峰峦图

庄闪闪

目录

1 峰峦图 **1**

1.1 数据结构 1

1.2 绘图教程 2

1.3 其他资料 16

1 峰峦图

上次可视化系列说了瀑布图。它可以用于展示拥有相同的 X 轴变量数据（如相同的时间序列）、不同的 Y 轴离散型变量（如不同的类别变量）和 Z 轴数值变量。

本节使用的**峰峦图**也可以很好地展示瀑布图的数据信息。它们对于可视化随时间或空间分布的变化非常有用。本节主要使用 `ggridges` 包中的 `geom_density_ridges()` 进行绘制峰峦图。详细介绍如下：

1.1 数据结构

这里使用 `base` 包中的 `diamonds` 数据集做例子。

```
# library
library(ggridges) # Ridgeline Plots in 'ggplot2', CRAN v0.5.2
```

```
library(ggplot2) # Create Elegant Data Visualisations Using the Grammar of Graphics, CR
head(diamonds)
```

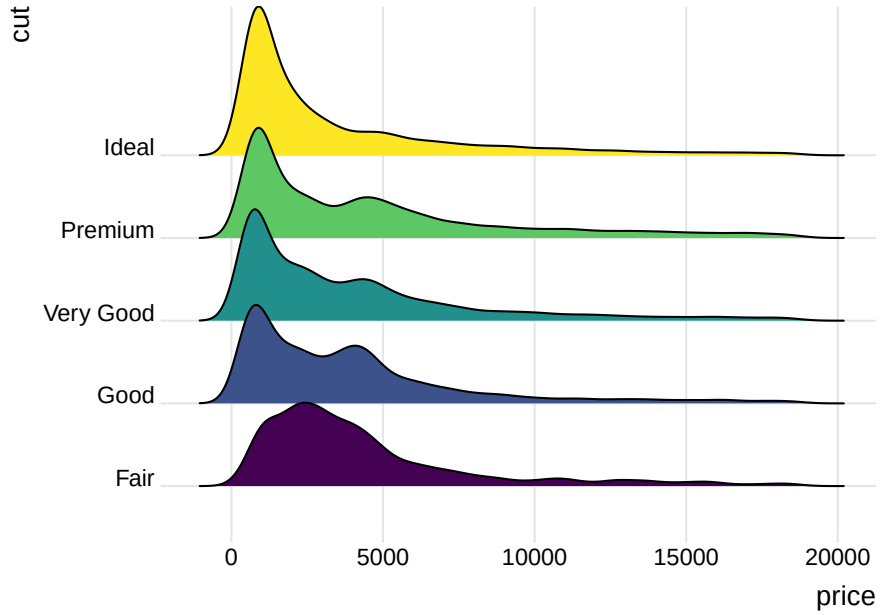
```
## # A tibble: 6 x 10
##   carat cut       color clarity depth table price     x     y     z
##   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1 0.23  Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
## 2 0.21  Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
## 3 0.23  Good      E     VS1      56.9    65   327  4.05  4.07  2.31
## 4 0.290 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
## 5 0.31  Good      J     SI2      63.3    58   335  4.34  4.35  2.75
## 6 0.24  Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
```

1.2 绘图教程

1.2.1 基础版本

使用 price 作为 x 轴, cut 为 y 轴，fill 参数也是设定为 cut。geom_density_ridges() 内部全部使用默认参数。使用了 ggridges 包中的主题 theme_ridges()。

