R 语言编程：基于 tidyverse

第 15 讲　ggplot2 绘图 I

张敬信

2022 年 2 月 14 日

哈尔滨商业大学

一. ggplot2 概述

ggplot2 是最流行的 R 可视化包，基于图层化语法：图形是一层一层的图层叠加而成，先进的绘图理念、优雅的语法代码、美观大方的生成图形，一直以来是 R 语言的名片。

1. ggplot2 **绘图语法是从数据产生图形的一系列语法：**
   - 选取整洁**数据**将其**映射**为**几何对象**（如点、线等），几何对象具有**美学特征**（如坐标轴、颜色等），若需要则对数据做**统计变换**，**调整标度**，将结果投影到**坐标系**，再根据喜好选择**主题**。

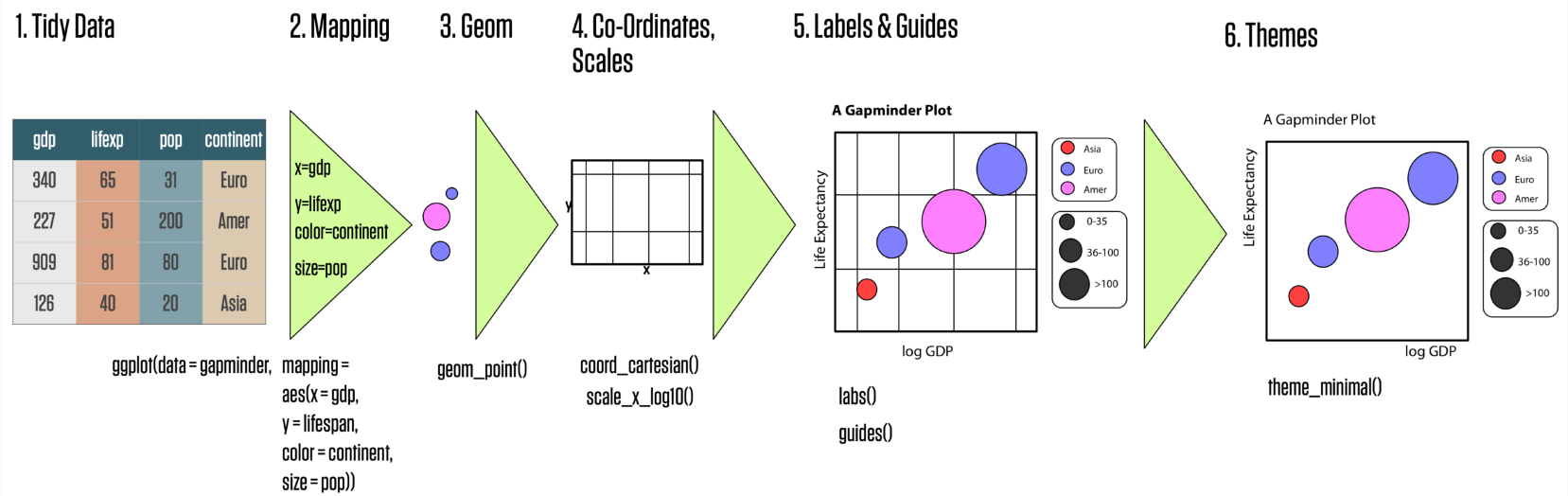


图 1: ggplot2 绘图流程

ggplot 的语法包括 10 个部件：

- **数据** (data)
- **映射** (mapping)
- **几何对象** (geom)
- 标度 (scale)
- 统计变换 (stats)
- 坐标系 (coord)
- 位置调整 (Position adjustments)
- 分面 (facet)
- 主题 (theme)
- 输出 (output)

其中前 3 个是必须的，其他部件 ggplot2 会自动帮你做好它认为”最优”的配置，当然也都可以手动定制。

ggplot2 基本绘图模板：

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

注意：添加图层的加号只能放在行尾，而不能放在下一行行头。

二. 数据、映射、几何对象

1. **数据** (data)
   - 用于绘图的数据，需要是整洁的数据框

```
library(tidyverse)
mpg
#> # A tibble: 234 x 11
#>   manufacturer model displ  year   cyl trans      drv   <i
#>   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <
#> 1 audi         a4      1.8  1999     4 auto(l5)   f
#> 2 audi         a4      1.8  1999     4 manual(m5) f
#> 3 audi         a4      2    2008     4 manual(m6) f
#> # ... with 231 more rows
```

- 用 ggplot() 创建一个坐标系统，先只提供数据，此时只是创建了一个空的图形

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

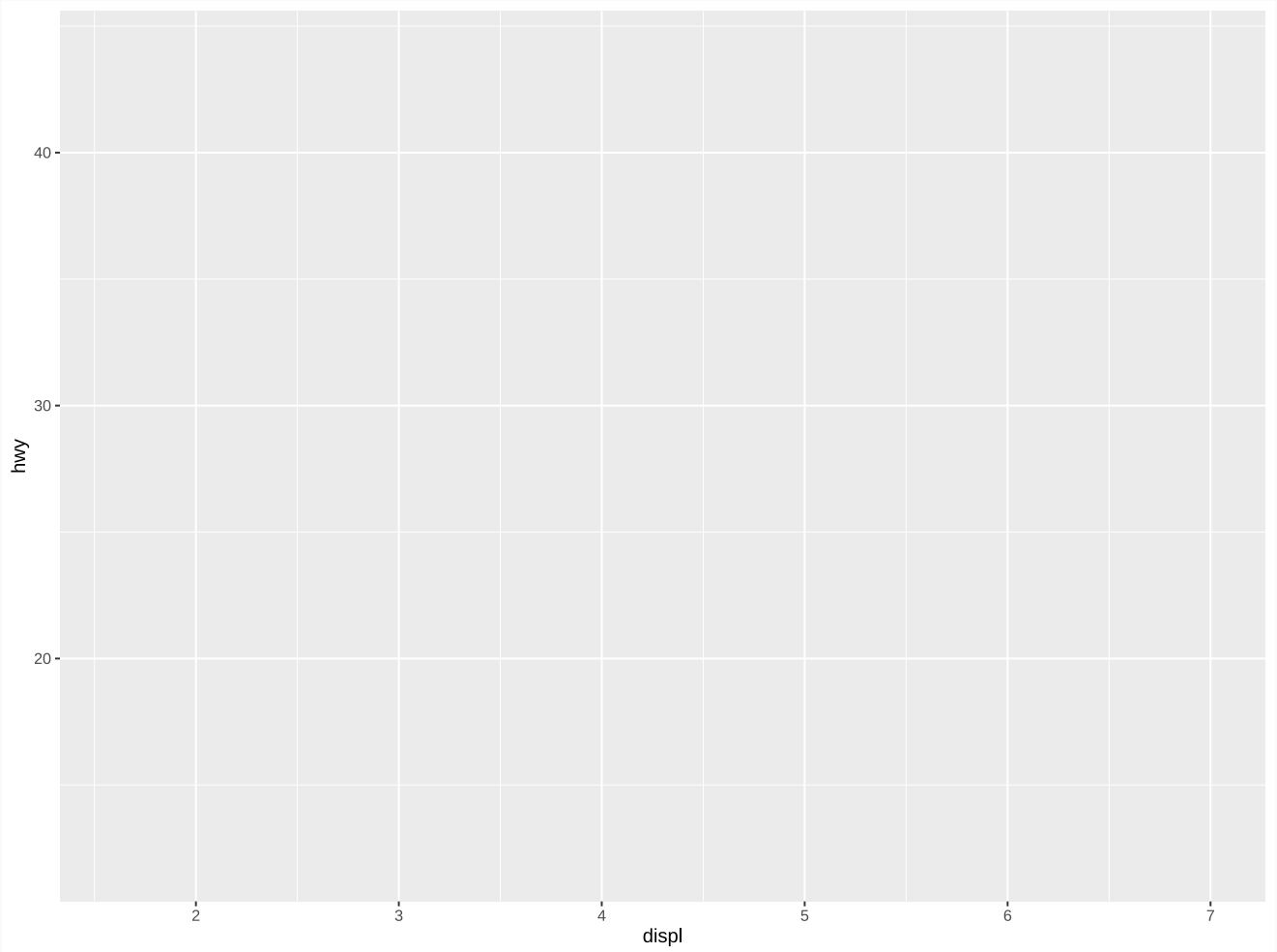
2. **映射** (mapping)

- `aes()`: 美学映射, 就是将数据集中的变量数据映射（关联）到相应的图形属性
- **映射**：指明了变量与图形所见元素之间的联系，告诉 `ggplot` 图形元素想要关联哪个变量数据
- 最常用的映射（美学）有：
    - `x`: x 轴
    - `y`: y 轴
    - `color`: 颜色
    - `size`: 大小
    - `shape`: 形状
    - `fill`: 填充
    - `alpha`: 透明度

