

ggplot2 数据可视化完全指南

WorkBuddy

2026-08-14

目录

1	ggplot2 简介	3
1.1	为什么选择 ggplot2	3
1.2	安装与加载	3
1.3	本文使用的内置数据	4
2	图形语法的核心思想	4
2.1	一张图 = 数据 + 映射 + 几何对象 +	4
2.2	基本模板	5
3	快速入门：第一张图	5
4	美学映射 aes()：用视觉属性表达数据	6
4.1	颜色 color	6
4.2	大小 size 与透明度 alpha	7
4.3	形状 shape	8
4.4	映射 vs 固定值	9

目录	2
5 常用几何对象 geom_*	10
5.1 散点图 geom_point()	10
5.2 折线图 geom_line() 与路径图 geom_path()	11
5.3 平滑曲线 geom_smooth()	11
5.4 柱状图 geom_bar() 与 geom_col()	12
5.5 直方图 geom_histogram() 与密度图 geom_density()	13
5.6 箱线图 geom_boxplot() 与小提琴图 geom_violin()	13
5.7 面积图 geom_area() 与热图 geom_tile()	14
5.8 常用 geom 速查表	15
6 标度 scale_*: 精细控制映射	15
6.1 颜色标度	16
6.2 坐标轴标度	16
6.3 坐标轴变换	17
7 分面 facet: 一页多图	18
7.1 facet_wrap(): 单变量分面	18
7.2 facet_grid(): 双变量网格分面	19
8 坐标系 coord_*	20
9 标题、标签与注释	21
10 主题 theme: 打造专业外观	22
10.1 内置主题	22
10.2 自定义 theme() 细节	23

1	GGPLOT2 简介	3
11	多图组合与保存	24
11.1	保存图片 ggsave()	24
11.2	多图拼接 patchwork	24
12	综合案例	24
13	常见技巧与避坑指南	26
14	进阶资源	26

1 ggplot2 简介

ggplot2 是 R 语言中最流行、最强大的数据可视化包，由 Hadley Wickham 基于 Leland Wilkinson 的著作 *The Grammar of Graphics*（图形语法）开发，是 **tidyverse** 生态系统的核心成员之一。

1.1 为什么选择 ggplot2

- 语法一致：所有图形都由相同的”积木”搭建，学会一个就等于学会全部；
- 层次分明：可以像搭积木一样，一层一层地叠加图形元素；
- 美观专业：默认主题和配色已经接近出版级别；
- 高度可定制：从坐标轴到图例，每个细节都可以调整；
- 生态丰富：ggthemes、ggrepel、patchwork、gganimate 等数百个扩展包。

1.2 安装与加载

```
install.packages("ggplot2") # 安装（只需一次）
library(ggplot2)           # 加载（每次会话都需要）
```

1.3 本文使用的内置数据

ggplot2 和 R 自带了多个示例数据集，本文主要使用：

```
# 汽车燃油效率数据 (ggplot2 内置)
data(mpg)
head(mpg, 3)
```

```
##      manufacturer model displ year cyl      trans drv  cty   hwy fl   class
## 1             audi   a4    1.8 1999   4    auto(l5)  f   18   29   p compact
## 2             audi   a4    1.8 1999   4 manual(m5)  f   21   29   p compact
## 3             audi   a4    2.0 2008   4 manual(m6)  f   20   31   p compact
```

```
# R 自带的汽车数据
head(mtcars, 3)
```

```
##              mpg  cyl disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4      21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag  21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710     22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
```

2 图形语法的核心理念

ggplot2 的灵魂是图形语法（Grammar of Graphics）：任何一张统计图形，都可以拆解为若干相互独立的组件（layer，图层），再组合而成。

2.1 一张图 = 数据 + 映射 + 几何对象 +

组件	函数	作用
数据（Data）	ggplot(data)	绘图所用的数据框

组件	函数	作用
美学映射 (Aesthetics)	<code>aes()</code>	把变量映射到 x、y、颜色、大小、形状等视觉属性
几何对象 (Geom)	<code>geom_*()</code>	决定画什么：点、线、柱、箱线图.....
统计变换 (Stat)	<code>stat_*()</code>	对数据进行汇总、平滑等计算
标度 (Scale)	<code>scale_*()</code>	控制映射的细节：颜色方案、坐标轴刻度等
坐标系 (Coord)	<code>coord_*()</code>	笛卡尔坐标、极坐标、翻转坐标等
分面 (Facet)	<code>facet_*()</code>	按变量把图拆成多个子图
主题 (Theme)	<code>theme_*()</code>	字体、背景、网格线等外观

2.2 基本模板

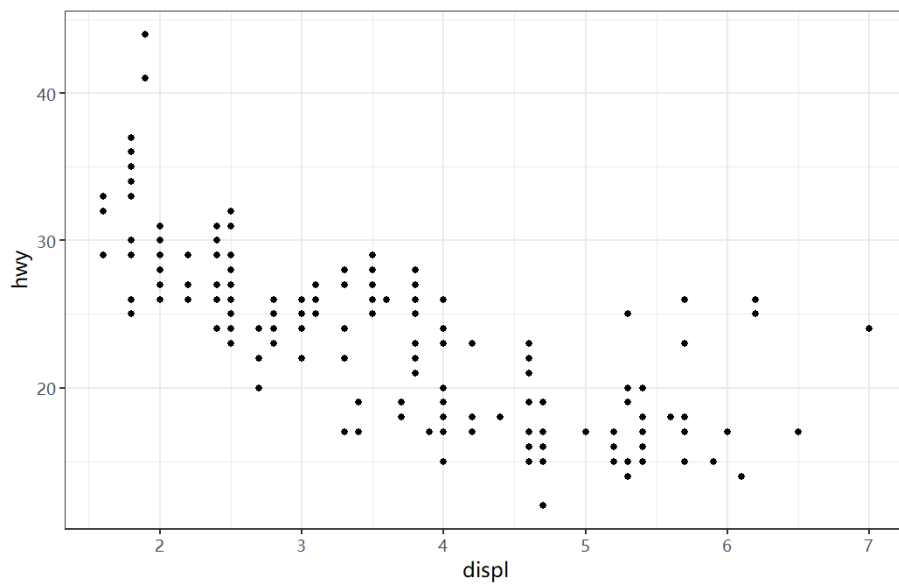
```
ggplot(data = < 数据 >, mapping = aes(< 映射 >)) +  
  < 几何对象函数 >() +  
  < 其他图层 >()
```