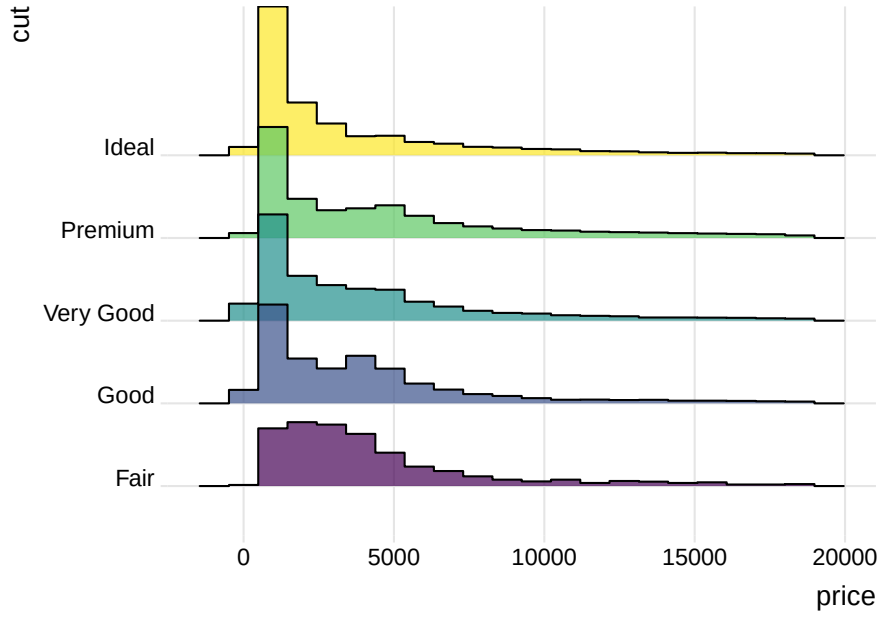
```
ggplot(diamonds, aes(x = price, y = cut, fill = cut)) +
    geom_density_ridges() +
    theme_ridges() +
    theme(legend.position = "none")
```



1.2.2 形状变化

如果不想绘制密度图，则可以使用 `stat="binline", bins=20` 绘制柱形图，其中 `bins=20` 表示每格格子大小。为了防止上下图片重叠，这里使用了透明度参数：`alpha=0.6`。

```
ggplot(diamonds, aes(x = price, y = cut, fill = cut)) +
    geom_density_ridges(alpha=0.7, stat="binline", bins=20) +
    theme_ridges() +
    theme(legend.position = "none")
```

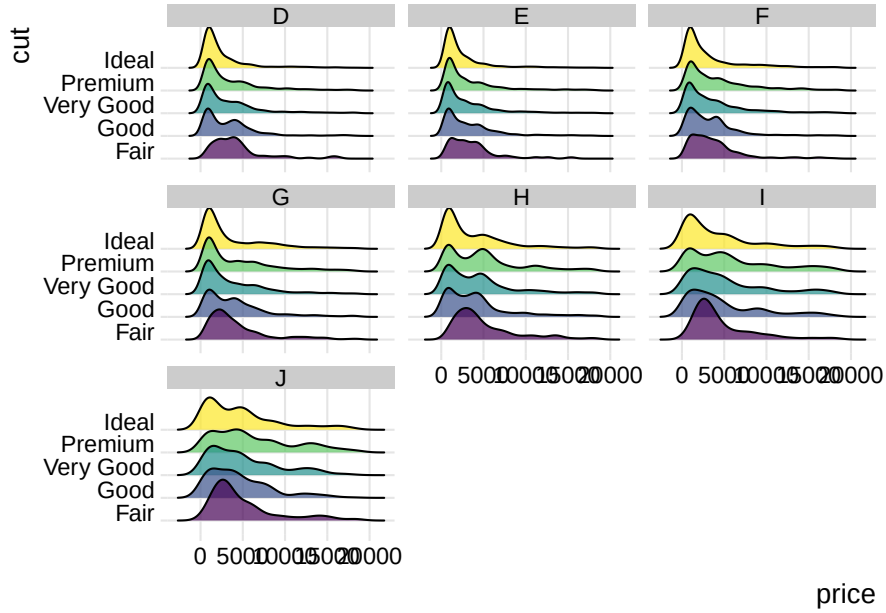


1.2.3 根据第三变量进行分面

```
ggplot(diamonds, aes(x = price, y = cut,fill = cut)) +
    geom_density_ridges(alpha=0.7) +
    facet_wrap(~color) +
  theme_ridges() +
    theme(legend.position = "none")
```

```
## Picking joint bandwidth of 559

## Picking joint bandwidth of 517

## Picking joint bandwidth of 585

## Picking joint bandwidth of 689

## Picking joint bandwidth of 755

## Picking joint bandwidth of 958
```

```
## Picking joint bandwidth of 1010
```