- 最需要的美学是 x 和 y, 分别映射到变量 displ 和 hwy, 再将美学 color 映射到 drv[1]，此时图形就有了坐标轴和网格线：

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy,
                                 color = drv))
```

[1] color 美学在绘制几何对象前还体现不出来.


40
30
20
hwy
2
3
4
5
6
7
displ

- 映射不是直接为出现在图形中的颜色、外形、线型等设定特定值，而是建立数据中的变量与可见的图形元素之间的联系
- 经常将图形的美学 `color, size` 等映射到数据集的分类变量，以实现不同分组用不同的美学来区分
- 若要为美学指定特定值，比如 `color = "red"`，是不能放在映射 `aes()` 中的

3. **几何对象** (Geometric)

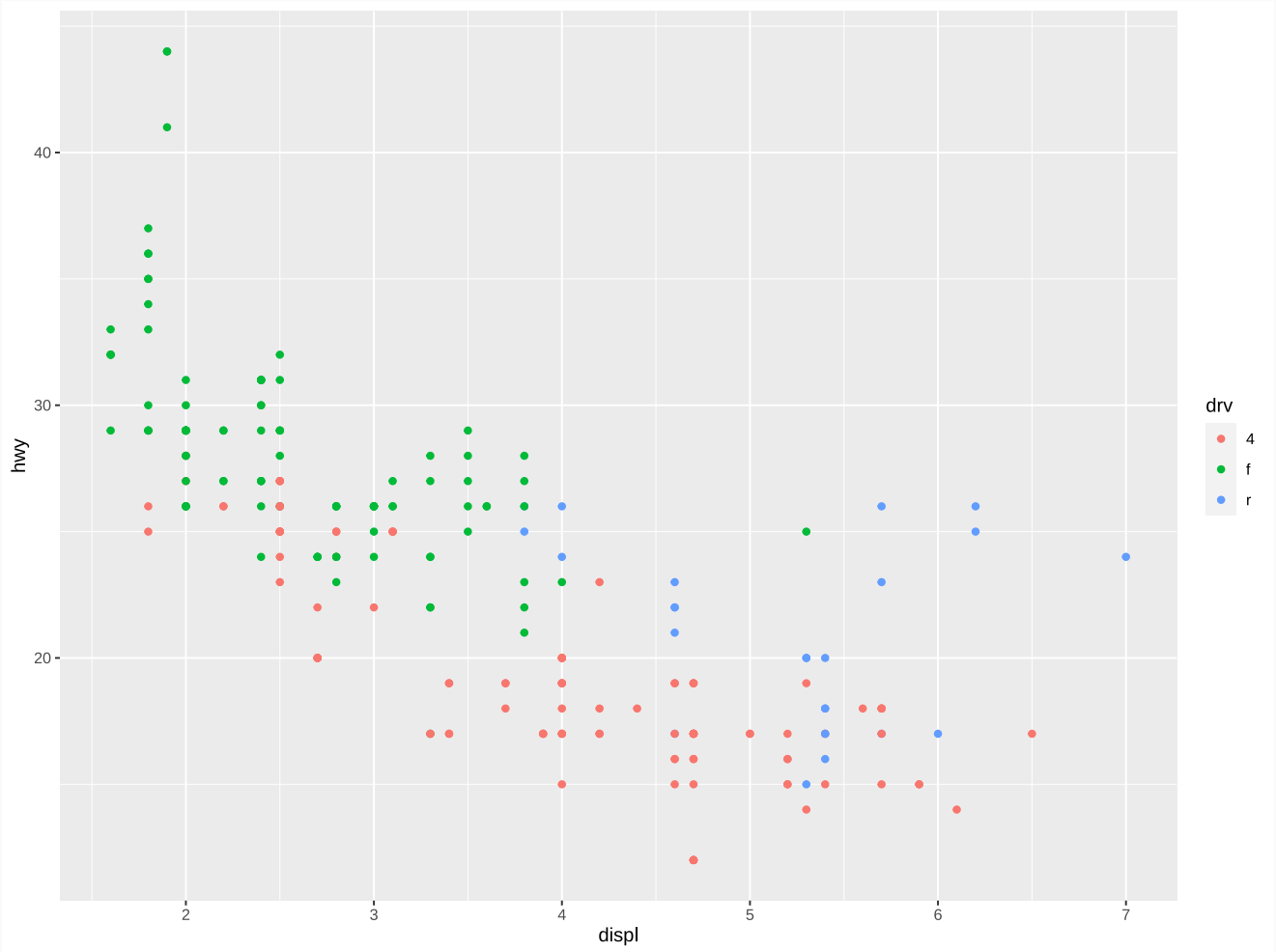
每个图形都是采用不同的视觉对象来表达数据，称为是 “几何对象”。不同类型的” 几何对象” 从不同角度来表达数据，如散点图、平滑曲线、线形图、条形图、箱线图等。

ggplot2 提供了 50 余种 “几何对象”，不同的几何对象支持的美学会有些不同，但均以 geom_xxxx( ) 的方式命名，常用的有：

- geom_point( ): 散点图
- geom_line( ): 折线图
- geom_smooth( ): 光滑（拟合）曲线
- geom_bar( )/geom_col( ): 条形图
- geom_histogram( ): 直方图
- geom_density( ): 概率密度图
- geom_boxplot( ): 箱线图
- geom_abline( ): 参考直线

- 要绘制几何对象，只需添加图层。绘制散点图：

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



- 美学映射也可以放在几何对象中，效果同上：

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

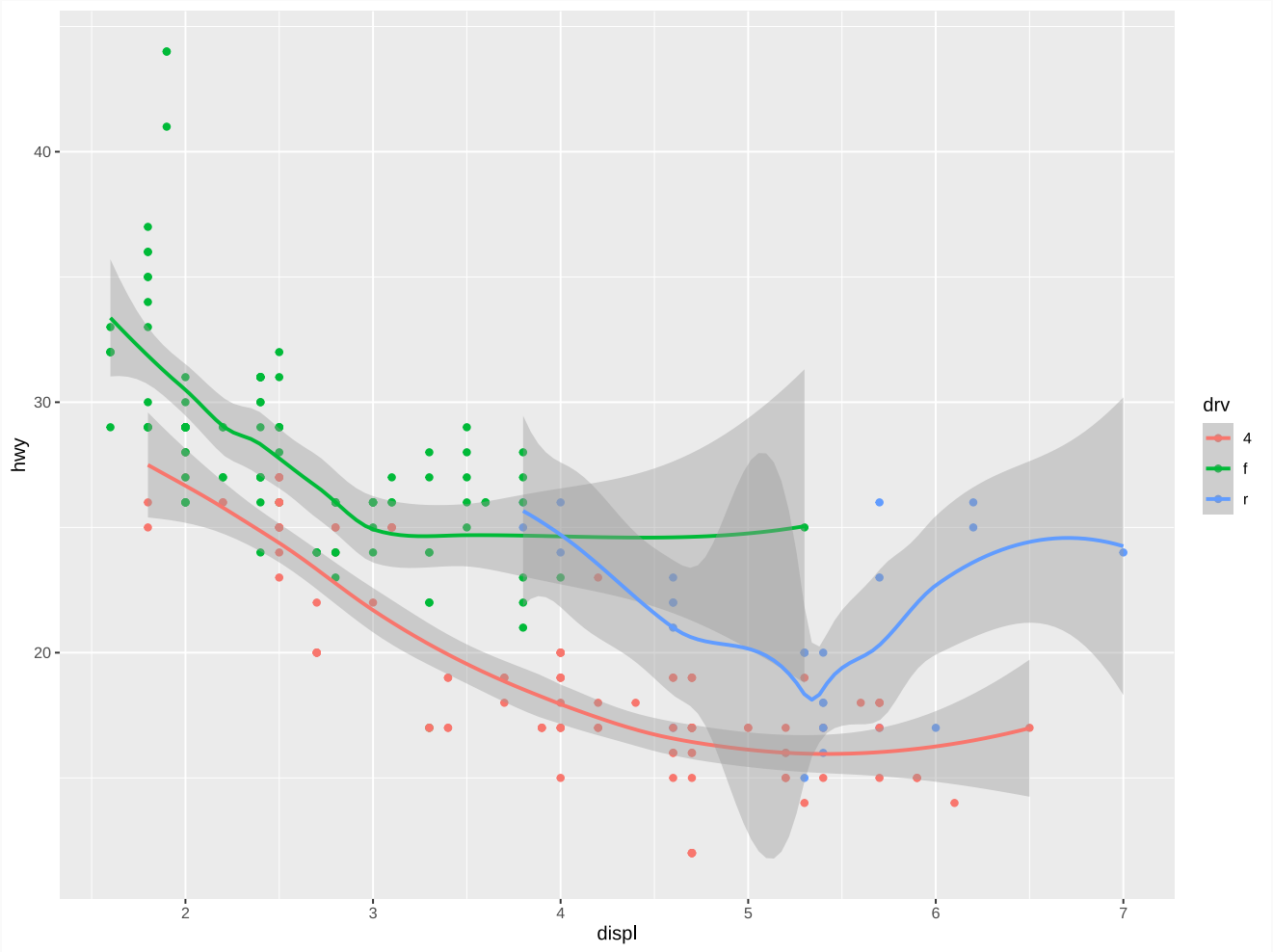
- 为图形美学设置特定值，但注意不能放在映射 aes() 中

