# 第 3 章 数据可视化

## Chapter 3 Data visualization

Qingyao Zhang

2026-04-29

## 目录

# 1 ggplot2 简介

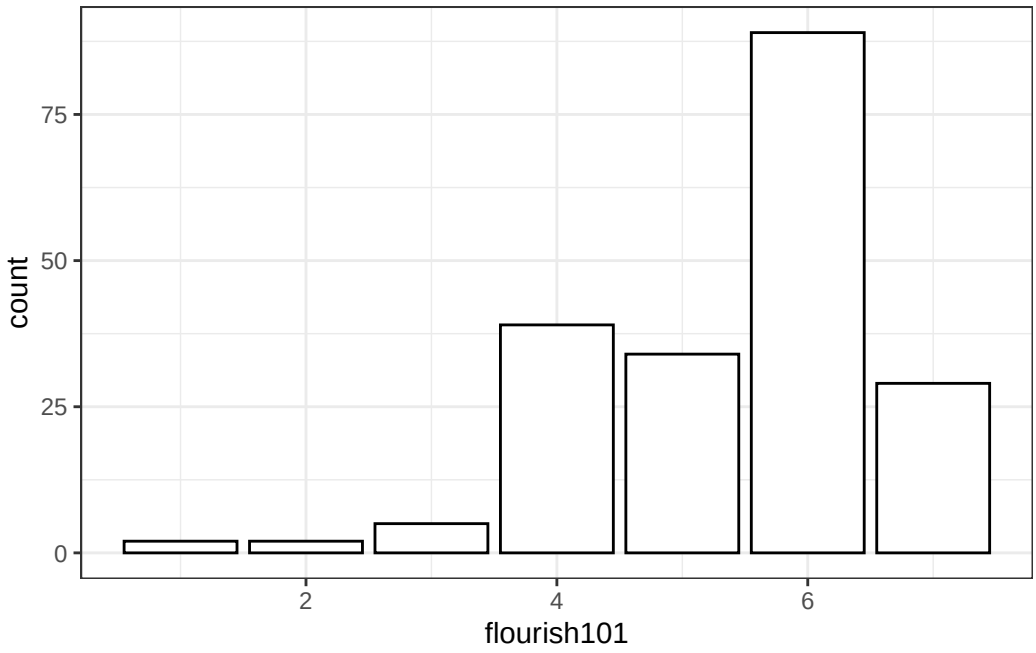
# 2 频数分布

## 2.1 简单频数分布与条形图

```
# 加载 well 数据
data("well", package = "Keng")
# flourish101 的得分
# flourish101 是心盛量表第 1 个时间点的第 1 个题项
well$flourish101
##   [1] 7 4 6 7 6 6 3 7 6 6 6 5 4 5 3 4 7 7 5 5 6 6 3 1 5 5 6 4 6 6 4 6 6 4 4 6 6
##  [38] 4 6 7 5 6 6 5 7 4 6 6 5 5 6 6 5 6 6 2 4 5 4 4 7 6 6 4 4 4 7 4 5 6 6 4 4 7
##  [75] 6 6 6 3 5 6 6 5 6 7 5 6 6 6 7 6 6 6 6 5 4 5 6 4 5 6 1 6 6 5 4 5 7 4 2 6 7
## [112] 6 6 4 7 7 7 6 5 6 6 6 4 6 6 4 7 5 7 5 4 7 7 6 7 4 5 6 5 6 6 7 6 4 6 4 5 6
## [149] 5 6 6 6 6 6 6 5 6 4 7 6 4 4 6 6 7 6 6 6 4 7 5 6 6 6 6 6 6 7 5 6 5 4 4 4 6
## [186] 6 7 5 6 4 6 6 6 7 5 6 6 4 3 4
```

```
# flourish101 的频数
# 参数 useNA：若存在缺失值，则统计缺失值的频数。
table(well$flourish101, useNA = "ifany")
##
##  1  2  3  4  5  6  7
##  2  2  5 39 34 89 29
```

```
# 作图
library(ggplot2)
ggplot(data = well, mapping = aes(x = flourish101)) +
  geom_bar()
```

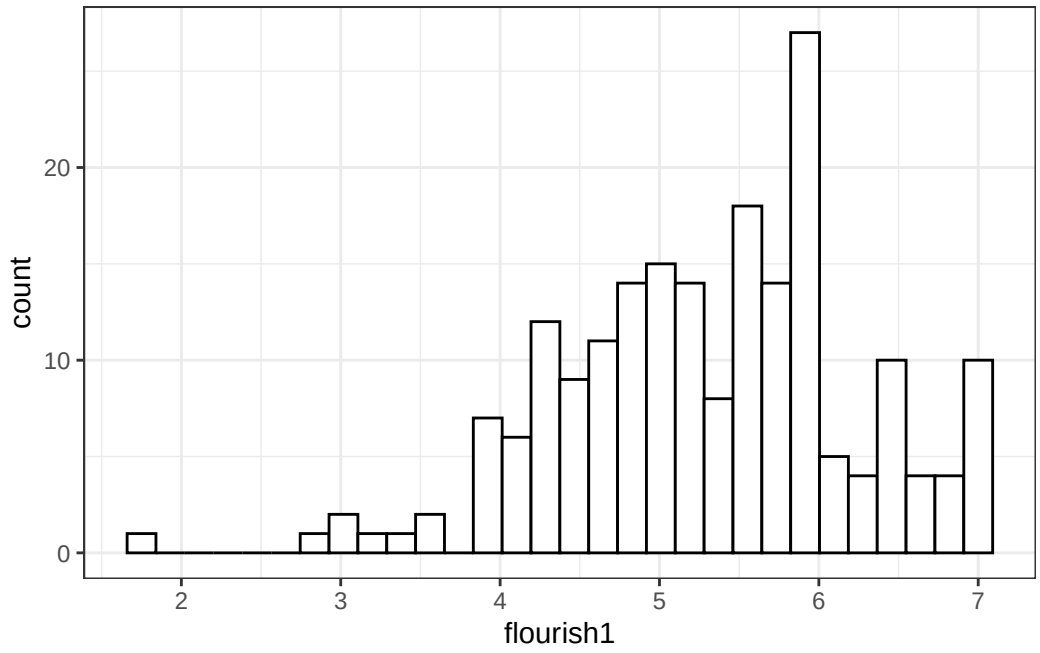



## 2.2 分组频数分布与直方图

```
range(well$flourish1)
## [1] 1.75 7.00
```

```
ggplot(well, aes(x = flourish1)) +
  geom_histogram()
```