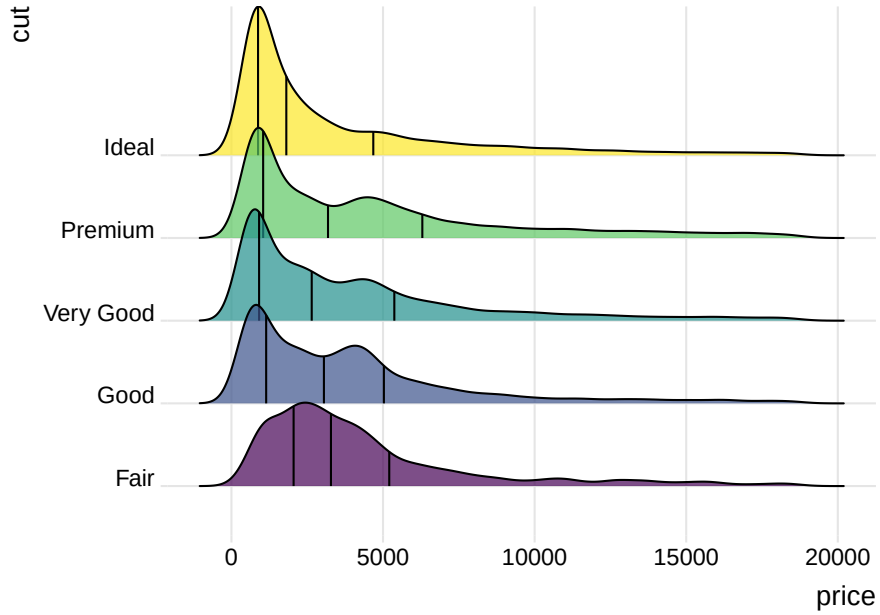


1.2.4 加入统计量

通过设置选项 quantile_lines = TRUE,我们可以使 stat_density_ridges 计算指示分位数的线的位置。默认情况下，绘制了三行，分别对应于第一，第二和第三四分位数：

```
ggplot(diamonds, aes(x = price, y = cut,fill = cut)) +
    geom_density_ridges(alpha=0.7,quantile_lines = TRUE) +
  theme_ridges() +
    theme(legend.position = "none")
```

```
## Picking joint bandwidth of 458
```

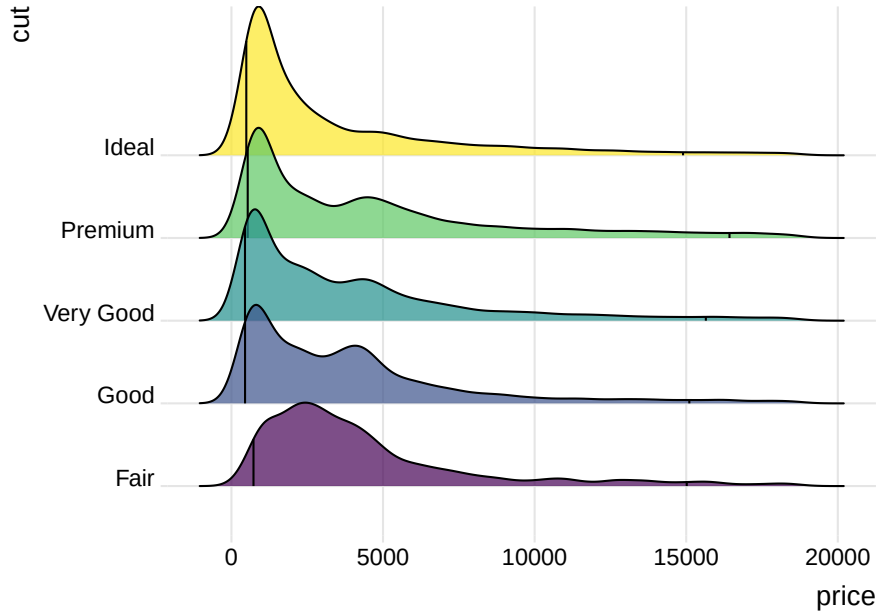


注意：quantiles=2 意味着在两个分位数之间的边界上有一条线即，(中位数)。

我们还可以通过切点而不是数字来指定分位数。例如，我们可以指出 2.5% 和 97.5%的尾巴。

```
ggplot(diamonds, aes(x = price, y = cut,fill = cut)) +
    geom_density_ridges(alpha=0.7,quantile_lines = TRUE,quantiles = c(0.025, 0.975)) +
  theme_ridges() +
    theme(legend.position = "none")
```