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

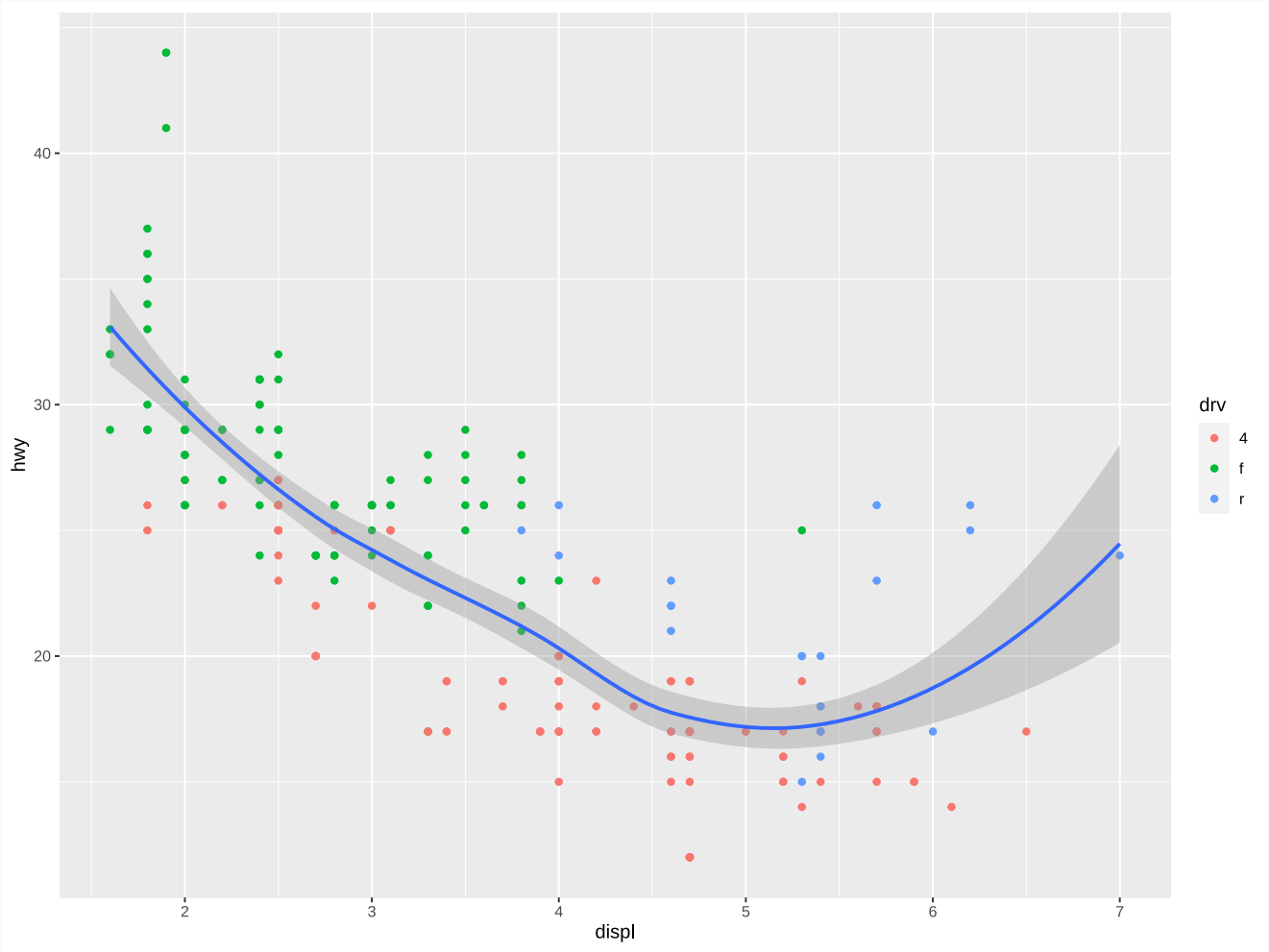

40
30
20
hwy
2
3
4
5
6
7
displ

- 图层是依次叠加的，再添加一个几何对象：光滑曲线，区分两种写法：

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



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



出现这种不同涉及 ggplot2“全局”与”局部”的约定：

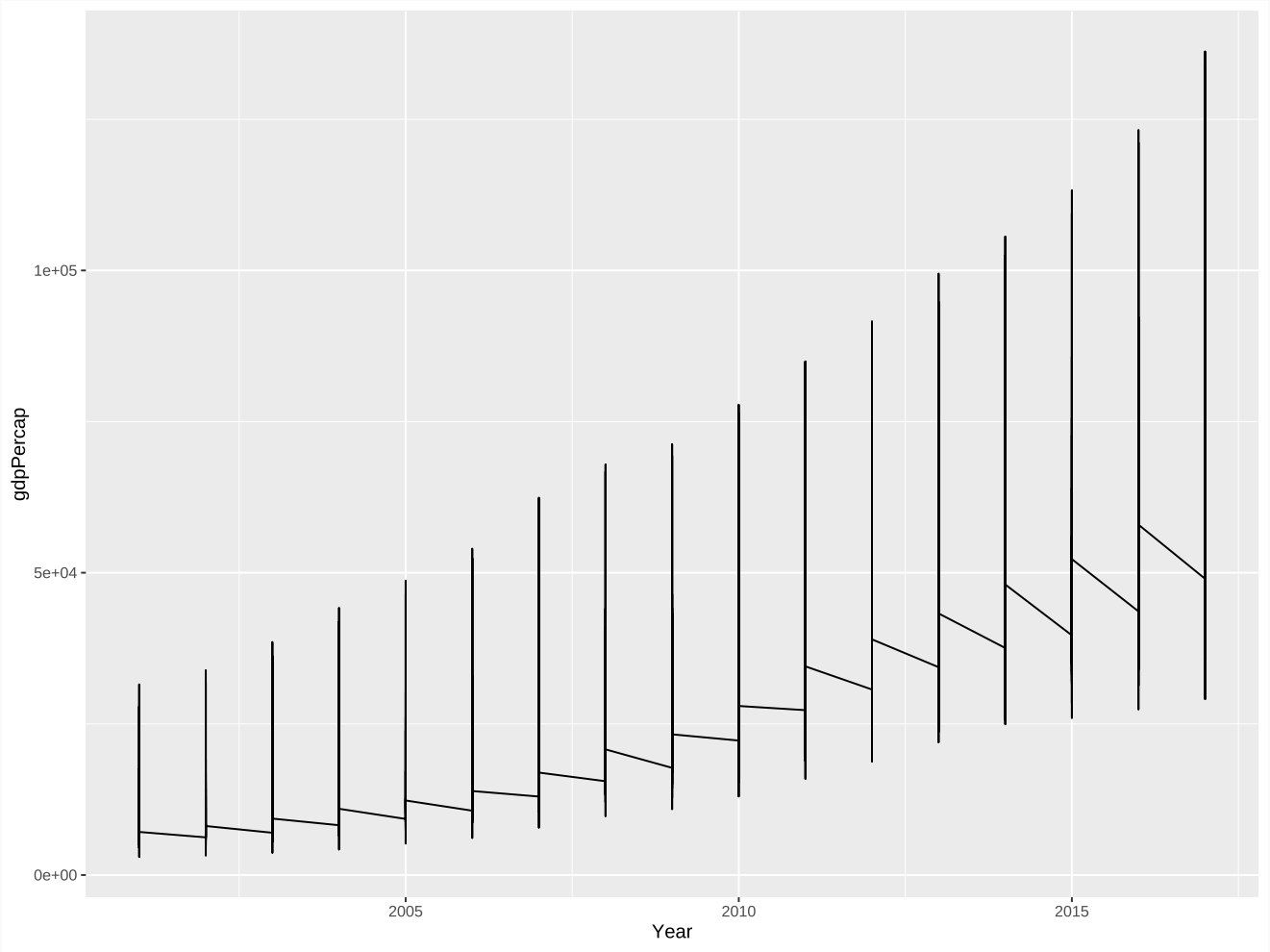
- ggplot()中的数据和映射，是全局的，可供所有几何对象共用；
- 而位于”几何对象”中的数据和映射，是局部的，只供该几何对象使用；
- “几何对象”优先使用局部的，局部没有则用全局的。

关于分组美学 (group)

- 前面用 `aes(color = drv)` 将颜色映射到分类变量 drv, 实际上就是实现了一种分组，对不同 drv 值的数据，按不同颜色分别绘图
- 来看另一种情况，针对分省数据绘制人均 GDP 与年份之间的折线图，若不区分省份，每个年份都对应 31 个省份人均 GDP 值

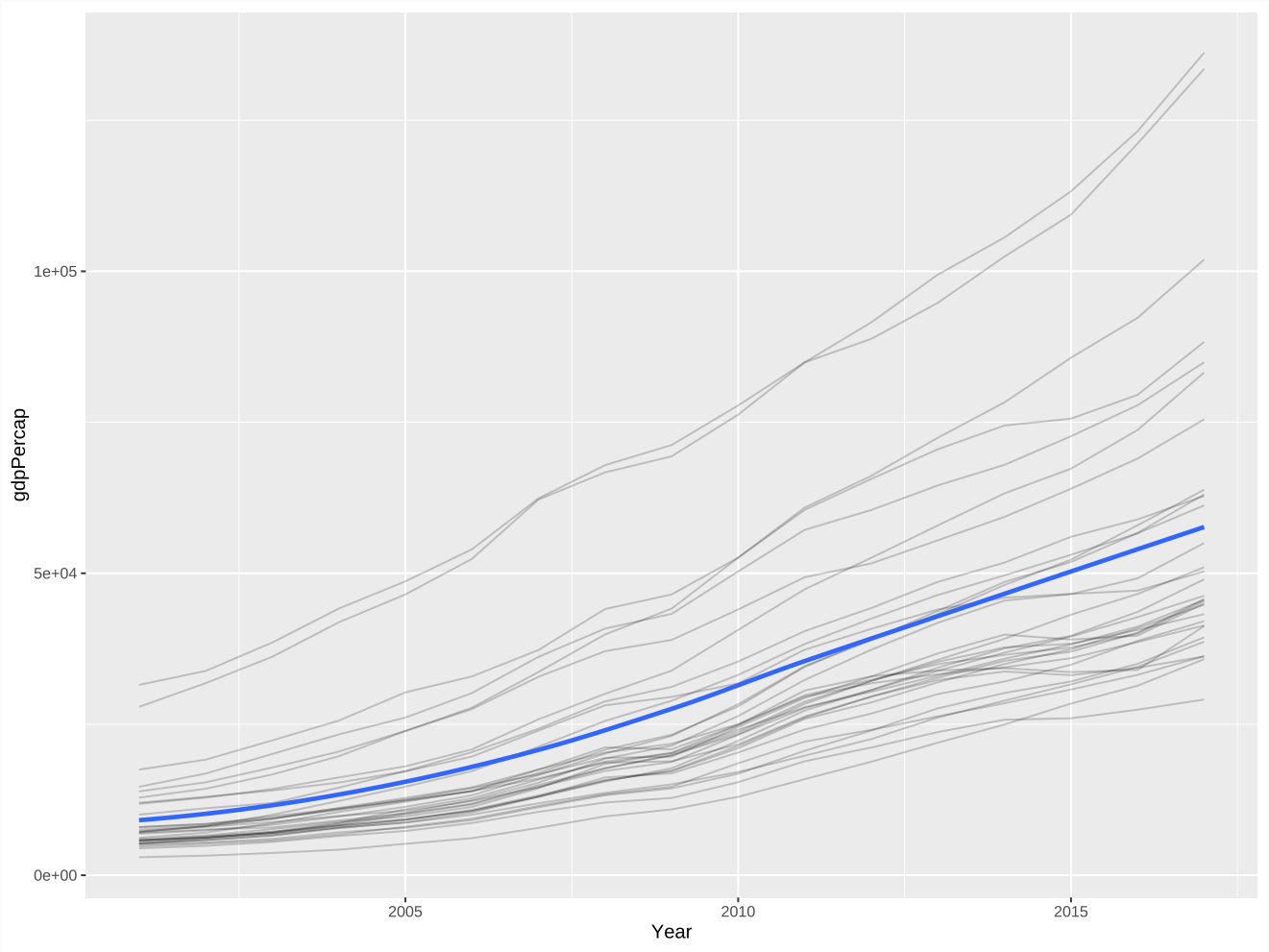
```
load("datas/ecostats.rda")
ecostats
#> # A tibble: 527 x 7
#>    Region  Year Electricity Investment Consumption Populati
#>    <chr>  <int>       <dbl>      <dbl>       <dbl>      <db
#> 1 安徽    2001        360.       893.        2739       61
#> 2 北京    2001        400.      1513.        9057       13
#> 3 福建    2001        439.      1173.        4770       34
#> # ... with 524 more rows
```

