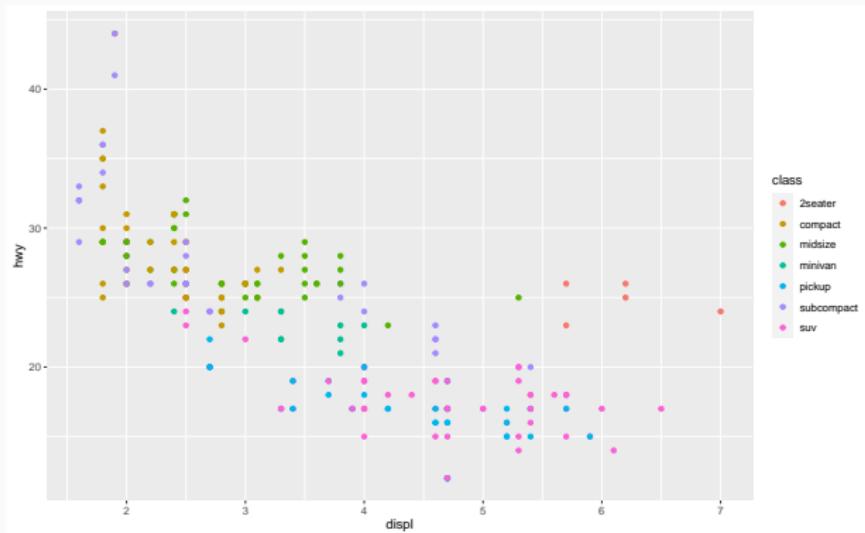
```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +  
  geom_point()
```



颜色映射

除了位置上的映射，ggplot2 还包含了颜色、形状及透明度等图形属性的映射

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



更多映射

大家试试下面代码呢

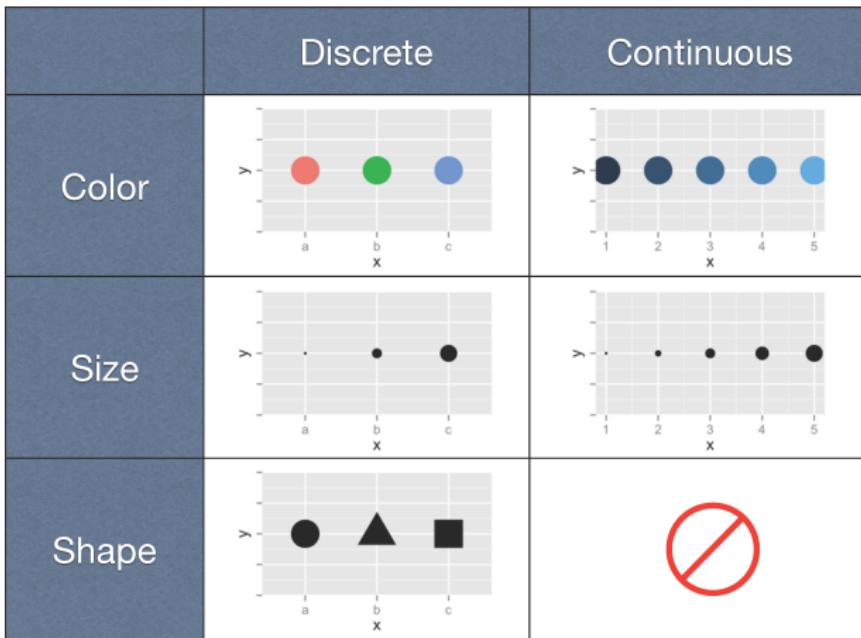
```
ggplot(data = mpg, aes(x = displ, y = hwy, size = class)) +  
  geom_point()
```

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

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

默认值

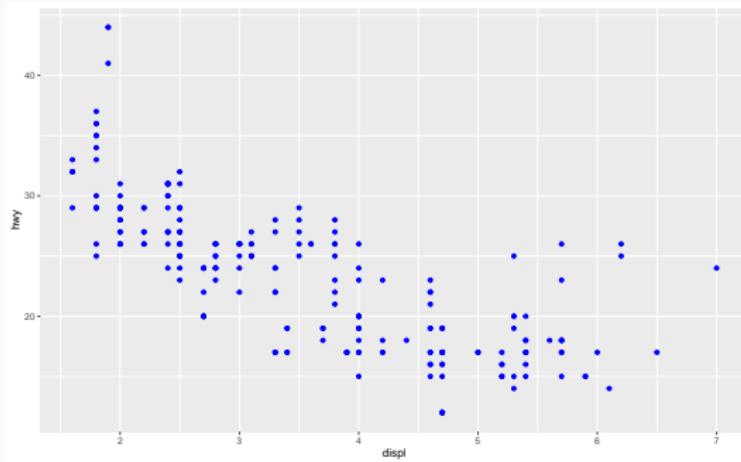
一些默认的设置



映射 vs. 设置

想把图中的点指定为某一种颜色，可以使用设置语句，比如

```
mpg %>%  
  ggplot(aes(displ, hwy)) +  
  geom_point(color = "blue")
```