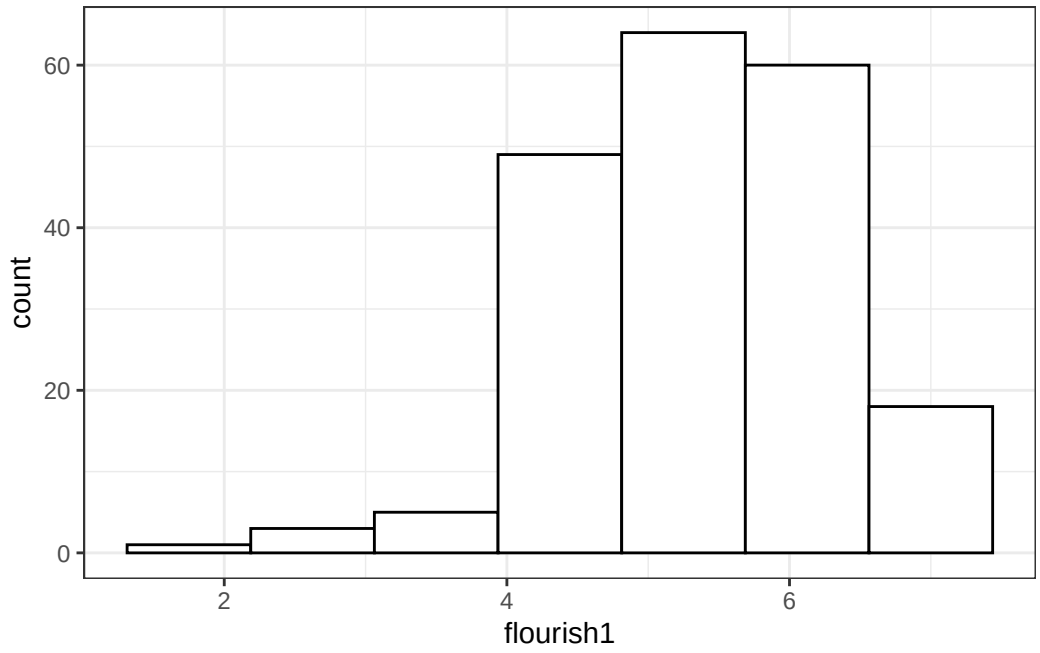
```
## `stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```




```
## `stat_bin()` using `bins = 30`. Pick better value `binwidth`.
```

```
fig_3_3 <- ggplot(well, aes(x = flourish1)) +
  geom_histogram(bins = 7)
fig_3_3
```

```
fig_3_3 <- ggplot(well, aes(x = flourish1)) +
  geom_histogram(bins = 7,
                 fill = "white",
                 color = "black") +
  theme_bw()
fig_3_3
```



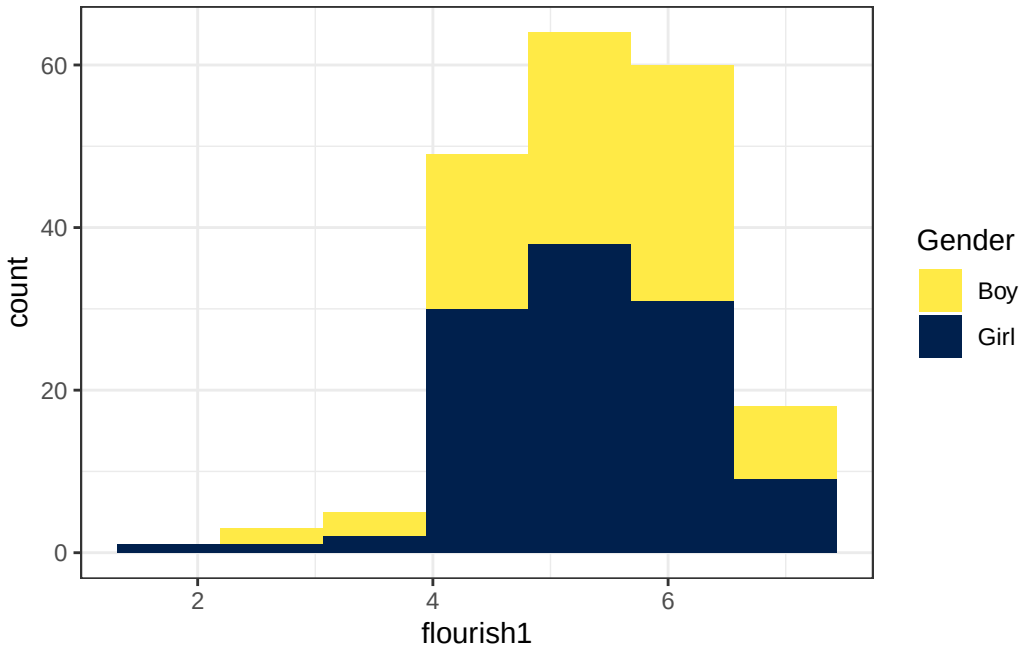

```r
ggsave("~/inf/心理与教育统计 _ 教材/Figs/fig_3_3.png", width = 6, height = 4, dpi = 600)
```

```r
# 将 ggplot_build 提取的内容存储在 ggplot_build_out 中
ggplot_build_out <- ggplot_build(fig_3_3)
# 使用 $ 提取 data，查看其中的第 [[1]] 个数据框的前 4 列
ggplot_build_out$data[[1]][1:4]
##   count     x   xmin   xmax
## 1     1 1.750 1.3125 2.1875
## 2     3 2.625 2.1875 3.0625
## 3     5 3.500 3.0625 3.9375
## 4    49 4.375 3.9375 4.8125
## 5    64 5.250 4.8125 5.6875
## 6    60 6.125 5.6875 6.5625
## 7    18 7.000 6.5625 7.4375
```

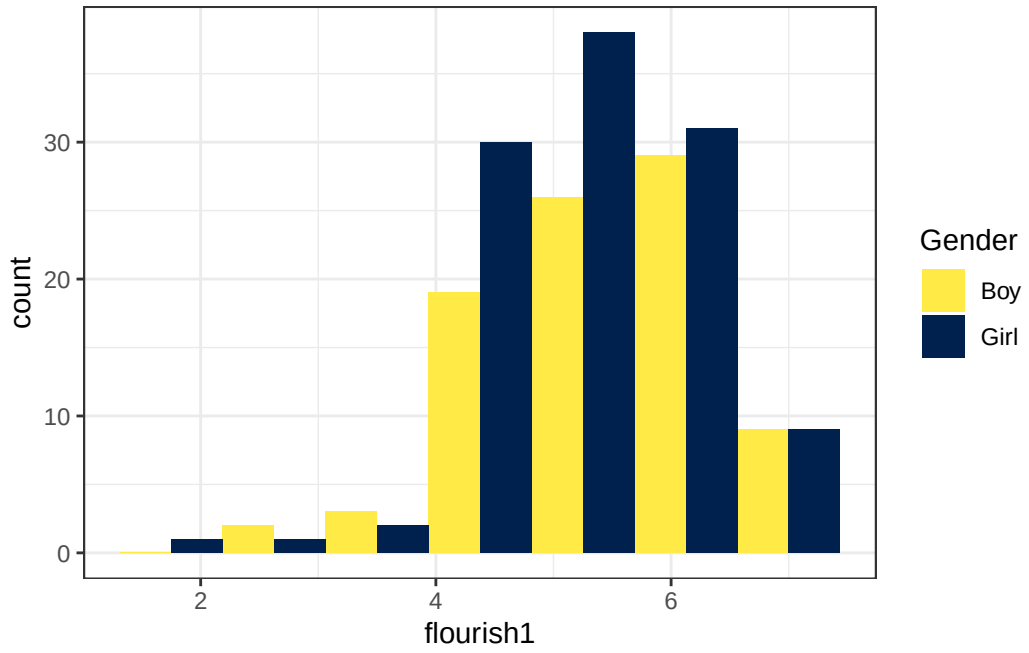
### 2.3 频数分布的差异

