

# R 语言编程：基于 tidyverse

## 第 15 讲 ggplot2 绘图 I

---

张敬信

2022 年 12 月 6 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**。

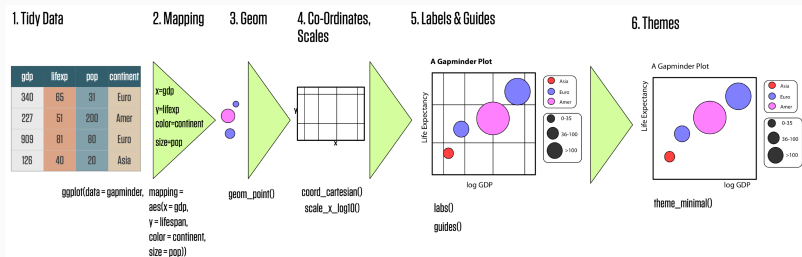


图 1: ggplot2 绘图流程

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)
```

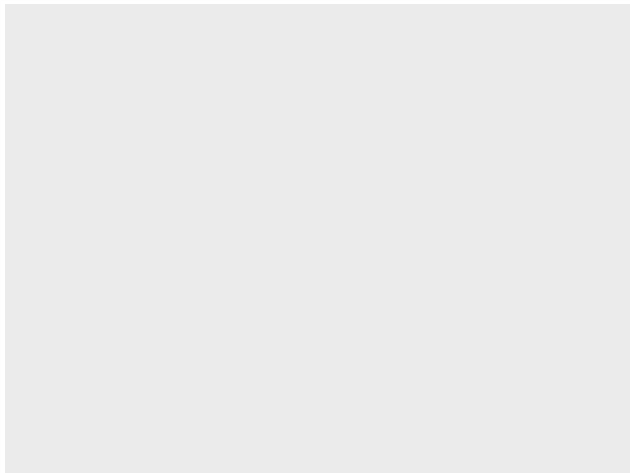
```
mpg
```

```
#> # A tibble: 234 x 11
```

```
#>   manufacturer model displ  year   cyl trans      drv
#>   <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

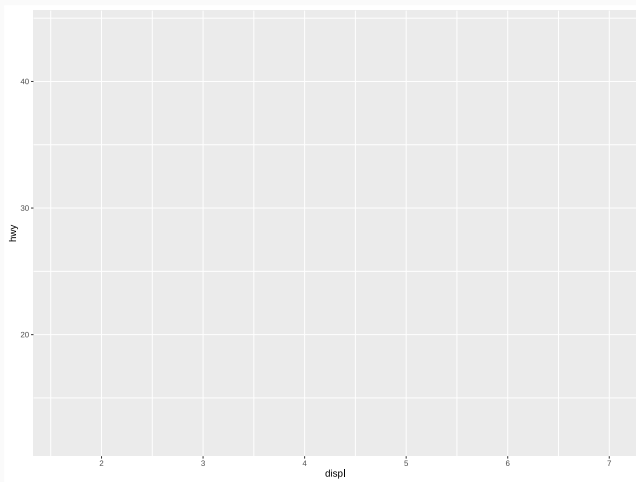


- 最需要的美学是 `x` 和 `y`, 分别映射到变量 `displ` 和 `hwy`, 再将美学 `color` 映射到 `drv`<sup>1</sup>, 此时图形就有了坐标轴和网格线:

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

---

<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.



- 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)

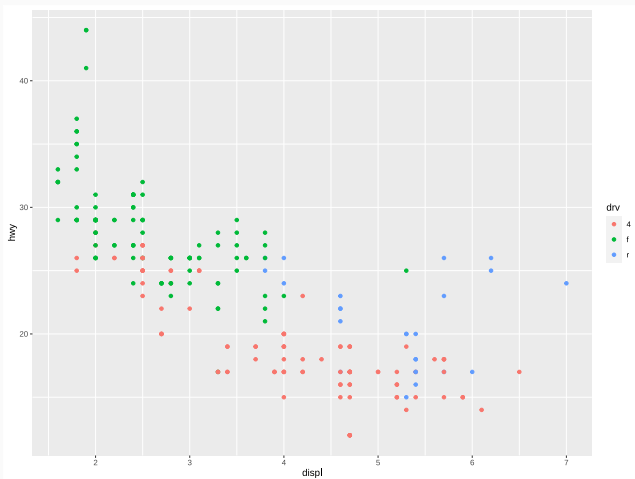
每个图形都是采用不同的视觉对象来表达数据，称为是“几何对象”。不同类型的“几何对象”从不同角度来表达数据，如散点图、平滑曲线、折线图、条形图、箱线图等。