注意：ggplot2 用 + 号（不是管道%>%）来连接各个图层。

3 快速入门：第一张图

以 mpg 数据集为例，画出发动机排量（displ）与高速油耗（hwy）的散点图：

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



三行代码，已经完成了：建立画布（`ggplot()`）、指定映射（`aes()`）、添加点图层（`geom_point()`）。

简写形式：`data =` 和 `mapping =` 可以省略：

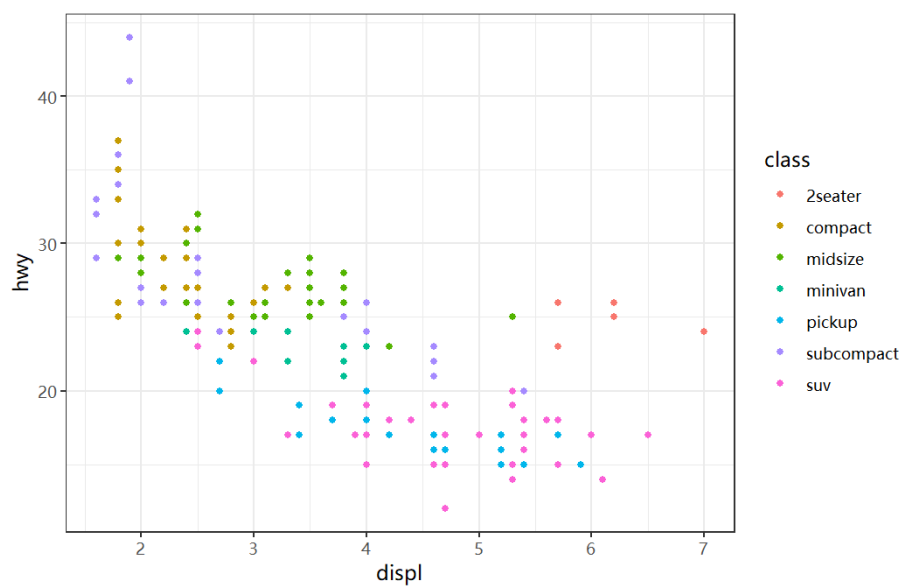
```
ggplot(mpg, aes(displ, hwy)) + geom_point()
```

4 美学映射 `aes()`: 用视觉属性表达数据

`aes()` 可以把变量映射到颜色、大小、形状、透明度等视觉通道，这是 `ggplot2` 与基础绘图最大的不同——图例会自动生成。

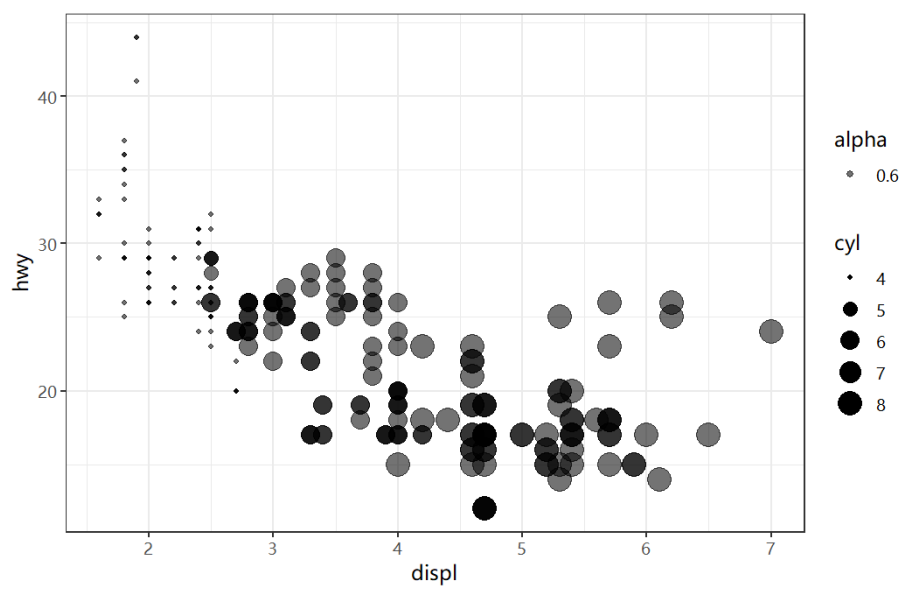
4.1 颜色 `color`

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



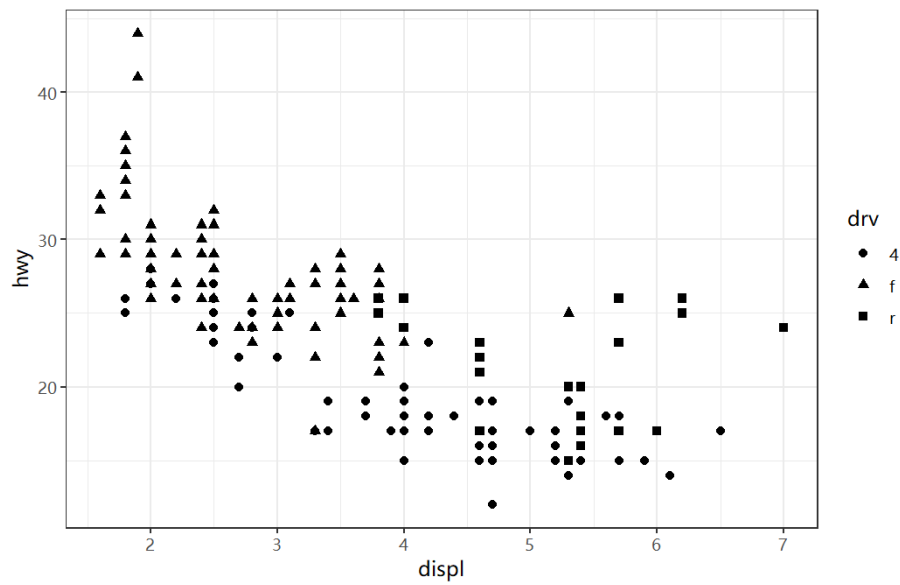
4.2 大小 size 与透明度 alpha

```
ggplot(mpg, aes(displ, hwy, size = cyl, alpha = 0.6)) +  
  geom_point()
```