```
ggplot(ecostats, aes(Year, gdpPercap)) +
  geom_line()
```



- 应该区分不同省份，这就需要显式地映射分组美学

```
ggplot(ecostats, aes(Year, gdpPercap)) +
  geom_line(aes(group = Region), alpha = 0.2) +
  geom_smooth(se = FALSE, size = 1.2)
```



三. 标度

- 通常 ggplot2 会自动根据输入变量选择最优的坐标刻度方案，若要手动设置或调整，就需要用到标度函数，统一格式为：

```
scale_<MAPPING>_<KIND>()
```

- 标度函数控制几何对象中的标度映射：不只是 x，y 轴，还有 color，fill，shape，size 产生的图例。它们是数据中的连续或分类变量的可视化表示，这需要关联到标度，所以要用到映射

- 常用的标度函数有：
  - scale_*_continuous()：* 为 x 或 y
  - scale_*_discrete()：* 为 x 或 y
  - scale_x_date()
  - scale_x_datetime()
  - scale_*_log10()，scale_*_sqrt()，scale_*_reverse()：* 为 x 或 y
  - scale_*_gradient()，scale_*_gradient2()：* 为 color，fill 等
- scales 包提供了很多现成的设置刻度标签风格的函数

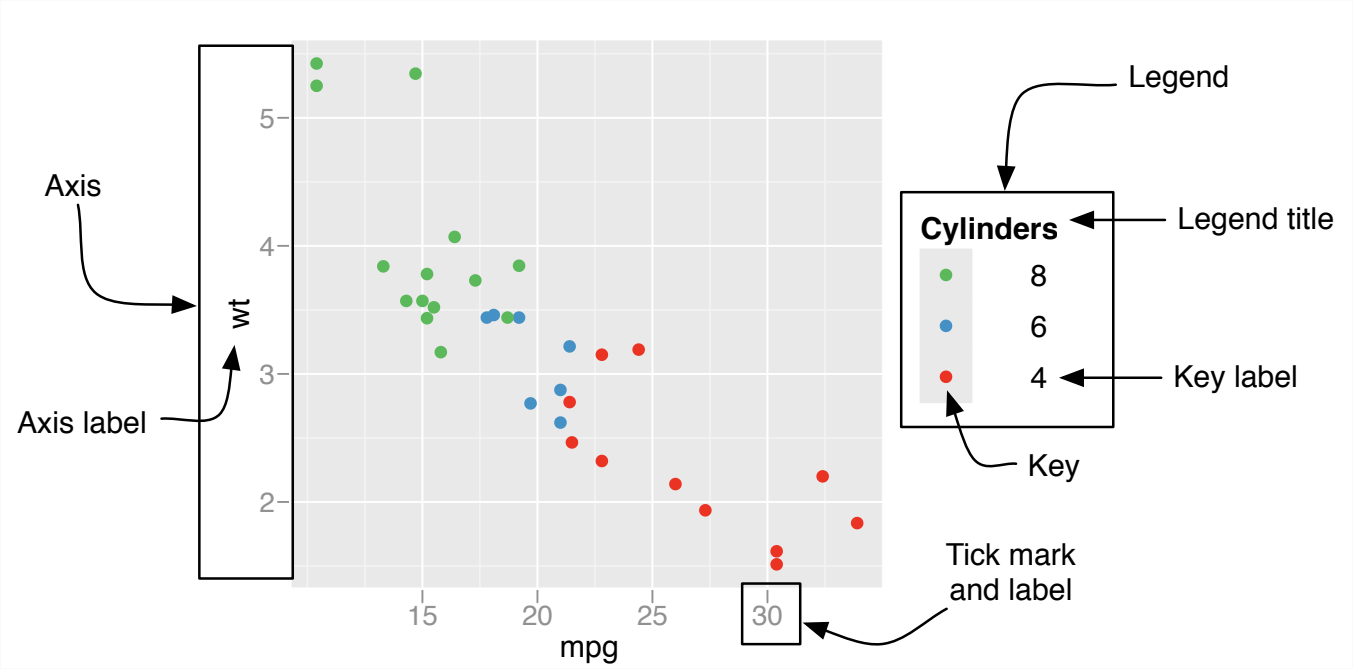


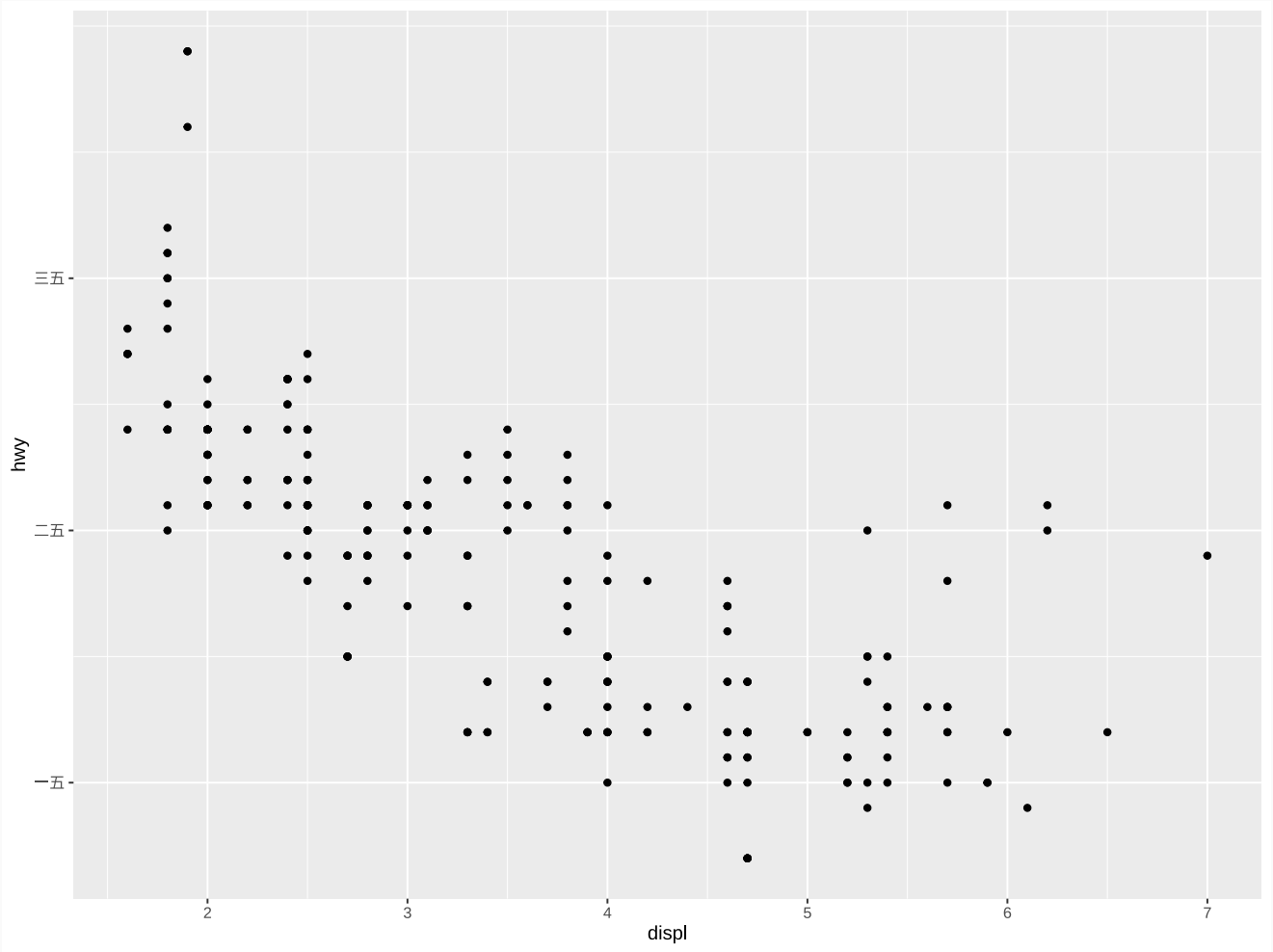
图 2: 图例与坐标轴的组件

1. 修改坐标轴刻度及标签

用 scale_*_continuous() 修改连续变量坐标轴的刻度和标签：

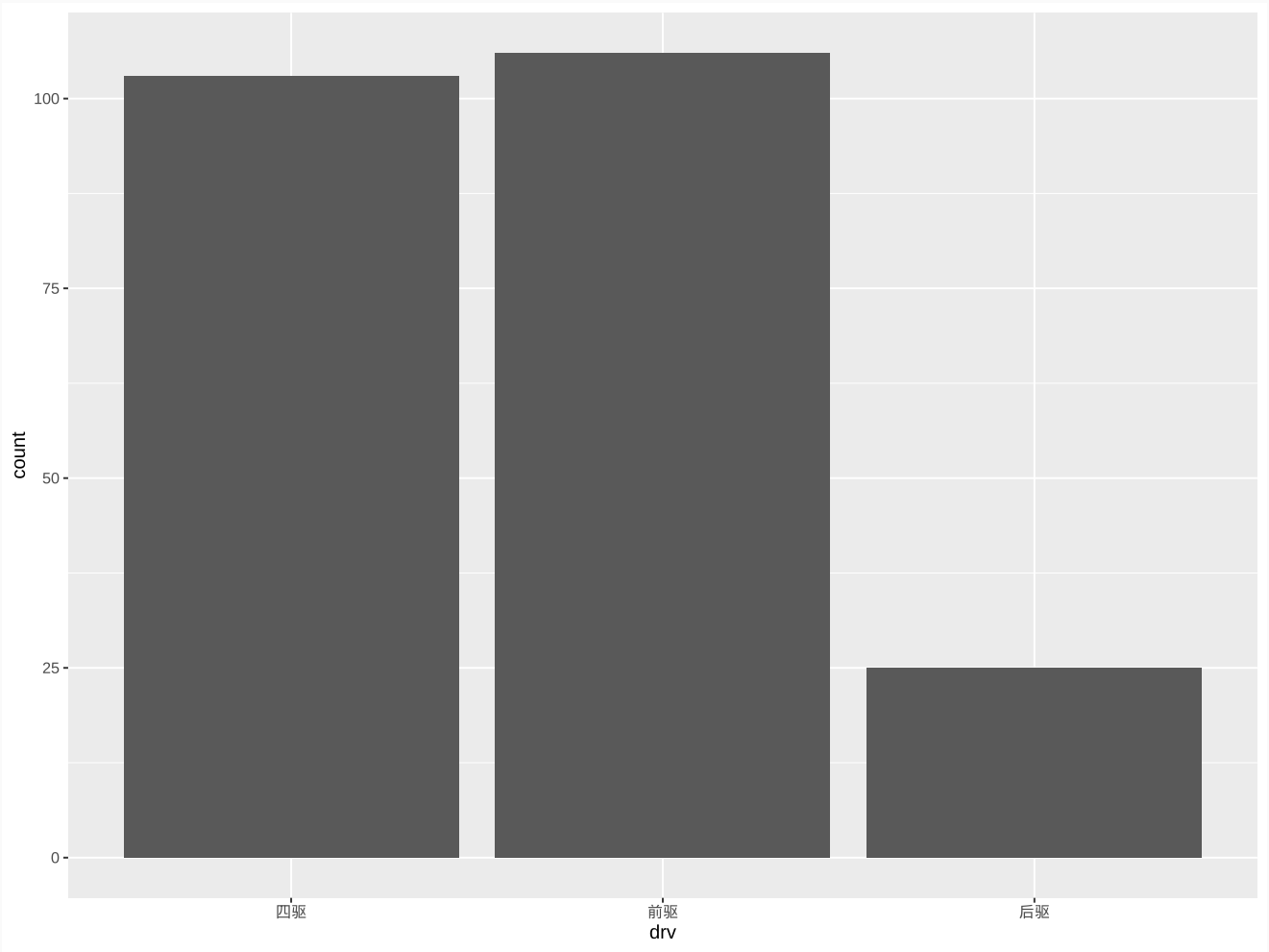
- 参数 breaks 设置各个刻度的位置
- 参数 labels 设置各个刻度对应的标签

```
ggplot(mpg, aes(displ, hwy)) +
  geom_point() +
  scale_y_continuous(breaks = seq(15, 40, by = 10),
                     labels = c(" 一五"," 二五"," 三五"))
```


三五
二五
一五
hwy
2
3
4
5
6
7
displ

用 scale_*_discrete() 修改离散变量坐标轴的标签

```
ggplot(mpg, aes(x = drv)) +
  geom_bar() +      # 条形图
  scale_x_discrete(labels = c("4" = " 四驱", "f" = " 前驱",
                              "r" = " 后驱"))
```

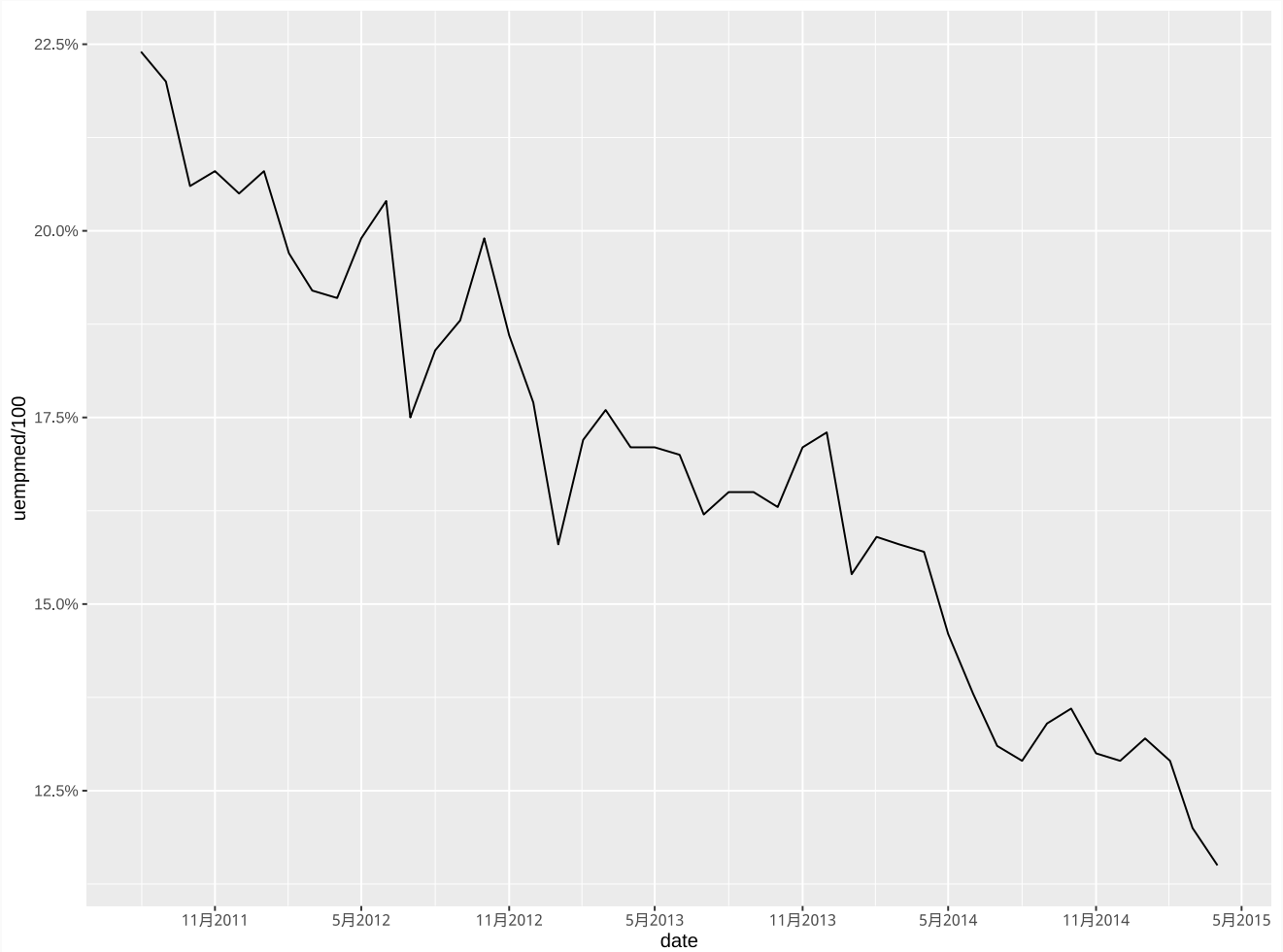

100
75
50
25
0
count
四驱
前驱
后驱
drv

用 scale_x_date() 设置日期刻度，参数：

- date_breaks 设置刻度间隔
- date_labels 设置标签的日期格式
- 借助 scales 包中的函数设置特殊格式，比如百分数 (percent)、科学计数法 (scientific)、美元格式 (dollar) 等

