

# R 语言编程：基于 tidyverse

## 第 16 讲 ggplot2 绘图 II

---

张敬信

2022 年 2 月 14 日

哈尔滨商业大学

## 四. 统计变换、坐标系、位置调整

### 1. 统计变换 (Statistics)

构建新的统计量进而绘图，称为“统计变换”，简称“统计”。比如，条形图、直方图都是先对数据分组，再计算分组频数绘图；平滑曲线用来根据数据拟合模型，进而绘制模型预测值.....

ggplot2 强大的一点就是，把统计变换直接融入绘图语法中，而不必先在外面对数据做统计变换，再回来绘图。

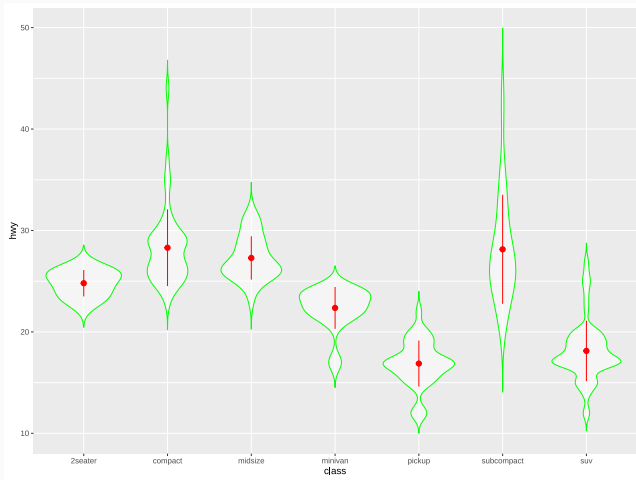
ggplot2 提供了 30 多种“统计”，均以 `stat_xxxx()` 的方式命名：

- 可在几何对象函数 `geom_*`() 中创建，通常直接使用后者即可：
  - `stat_bin()`: `geom_bar()`, `geom_freqpoly()`, `geom_histogram()`
  - `stat_bindot()`: `geom_dotplot()`
  - `stat_boxplot()`: `geom_boxplot()`
  - `stat_contour()`: `geom_contour()`
  - `stat_quantile()`: `geom_quantile()`
  - `stat_smooth()`: `geom_smooth()`
  - `stat_sum()`: `geom_count()`

- 不能在几何对象函数 `geom_*()` 中创建:
  - `stat_ecdf()`: 计算经验累积分布图
  - `stat_function()`: 根据  $x$  值的函数计算  $y$  值
  - `stat_summary()`: 在  $x$  唯一值处汇总  $y$  值
  - `stat_qq()`: 执行 Q-Q 图计算
  - `stat_spoke()`: 转换极坐标的角度和半径为直角坐标位置
  - `stat_unique()`: 剔除重复行

用 `stat_summary()` 做统计汇总并绘图

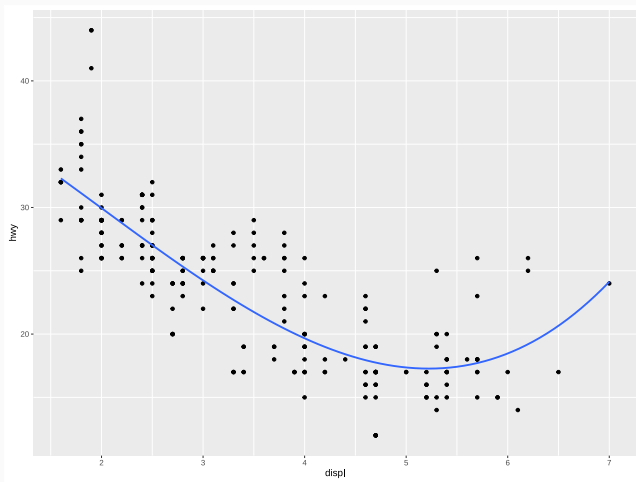
```
ggplot(mpg, aes(x = class, y = hwy)) +  
  geom_violin(trim = FALSE, alpha = 0.5, color = "green") +  
  stat_summary(fun = mean,  
               fun.min = function(x) {mean(x) - sd(x)},  
               fun.max = function(x) {mean(x) + sd(x)},  
               geom = "pointrange", color = "red")
```



