Data Analysis Tools and Practice(Using R)

2020.04.02

ggplott画图2

北京大学 软件与微电子学院
School of Software and Microelectronics, Peking University

Huiping Sun(孙惠平)
sunhp@ss.pku.edu.cn

上次课程内容回顾

- ggplot2
- qplot()：
  - ✴ data; log; colour; shape; alpha;
- geom：
  - ✴ point; smooth; jitter; boxplot;path; line; histogram; freqpoly; density; bar;
  - ✴ binwidth; fill; weight; scale_y_continous(); smooth;
- facets：
- ggplot():
  - ✴ +; %+%; layer(); geom_xxx(); stat_xxx(); aes(); group;

作业讲解

练习 – 0022

- gdp_long.txt
- 做折线图（网格、特殊线，图例的不同位置）
- 条形图（正常、堆积、横向、颜色宽度等、显示数字、误差线）

- cityrain.csv
- 做折线图（边界标注，slide，mar和bty的含义）

练习 - 0025

- dapengde_DummyR_PM25.csv是2003 年 8 月在北京城区的三个高度（8 米，100 米，325 米）测得的 PM2.5 的质量浓度日变化的统计数据，共 4 列 25 行。
  - ➡ 请画出一条折线表示h8和time的关系，要求是"time"和"pm2.5" 分别是x轴的名称和y轴的名称，lty=1（表示line的type为1，表示直线）y轴的范围是0到200.
  - ➡ 在上图增加一条折线(使用lines()函数)表示h100和time的关系，要求颜色为红色，线型为虚线(lty=2)
  - ➡ 在上图中增加图例来表示上边画的两条折线，其中图例位置为（x=15，y=180）位置处，内容为8m和100m,两条折线分别为黑色直线和红色虚线。
  - ➡ 画出x轴，刻度指定为和时间相对应的24个小时。
  - ➡ 与h8和h100两条折线相对应，画出其对应的y轴均值的水平线。

- 基本绘图、qplot、ggplot

练习 - 0026

- 使用数据集airquality回答下列问题
  1）使用str()函数来观察airquality这个数据的变量有那些:
  2）用函数计算第三个变量（风速）的平均值，最小值，最大值和标准差:
  3）使用pdf("mygraph.pdf") 将上面的图形保存到你的作业文件夹（本地硬盘）
  4）用plot()函数创建风速与风度的散点图：添加回归曲线和标题“Weather in NYC”：

- 使用R自带的数据集cars画出散点图，颜色设置为彩虹色，形状为编码为1:10的图形。主标题为“speed and diantance”，主标题颜色为蓝色，主标题缩放比例为1.5，字体为2，副标题为“scatter plot”，副标题颜色为灰色，主标题缩放比例为1.2

- 画出数据框cars的speed列的频率直方图，主标题为“speed hist”，主标题颜色为蓝色，主标题缩放比例为1.5，字体为2，副标题为“histogram exercise”，副标题颜色为灰色，主标题缩放比例为1.2，y轴范围为0到0.1
  添加密度曲线，要求颜色为红色，线段类型为虚线，宽度为2

- 使用R数据集VADeaths,查看这个数据集，画出各个年龄段死亡率的箱型图，要求并排排列，颜色为前4个彩虹色，添加图例，图例名称为VADeaths的列名，y轴范围为0到100，主标题为“VADeaths barplot”，主标题颜色为蓝色，主标题字体为2，副标题为“barplot exercise”，副标题颜色为灰色，主标题缩放比例为1.5