更多设置

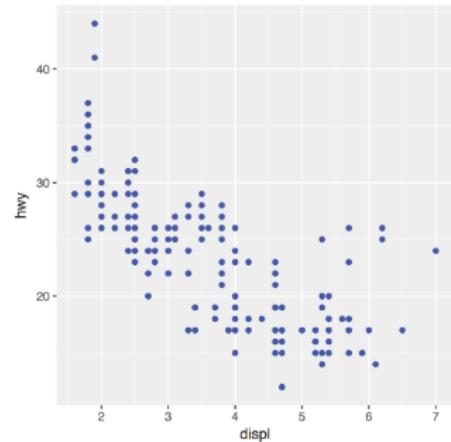
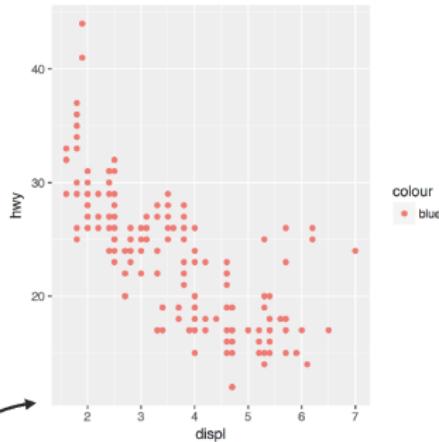
大家也可以试试下面

```
ggplot(mpg, aes(displ, hwy)) + geom_point(size = 5)
```

```
ggplot(mpg, aes(displ, hwy)) + geom_point(shape = 2)
```

```
ggplot(mpg, aes(displ, hwy)) + geom_point(alpha = 0.5)
```

提问



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

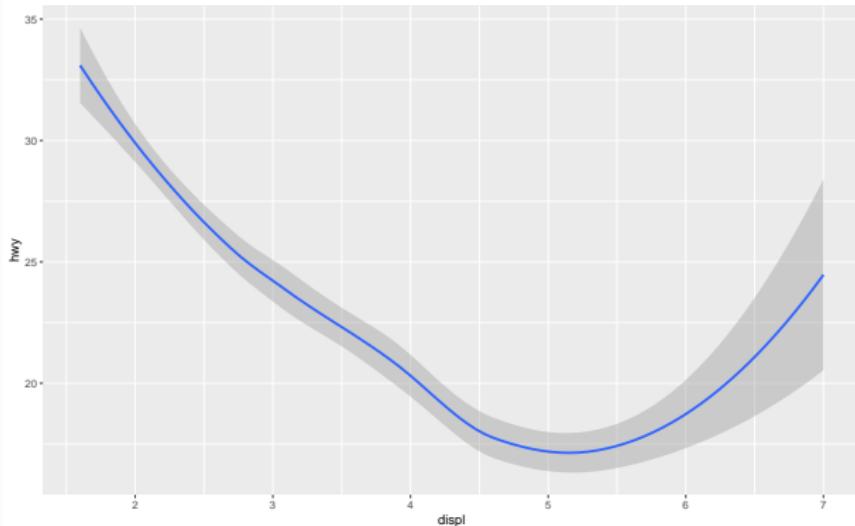
```
ggplot(mpg) + geom_point(aes(x = displ, y = hwy), color = "blue")
```

思考下 `aes(color = "blue")` 为什么会红色的点？

几何对象

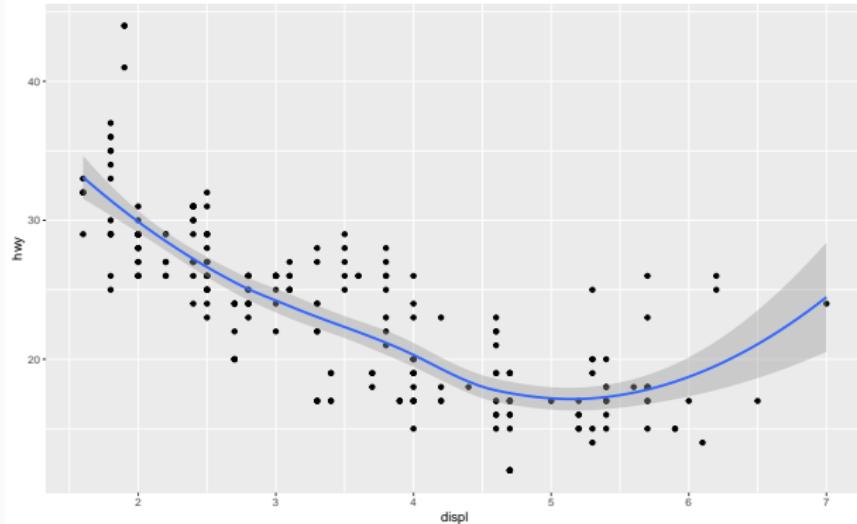
`geom_point()` 可以画散点图，也可以使用 `geom_smooth()` 绘制平滑曲线