用 `stat_smooth()`, 与 `geom_smooth()` 相同, 添加光滑曲线:

- `method`: 指定平滑曲线的统计函数, 如 `lm` 线性回归, `glm` 广义线性回归, `loess` 多项式回归, `gam` 广义加法模型 (`mgcv` 包), `rlm` 稳健回归 (`MASS` 包) 等
- `formula`: 指定平滑曲线的方程, 如 `y ~ x`, `y ~ poly(x, 2)`, `y ~ log(x)`, 需要与 `method` 参数搭配使用
- `se`: 设置是否绘制置信区间

```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point() +  
  stat_smooth(method = "lm",  
              formula = y ~ splines::bs(x, 3),  
              se = FALSE)      # 不绘制置信区间
```





## 2. 坐标系 (Coordinante)

ggplot2 默认是直角坐标系 `coord_cartesian()`, 常用的坐标系操作还有:

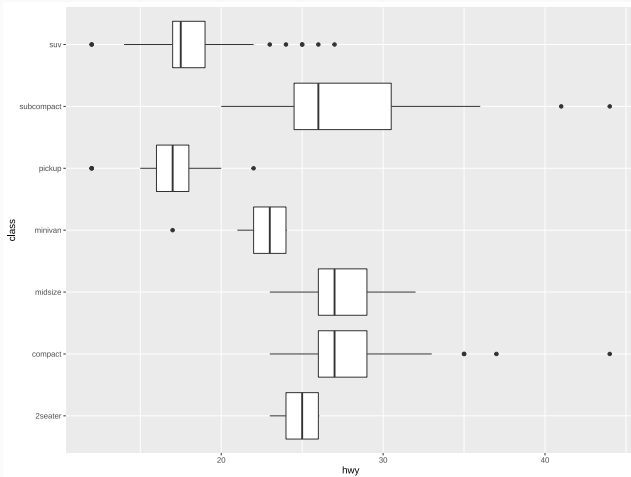
- `coord_flip()`: 坐标轴翻转, 即 x 轴与 y 轴互换, 比如绘制水平条形图
- `coord_fixed()`: 固定  $\text{ratio} = y / x$  的比例
- `coord_polar()`: 转化为极坐标系, 比如条形图转为极坐标系即为饼图
- `coord_trans()`: 彻底的坐标变换, 不同于 `scale_x_log10()` 等
- `coord_map()`, `coord_quickmap()`: 与 `geom_polygon()` 连用, 控制地图的坐标投影
- `coord_sf()`: 与 `geom_sf()` 连用, 控制地图的坐标投影

- 坐标轴翻转，从水平图到竖直图：

```
ggplot(mpg, aes(class, hwy)) +  
  geom_boxplot() +  
  coord_flip()
```

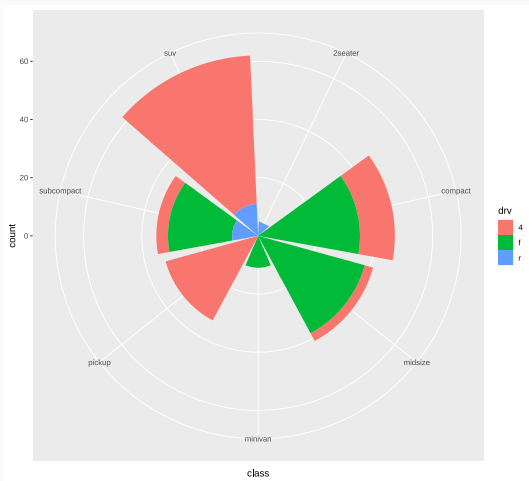
# 箱线图

# 从竖直变成水平



- 直角坐标下的条形图，转化为极坐标下的风玫瑰图：

```
ggplot(mpg, aes(class, fill = drv)) +  
  geom_bar() +  
  coord_polar()
```