4.3 形状 shape

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

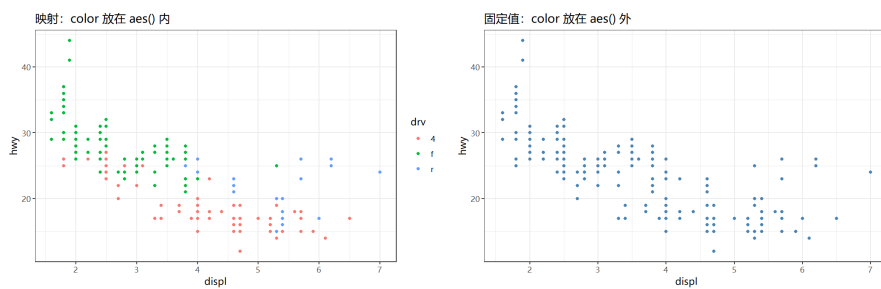



4.4 映射 vs 固定值

这是一个常见易错点：放在 `aes()` 里是“按变量映射”，放在 `aes()` 外是“给所有元素设定固定值”：

```
# 按 drv 变量映射颜色（自动生成图例）
ggplot(mpg, aes(displ, hwy, color = drv)) +
  geom_point() +
  labs(title = "映射: color 放在 aes() 内")

# 所有点统一为蓝色（无图例）
ggplot(mpg, aes(displ, hwy)) +
  geom_point(color = "steelblue") +
  labs(title = "固定值: color 放在 aes() 外")
```



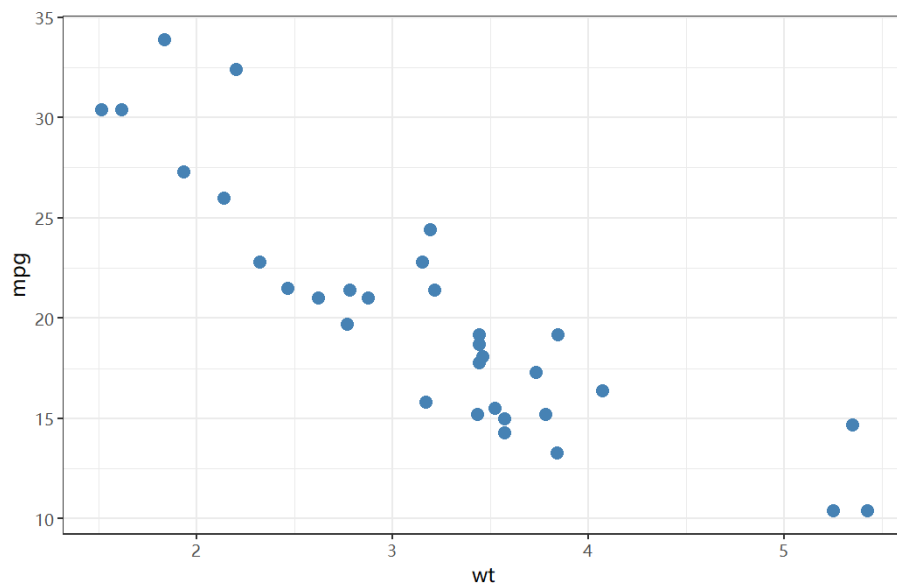
5 常用几何对象 *geom_**

几何对象决定图形的”形状”。常用的有：

5.1 散点图 *geom_point()*

适合展示两个连续变量的关系：

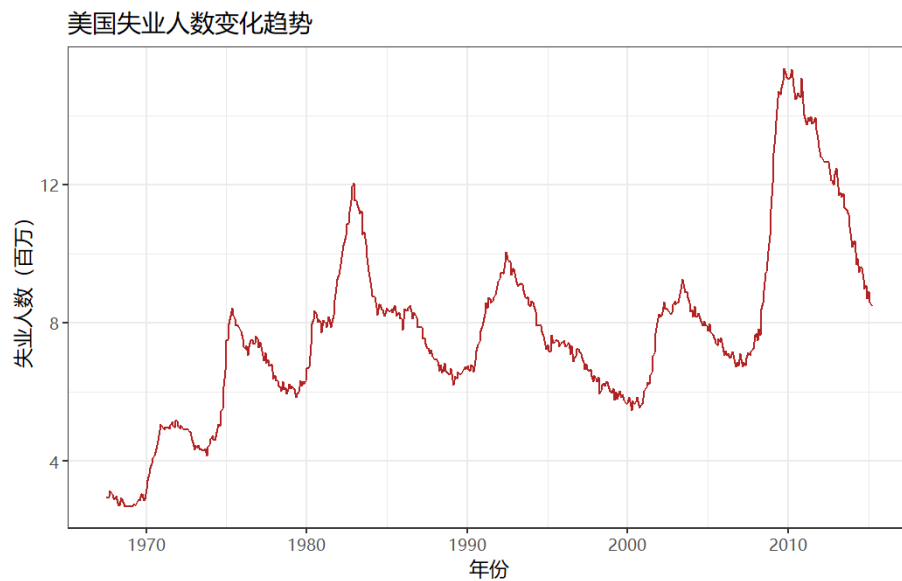
```
ggplot(mtcars, aes(wt, mpg)) +  
  geom_point(color = "steelblue", size = 3)
```



5.2 折线图 `geom_line()` 与路径图 `geom_path()`

`geom_line()` 按 x 轴顺序连线，适合时间序列；`geom_path()` 按数据原始顺序连线。

```
ggplot(economics, aes(date, unemploy / 1000)) +  
  geom_line(color = "firebrick") +  
  labs(x = " 年份", y = " 失业人数 (百万)",  
       title = " 美国失业人数变化趋势")
```