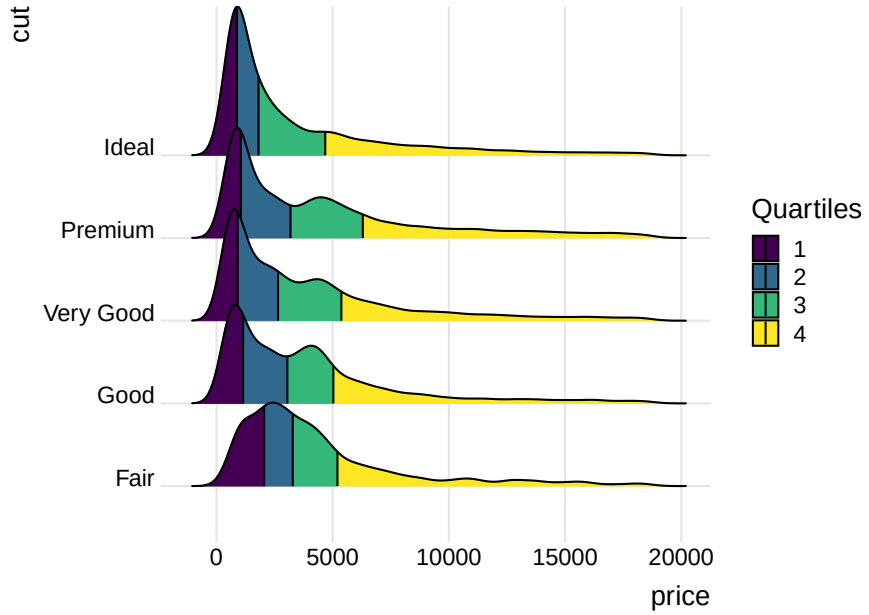
```
## Picking joint bandwidth of 458
```



使用 stat_density_ridges，计算 stat(quantile)，通过分位数进行着色,。注意，仅当 calc_ecdf = TRUE 时才能计算。

```
ggplot(diamonds, aes(x = price, y = cut,fill = factor(stat(quantile)))) +
    stat_density_ridges(
      geom = "density_ridges_gradient",
      calc_ecdf = TRUE,
    quantiles = 4, quantile_lines = TRUE) +
  theme_ridges() +
  scale_fill_viridis_d(name = "Quartiles")
```

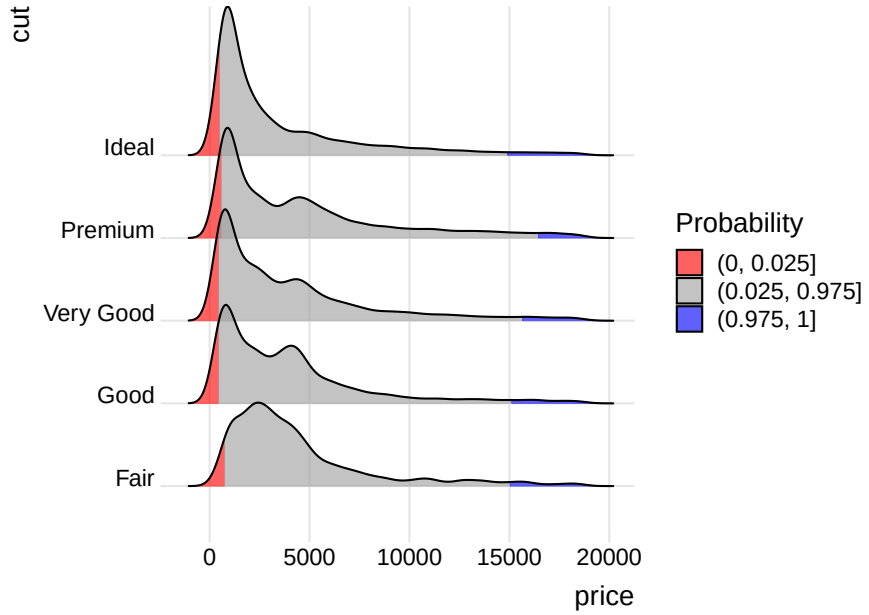
```
## Picking joint bandwidth of 458
```



我们可以使用相同的方法来突出分布的尾部。

```
ggplot(diamonds, aes(x = price, y = cut,fill = factor(stat(quantile)))) +
    stat_density_ridges(
      geom = "density_ridges_gradient",
      calc_ecdf = TRUE,
    quantiles = c(0.025, 0.975)) +
  theme_ridges() +
    scale_fill_manual(
    name = "Probability", values = c("#FF0000A0", "#A0A0A0A0", "#0000FFA0"),
    labels = c("(0, 0.025]", "(0.025, 0.975]", "(0.975, 1]")
  )
```