```r
well$Gender <- factor(well$gender100, labels = c("Boy", "Girl"))
```

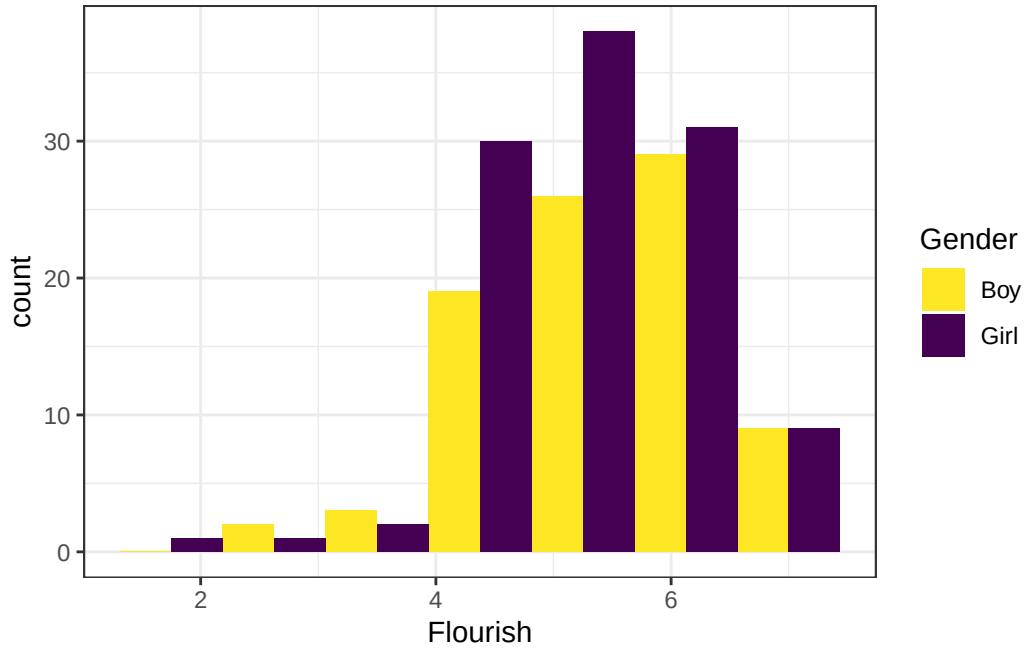
```r
ggplot(well, aes(x = flourish1, fill = Gender)) +
  geom_histogram(bins = 7)
```



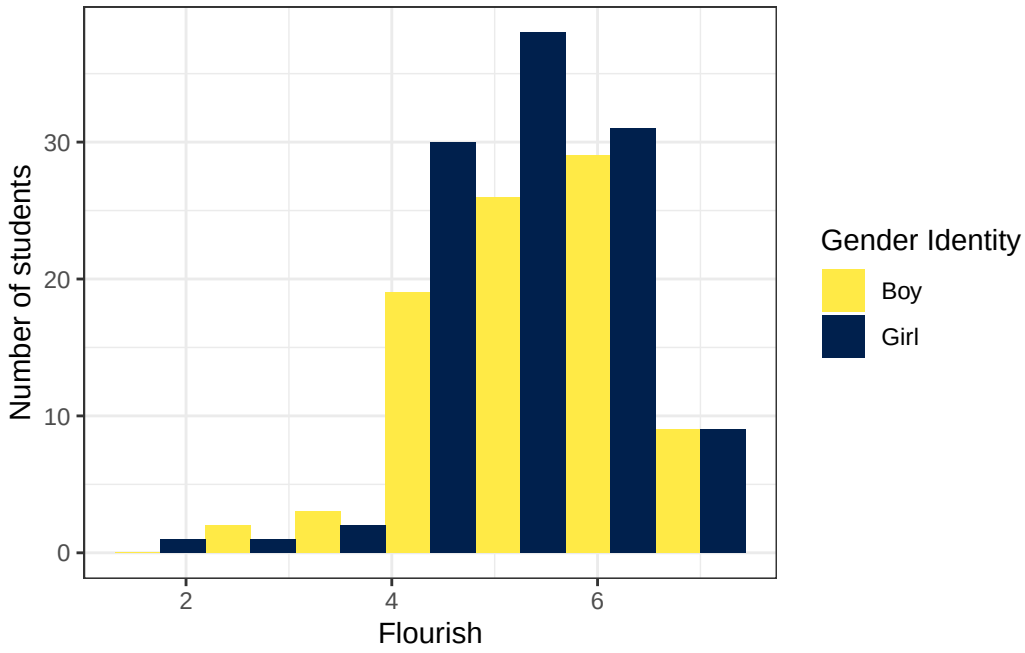

```r
ggplot(well, aes(x = flourish1, fill = Gender)) +
  geom_histogram(bins = 7, position = "dodge")
```