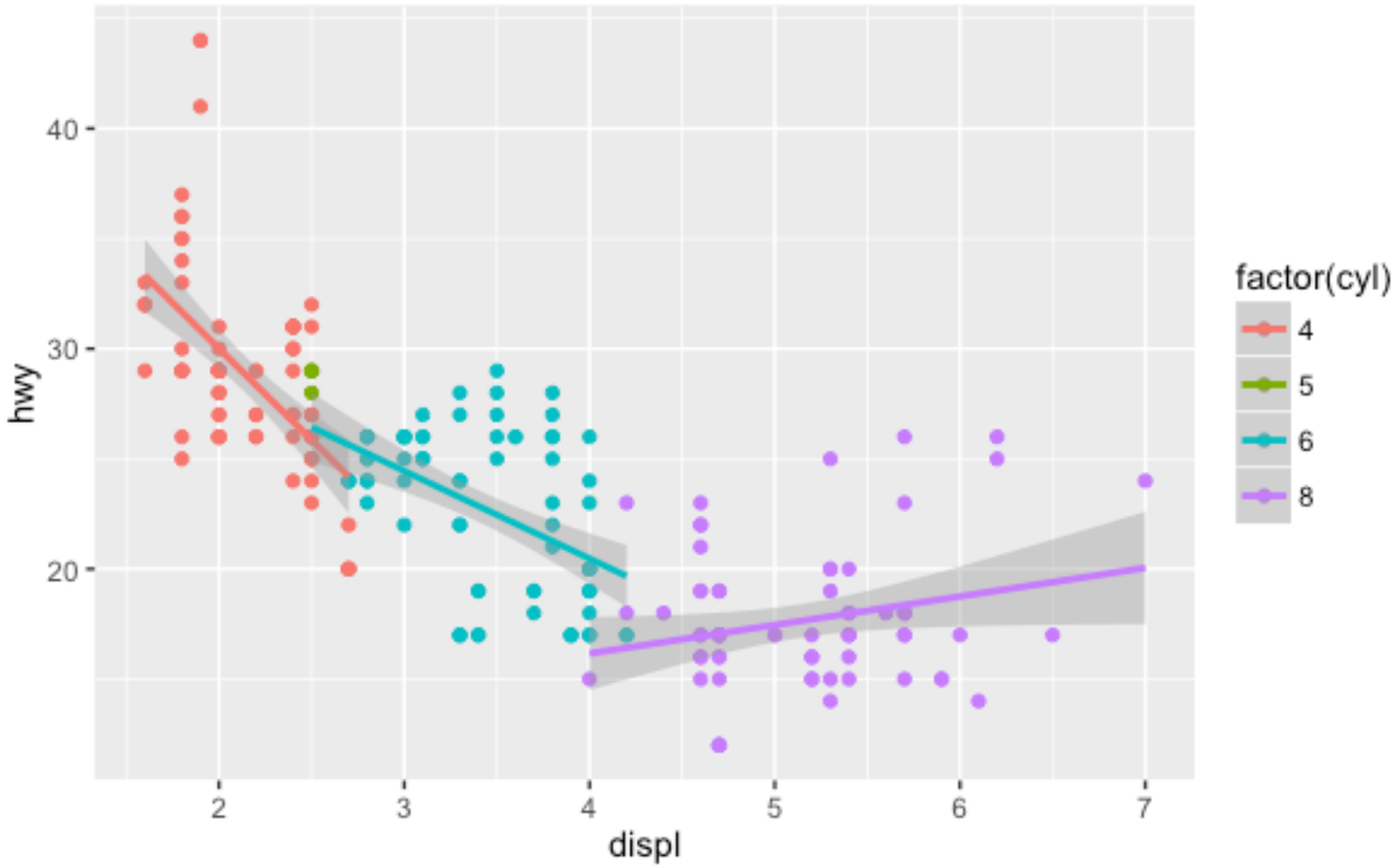
- 基本绘图、qplot、ggplot

工具箱

CH5

图层用途

- 展示数据本身
- 展示数据的统计摘要
- 添加额外的元数据、上下文信息和注解



基本图形类型