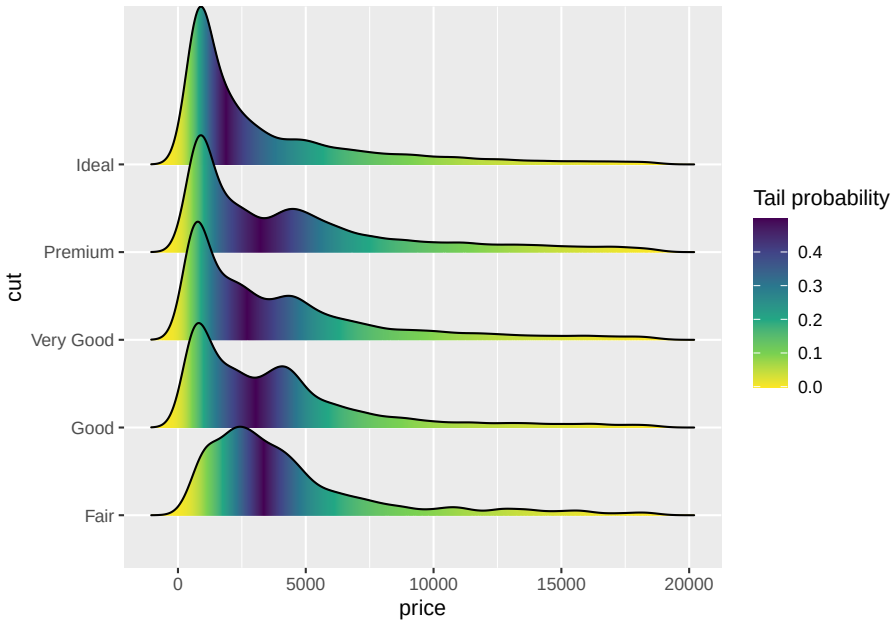
```
## Picking joint bandwidth of 458
```



最后，当 `calc_ecdf = TRUE` 时，我们还可以计算 `stat(ecdf)`，它表示该分布的经验累积密度函数。我们将其概率直接映射到颜色上。

```
ggplot(diamonds, aes(x = price, y = cut,fill = 0.5 - abs(0.5 - stat(ecdf)))) +
    stat_density_ridges(geom = "density_ridges_gradient", calc_ecdf = TRUE) +
  scale_fill_viridis_c(name = "Tail probability", direction = -1)
```

```
## Picking joint bandwidth of 458
```

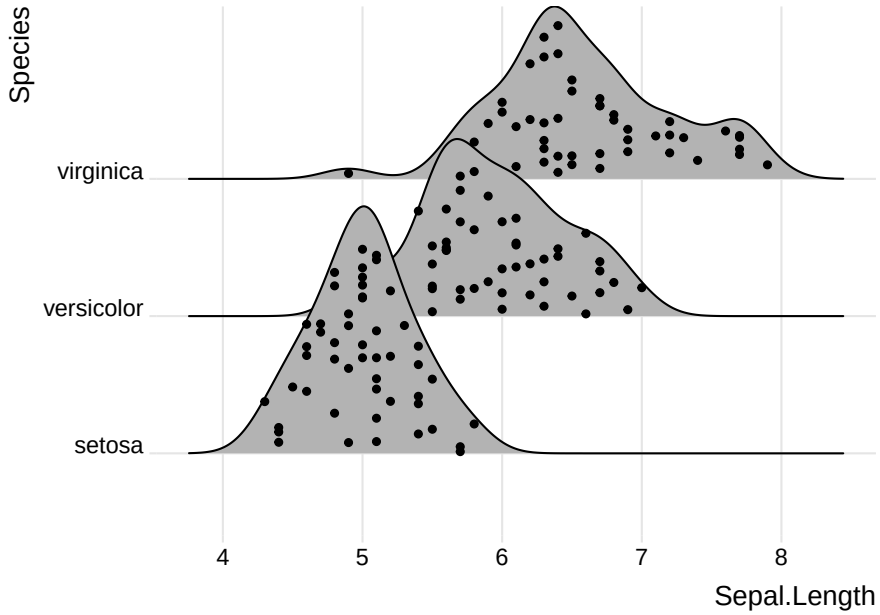


1.2.5 加入抖动点

stat_density_ridges() 还提供了可视化生成分布的原始数据点的选项。可以通过设置 jittered_points = TRUE 实现。为了只管我们这里使用 iris 包。

```
ggplot(iris, aes(x = Sepal.Length, y = Species)) +
  geom_density_ridges(jittered_points = TRUE)+
    theme_ridges() +
    theme(legend.position = "none")
```