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



图层叠加

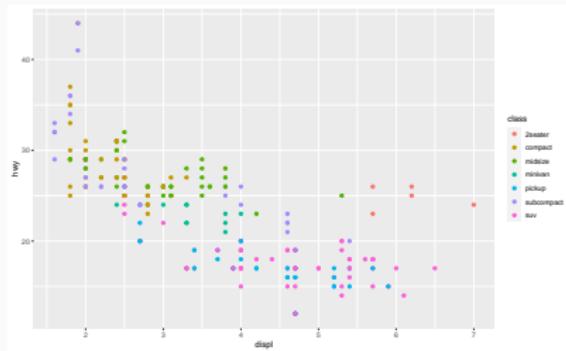
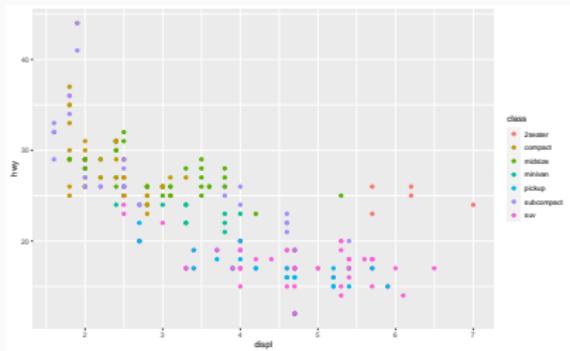
```
ggplot(data = mpg, aes(x = displ, y = hwy)) +  
  geom_point() +  
  geom_smooth()
```



Global vs. Local

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

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



大家可以看到，以上两段代码出来的图是一样，但背后的含义却不同。

Global vs. Local

- 如果映射关系 `aes()` 写在 `ggplot()` 里，那么 `x = displ, y = hwy, color = class` 为全局变量

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

- `geom_point()` 中缺少所绘图所需要的映射关系，就会继承全局变量的映射关系

Global vs. Local

- 如果映射关系 `aes()` 写在几何对象 `geom_point()` 里，就为局部变量。

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

- `geom_point()` 绘图所需要的映射关系已经存在，就不会继承全局变量的映射关系

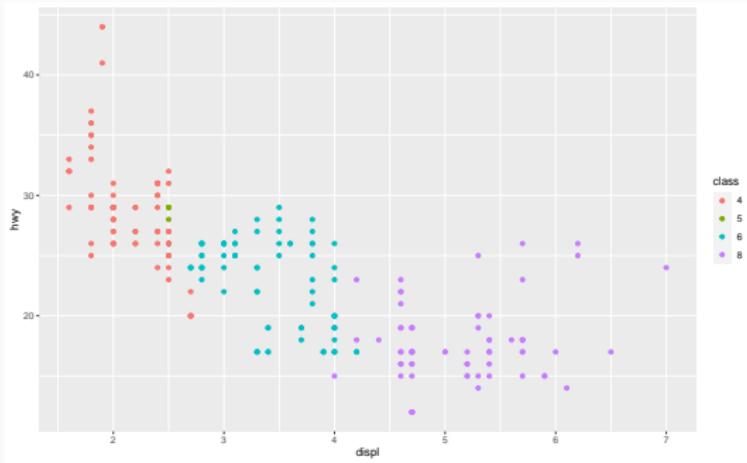
Global vs. Local

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

这里的 `geom_point()` 和 `geom_smooth()` 都会从全局变量中继承映射关系。

Global vs. Local

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



局部变量中的映射关系 `aes(color =)` 已经存在，因此不会从全局变量中继承，沿用当前的映射关系。

提问

大家细细体会下，下面两段代码的区别

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

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

保存图片

可以使用 `ggsave()` 函数，将图片保存为所需要的格式，如 “.pdf”, “.png” 等

```
p <- ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_smooth(method = lm) +
  geom_point(aes(color = class)) +
  ggtitle("This is my first plot")

ggsave(
  filename = "myplot.pdf",
  plot = p,
  width = 8,
  height = 6
)
```