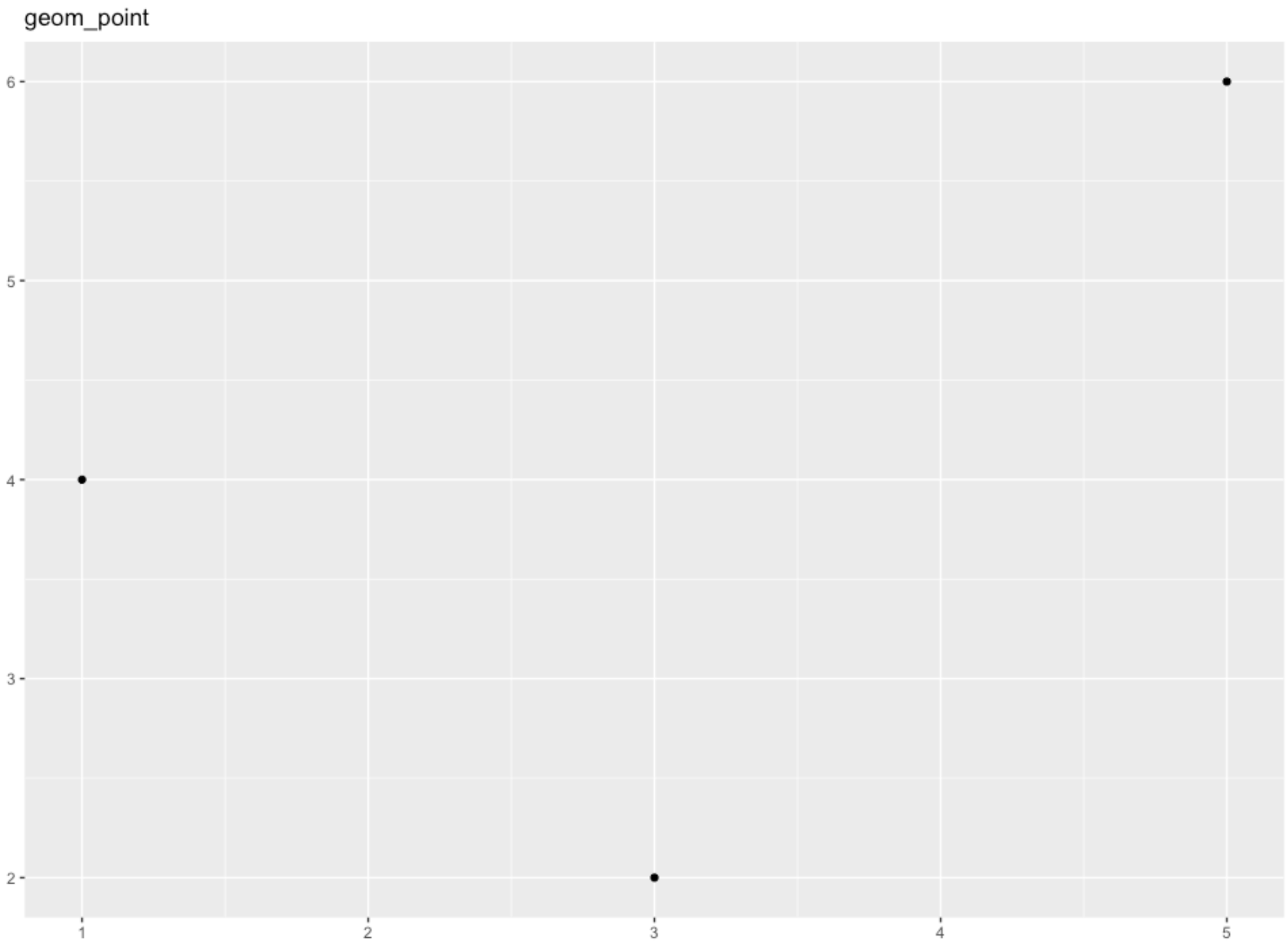
- geom_area()：面积图
- geom_bar(stat="identity")：条形图
- geom_line()：线条图
- geom_point()：散点图
- geom_text()：添加标签
- geom_tile()：色深图、水平图