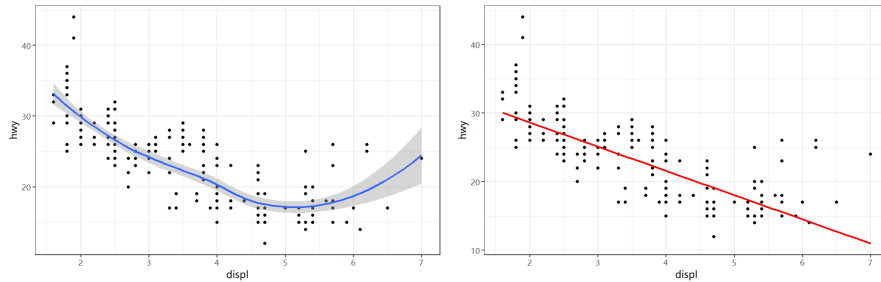
5.3 平滑曲线 `geom_smooth()`

为散点图添加拟合曲线，默认带 95% 置信区间：

```
# 默认方法（数据量小时为 loess 局部回归）  
ggplot(mpg, aes(displ, hwy)) +  
  geom_point() +  
  geom_smooth()
```

线性回归 + 去掉置信区间

```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point() +  
  geom_smooth(method = "lm", se = FALSE, color = "red")
```



5.4 柱状图 `geom_bar()` 与 `geom_col()`

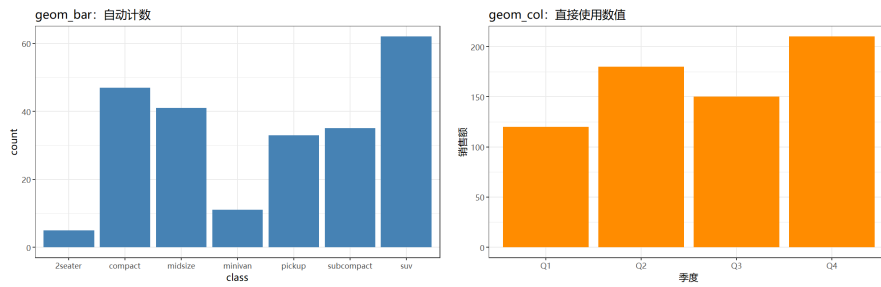
- `geom_bar()`: 自动统计每组的频数 (y 轴是 count);
- `geom_col()`: 直接使用数据中的数值作为柱高。

`geom_bar`: 自动计数

```
ggplot(mpg, aes(class)) +  
  geom_bar(fill = "steelblue") +  
  labs(title = "geom_bar: 自动计数")
```

`geom_col`: 使用已有的数值

```
df <- data.frame(季度 = c("Q1", "Q2", "Q3", "Q4"),  
                 销售额 = c(120, 180, 150, 210))  
ggplot(df, aes(季度, 销售额)) +  
  geom_col(fill = "darkorange") +  
  labs(title = "geom_col: 直接使用数值")
```

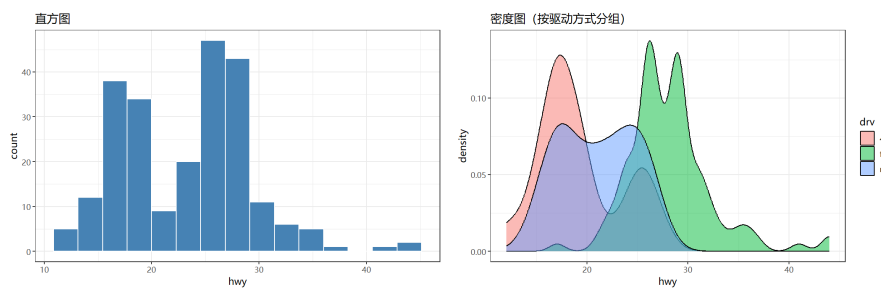


5.5 直方图 `geom_histogram()` 与密度图 `geom_density()`

展示单个连续变量的分布:

```
ggplot(mpg, aes(hwy)) +
  geom_histogram(bins = 15, fill = "steelblue", color = "white") +
  labs(title = "直方图")

ggplot(mpg, aes(hwy, fill = drv)) +
  geom_density(alpha = 0.5) +
  labs(title = "密度图 (按驱动方式分组)")
```

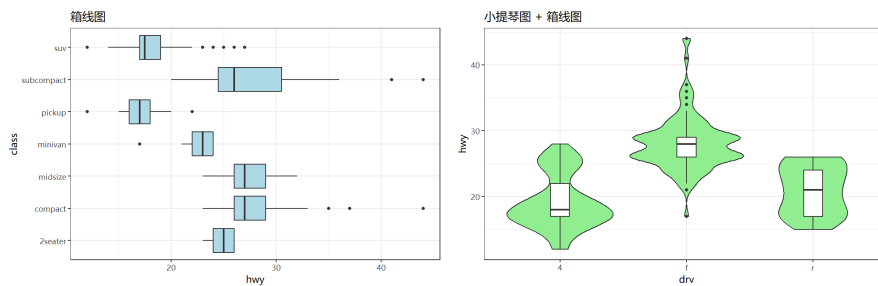


5.6 箱线图 `geom_boxplot()` 与小提琴图 `geom_violin()`

比较不同组别的分布:

```
ggplot(mpg, aes(class, hwy)) +
  geom_boxplot(fill = "lightblue") +
  coord_flip() +
  labs(title = "箱线图")

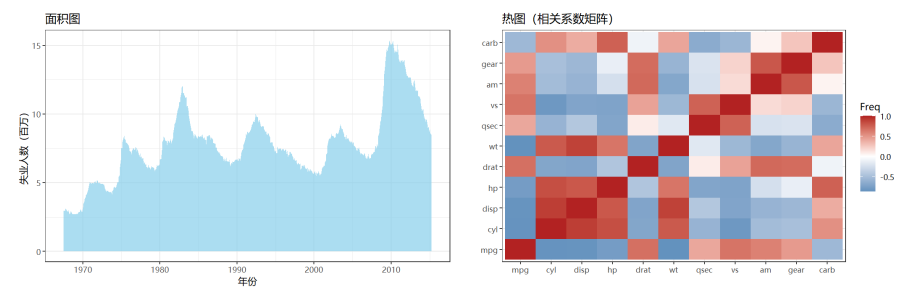
ggplot(mpg, aes(drv, hwy)) +
  geom_violin(fill = "lightgreen") +
  geom_boxplot(width = 0.15, fill = "white") +
  labs(title = "小提琴图 + 箱线图")
```