```
economics
#> # A tibble: 574 x 6
#>   date         pce    pop psavert uempmed unemploy
#>   <date>     <dbl>  <dbl>   <dbl>   <dbl>    <dbl>
#> 1 1967-07-01  507. 198712    12.6     4.5     2944
#> 2 1967-08-01  510. 198911    12.6     4.7     2945
#> 3 1967-09-01  516. 199113    11.9     4.6     2958
#> # ... with 571 more rows
```

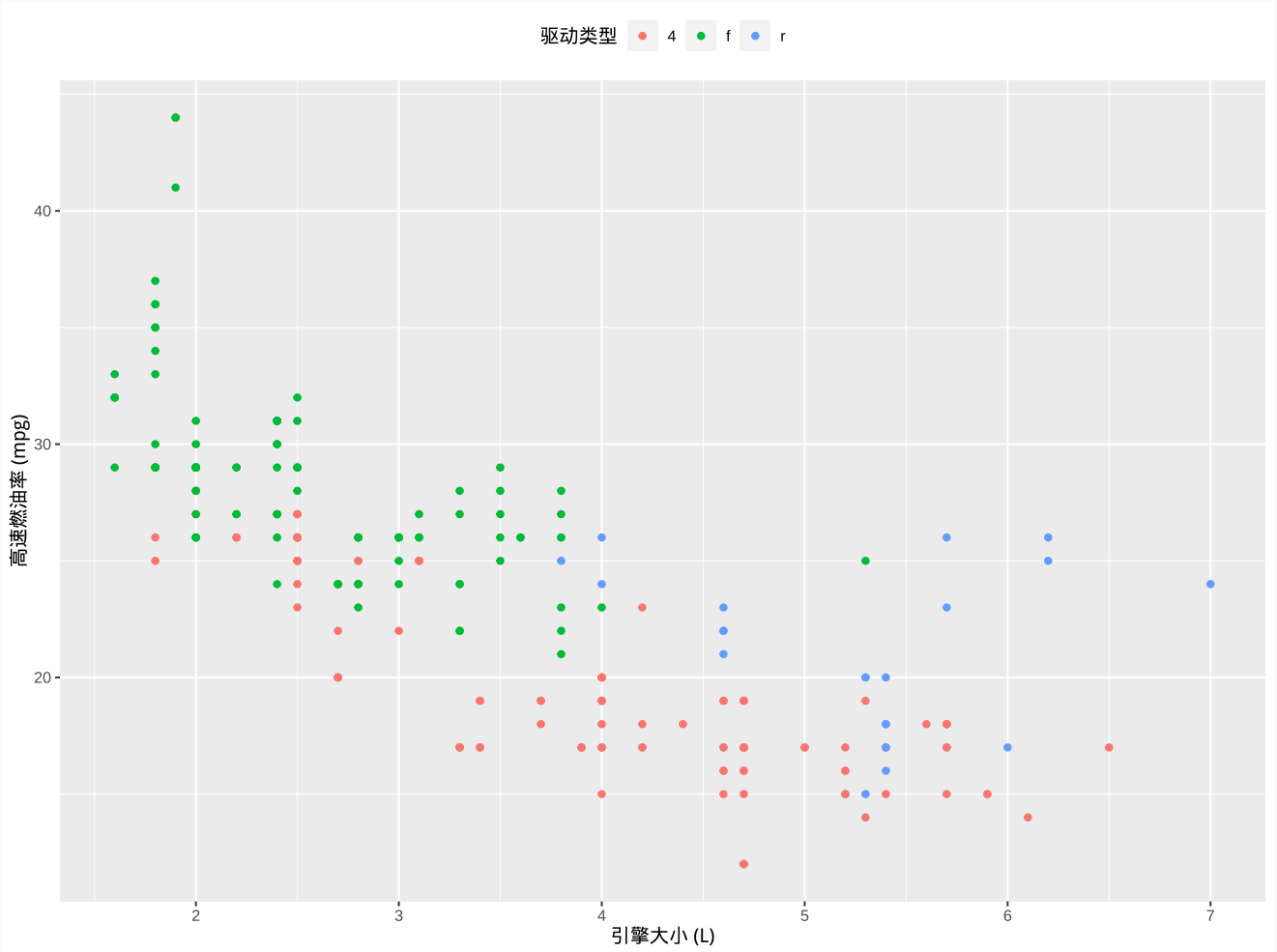
```r
ggplot(tail(economics, 45), aes(date, uempmed / 100)) +
  geom_line() +
  scale_x_date(date_breaks = "6 months", date_labels = "%b%Y"
  scale_y_continuous(labels = scales::percent)
```


22.5%
20.0%
17.5%
15.0%
12.5%
uempmed/100
11月2011
5月2012
11月2012
5月2013
11月2013
5月2014
11月2014
5月2015
date

2. 修改坐标轴标签、图例名及图例位置

- 用 `labs()` 函数的参数 `x, y`, 或者函数 `xlab(), ylab()`, 设置 x 轴、y 轴标签
- 若前面已使用 `color` 美学，则可以在 `labs()` 函数中使用参数 `color` 修改颜色的图例名
- 图例位置是在 `theme` 图层通过参数 `legend.position` 设置，可选取值有"none", "left", "right", "bottom", "top".

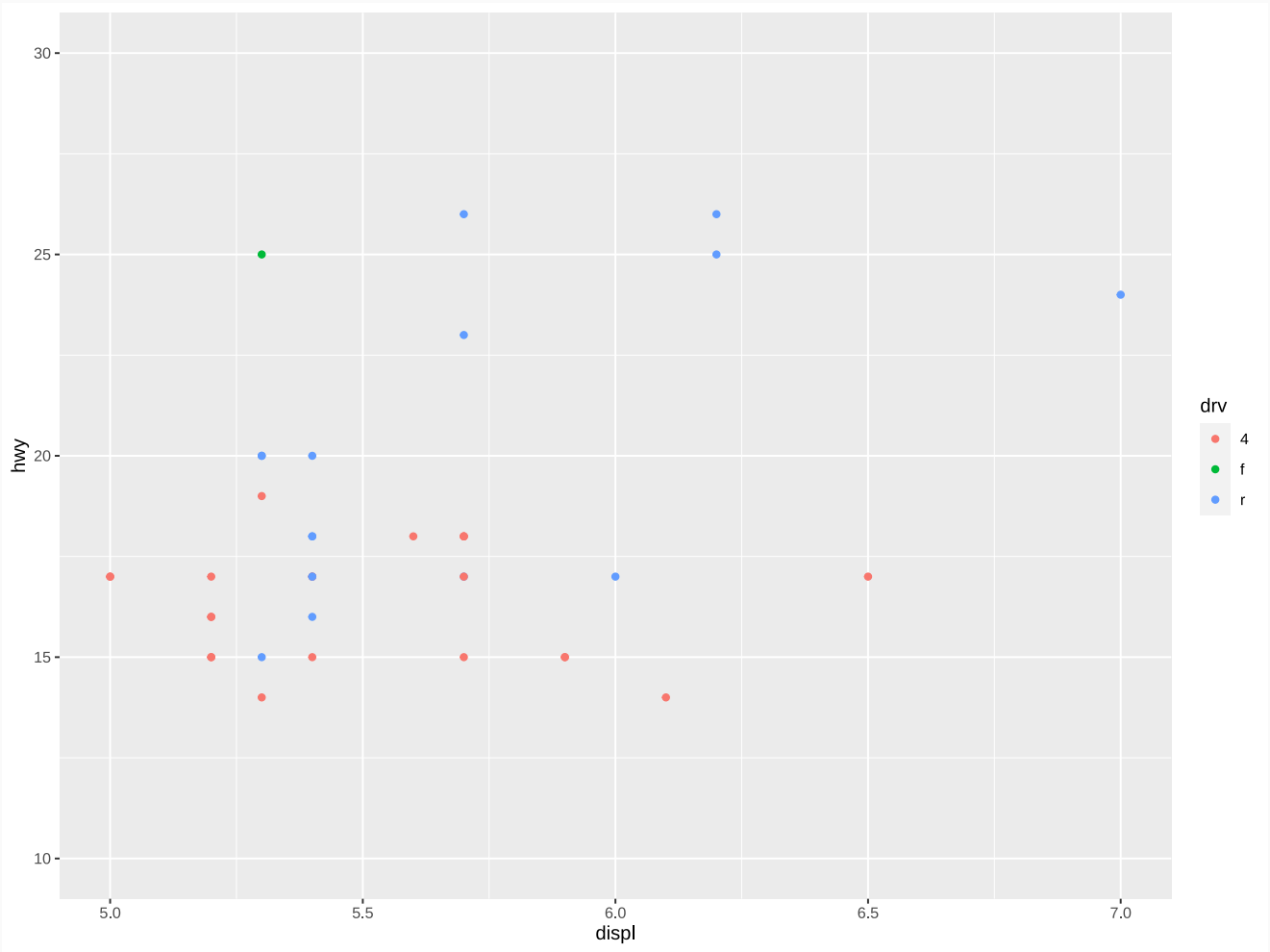
```r
ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(color = drv)) +
  labs(x = " 引擎大小 (L)", y = " 高速燃油率 (mpg)",
       color = " 驱动类型") +       # 或者
  # xlab(" 引擎大小 (L)") + ylab(" 高速燃油率 (mpg)")
  theme(legend.position = "top")
```


驱动类型
4
f
r
高速燃油率 (mpg)
40
30
20
2
3
4
5
6
7
引擎大小 (L)

3. 设置坐标轴范围

用 coord_cartesian() 函数的参数 xlim 和 ylim, 或者用 xlim(), ylim() 函数，设置 x 轴和 y 轴的范围：