colour size shape fill

```
> df <- data.frame(
+    x = c(3, 1, 5),
+    y = c(2, 4, 6),
+    label = c("a","b","c")
+ )
> p <- ggplot(df, aes(x, y, label = label)) +
+    xlab(NULL) + ylab(NULL)
```

散点图

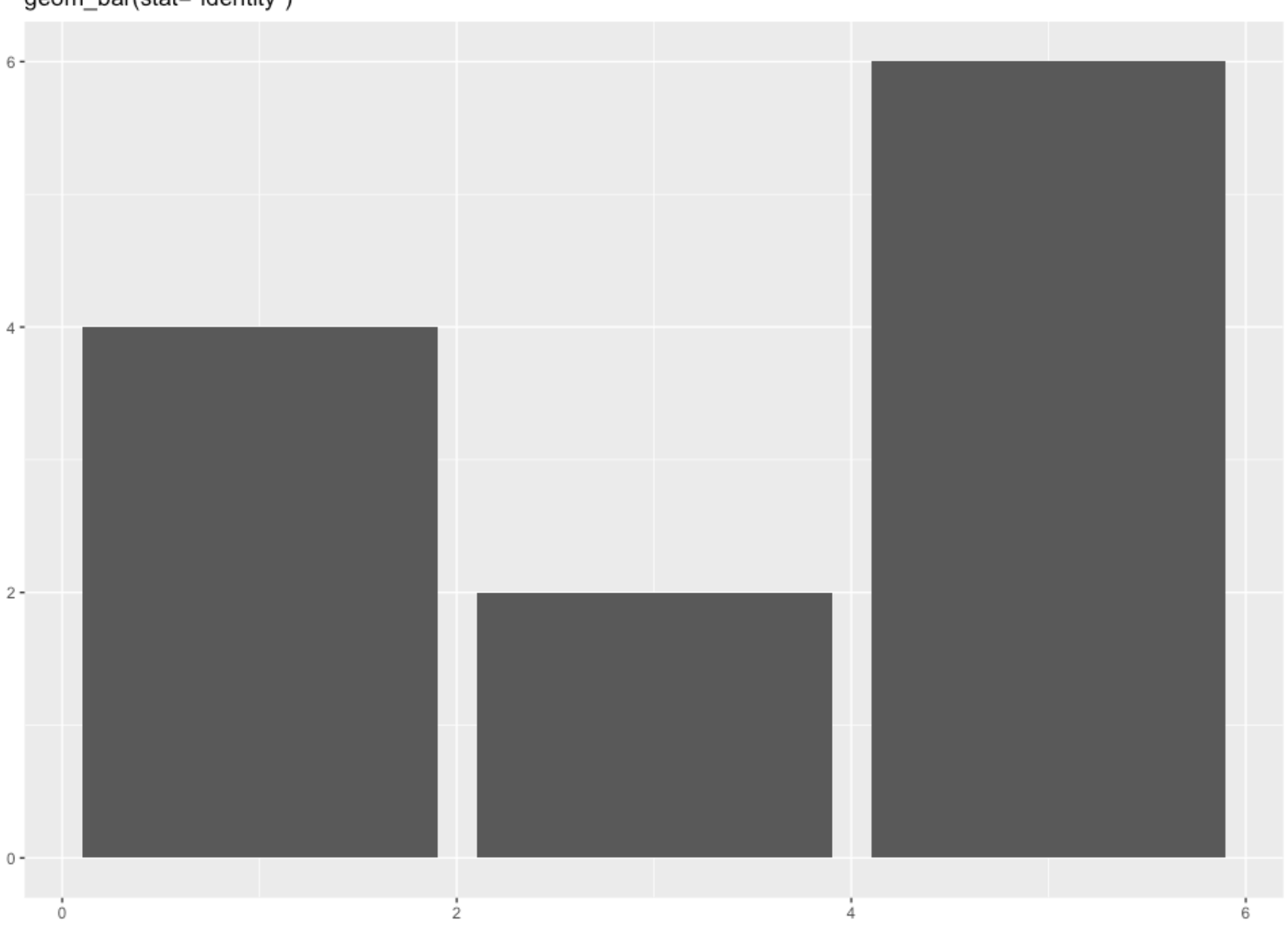
```
> p + geom_point() + labs(title = "geom_point")
```