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()/geom_hline()/geom_vline`: 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()
```

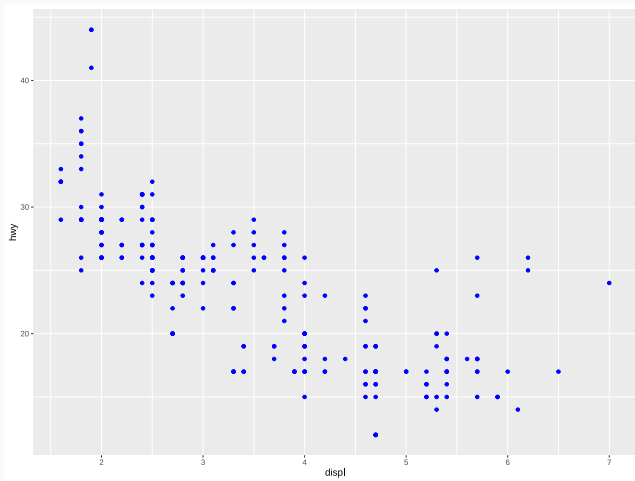


- 美学映射也可以放在几何对象中，效果同上：

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

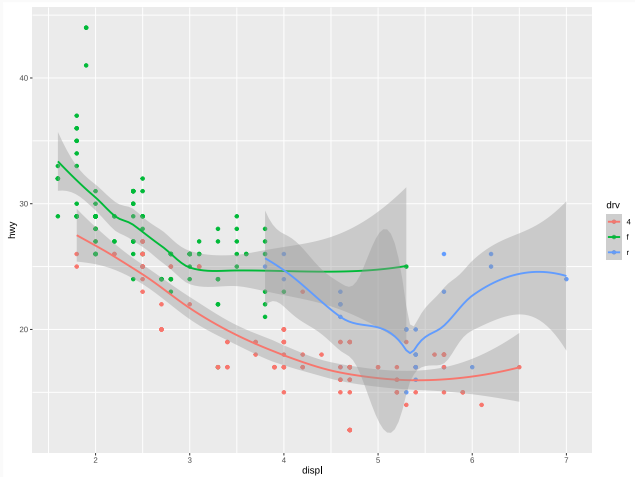
- 为图形美学设置特定值，但注意不能放在映射 aes() 中

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



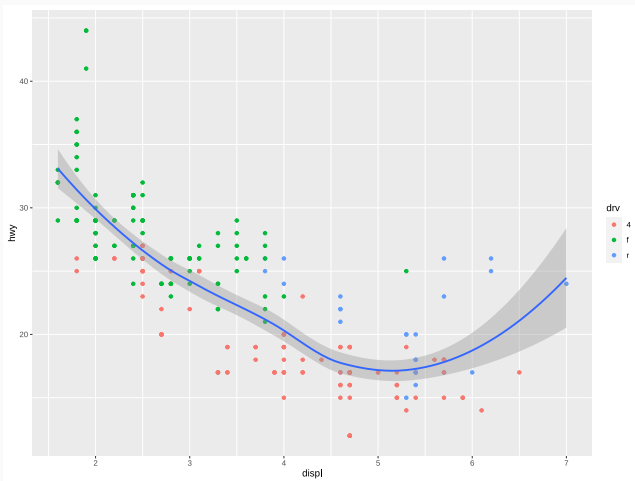
- 图层是依次叠加的，再添加一个几何对象：光滑曲线，区分两种写法：

```
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("data/ecostats.rda")
```

```
ecostats
```

```
#> # A tibble: 527 x 7
```

```
#>   Region Year Electricity Investment Consumption Population
```

```
#>   <chr>   <int>         <dbl>         <dbl>         <dbl>
```

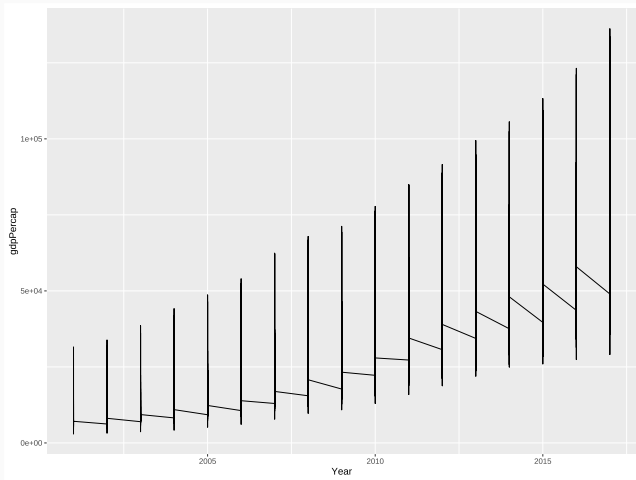
```
#> 1 安徽      2001          360.           893.           2739
```

```
#> 2 北京      2001          400.          1513.           9057
```

```
#> 3 福建      2001          439.          1173.           4770
```

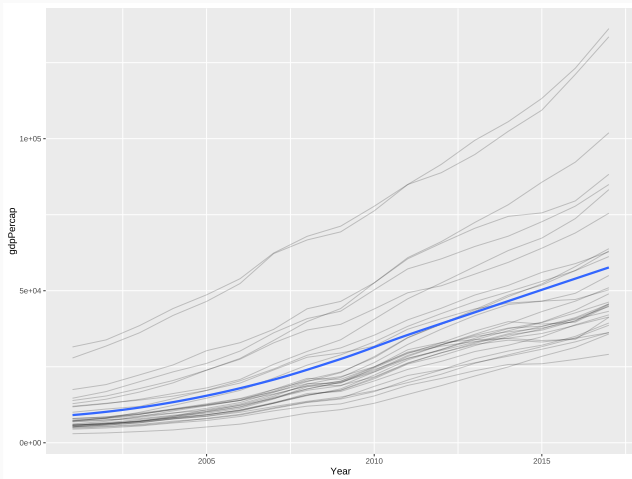
```
#> # ... with 524 more rows
```