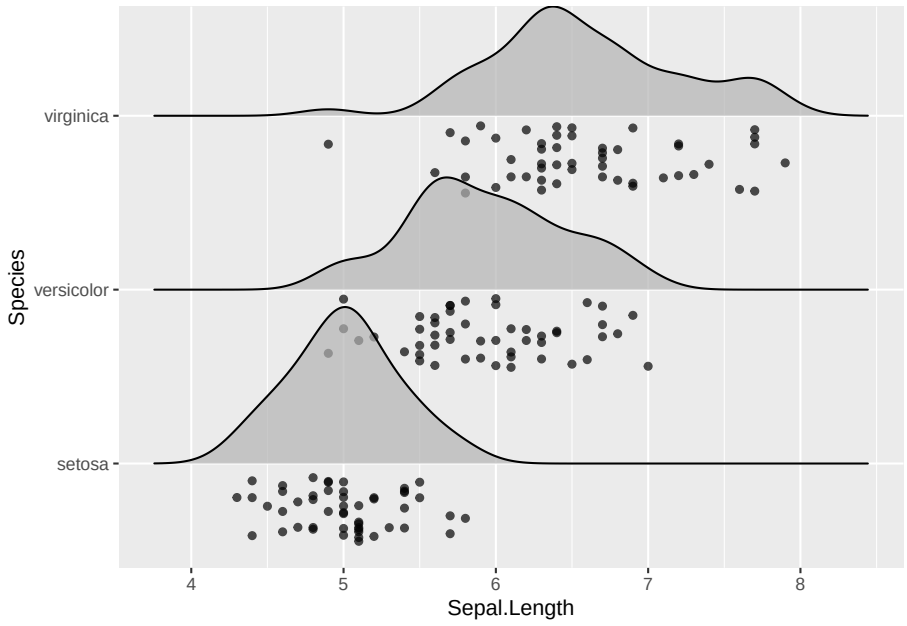
```
## Picking joint bandwidth of 0.181
```



当然可以将其放在密度函数的下方，通过使用 `position = "raincloud"` 参数。

```
ggplot(iris, aes(x = Sepal.Length, y = Species)) +
  geom_density_ridges(
    jittered_points = TRUE, position = "raincloud",
    alpha = 0.7, scale = 0.9
  )
```

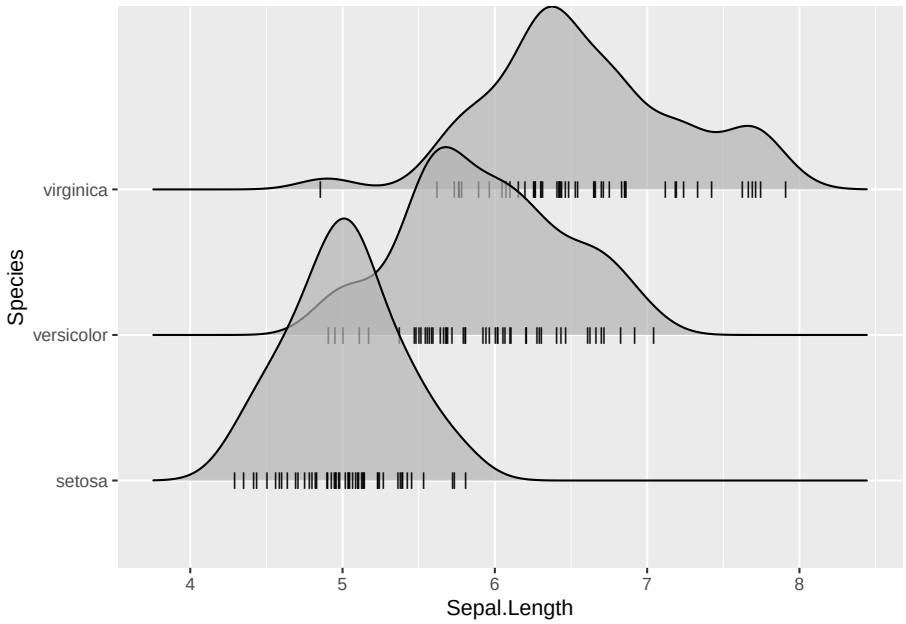
```
## Picking joint bandwidth of 0.181
```