5.7 面积图 `geom_area()` 与热图 `geom_tile()`

```
ggplot(economics, aes(date, unemploy / 1000)) +
  geom_area(fill = "skyblue", alpha = 0.7) +
  labs(title = "面积图", x = "年份", y = "失业人数 (百万)")

# 相关矩阵热图
cor_mat <- cor(mtcars)
cor_df <- as.data.frame(as.table(cor_mat))
ggplot(cor_df, aes(Var1, Var2, fill = Freq)) +
  geom_tile() +
  scale_fill_gradient2(low = "steelblue", high = "firebrick", mid = "white") +
  labs(title = "热图 (相关系数矩阵)", x = "", y = "")
```



5.8 常用 `geom` 速查表

geom	图形	适用场景
<code>geom_point()</code>	散点图	两个连续变量的关系
<code>geom_line()</code>	折线图	时间序列、趋势
<code>geom_smooth()</code>	平滑曲线	拟合关系、趋势
<code>geom_bar()</code>	柱状图 (计数)	分类变量频数
<code>geom_col()</code>	柱状图 (数值)	已汇总的数值
<code>geom_histogram()</code>	直方图	连续变量分布
<code>geom_density()</code>	密度图	分布形状比较
<code>geom_boxplot()</code>	箱线图	分组分布比较
<code>geom_violin()</code>	小提琴图	分布密度比较
<code>geom_area()</code>	面积图	随时间的累积量
<code>geom_tile()</code>	热图	矩阵、网格数据
<code>geom_text()</code> / <code>geom_label()</code>	文字标签	在图中添加文字

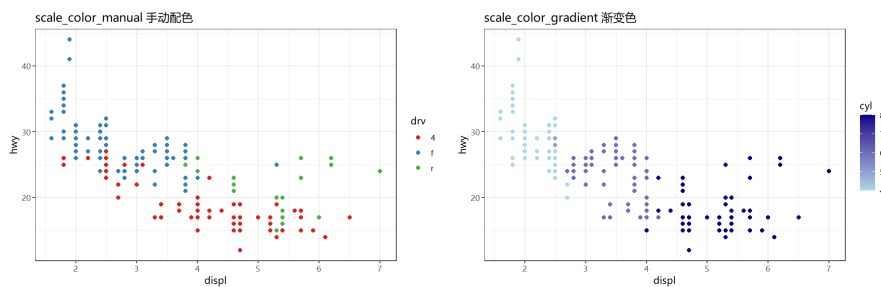
6 标度 `scale_*`: 精细控制映射

标度函数控制” 变量 → 视觉属性” 的具体方式。命名规律：`scale_ + 美学属性 + _ + 类型`。

6.1 颜色标度

```
# 离散变量: 手动指定颜色
ggplot(mpg, aes(displ, hwy, color = drv)) +
  geom_point(size = 2) +
  scale_color_manual(values = c("4" = "#E41A1C", "f" = "#377EB8", "r" = "#4DAF4A")) +
  labs(title = "scale_color_manual 手动配色")

# 连续变量: 渐变色
ggplot(mpg, aes(displ, hwy, color = cyl)) +
  geom_point(size = 2) +
  scale_color_gradient(low = "lightblue", high = "darkblue") +
  labs(title = "scale_color_gradient 渐变色")
```



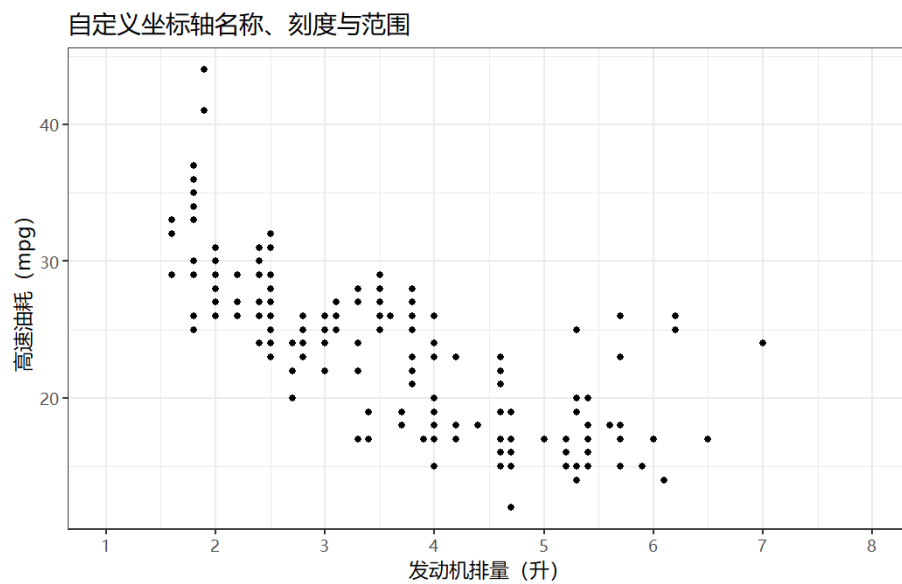
推荐使用 `scale_color_brewer()` / `scale_fill_brewer()` (ColorBrewer 专业配色) 或 `scale_color_viridis_d()` (色盲友好的 viridis 配色)。

6.2 坐标轴标度

```
ggplot(mpg, aes(displ, hwy)) +
  geom_point() +
  scale_x_continuous(name = " 发动机排量 (升) ",
    breaks = seq(0, 8, by = 1),
    limits = c(1, 8)) +
```