```
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

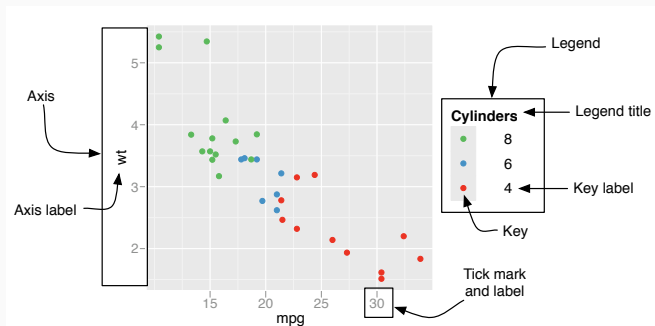


图 2: 图例与坐标轴的组件

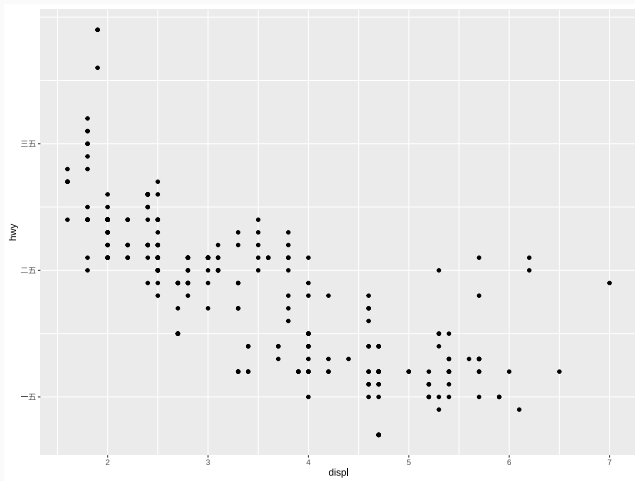


## 1. 修改坐标轴刻度及标签

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

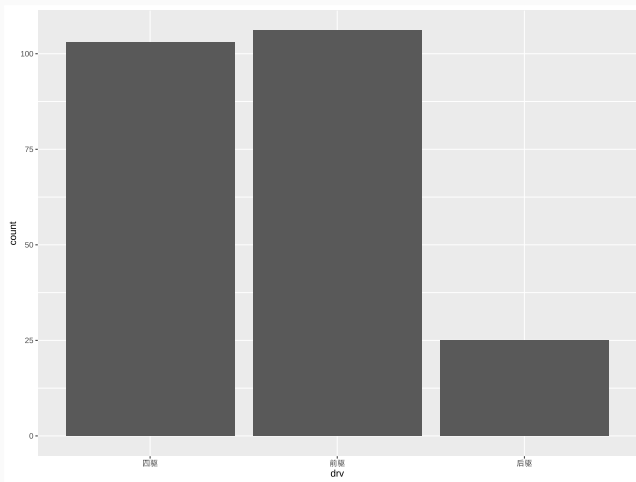
- 参数 `breaks` 设置各个刻度的位置
- 参数 `labels` 设置各个刻度对应的标签

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



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"))
```



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>
```

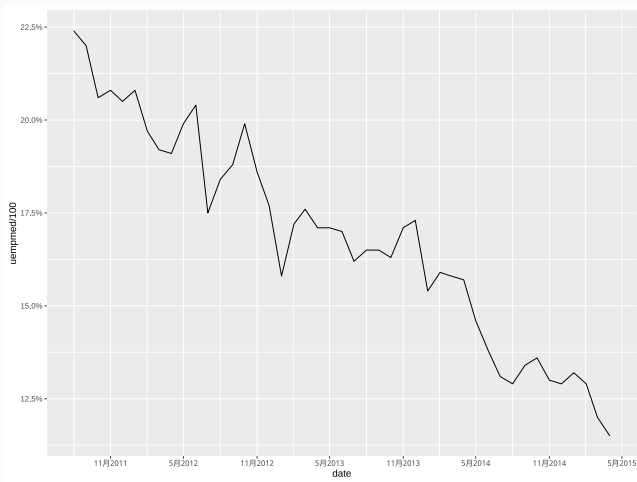
```
#> 1 1967-07-01  507. 198712    12.6     4.5    2944
```

```
#> 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
```

```
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)
```



## 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”.



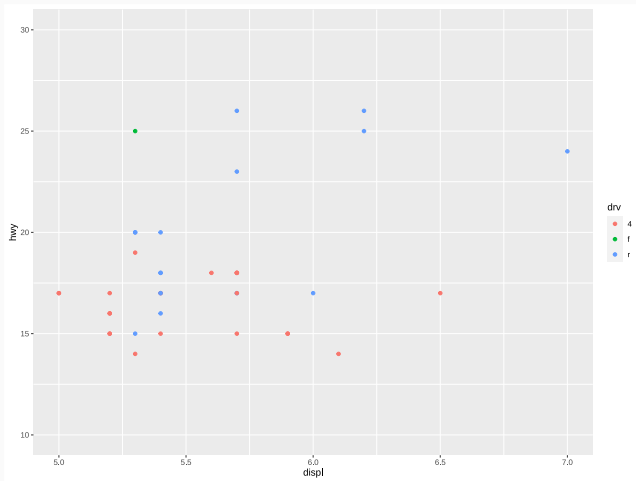
```
ggplot(mpg, aes(displ, hwy)) +  
  geom_point(aes(color = drv)) +  
  labs(x = " 引擎大小 (L)", y = " 高速燃油率 (mpg)",  
        color = " 驱动类型") +      # 或者  
  # xlab(" 引擎大小 (L)") + ylab(" 高速燃油率 (mpg)")  
  theme(legend.position = "top")
```