条形图

```
> p + geom_bar(stat="identity") +
+   labs(title = "geom_bar(stat=\"identity\")")
```

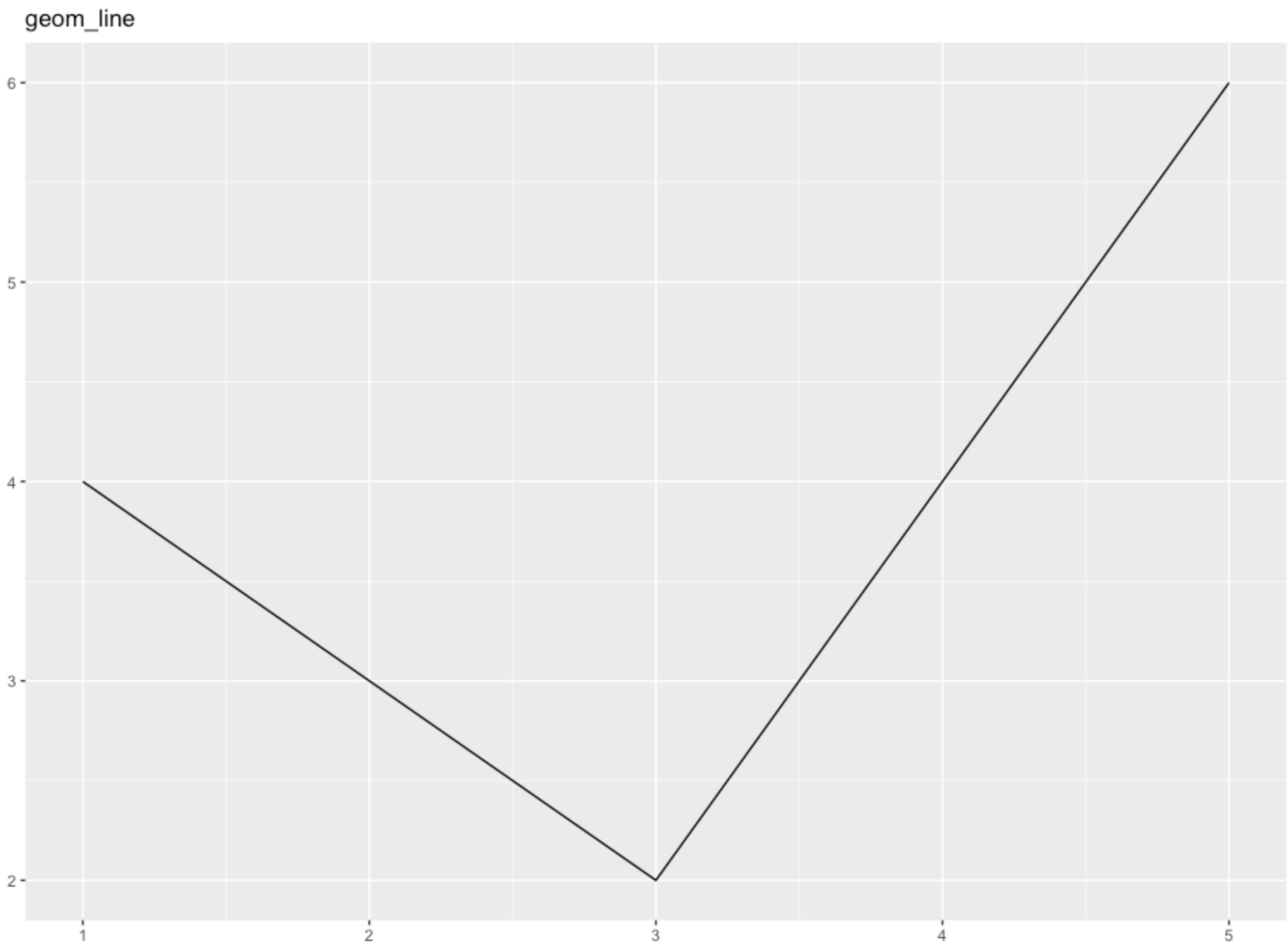
stat = identity



线条图

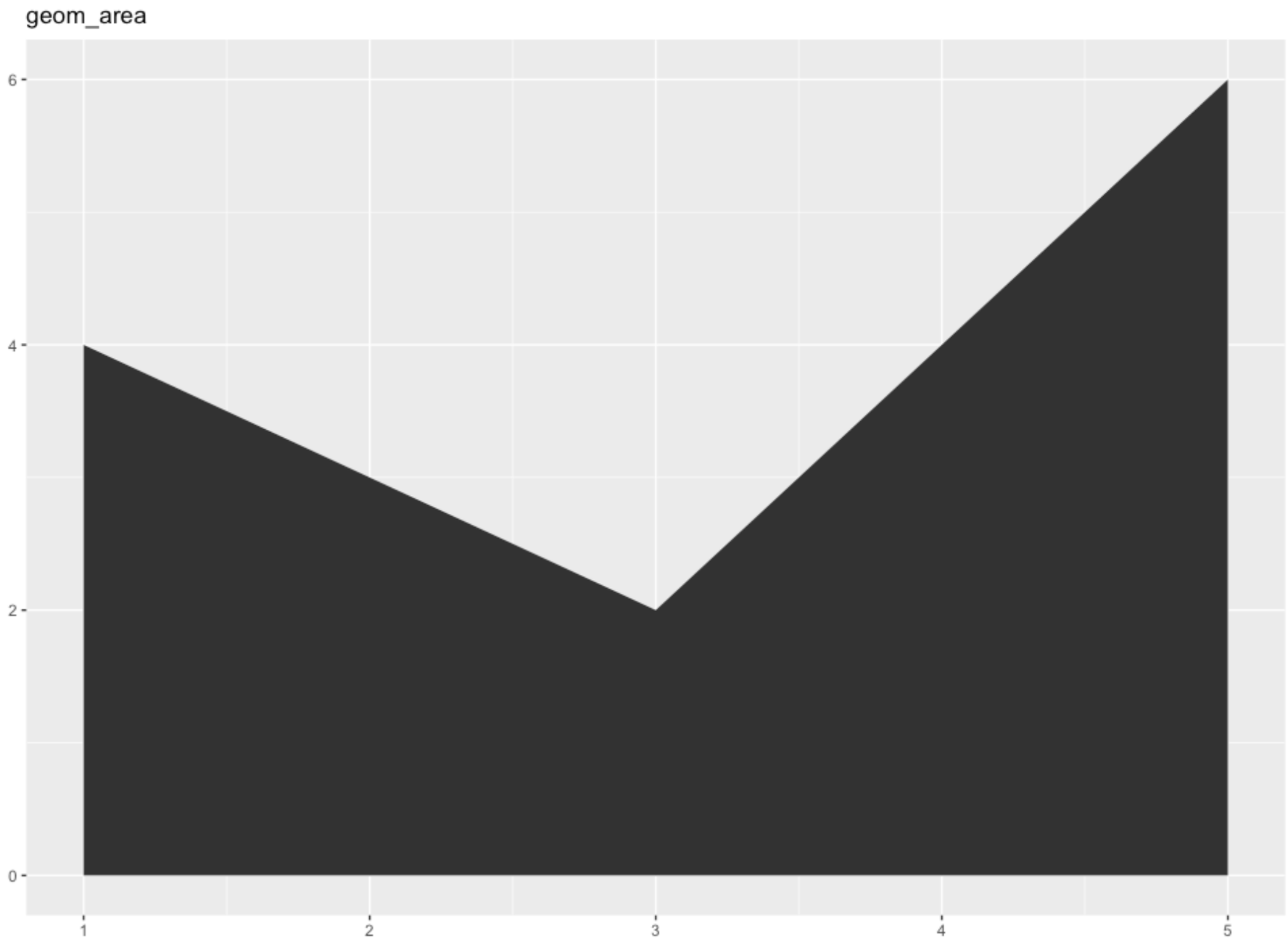
```
> p + geom_line() + labs(title = "geom_line")
```