```
ggplot(well, aes(x = flourish1, fill = Gender)) +
  geom_histogram(bins = 7, position = "dodge") +
  scale_x_continuous(name = "Flourish")
```



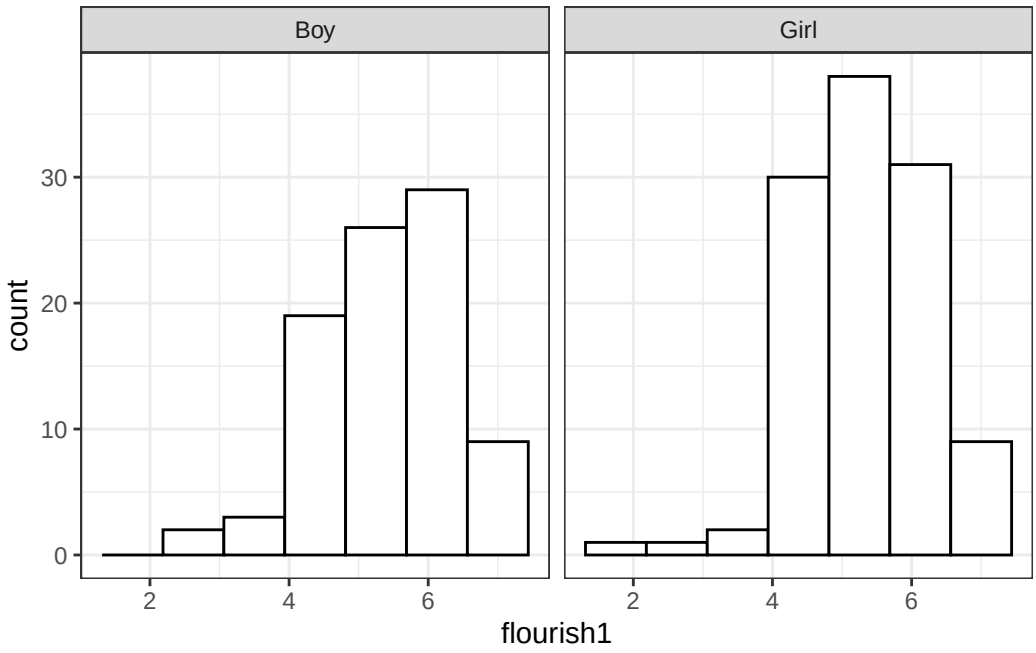
```
ggplot(well, aes(x = flourish1, fill = Gender)) +
  geom_histogram(bins = 7, position = "dodge") +
  labs(x = "Flourish", y = "Number of students", fill = "Gender Identity")
```




有些读者可能尝试了将量尺的名称修改为中文，却发现图中的中文无法正常显示。解决该问题的最简单的方法是：图中文字用英文或字母缩写，然后在图注中用中文注明图中英文或字母缩写的含义。这一做法同时也有利于保持统计图的简洁。解决该问题的另一方法是使用 `showtext` 程序包，该方法略微复杂，请参考本书数字资源。

比较性别差异的另一种方式是使用分面函数 `facet_wrap()` 将数据分成男、女两组，然后作出两幅子图。在下面的代码中，参数 `facets = vars(gender)` 设定按性别分组，其中 `vars()` 的作用是声明在 `well` 数据中寻找 `Gender` 变量。

```
ggplot(well, aes(x = flourish1)) +
  geom_histogram(bins = 7) +
  facet_wrap(facets = vars(Gender))
```




facet_wrap() 的常用参数还包括 labeller、nrow、ncol。请读者自行将 labeller 的值修改为"label_both" 并运行代码，观察"label_both" 带来的变化。

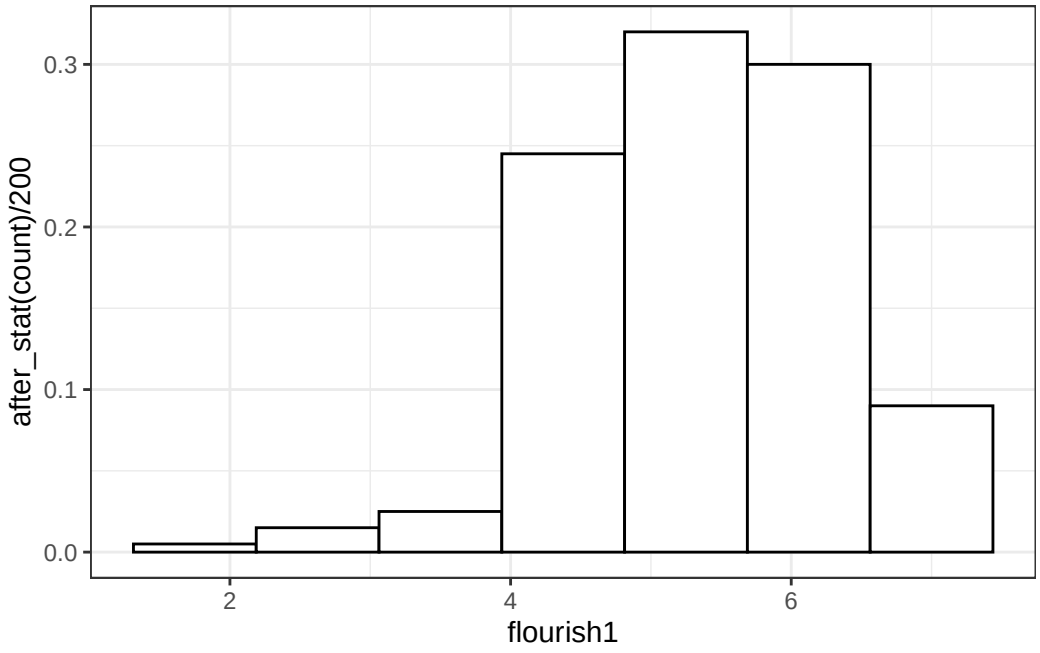
```
ggplot(well, aes(x = flourish1)) +
  geom_histogram(bins = 7) +
  facet_wrap(facets = vars(Gender), labeller = "label_both")
```

## 2.4 相对频数分布

在比较不同性别的频数分布时，我们容易发现这样的问题：男生、女生的总人数不同，直接比较频数意义不大，我们需要比较不同性别的相对频数（比例）。下面我们以抑郁得分 flourish1 为例绘制相对频数分布图。我们可以首先使用 nrow() 得到 depress 的样本量为 200，之后我们在 geom_histogram() 函数中增加参数 aes(y = after_stat(count)/200)。geom_histogram() 作图时会自动统计 flourish1 各分数段的频数，这些频数没有外显地输出给我们。我们首先使用 after_stat(count) 将这些频数提取出来，之后将其除以总人数。参数 aes(y = after_stat(count)/200) 的作用是将 y 轴上的变量从默认的频数 count 修改为相对频数。注意，下面使用了两组 aes()，ggplot() 中的 aes() 所设定的映射关系对后续所有函数皆有效，geom_histogram() 中的 aes() 所设定的映射关系仅对当前函数有效。