### 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)
```



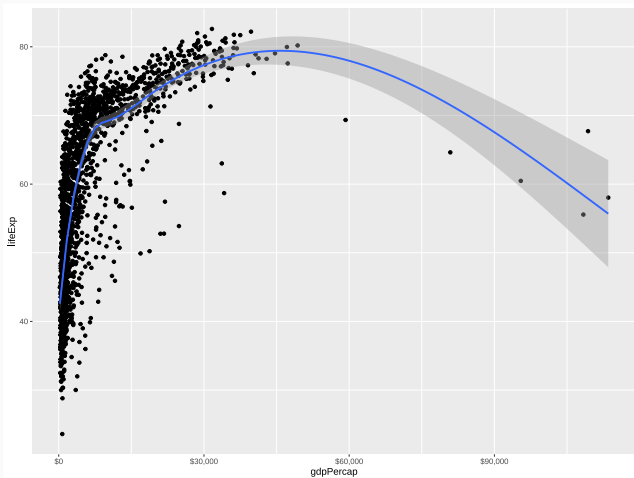
## 4. 变换坐标轴

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。

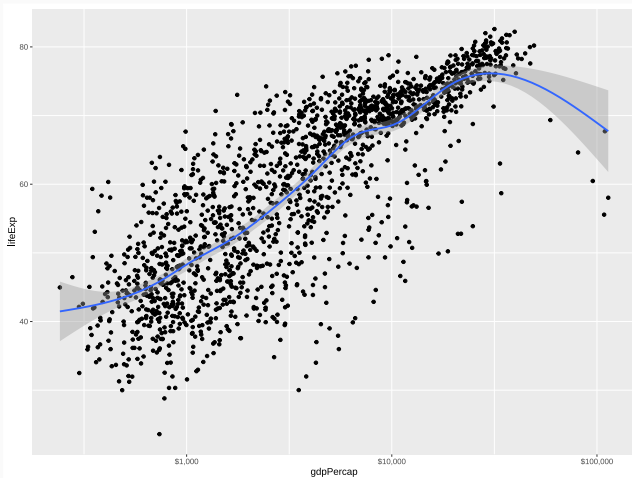
ggplot2 提供的坐标变换函数 `scale_x_log10()` 等是变换坐标系，能够在视觉效果相同的情况下，使用原始数据的坐标刻度