```
ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(color = drv)) +
  coord_cartesian(xlim = c(5, 7), ylim = c(10, 30))  # 或者
  # xlim(5, 7) + ylim(10, 30)
```


30
25
20
15
10
hwy
5.0
5.5
6.0
6.5
7.0
displ
drv
4
f
r

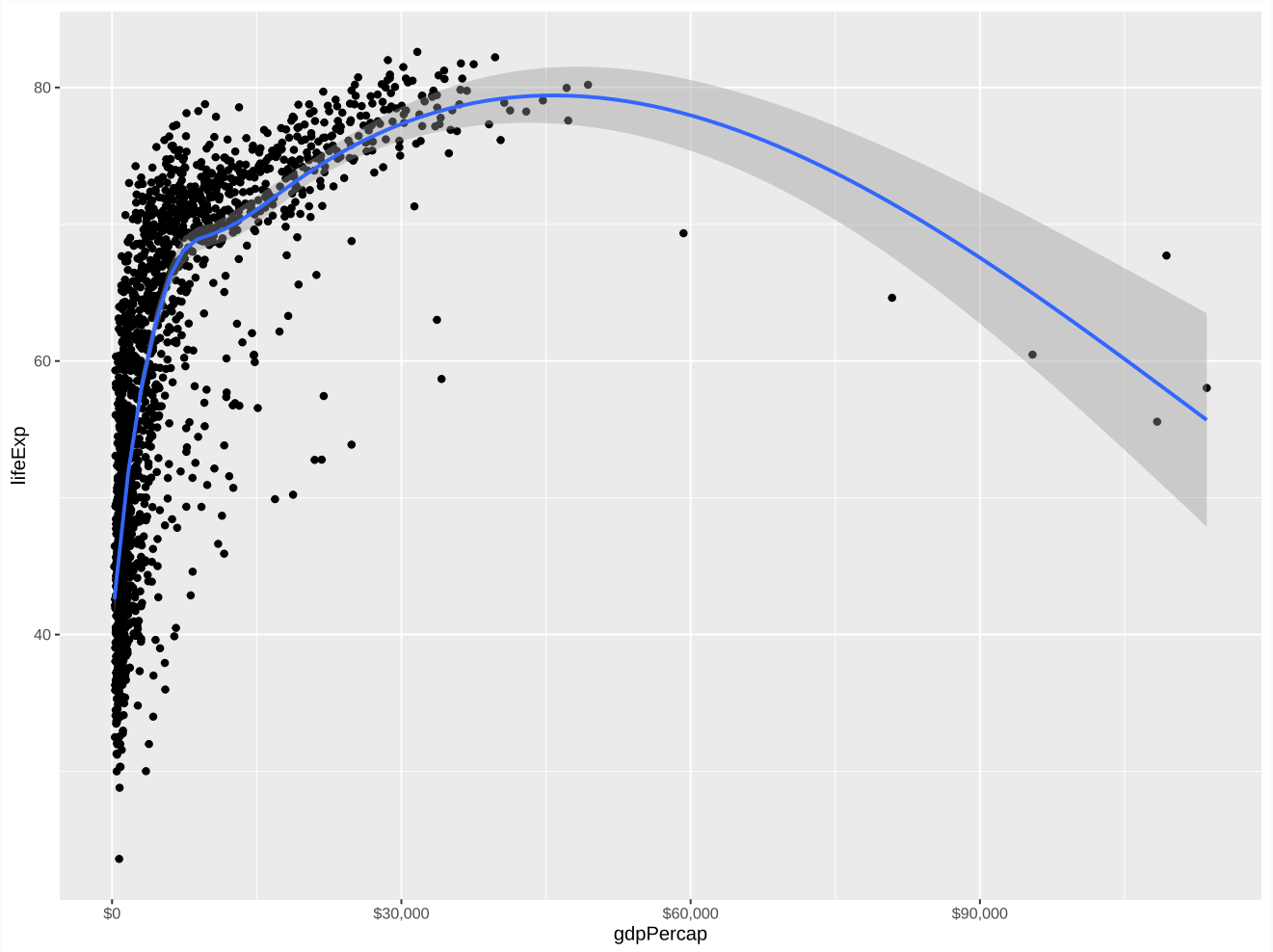
4. 变换坐标轴

变换数据再绘图，比如对数变换，坐标刻度也会变成变换之后的，这使得图形不好理解。

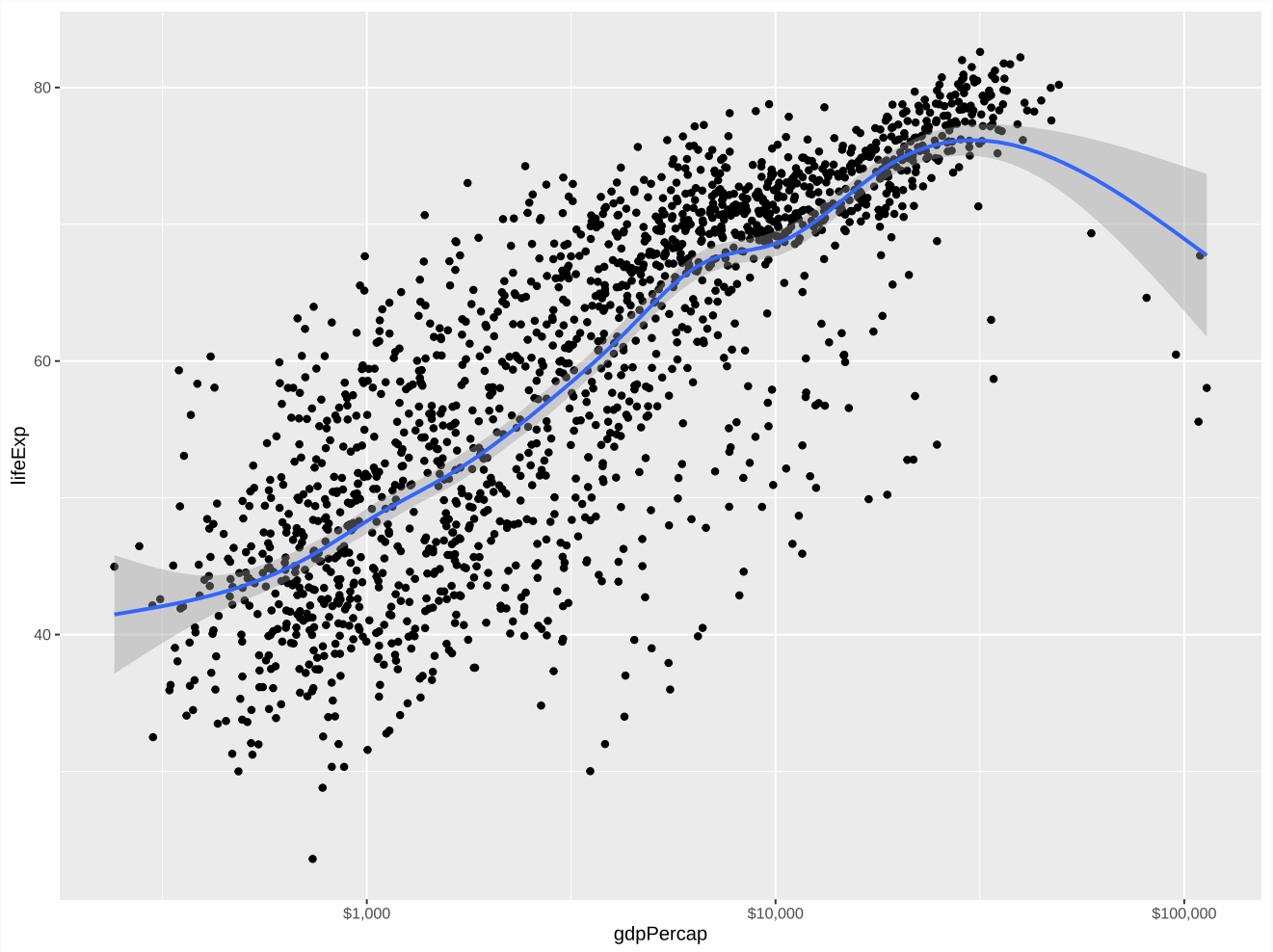
ggplot2 提供的坐标变换函数 scale_x_log10() 等是变换坐标系，能够在视觉效果相同的情况下，使用原始数据的坐标刻度

```
load("datas/gapminder.rda")
p = ggplot(gapminder, aes(gdpPercap, lifeExp)) +
  geom_point() +
  geom_smooth()
```

```
p + scale_x_continuous(labels = scales::dollar)
```



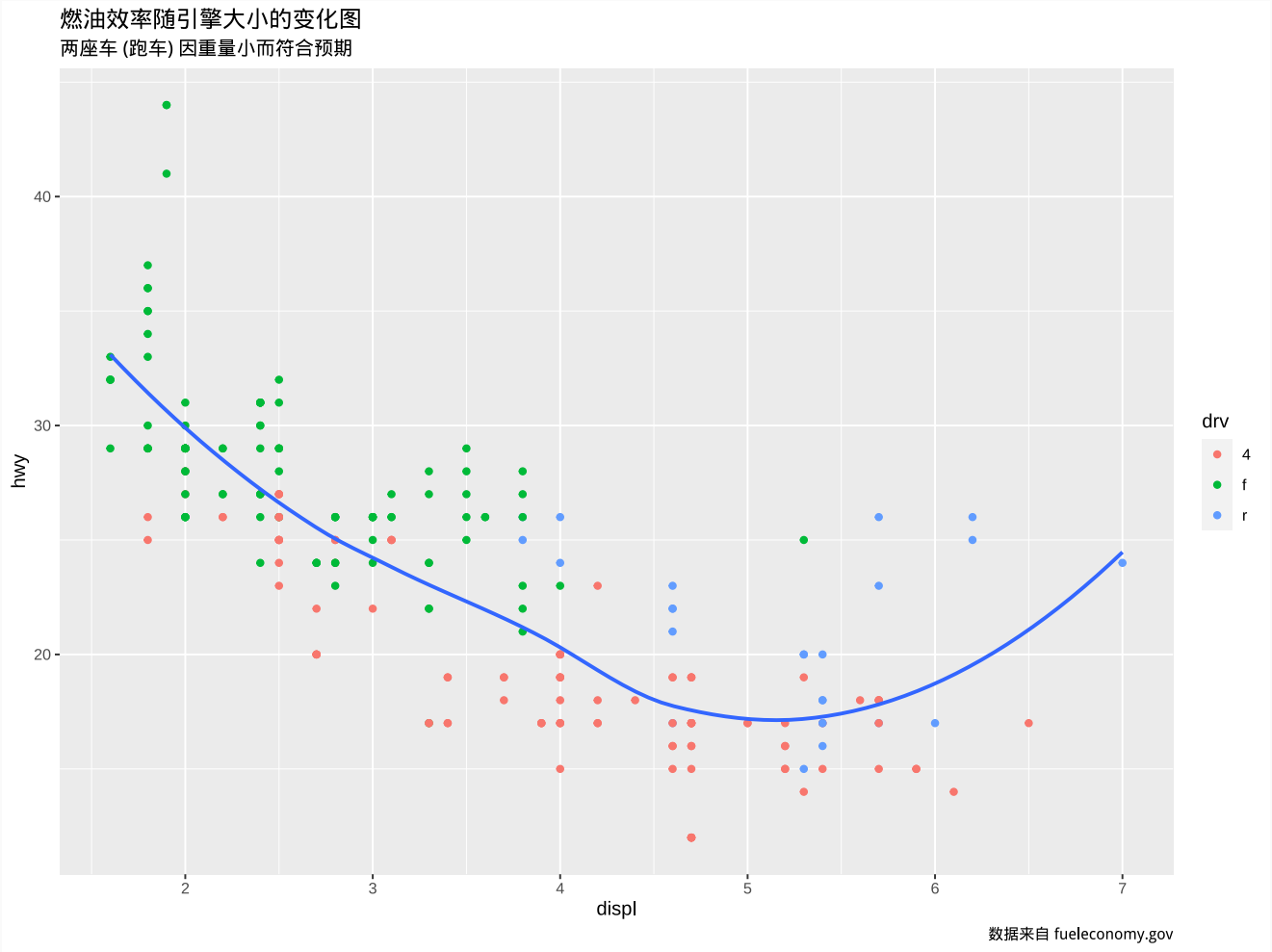
```
p + scale_x_log10(labels = scales::dollar)
```



5. 设置图形标题

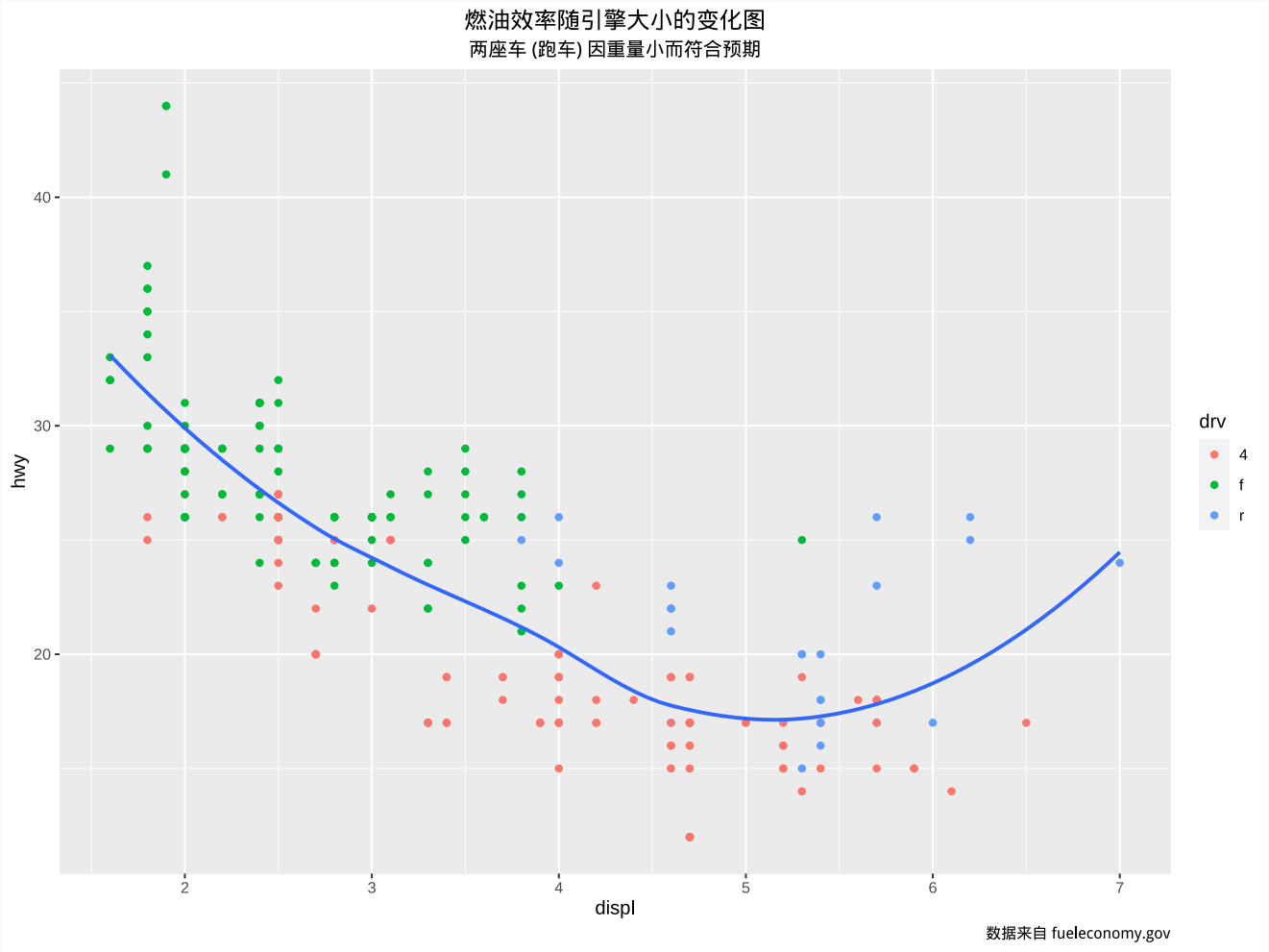
用 labs() 函数的参数 title, subtitle, caption 设置标题、副标题、脚注标题（默认右下角）