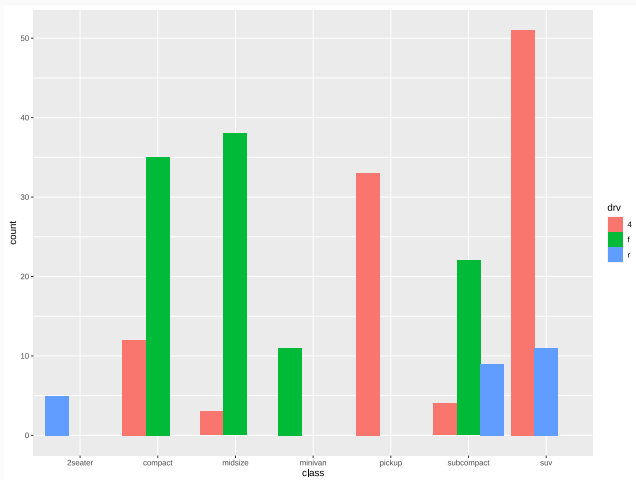


### 3. 位置调整 (Position adjustments)

条形图中的条形位置调整:

- `position_stack()`: 竖直堆叠
- `position_fill()`: 竖直 (百分比) 堆叠, 按比例放缩保证总高度为 1
- `position_dodge()`, `position_dodge2()`: 水平堆叠

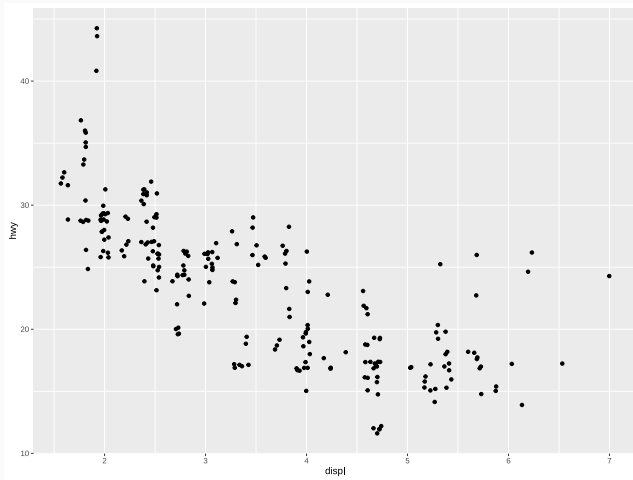
```
ggplot(mpg, aes(class, fill = drv)) +  
  geom_bar(position = position_dodge(preserve = "single"))  
# geom_bar(position = "dodge")
```



散点图中的散点位置调整:

- `position_nudge()`: 将散点移动固定的偏移量
- `position_jitter()`: 给每个散点增加一点随机噪声 (抖散图)
- `position_jitterdodge()`: 增加一点随机噪声并躲避组内的点, 特别用于箱线图 + 散点图

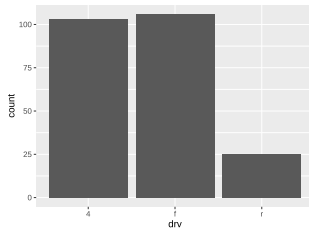
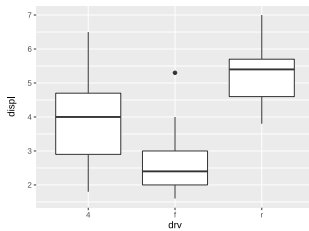
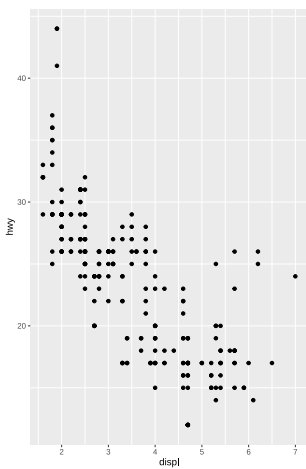
```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point(position = "jitter")    # 避免有散点重叠
```



- 用 patchwork 包排布多个图形

```
library(patchwork)
p1 = ggplot(mpg, aes(displ, hwy)) +
  geom_point()
p2 = ggplot(mpg, aes(drv, displ)) +
  geom_boxplot()
p3 = ggplot(mpg, aes(drv)) +
  geom_bar()
p1 | (p2 / p3)
```