```
load("data/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

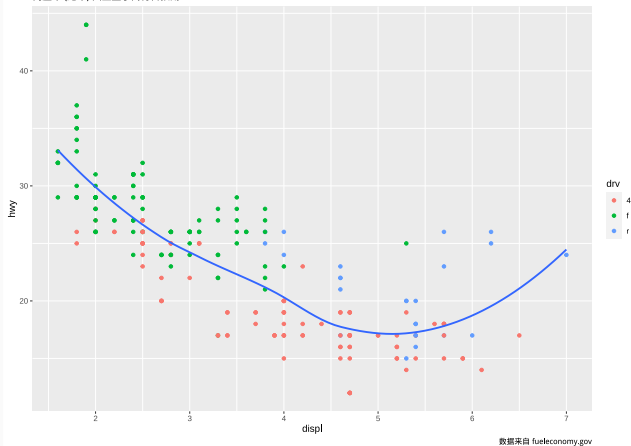
用 `labs()` 函数的参数 `title`, `subtitle`, `caption` 设置标题、副标题、脚注标题（默认右下角）

```
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

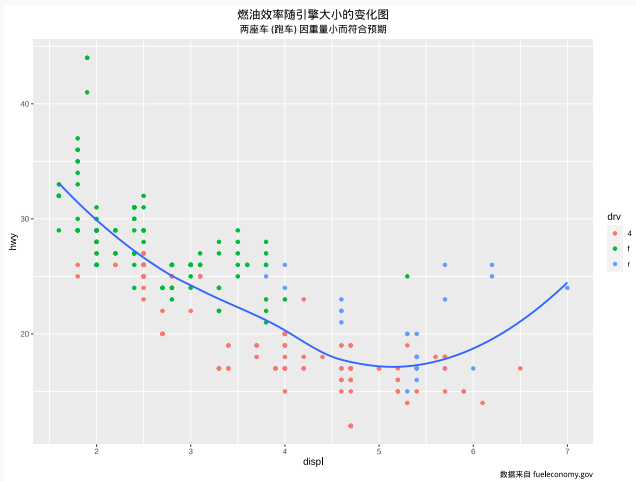


燃油效率随引擎大小的变化图  
两驱车(跑车) 因重量小而符合预期



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<sup>2</sup>。

### (1) 离散变量

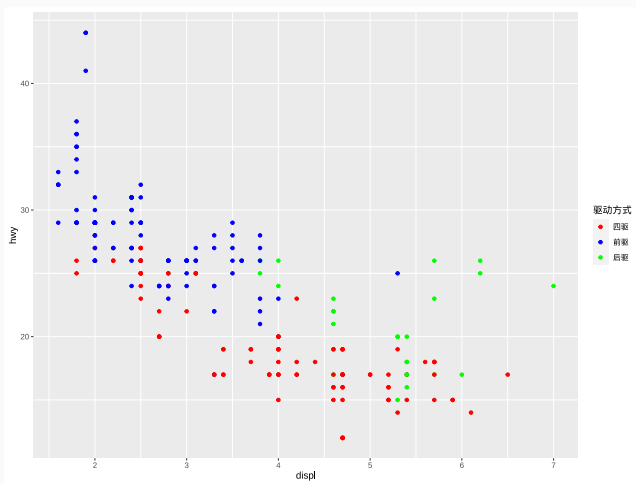
- manual: 直接指定分组使用的颜色
- hue: 通过改变色相 (hue) 饱和度 (chroma) 亮度 (luminosity) 来调整颜色
- brewer: 使用 ColorBrewer 的颜色
- grey: 使用不同程度的灰色

---

<sup>2</sup>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)`.

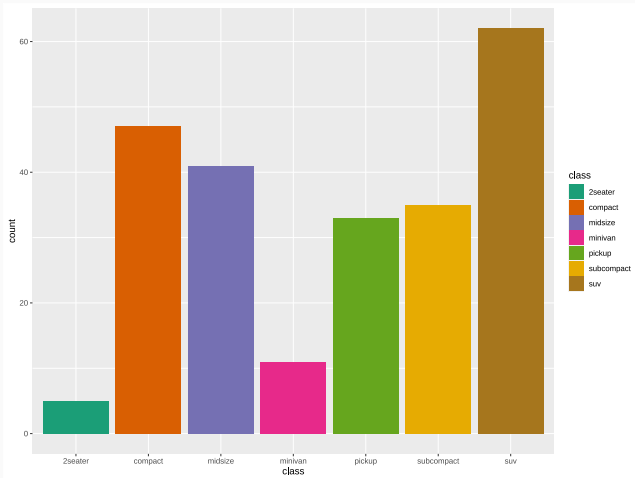
用 `scale_*_manual()` 手动设置颜色, 并修改图例及其标签

```
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"))
```



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
```

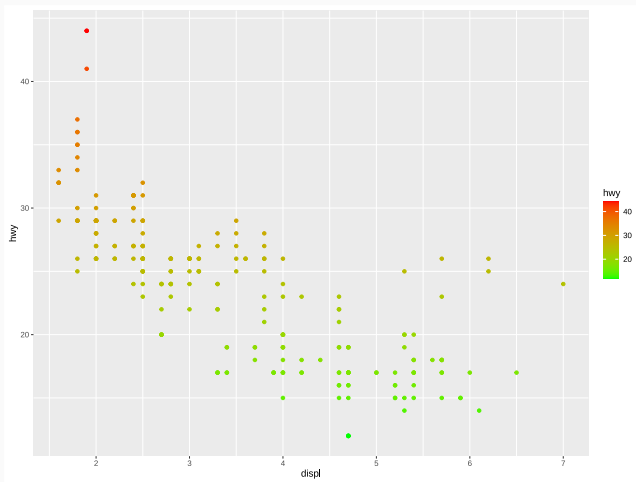


## (2) 连续变量

- `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

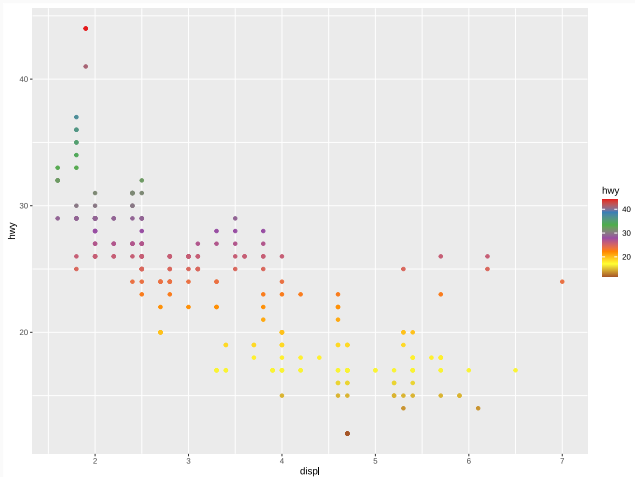
```
ggplot(mpg, aes(displ, hwy, color = hwy)) +  
  geom_point() +  
  scale_color_gradient(low = "green", high = "red")
```