```
nrow(well)
ggplot(well, aes(x = flourish1)) +
  geom_histogram(aes(y = after_stat(count)/200),
                 bins = 7)
```

```
## [1] 200
```




由图可见，直方图的形状没有发生变化，但纵坐标缩小成了比例，最大值约为 $0.20$。

## 2.5 累积概率分布

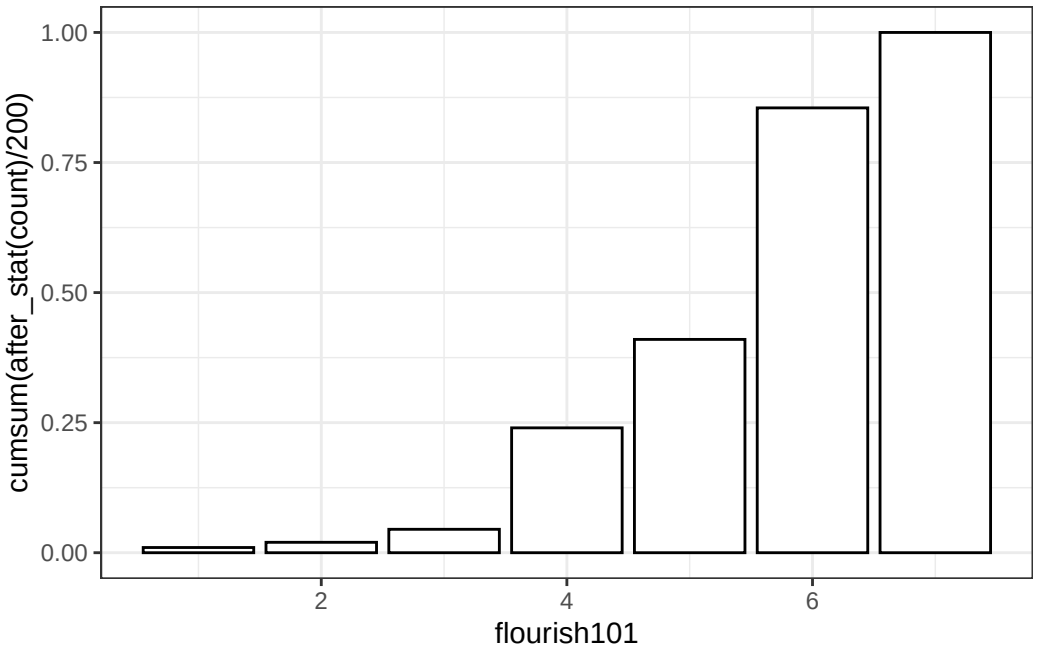
累积频数与累积概率有助于我们了解各分数段人数的变化。累计频数指的是某个得分及以下的人数，累计概率指的是累计频数占总数的比例。

以题项 `flourish101` 为例，其 $4$ 个数值的频数、概率、累积频数与累积概率见下表:

| 序号 | 分数 | 频数 | 概率 | 累计频数 | 累计概率 |
|---|---|---|---|---|---|
| 1 | 1 | 50 | 0.2874 | 50 | 0.2874 |
| 2 | 2 | 101 | 0.5805 | 151 | 0.8678 |
| 3 | 3 | 15 | 0.0862 | 166 | 0.9540 |
| 4 | 4 | 8 | 0.0460 | 200 | 1.0000 |

对于抑郁，我们关心中重度抑郁的人数。$2$ 分及以下的累计频数为 $151$，累计概率为 $86.78\%$，$3$ 分以上的人数为 $23$。即，中重度抑郁的人数约 $23$，占比约 $13.22\%$。我们可以使用累计概率图呈现累计概率随抑郁得分升高的变化趋势。