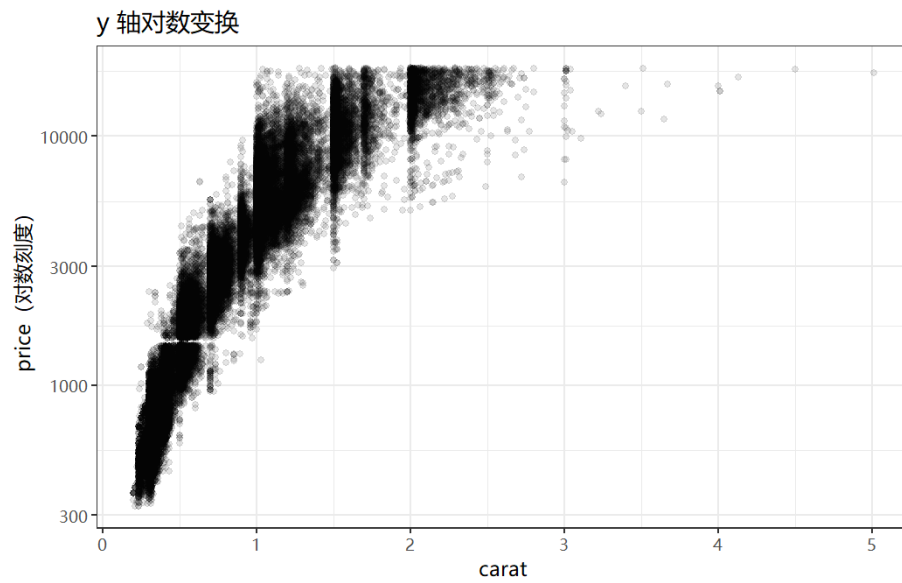


```
scale_y_continuous(name = " 高速油耗 (mpg) ") +  
labs(title = " 自定义坐标轴名称、刻度与范围")
```



6.3 坐标轴变换

```
ggplot(diamonds, aes(carat, price)) +  
  geom_point(alpha = 0.1) +  
  scale_y_log10() +  
  labs(title = "y 轴对数变换", y = "price (对数刻度)")
```

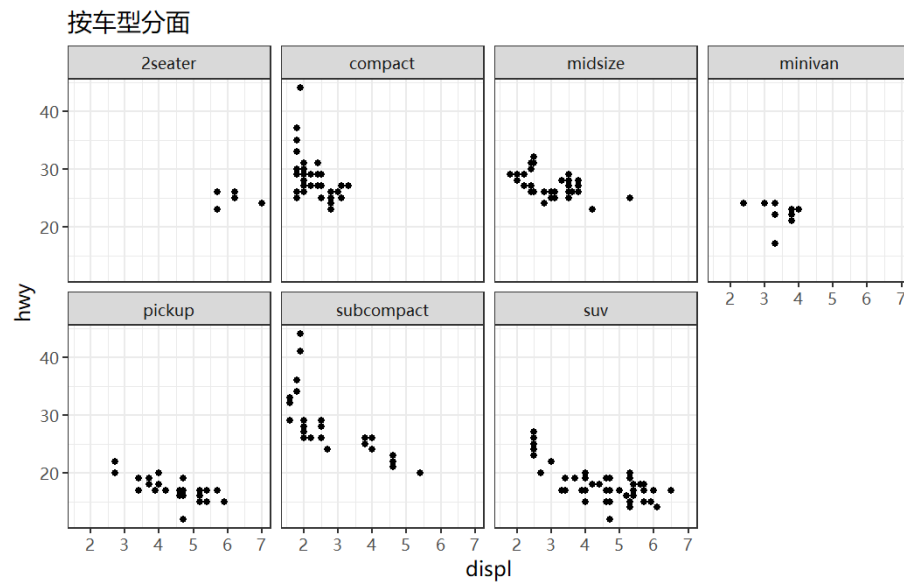


7 分面 facet: 一页多图

分面按分类变量把数据拆成多个子图，是探索分组差异的利器。

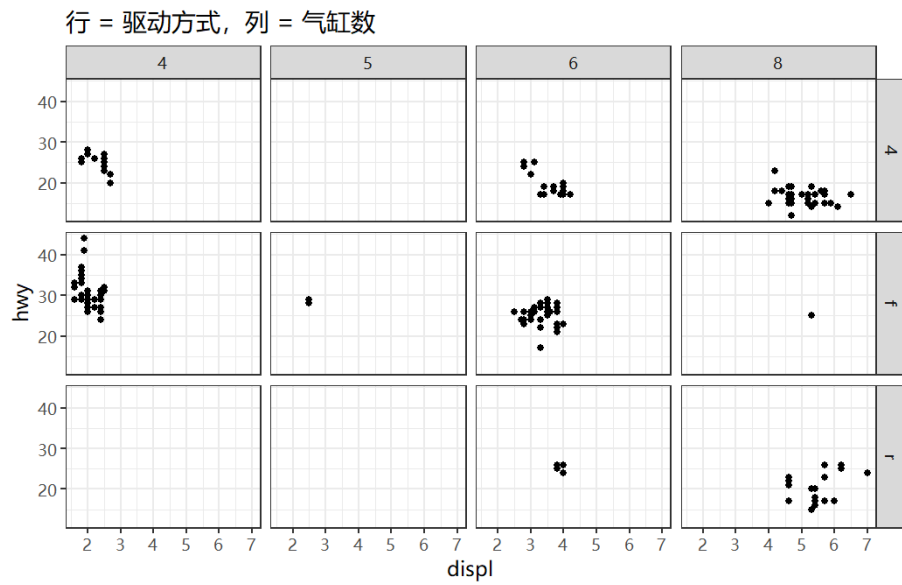
7.1 facet_wrap(): 单变量分面

```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point() +  
  facet_wrap(~ class, ncol = 4) +  
  labs(title = "按车型分面")
```



7.2 facet_grid(): 双变量网格分面

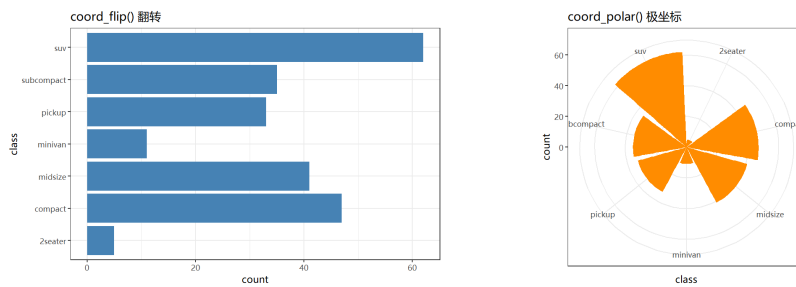
```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point() +  
  facet_grid(drv ~ cyl) +  
  labs(title = "行 = 驱动方式, 列 = 气缸数")
```