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")
```

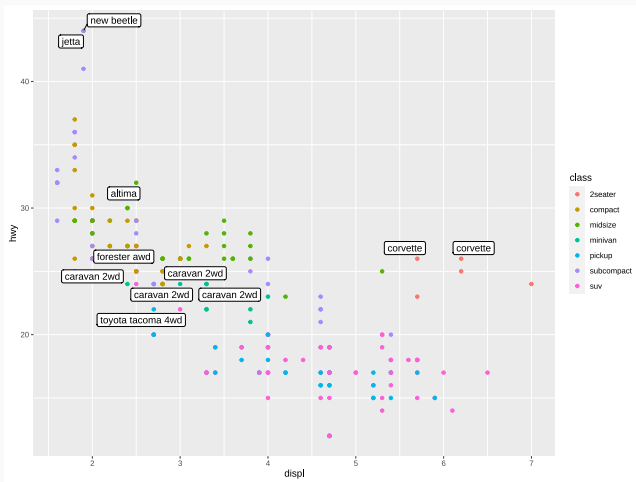


## 7. 添加文字标注

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))
```



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")
```



## 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：

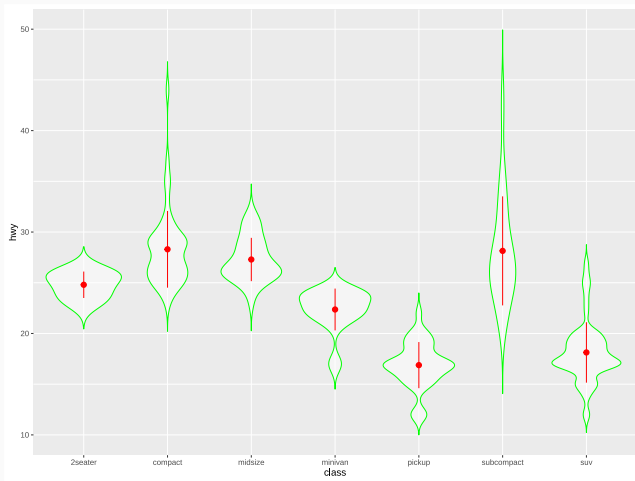
- 可在几何对象函数 `geom_*`() 中创建，通常直接使用后者即可：
  - `stat_bin()`: `geom_bar()`, `geom_freqpoly()`,  
`geom_histogram()`
  - `stat_bindot()`: `geom_dotplot()`
  - `stat_boxplot()`: `geom_boxplot()`
  - `stat_contour()`: `geom_contour()`
  - `stat_quantile()`: `geom_quantile()`
  - `stat_smooth()`: `geom_smooth()`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

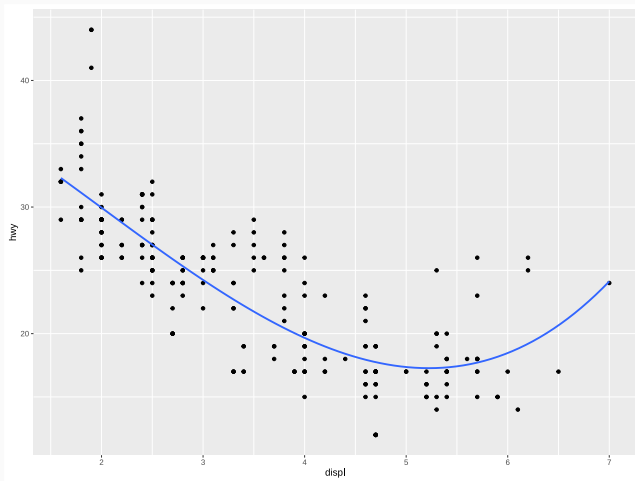
```
ggplot(mpg, aes(x = class, y = hwy)) +  
  geom_violin(trim = FALSE, alpha = 0.5, color = "green") -  
  stat_summary(fun = mean,  
               fun.min = function(x) {mean(x) - sd(x)},  
               fun.max = function(x) {mean(x) + sd(x)},  
               geom = "pointrange", color = "red")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



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：

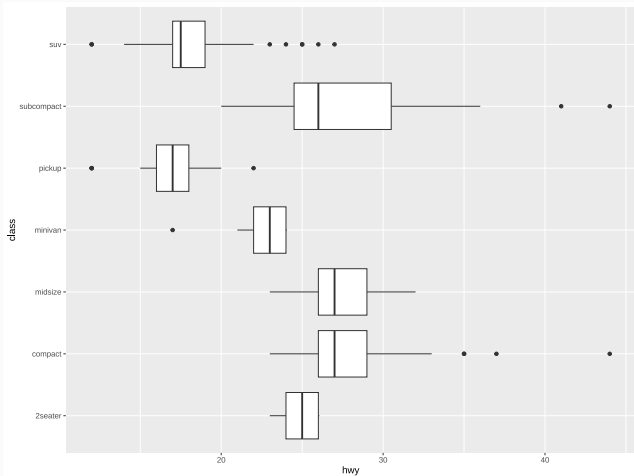
- `coord_flip()`: 坐标轴翻转，即 x 轴与 y 轴互换，比如绘制水平条形图
- `coord_fixed()`: 固定  $\text{ratio} = y / x$  的比例
- `coord_polar()`: 转化为极坐标系，比如条形图转为极坐标系即为饼图
- `coord_trans()`: 彻底的坐标变换，不同于 `scale_x_log10()` 等
- `coord_map()`, `coord_quickmap()`: 与 `geom_polygon()` 连用，控制地图的坐标投影
- `coord_sf()`: 与 `geom_sf()` 连用，控制地图的坐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