```r
ggplot(well, aes(x = flourish101)) +
  geom_bar(aes(y = cumsum(after_stat(count)/200)))
```

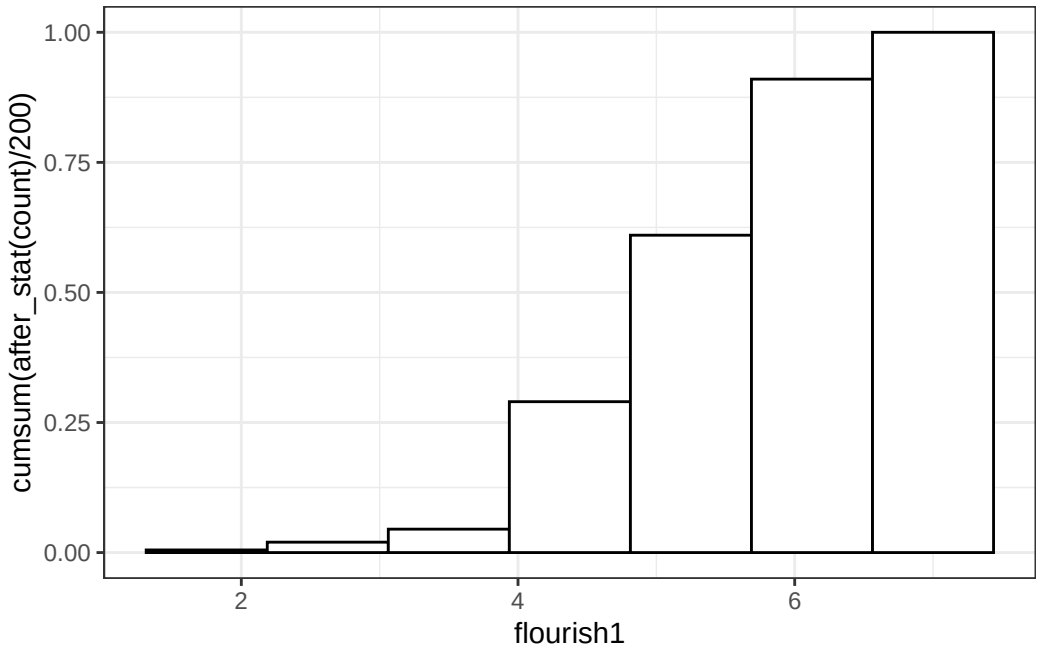



以抑郁得分 `flourish1` 为例，其十个分数段的频数、概率、累积频数与累积概率见下表:

| 序号 | 分数段 | 频数 | 概率 | 累计频数 | 累计概率 |
|---|---|---|---|---|---|
| 1 | (1.05, 1.25] | 4 | 0.0230 | 4 | 0.0230 |
| 2 | (1.25, 1.45] | 14 | 0.0805 | 18 | 0.1034 |
| 3 | (1.45, 1.65] | 34 | 0.1954 | 52 | 0.2989 |
| 4 | (1.65, 1.85] | 30 | 0.1724 | 82 | 0.4713 |
| 5 | (1.85, 2.05] | 34 | 0.1954 | 116 | 0.6667 |
| 6 | (2.05, 2.25] | 29 | 0.1667 | 145 | 0.8333 |
| 7 | (2.25, 2.45] | 16 | 0.0920 | 161 | 0.9253 |
| 8 | (2.45, 2.65] | 6 | 0.0345 | 167 | 0.9598 |
| 9 | (2.65, 2.85] | 4 | 0.0230 | 171 | 0.9828 |
| 10 | (2.85, 3.05] | 3 | 0.0172 | 200 | 1.0000 |

可见，2.45 及以下的累计频数为 161，累计概率为 92.53%，那么，2.45 以上的人数为 13。即，中重度抑郁的人数约 13 ，占比约 7.47%。下面用累计概率图呈现累计概率随抑郁得分升高的变化趋势。

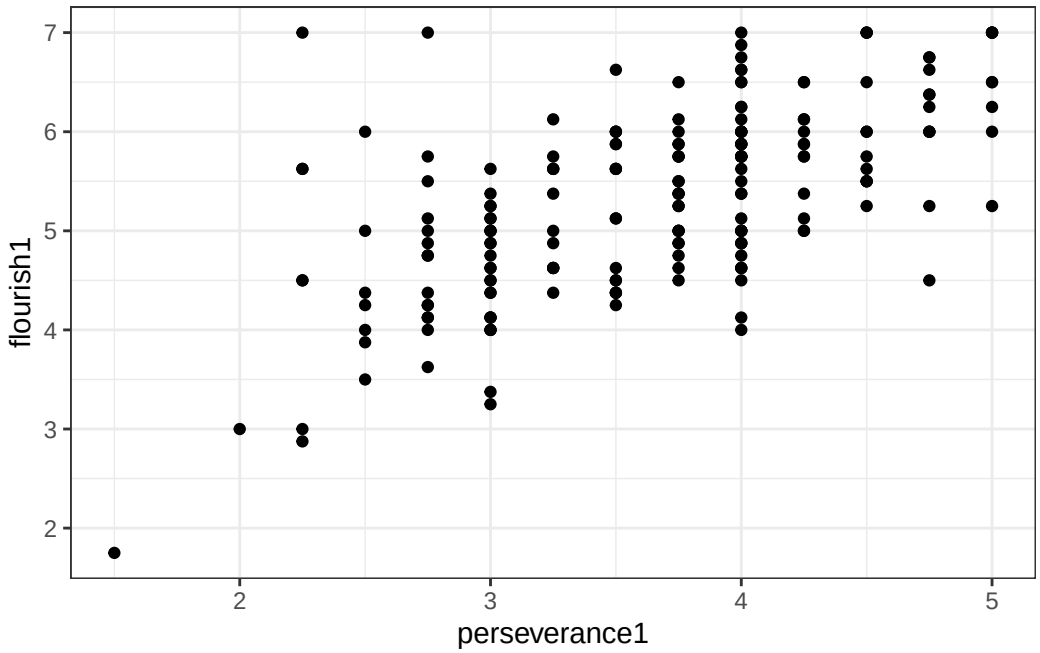
```r
ggplot(well, aes(x = flourish1)) +
  geom_histogram(aes(y = cumsum(after_stat(count)/200)),
                 bins = 7)
```




# 3 变化趋势

## 3.1 变化趋势与散点图

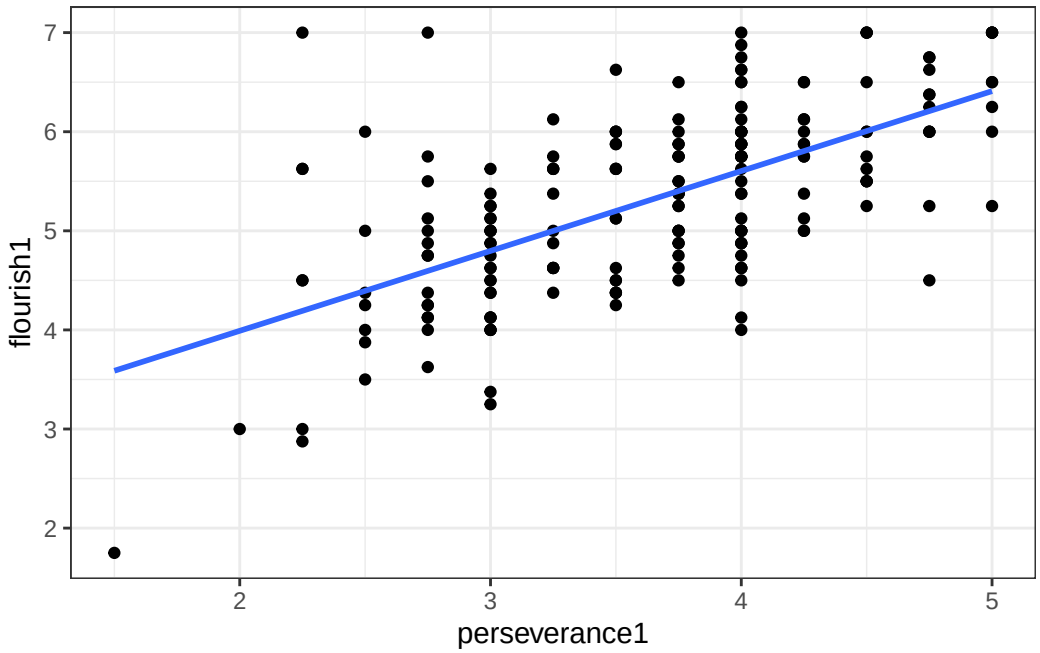
```r
ggplot(well, aes(x = perseverance1, y = flourish1)) +
  geom_point()
```




可见，所有的点的布局呈现出椭圆形，抑郁与情绪应对是相关的，抑郁水平随着情绪应对水平的升高而升高，我们将这种关系称为正相关关系。我们可以使用 geom_smooth() 函数绘制出平滑的趋势线，这样会使得抑郁（y 轴变量）与情绪应对（x 轴变量）的关系趋势更加明显。在这里，读者不必深究 geom_smooth() 及其参数，本书后面的章节会介绍相关内容。

```
ggplot(well, aes(x = perseverance1, y = flourish1)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```