8 坐标系 `coord_*`

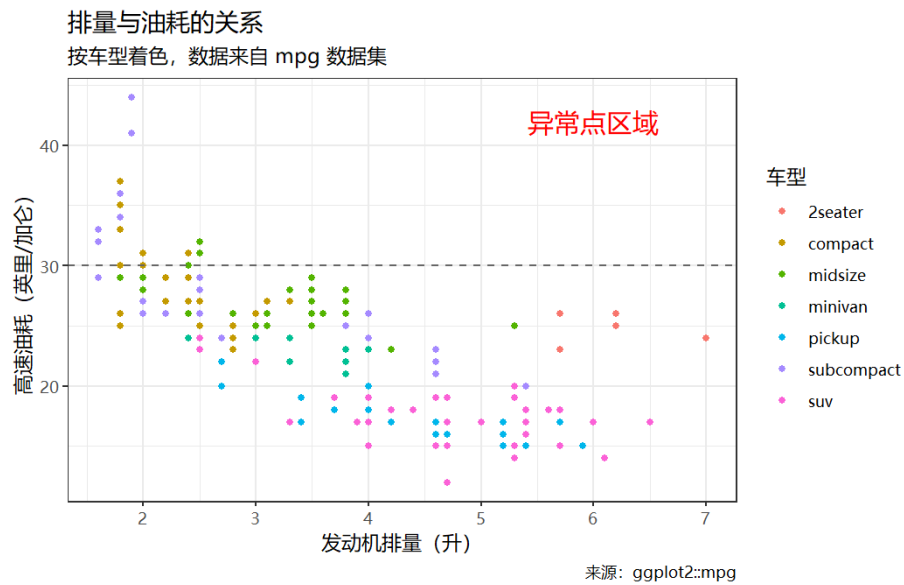
```
# coord_flip: 翻转坐标轴 (横向柱状图)
ggplot(mpg, aes(class)) +
  geom_bar(fill = "steelblue") +
  coord_flip() +
  labs(title = "coord_flip() 翻转")

# coord_polar: 极坐标 (柱状图 → 玫瑰图)
ggplot(mpg, aes(class)) +
  geom_bar(fill = "darkorange") +
  coord_polar() +
  labs(title = "coord_polar() 极坐标")
```



9 标题、标签与注释

```
p <- ggplot(mpg, aes(displ, hwy, color = class)) +
  geom_point() +
  labs(
    title = " 排量与油耗的关系",
    subtitle = " 按车型着色, 数据来自 mpg 数据集",
    x = " 发动机排量 (升) ",
    y = " 高速油耗 (英里/加仑) ",
    color = " 车型", # 修改图例标题
    caption = " 来源: ggplot2::mpg"
  ) +
  annotate("text", x = 6, y = 42,
    label = " 异常点区域", color = "red", size = 5) +
  geom_hline(yintercept = 30, linetype = "dashed", color = "grey40")
p
```

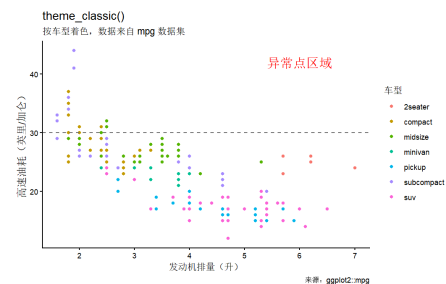
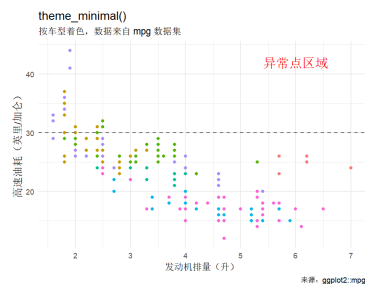


- `labs()`: 一站式设置标题、副标题、坐标轴、图例标题、脚注;
- `annotate()`: 在指定位置添加注释 (`text` / `rect` / `segment` 等);
- `geom_hline()` / `geom_vline()` / `geom_abline()`: 参考线。

10 主题 theme: 打造专业外观

10.1 内置主题

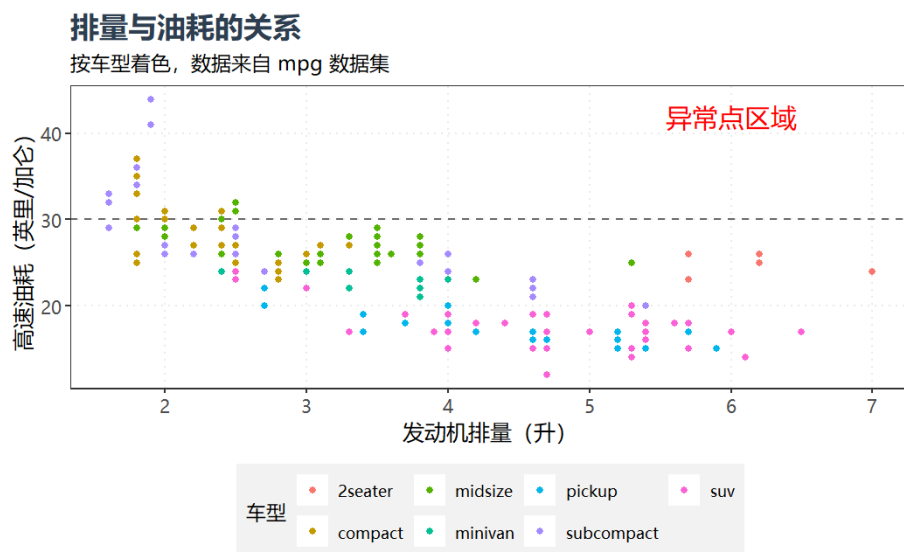
```
p + theme_minimal() + labs(title = "theme_minimal()")
p + theme_classic() + labs(title = "theme_classic()")
```



常用内置主题: `theme_bw()`、`theme_gray()` (默认)、`theme_minimal()`、`theme_classic()`、`theme_light()`、`theme_dark()`、`theme_void()`。
`ggthemes` 包还提供《经济学人》、`FiveThirtyEight` 等风格。