- 坐标轴翻转，从水平图到竖直图：

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

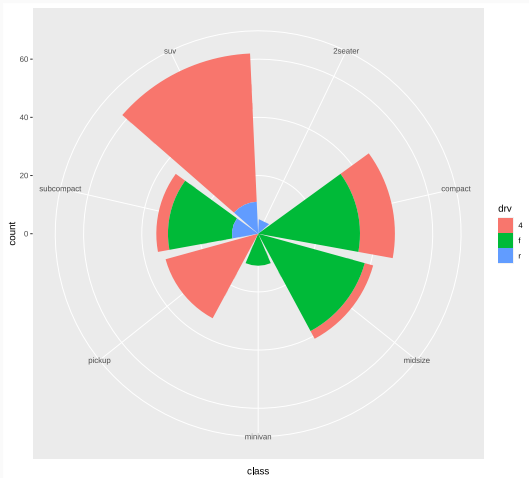
# 箱线图

# 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()
```



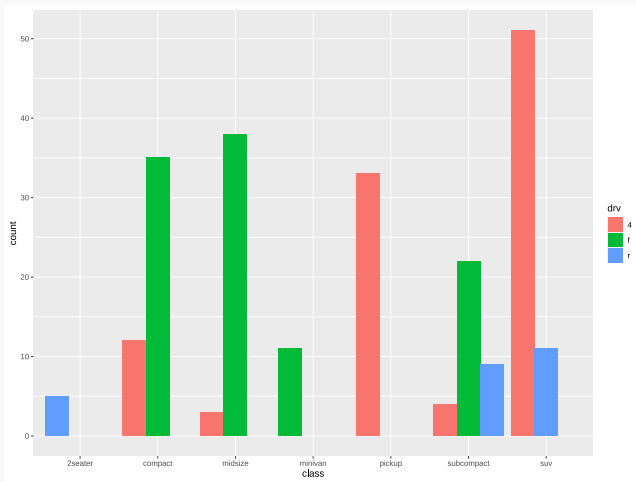
### 3. 位置调整 (Position adjustments)

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")
```

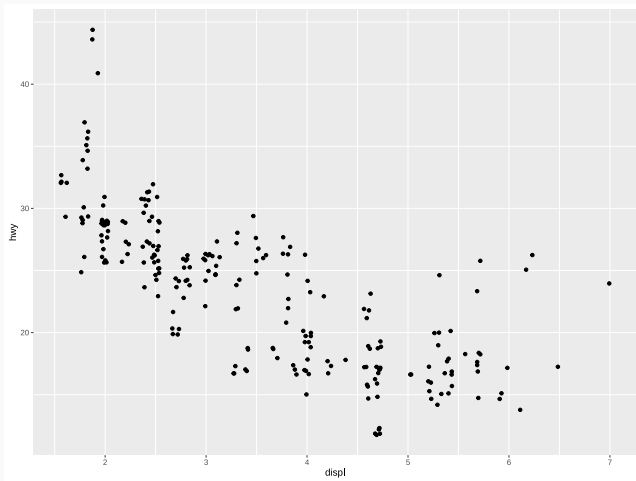




散点图中的散点位置调整:

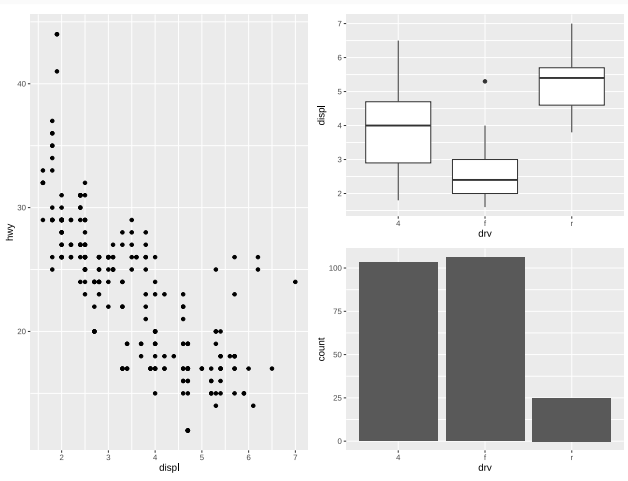
- `position_nudge()`: 将散点移动固定的偏移量
- `position_jitter()`: 给每个散点增加一点随机噪声 (抖散图)
- `position_jitterdodge()`: 增加一点随机噪声并躲避组内的点, 特别用于箱线图 + 散点图

```
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)

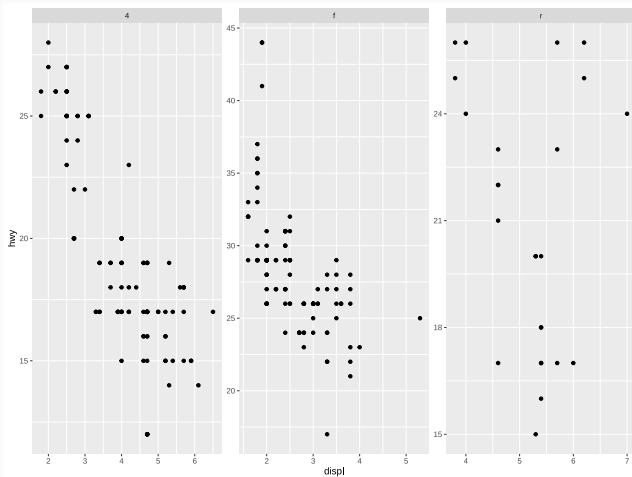
利用分类变量将图形分为若干个“面”（子图），即对数据分组再分别绘图，称为“分面”。

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

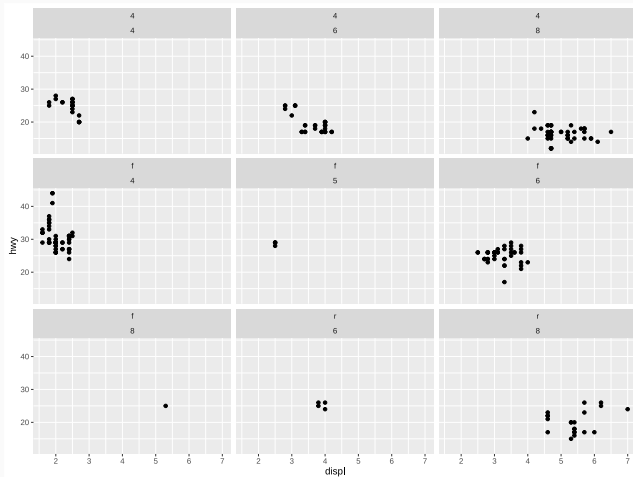
封装分面，先生成一维的面板系列，再封装到二维中。

- 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)
```