```
p = ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(color = drv)) +
  geom_smooth(se = FALSE) +
  labs(title = " 燃油效率随引擎大小的变化图",
       subtitle = " 两座车（跑车）因重量小而符合预期",
       caption = " 数据来自 fueleconomy.gov")
p
```


燃油效率随引擎大小的变化图
两座车 (跑车) 因重量小而符合预期
hwy
40
30
20
2
3
4
5
6
7
displ
drv
4
f
r
数据来自 fueleconomy.gov

图形标题默认居左，若要居中，需要加 theme 图层专门设置

```
p + theme(plot.title = element_text(hjust = 0.5), # 标题居中
          plot.subtitle = element_text(hjust = 0.5))
```



6. 设置 fill, color 颜色

数据的某个维度信息可以通过颜色来展示，颜色直接影响图形的美感。可以直接使用颜色值，但是更建议使用 RColorBrewer（调色板）或 colorspace 包[2]。

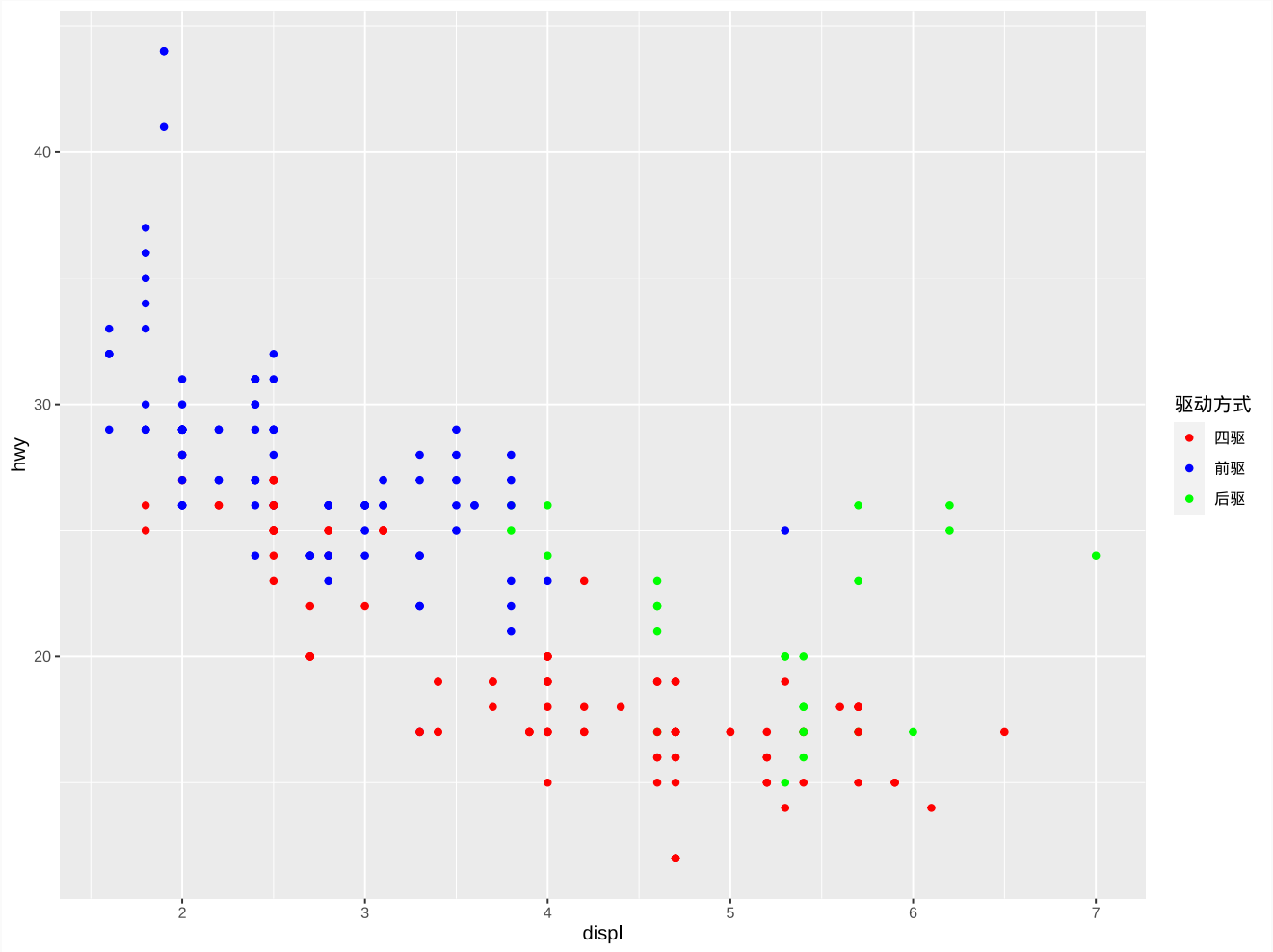
(1) 离散变量

- manual: 直接指定分组使用的颜色
- hue: 通过改变色相 (hue) 饱和度 (chroma) 亮度 (luminosity) 来调整颜色
- brewer: 使用 ColorBrewer 的颜色
- grey: 使用不同程度的灰色

[2]查看所有可用的调色版：RColorBrewer::display.brewer.all(); 查看所有可用的颜色空间：hcl_palettes::hcl_palettes(plot = TRUE).

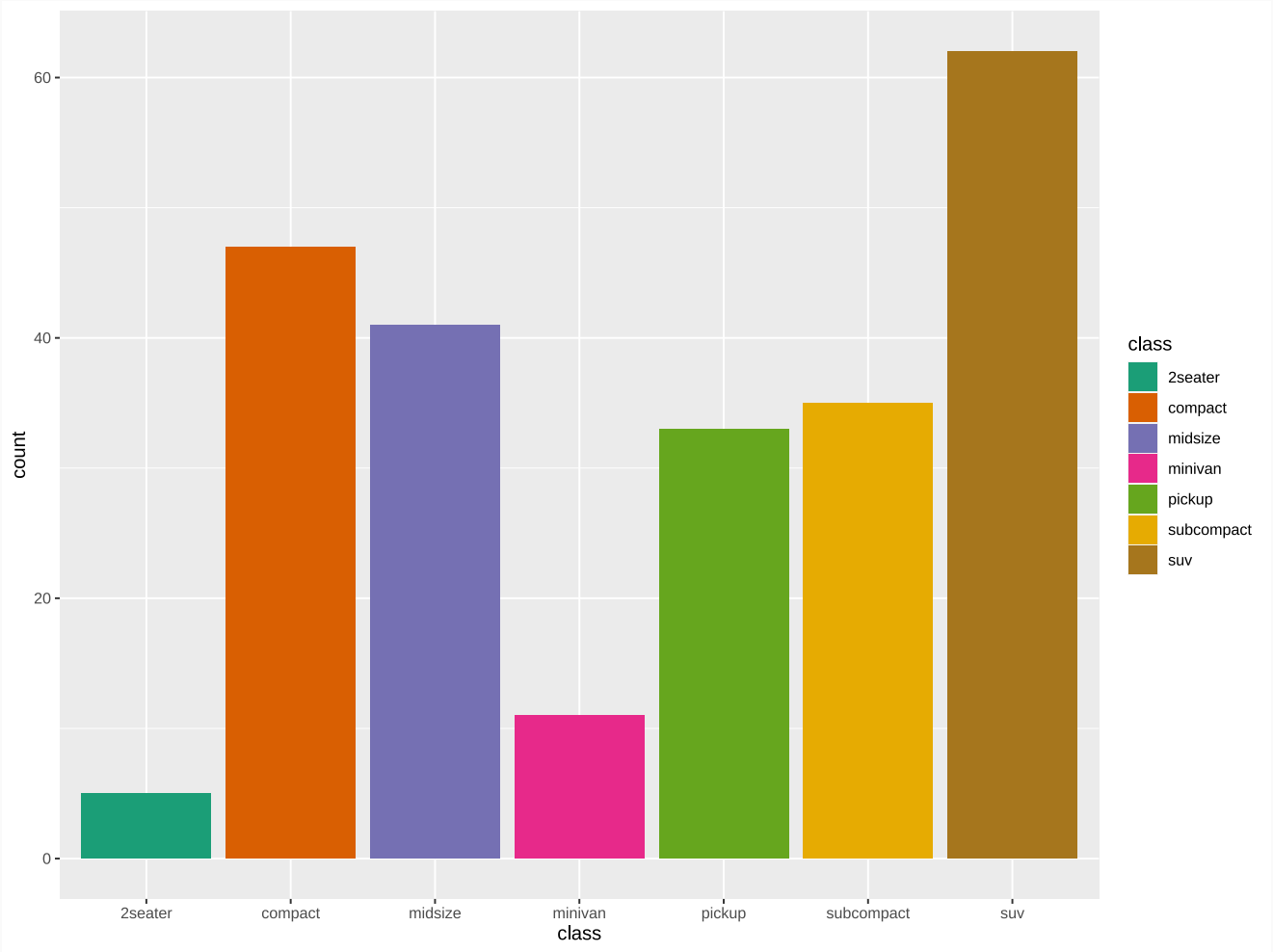
用 scale_*_manual() 手动设置颜色，并修改图例及其标签

```
ggplot(mpg, aes(displ, hwy, color = drv)) +
    geom_point() +
    scale_color_manual(" 驱动方式",          # 修改图例名
                       values = c("red", "blue", "green"),
                       # breaks = c("4", "f", "r"),
                       labels = c(" 四驱", " 前驱", " 后驱"))
```


hwy
40
30
20
2
3
4
5
6
7
displ
驱动方式
四驱
前驱
后驱

用 scale_*_brewer() 调用调色版中的颜色

```
ggplot(mpg, aes(x = class, fill = class)) +
  geom_bar() +
  scale_fill_brewer(palette = "Dark2")  # 使用 Dark2 调色版
```



(2) 连续变量

- gradient: 设置二色渐变色
- gradient2: 设置三色渐变色
- distiller: 使用 ColorBrewer 的颜色
- identity 使用 color 变量对应的颜色，对离散型和连续型都有效

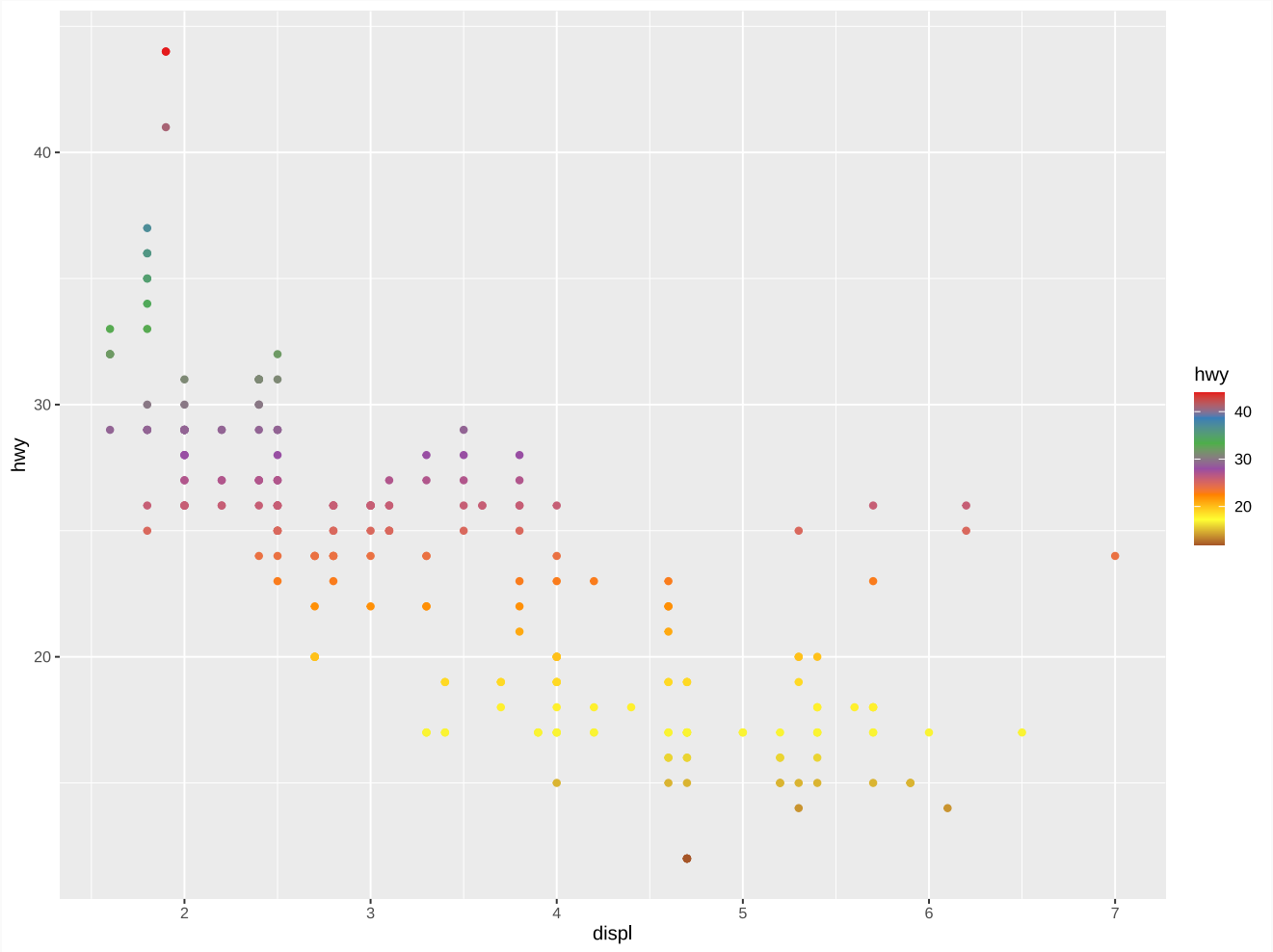
用 scale_color_gradient() 设置二色渐变色