我们还可以模拟地毯形式：

```
ggplot(iris, aes(x = Sepal.Length, y = Species)) +
  geom_density_ridges(
    jittered_points = TRUE,
    position = position_points_jitter(width = 0.05, height = 0),
    point_shape = '|', point_size = 3, point_alpha = 1, alpha = 0.7,
  )
```

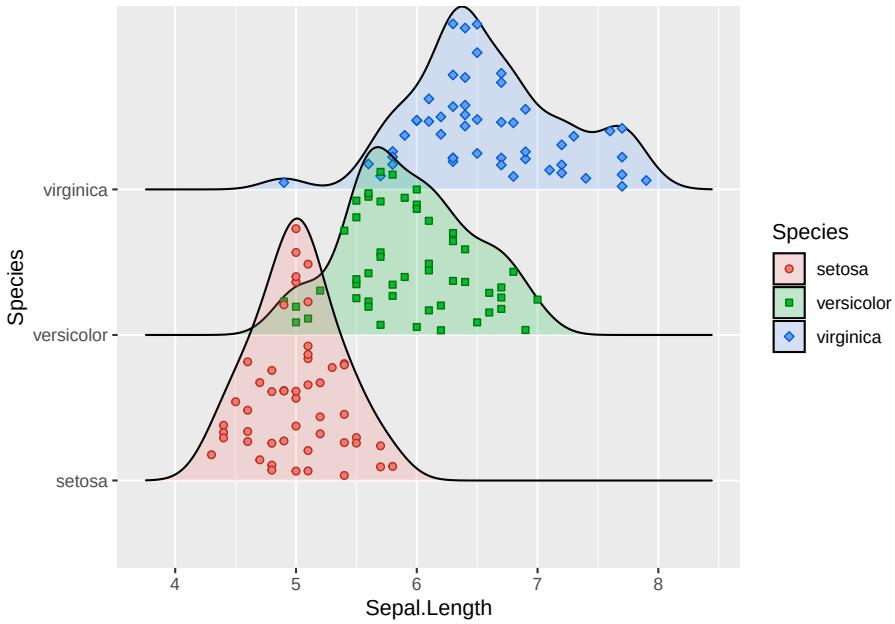
```
## Picking joint bandwidth of 0.181
```



可以使用 ggridges 提供的特殊比例来设置抖动点的样式。首先，`scale_discrete_manual()` 可用于制作具有任意形状和比例的图形。`scale_point_color_hue()`。

```
ggplot(iris, aes(x = Sepal.Length, y = Species, fill = Species)) +
  geom_density_ridges(
    aes(point_color = Species, point_fill = Species, point_shape = Species),
    alpha = .2, point_alpha = 1, jittered_points = TRUE
  ) +
  scale_point_color_hue(l = 40) +
  scale_discrete_manual(aesthetics = "point_shape", values = c(21, 22, 23))
```