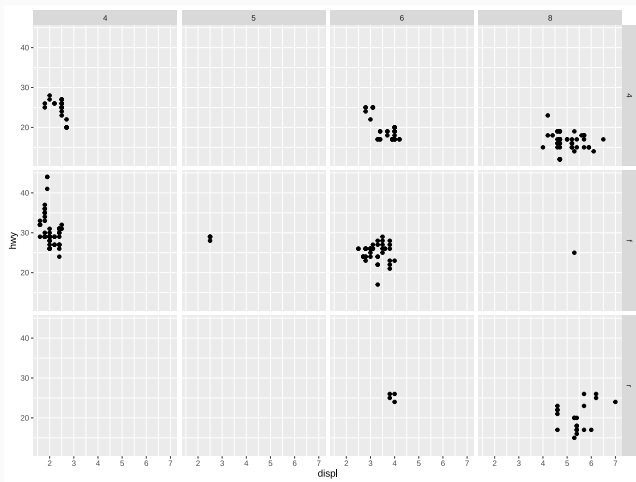


## (2) facet\_grid()

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>3</sup>:

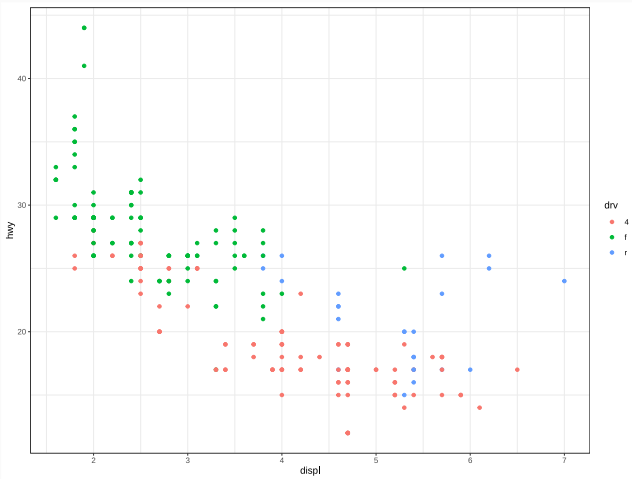
- `theme_bw()`
- `theme_light()`
- `theme_classic()`
- `theme_gray()`: 默认
- `theme_linedraw()`
- `theme_dark()`
- `theme_minimal()`
- `theme_void()`

---

<sup>3</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

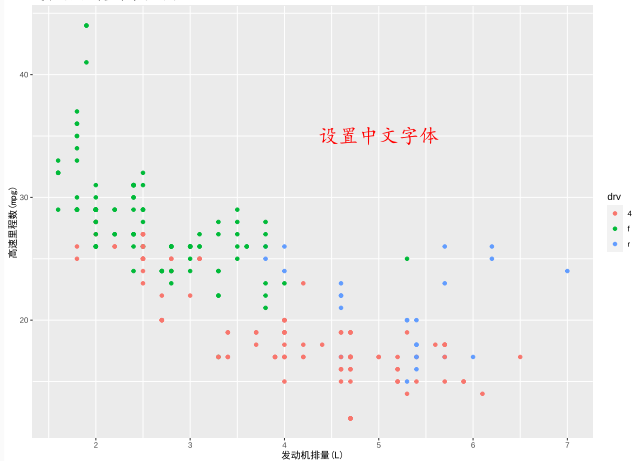
载入字体后，再执行一下 `showtext_auto()` (自动启用/关闭功能)，就可以使用该字体了。

`ggppplot2` 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), (王敏杰, 2021), 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.

王敏杰 (2021). 数据科学中的 R 语言. bookdown.org.

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

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