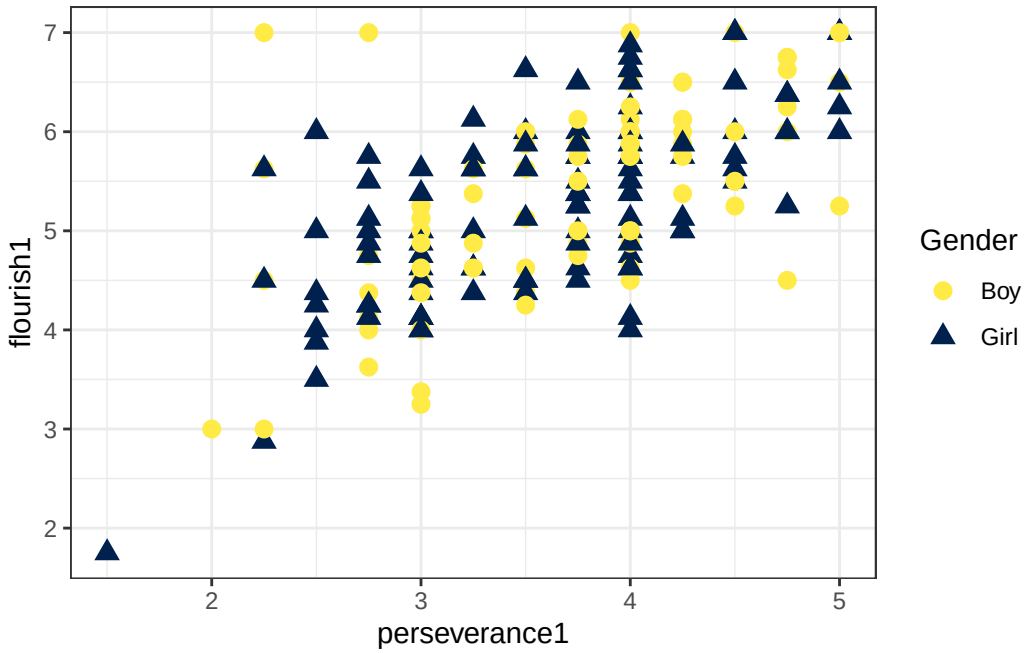
```
## `geom_smooth()` using formula = 'y ~ x'
```




```
## `geom_smooth()` using formula = 'y ~ x'
```

另外，我们还可以对比不同性别中抑郁与情绪应对的关系趋势。geom_point() 通过参数 aes(color = Gender, shape = Gender) 设定用不同颜色、不同形状表示不同性别。为了使点的形状易于区分，我们使用参数 size = 3 将点的大小设置为 3mm。注意：此时的 size 是一个常量，而非变量，因此，参数 size = 3 要写在 aes() 函数的外部。

```
ggplot(well, aes(x = perseverance1, y = flourish1)) +
  geom_point(aes(color = Gender, shape = Gender), size = 3)
```



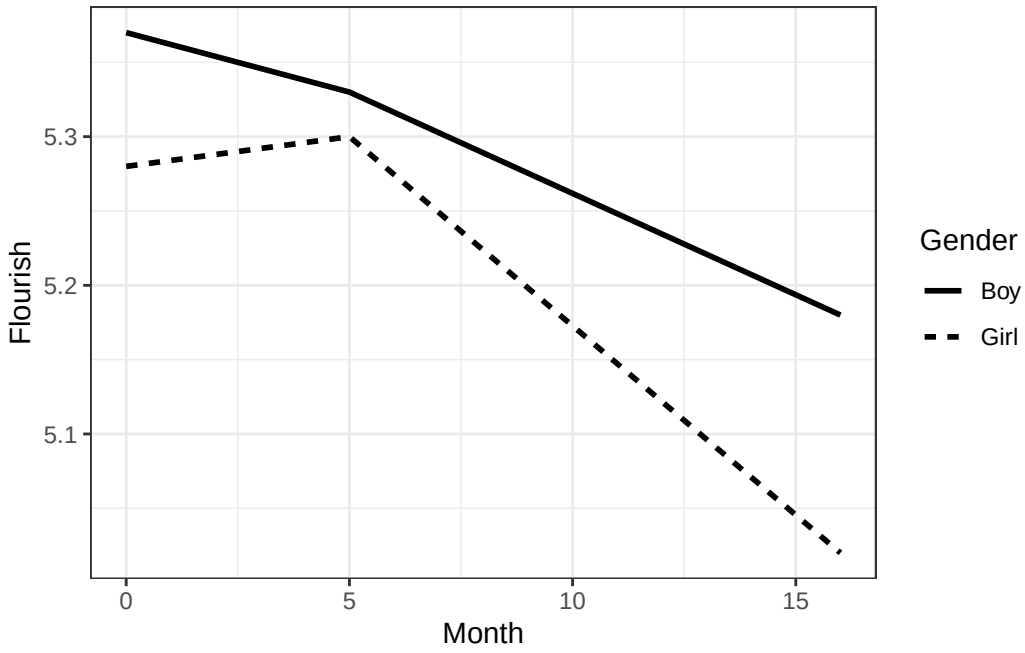

### 3.2 变化趋势与折线图

随着年龄的变化，抑郁水平的变化趋势是怎样的？我们可以使用折线图描绘抑郁水平随年龄变化的趋势。首先，我们创建一个包含性别、年龄、抑郁水平的数据框：

```
dat_trend <- data.frame(
  Gender = factor(c("Boy", "Boy", "Boy", "Girl", "Girl", "Girl")),
  Month = c(c(0, 5, 16), c(0, 5, 16)),
  Flourish = c(5.37, 5.33, 5.18, 5.28, 5.30, 5.02))
```