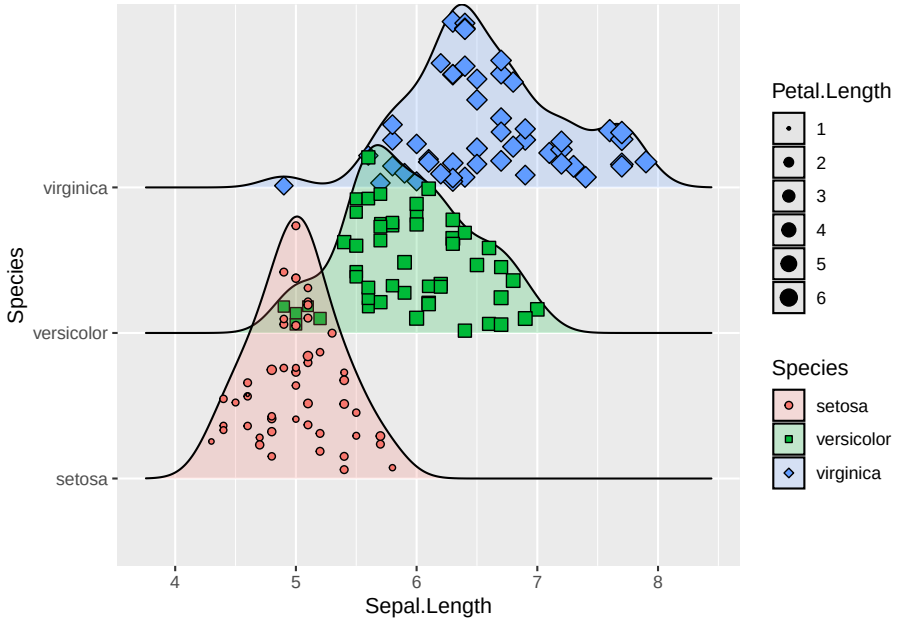
```
## Picking joint bandwidth of 0.181
```



如果你还想再加入一个变量进行可视化，可以在 geom_density_ridges() 加入，例如：point_shape = Species, point_fill = Species, point_size = Petal.Length。

```
ggplot(iris, aes(x = Sepal.Length, y = Species, fill = Species)) +
  geom_density_ridges(
    aes(point_shape = Species, point_fill = Species, point_size = Petal.Length),
    alpha = .2, point_alpha = 1, jittered_points = TRUE
  ) +
  scale_point_color_hue(l = 40) + scale_point_size_continuous(range = c(0.5, 4)) +
  scale_discrete_manual(aesthetics = "point_shape", values = c(21, 22, 23))
```

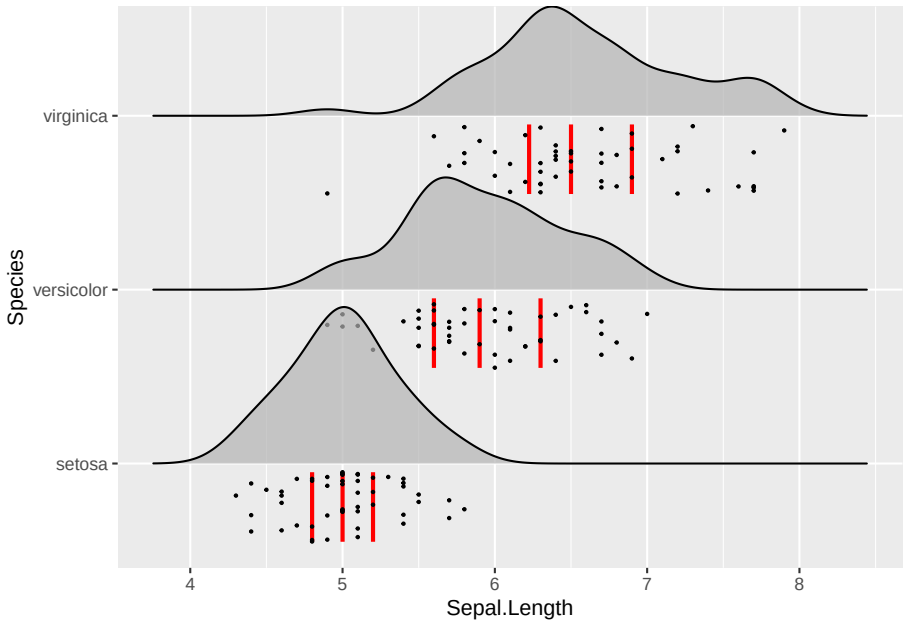
```
## Picking joint bandwidth of 0.181
```



另外一种有趣的可视化是通过 `vline_xxx` 构造以下图形。

```
ggplot(iris, aes(x = Sepal.Length, y = Species)) +
  geom_density_ridges(
    jittered_points = TRUE, quantile_lines = TRUE, scale = 0.9, alpha = 0.7,
    vline_size = 1, vline_color = "red",
    point_size = 0.4, point_alpha = 1,
    position = position_raincloud(adjust_vlines = TRUE)
  )
```

```
## Picking joint bandwidth of 0.181
```



1.3 其他资料

对于该包的其他有趣函数与可视化可参考以下资料：- Introduction to ggridges

- RDocumentation-ggridges
- Basic ridgeline plot