group



面积图

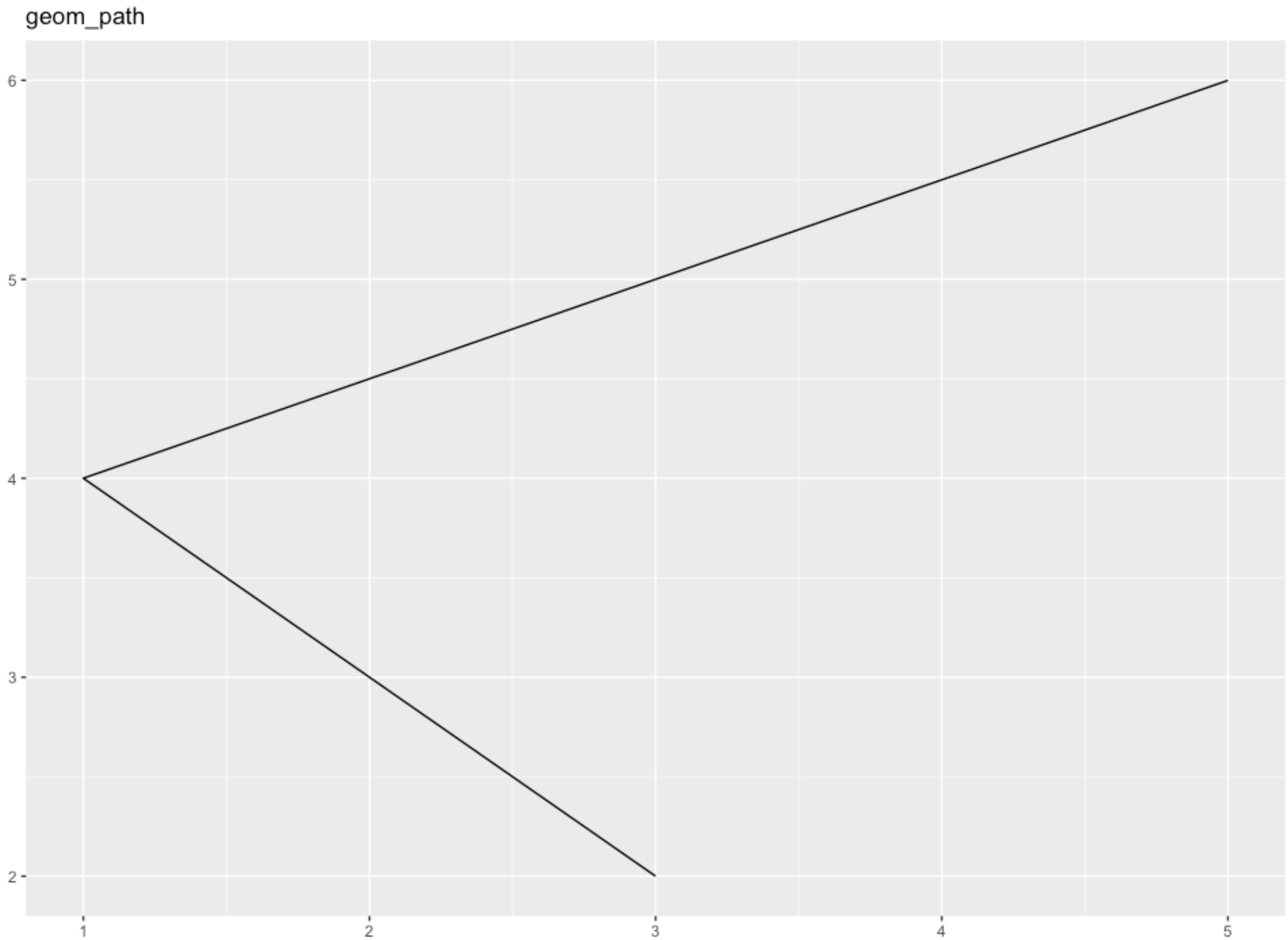
```
> p + geom_area() + labs(title = "geom_area")
```



路径图

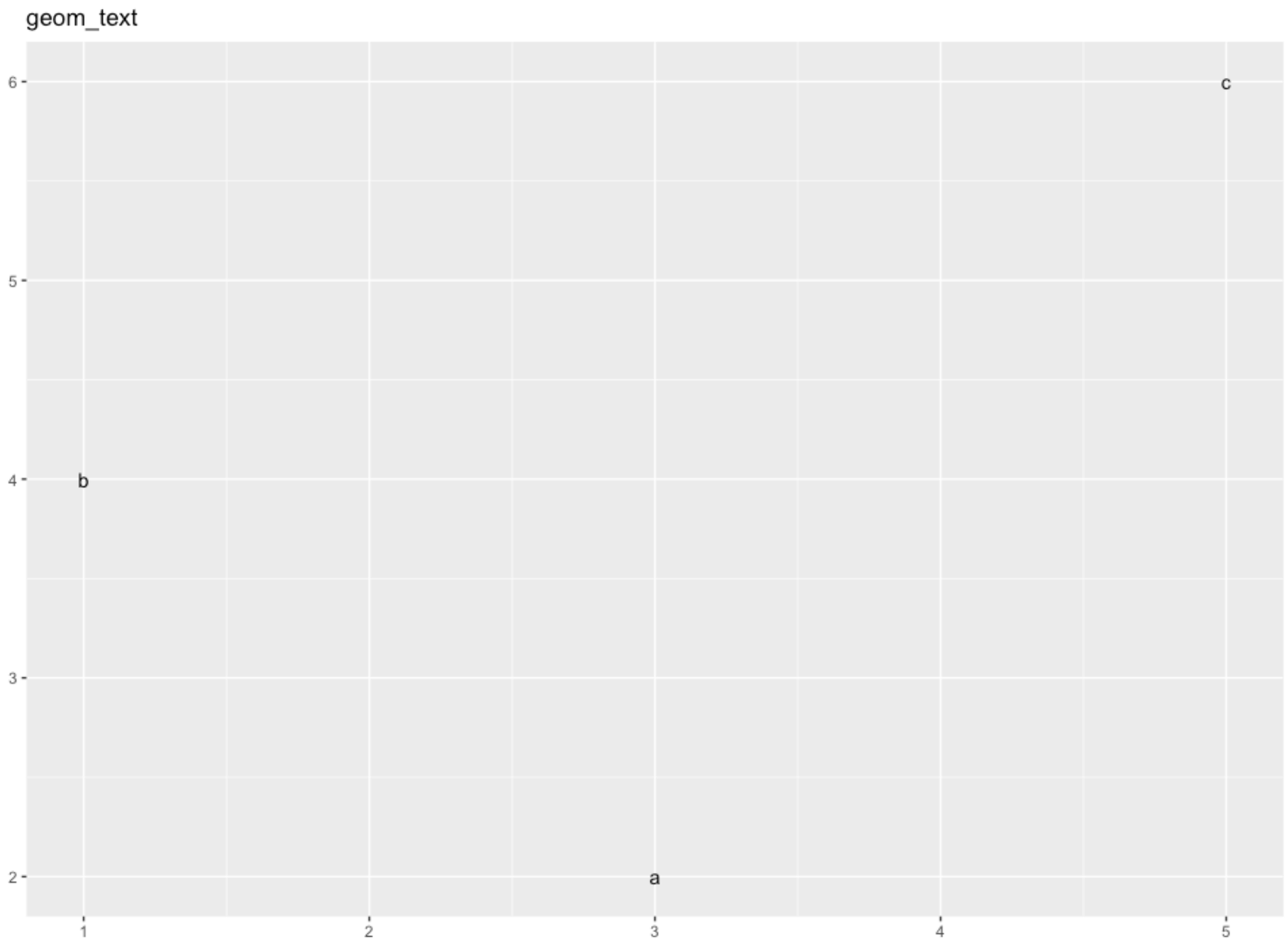
```
> p + geom_path() + labs(title = "geom_path")
```