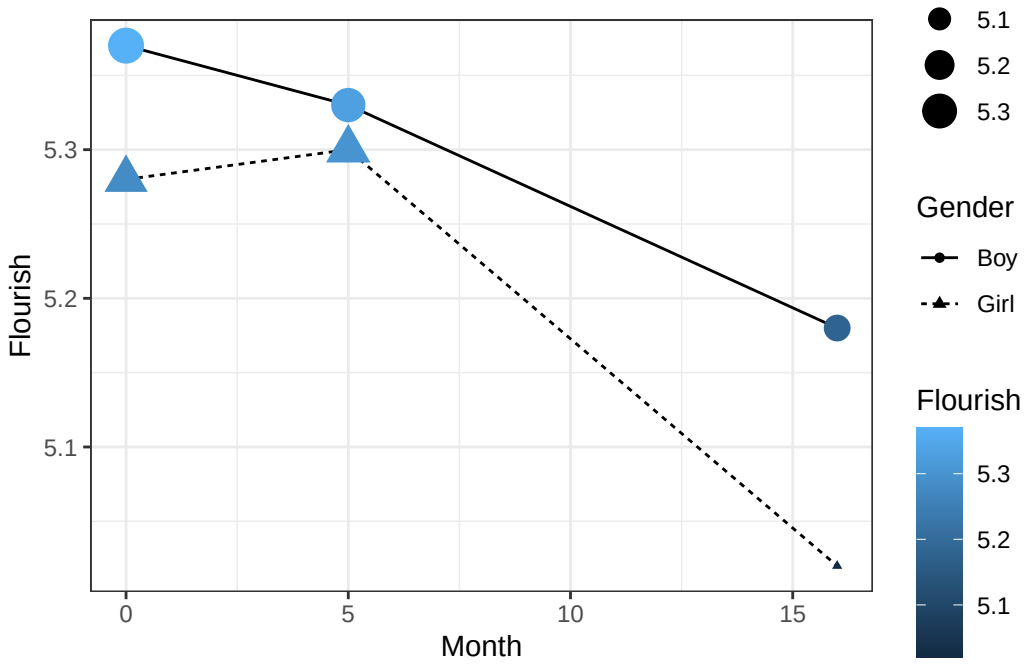
接下来，我们使用 `geom_line()` 绘制折线图。同时，我们通过参数 `linetype = Gender` 使用不同类型的线表示不同的性别，读者可使用`?linetype` 查询 ggplot2 中线的类型。ggplot2 中线的宽度默认为 0.5 个单位，为了使不同类型的线在视觉上易于区分，我们使用参数 `linewidth = 1` 将线的宽度设置为 1 个单位。作图后可见，整体上抑郁随着年龄的增长而降低。

```
ggplot(dat_trend, aes(Month, Flourish)) +
  geom_line(aes(linetype = Gender), linewidth = 1)
```




接下来，我们使用 geom_point() 在折线图的基础上增加一层散点图，同时用 aes(color = Depression, size = Depression, shape = Gender) 设定映射关系，用点的颜色与大小表示抑郁得分的高低。

```
ggplot(dat_trend, aes(Month, Flourish)) +
  geom_line(aes(linetype = Gender)) +
  geom_point(aes(color = Flourish, size = Flourish, shape = Gender))
```




# 4 适配黑白印刷的美化技术

## 4.1 颜色

在条形图中，ggplot2 默认使用深灰色填充矩形。为了避免黑白印刷品上出现大片的深灰色，下面的代码使用参数 `fill = "white"` 将填充色改为白色，使用参数 `color = "#000000"` 将矩形边框的颜色改为黑色。`fill = "white"` 通过颜色的名称设定颜色，读者可以使用 `colors()` 查询 R 语言可用的颜色名称。`color = "#000000"` 通过颜色的十六进制（HEX）RGB 值设定颜色，黑色的十六进制 RGB 值为 #000000。若读者想用青花瓷的蓝色，读者可在电脑上打开青花瓷的照片，然后使用 WPS 的取色器查询青花瓷颜色的 RGB 值。

```
ggplot(well, mapping = aes(x = flourish101)) +
  geom_bar(fill = "white", color = "#000000")
```

在黑白印刷品上，原本彩色的统计图只剩下黑色、灰色与白色。如果统计图将采用黑白印刷，而我们又需要用不同的颜色表示变量不同的数值，我们可以针对不同的情况采取不同的方法。若变量为连续变量，我们可以使用 ggplot2 默认的单色渐变色，例如图。单色渐变色采用黑白印刷后会呈现出深浅不一的灰色。另外，我们也可以使用不同的透明度来表示变量不同的取值。下面的代码用透明度表示性别，并将直方图的填充色设置为靛蓝（色值为”#375392”）：

```
ggplot(well, aes(x = perseverance1, y = flourish1, alpha = flourish1)) +
  geom_point(color = "#375392")
```

当变量为离散变量时，若用不同的彩色表示离散变量的不同取值，这些彩色在黑白印刷品上可能呈现为同一种灰色。因此，在这种情况下我们要谨慎地选择颜色。比较简单且直接的方法是通过 `scale_fill_grey()` 与 `scale_color_grey()` 使用灰色渐变色。下面的代码 `scale_fill_grey()` 使用不同灰度表示不同性别，参数 `start = 0.8` 设定渐变色的起点灰色，`end = 0.2` 设定渐变色的终点灰色，`start` 与 `end` 的数值越小，相应的灰色越深。

```
ggplot(well, aes(x = flourish1, fill = Gender)) +
  geom_histogram(bins = 7) +
  scale_fill_grey(start = 0.8, end = 0.2)
```

若想使用多种彩色，同时想保证黑白印刷的良好效果，我们可使用 Viridis 系列的量尺函数。这些量尺函数会自动选用对色盲人群友好的颜色，这些颜色在黑白印刷时仍能保持良好的灰度对比。下面的代码通过 `scale_fill_viridis_c()` 设置填充色，并调整了两个参数的默认值。参数 `direction = -1` 的作用是反转选择颜色的顺序，使深色在下、浅色在上，更具沉稳感。参数 `option = "E"` 的作用是从序号为 E 的调色板中选取颜色。

```
ggplot(well, aes(x = flourish1, fill = Gender)) +
  geom_histogram(bins = 7) +
  scale_fill_viridis_d(direction = -1, option = "E")
```

### 4.2 主题

ggplot2 的默认主题 `theme_grey()` 可能不符合我们的需求，我们可以使用主题函数进行修改。读者可运行命令`?theme` 查阅 R 文档，选择符合自己需要的其他主题。下面的代码使用了 `theme_bw()`，即，黑白（black and white）主题：

```
ggplot(well, mapping = aes(x = flourish101, fill = Gender)) +
  geom_bar() +
  theme_bw()
```

## 5 保存图片

我们使用 `ggsave()` 保存刚作好的图。在下面的代码中，文件格式设定为 “png”，文件名设定为 “plot.pdf”，图片的长与高设定为 6 英寸与 4 英寸，图片的分辨率设定为 600DPI。`ggsave()` 默认以英寸为图片大小的单位，1 英寸等于 2.54 厘米。

```r
ggsave("plot.png", width = 6, height = 4, dpi = 600)
```

## 6 总结

本章通过频数分布图评估了抑郁的一般水平以及中重度抑郁的情况，通过散点图与折线图探索了抑郁的变化趋势。同时，本章也介绍了 ggplot2 中常用的作图技术，这些技术基本能满足读者的工作需要。作为一本统计书，本章难以对 ggplot2 丰富且灵活的功能进行全面的介绍。读者在掌握本书的内容后，可以在 AI 的辅助下对 ggplot2 进行拓展学习。例如：读者可以询问 AI："在 ggplot2 中如何修改 x 轴的刻度及其标签？"