## 五. 分面、主题、输出

### 1. 分面 (Facet)

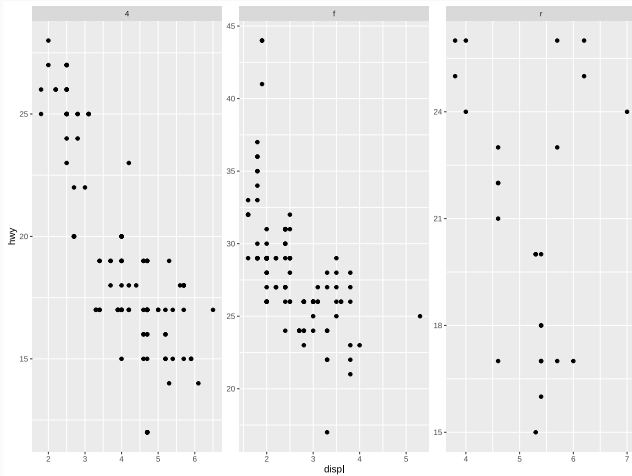
利用分类变量将图形分为若干个“面”（子图），即对数据分组再分别绘图，称为“分面”。

#### (1) `facet_wrap()`

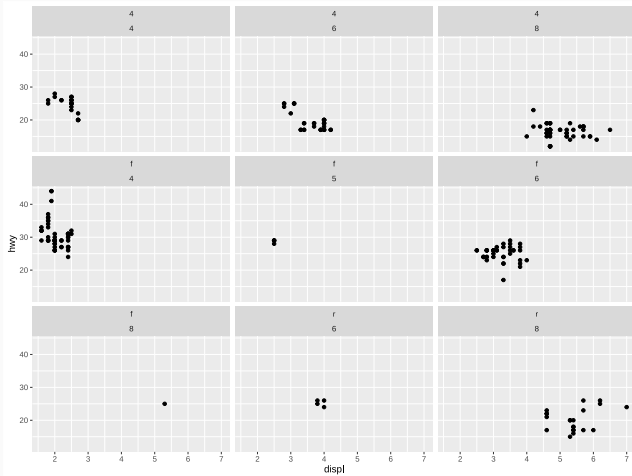
封装分面，先生成一维的面板系列，再封装到二维中。

- 分面形式: `~ 分类变量, ~ 分类变量 1 + 分类变量 2`
- `scales` 参数设置是否共用坐标刻度, "fixed" (默认, 共用), "free" (不共用), 也可以用 `free_x`, `free_y` 单独设置
- 参数 `nrow` 和 `ncol` 可设置子图的放置方式

```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point() +  
  facet_wrap(~ drv, scales = "free")
```



```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point() +  
  facet_wrap(~ drv + cyl)
```

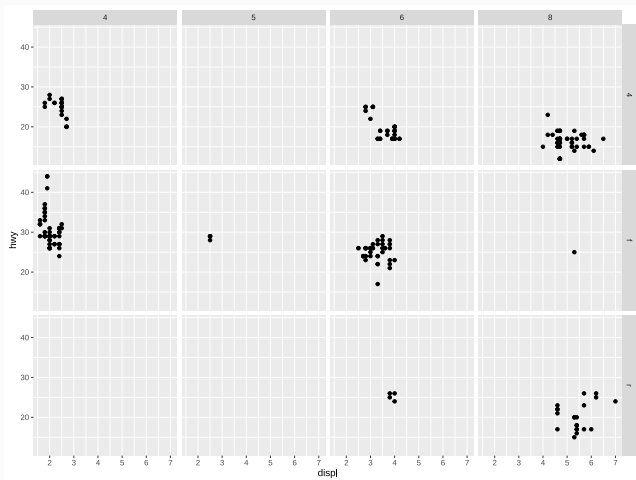


## (2) facet\_grid()

网格分面，生成二维的面板网格，面板的行与列通过分面变量定义。

- 分面形式：行分类变量 ~ 列分类变量

```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point() +  
  facet_grid(drv ~ cyl)
```



## 2. 主题 (theme)

你可以为图形选择不同风格的主题 (外观), `ggplot2` 提供了 8 套可选主题<sup>1</sup>:

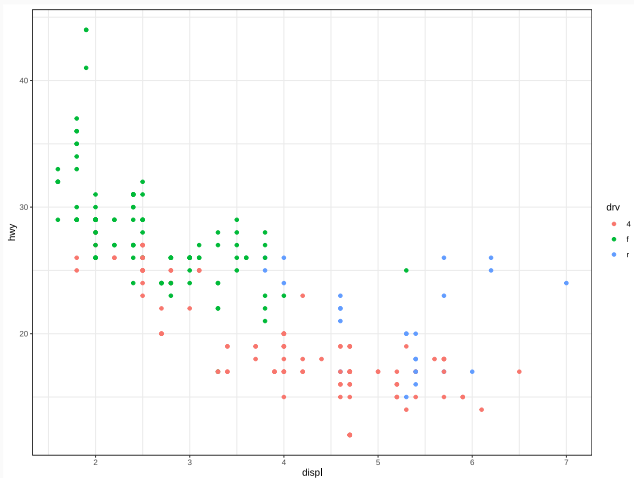
- `theme_bw()`
- `theme_light()`
- `theme_classic()`
- `theme_gray()`: 默认
- `theme_linedraw()`
- `theme_dark()`
- `theme_minimal()`
- `theme_void()`

---

<sup>1</sup>也可以用 `theme()` 函数定制自己的主题, 更多的主题还有 `ggthemes`, `ggpubr` 包, 其中包含一些顶级期刊专用绘图主题.

- 使用或修改主题，只需要添加主题图层

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





### 3. 输出 (output)

- 用 `ggsave()` 函数, 将当前图形保存为想要格式的图形文件, 如 png, pdf 等

```
ggsave("my_plot.pdf", width = 8, height = 6, dpi = 300)
```

**注:** 参数 `width` 和 `height` 通常只设置其中一个, 另一个自动, 以保持原图形宽高比。

关于图形中使用中文字体导出到 pdf 等图形文件出现乱码问题的解决办法。

出现中文乱码是因为 R 环境只载入了“sans (Arial)”“serif (Times New Roman)”“mono (Courier New)”三种英文字体，没有中文字体可用。

解决办法是用 `showtext` 包从系统字体中载入中文字体。

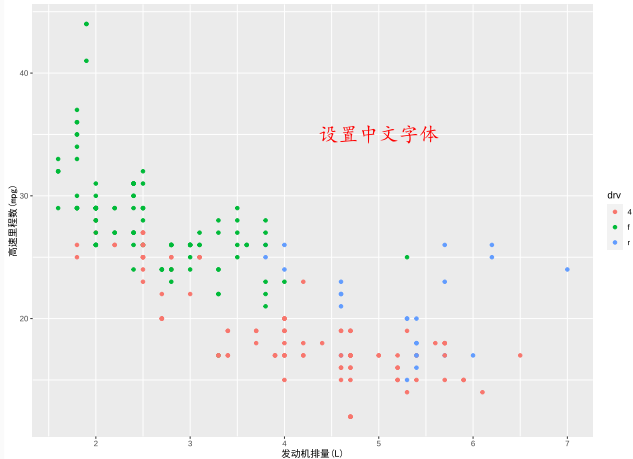
- `font_paths()`: 查看系统字体路径，windows 默认是 `C:\Windows\Fonts`
- `font_files()`: 查看系统自带的所有字体文件
- `font_add()`: 从系统字体中载入字体，需提供 `family` 名字，字体路径

载入字体后，再执行一下 `showtext_auto()` (自动启用/关闭功能)，就可以使用该字体了。

`ggplot2` 中各种设置主题、文本相关的函数 `*_text()`，`annotate()` 等，都提供了 `family` 参数，设定为 `font_add()` 中一致的 `family` 名字即可。

```
library(showtext)
font_add("heiti", "simhei.ttf")
font_add("kaiti", "simkai.ttf")
showtext_auto()
ggplot(mpg, aes(displ, hwy, color = drv)) +
  geom_point() +
  theme(axis.title = element_text(family = "heiti"),
        plot.title = element_text(family = "kaiti")) +
  xlab(" 发动机排量 (L)") +
  ylab(" 高速里程数 (mpg)") +
  ggtitle(" 汽车发动机排量与高速里程数") +
  annotate("text", 5, 35, family = "kaiti", size = 8,
         label = " 设置中文字体", color = "red")
ggsave("images/font_example.pdf", width = 7, height = 4)
```

汽车发动机排量与高速里程数



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*.