10.2 自定义 theme() 细节

```
p +
  theme(
    plot.title = element_text(size = 16, face = "bold", color = "#2C3E50"),
    axis.title = element_text(size = 12),
    axis.text = element_text(size = 10, color = "grey30"),
    legend.position = "bottom", # 图例放到底部
    legend.background = element_rect(fill = "grey95"),
    panel.grid.minor = element_blank(), # 去掉次网格线
    panel.grid.major = element_line(linetype = "dotted")
  )
```



来源: `ggplot2::mpg`

`legend.position` 可选 "top"、"bottom"、"left"、"right"、"none" (隐藏图例)。

11 多图组合与保存

11.1 保存图片 ggsave()

```
# 保存最近绘制的一张图
ggsave("my_plot.png", width = 8, height = 5, dpi = 300)

# 保存指定的图, 支持 png / pdf / svg / jpg 等格式
ggsave("my_plot.pdf", plot = p, width = 8, height = 5)
```

11.2 多图拼接 patchwork

```
install.packages("patchwork")
library(patchwork)

p1 <- ggplot(mpg, aes(displ, hwy)) + geom_point()
p2 <- ggplot(mpg, aes(class)) + geom_bar()

p1 + p2          # 左右并排
p1 / p2          # 上下堆叠
(p1 | p2) / p2   # 自由组合
```

12 综合案例

把前面学到的内容组合起来, 完成一张信息丰富的分组图:

```
ggplot(mpg, aes(displ, hwy, color = drv)) +
  geom_point(size = 2, alpha = 0.8) +
  geom_smooth(method = "lm", se = FALSE, linewidth = 0.8) +
  facet_wrap(~ drv, labeller = labeller(drv = c(
    "4" = " 四驱", "f" = " 前驱", "r" = " 后驱")))) +
```

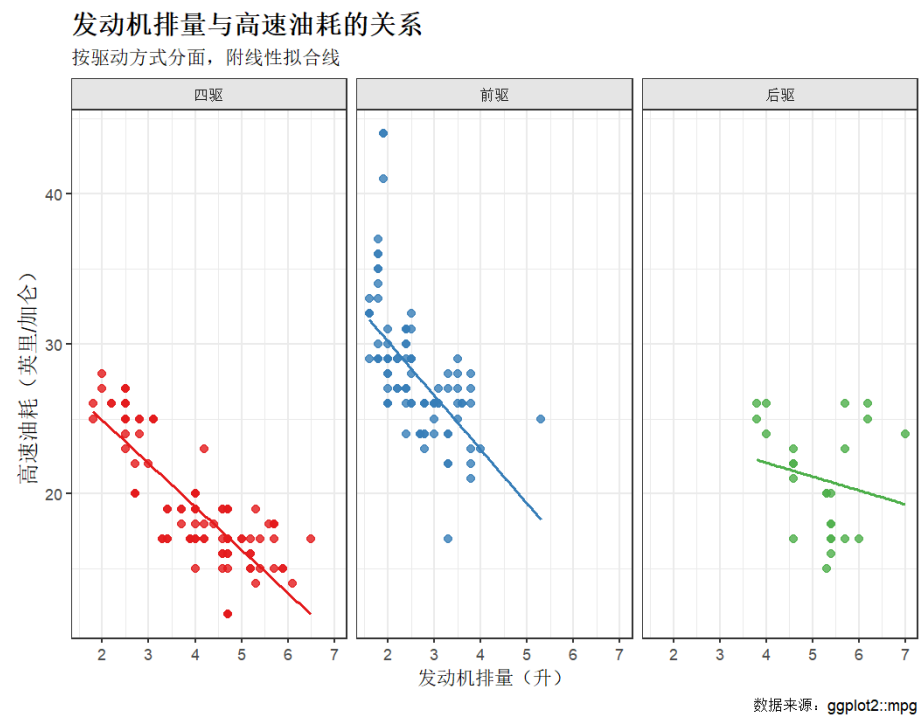


```

scale_color_manual(values = c("4" = "#E41A1C",
                              "f" = "#377EB8",
                              "r" = "#4DAF4A")) +

labs(
  title = " 发动机排量与高速油耗的关系",
  subtitle = " 按驱动方式分面，附线性拟合线",
  x = " 发动机排量（升）",
  y = " 高速油耗（英里/加仑）",
  caption = " 数据来源: ggplot2::mpg"
) +
theme_bw() +
theme(
  plot.title = element_text(face = "bold", size = 15),
  strip.background = element_rect(fill = "grey90"),
  legend.position = "none"
)

```



这张图用到了：散点图层 + 拟合线图层、颜色映射与手动配色、分面与中文分面标签、标题标签、主题微调——正是图形语法“分层叠加”思想的完整体现。

13 常见技巧与避坑指南

1. 用 + 连接图层，不要混用管道%>%；+ 必须放在行尾。
2. 映射 vs 固定值：按变量变化放 `aes()` 内，统一设定放 `aes()` 外。
3. `geom_bar()` vs `geom_col()`：要数频数用前者，数据已有数值用后者。
4. 因子决定顺序：柱状图、箱线图的顺序由因子水平决定，可用 `factor(x, levels = ...)` 或 `forcats::fct_reorder()` 调整。
5. 分组不显式：折线图按组连线时，指定 `aes(group = 分组变量)`。
6. 重叠点太多：用 `alpha` 透明度、`geom_jitter()` 抖动或 `geom_hex()` 二维分箱。
7. 去掉图例：`theme(legend.position = "none")` 或 `guides(color = "none")`。
8. 中文乱码 (Windows)：先注册字体再使用，例如 `windowsFonts(yahei = windowsFont("Microsoft YaHei"))` 后设置 `theme_set(theme_bw(base_family = "yahei"))`；或使用 `showtext` 包获得跨平台一致的字体效果。

14 进阶资源

- 官方文档：<https://ggplot2.tidyverse.org>
- 《ggplot2: Elegant Graphics for Data Analysis》(Hadley Wickham, 第3版, 免费在线阅读)
- 《R for Data Science》可视化章节：<https://r4ds.hadley.nz>
- R Graph Gallery: <https://r-graph-gallery.com> (大量可直接套用的图例代码)
- 扩展包索引：<https://exts.ggplot2.tidyverse.org>