```
ggplot(mpg, aes(displ, hwy, color = hwy)) +
  geom_point() +
  scale_color_gradient(low = "green", high = "red")
```


hwy
40
30
20
2
3
4
5
6
7
displ
hwy
40
30
20

用 scale_*_distiller() 调用调色版中的颜色

```
ggplot(mpg, aes(displ, hwy, color = hwy)) +
  geom_point()  +
  scale_color_distiller(palette = "Set1")
```

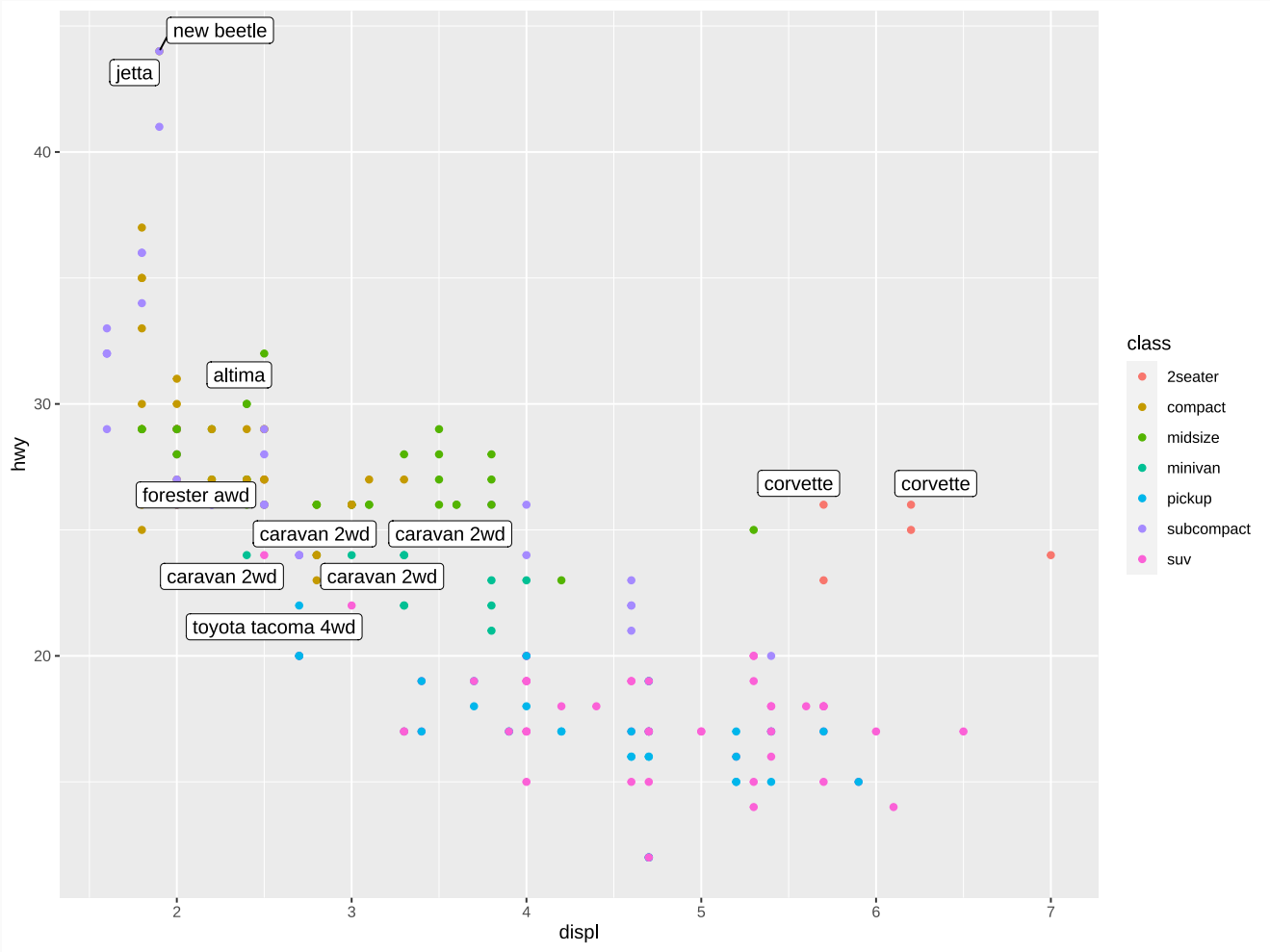


7. 添加文字标注

ggrepel 包提供了 geom_label_repel() 和 geom_text_repel() 函数，为图形添加文字标注。

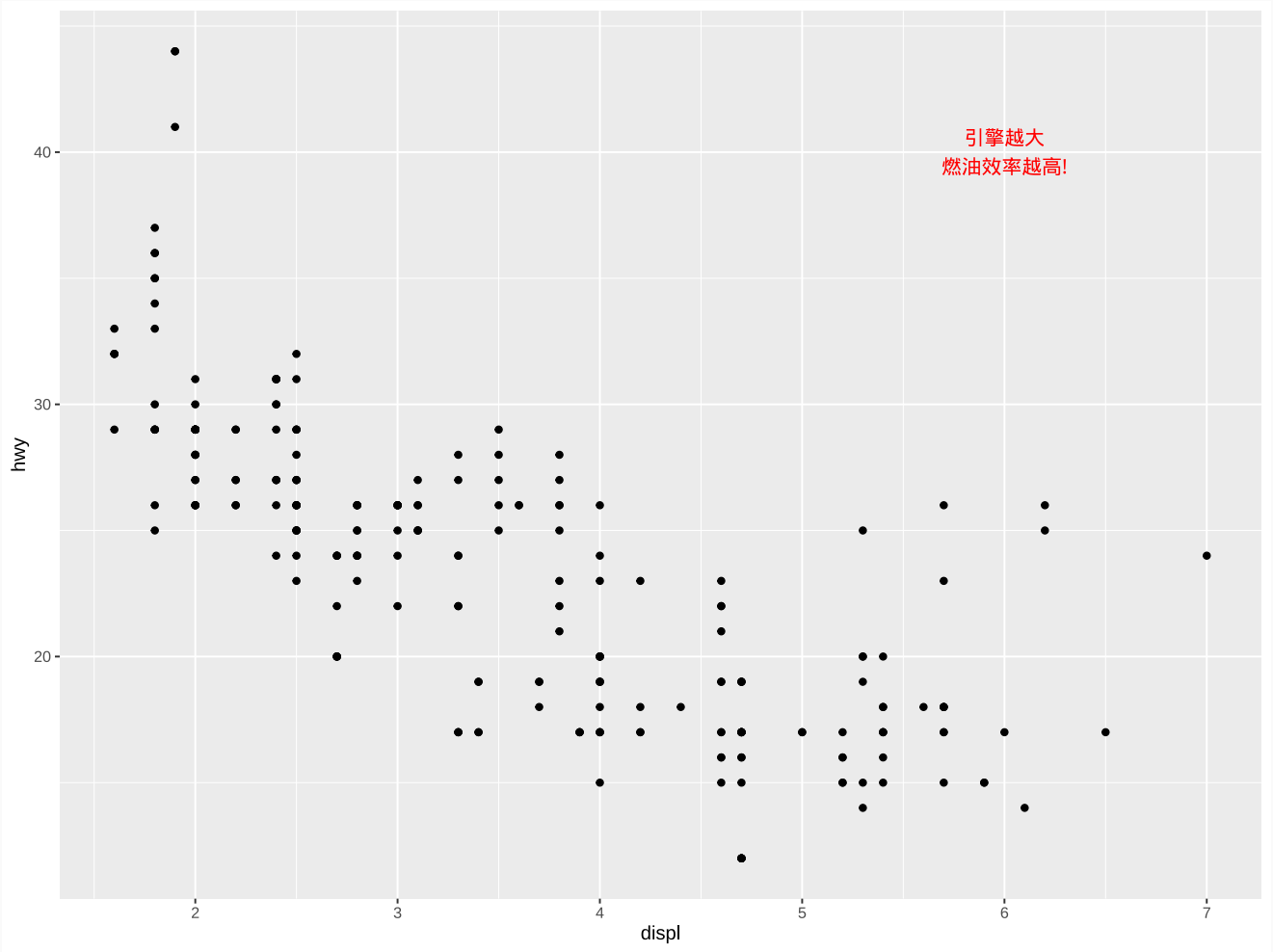
首先要准备好标记点的数据，然后增加文字标注的图层，需要提供标记点数据，以及要标注的文字给 label 美学，若来自数据变量，则需要用映射。

```
library(ggrepel)
best_in_class = mpg %>%    # 选取每种车型 hwy 值最大的样本
  group_by(class) %>%
  slice_max(hwy, n = 1)
ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(color = class)) +
  geom_label_repel(data = best_in_class,
                   aes(label = model))
```


new beetle
jetta
altima
forester awd
caravan 2wd
caravan 2wd
caravan 2wd
caravan 2wd
toyota tacoma 4wd
corvette
corvette
hwy
40
30
20
displ
2
3
4
5
6
7
class
2seater
compact
midsize
minivan
pickup
subcompact
suv

若要在图形某坐标位置添加文本注释，则用 annotate() 函数，需要提供添加文本的中心坐标位置，和要添加的文字内容

```
ggplot(mpg, aes(displ, hwy)) +
  geom_point() +
  annotate(geom = "text", x = 6, y = 40,
           label = " 引擎越大\n燃油效率越高!", size = 4,
           color = "red")
```


引擎越大
燃油效率越高!
hwy
40
30
20
displ
2
3
4
5
6
7

本篇主要参阅 (张敬信, 2022), (Wickham, 2020), (Hadley Wickham, 2017), (?), A Practical Introduction to Data Visualization with ggplot2, 模板感谢 (黄湘云, 2021), (谢益辉, 2021).

参考文献

Hadley Wickham, G. G. (2017). *R for Data Science*. O' Reilly, 1 edition. ISBN 978-1491910399.

Wickham, H. (2020). *ggplot2: Elegant Graphics for Data Analysis*. Springer, 3 edition.

张敬信 (2022). *R* 语言编程：基于 *tidyverse*. 人民邮电出版社, 北京.

谢益辉 (2021). *rmarkdown: Dynamic Documents for R*.

黄湘云 (2021). *Github: R-Markdown-Template*.