数据出现顺序



添加标签

```
> p + geom_text() + labs(title = "geom_text")
```



label

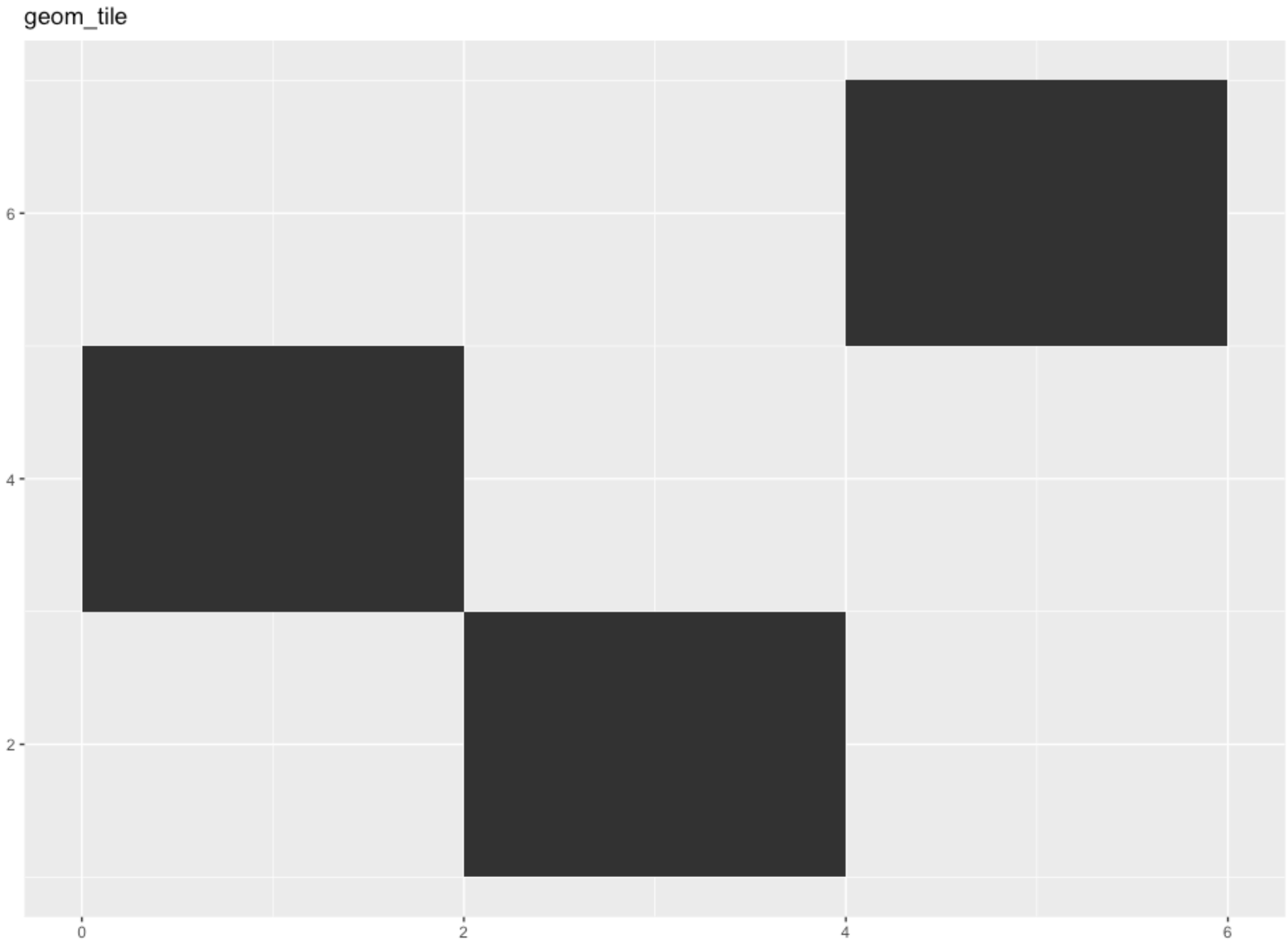
hjust

vjust

angle

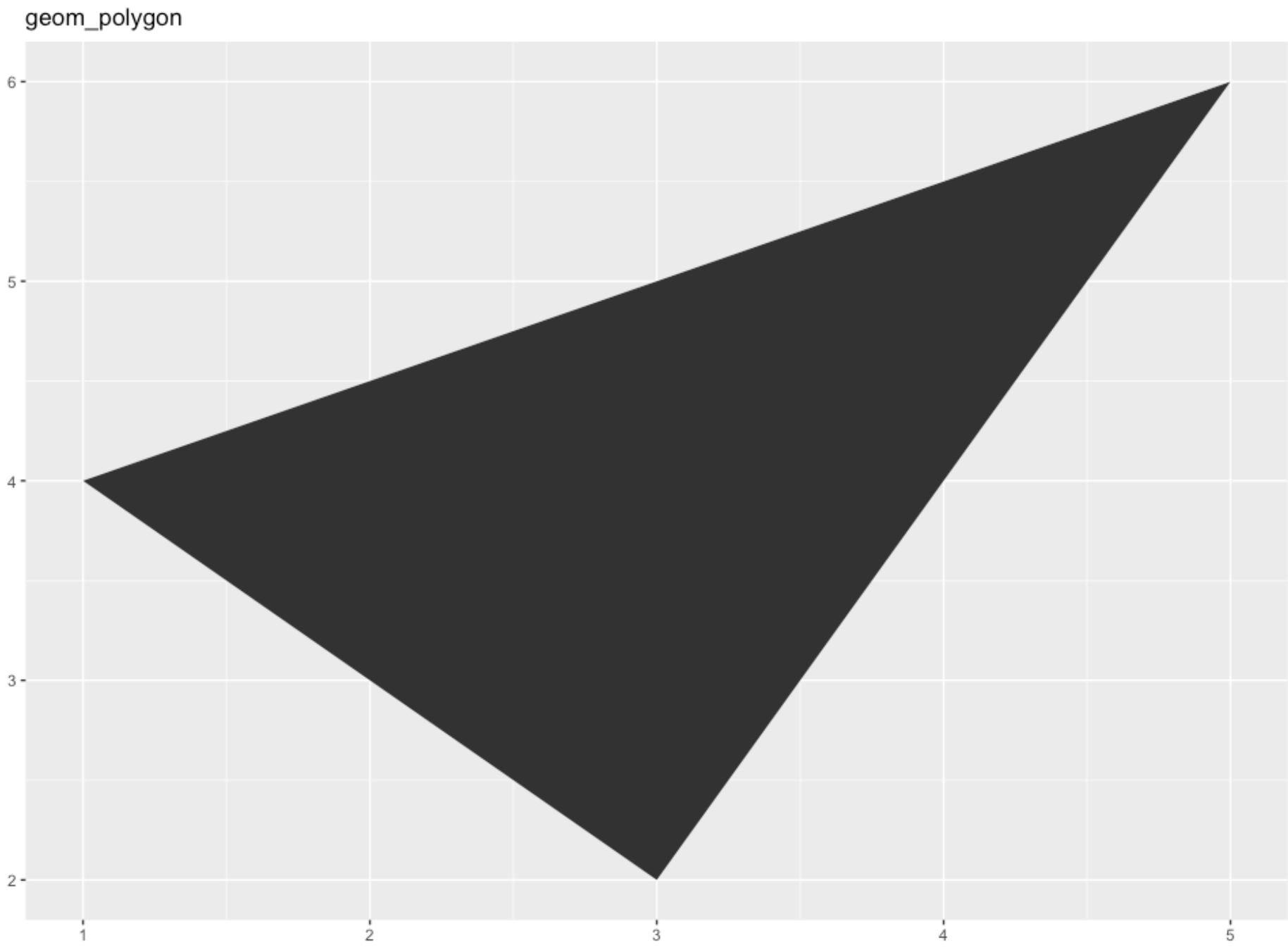
色深图 / 水平图

```
> p + geom_tile() + labs(title = "geom_tile")
```



多边形图

```
> p + geom_polygon() + labs(title = "geom_polygon")
```



钻石数据集

carat	cut	color	clarity	depth	table	price	x	y	z
0.2	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
0.2	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
0.2	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
0.2	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
0.2	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75
0.2	Very Good	J	VVS2	62.8	57.0	336	3.94	3.96	2.48

carat：克拉重量

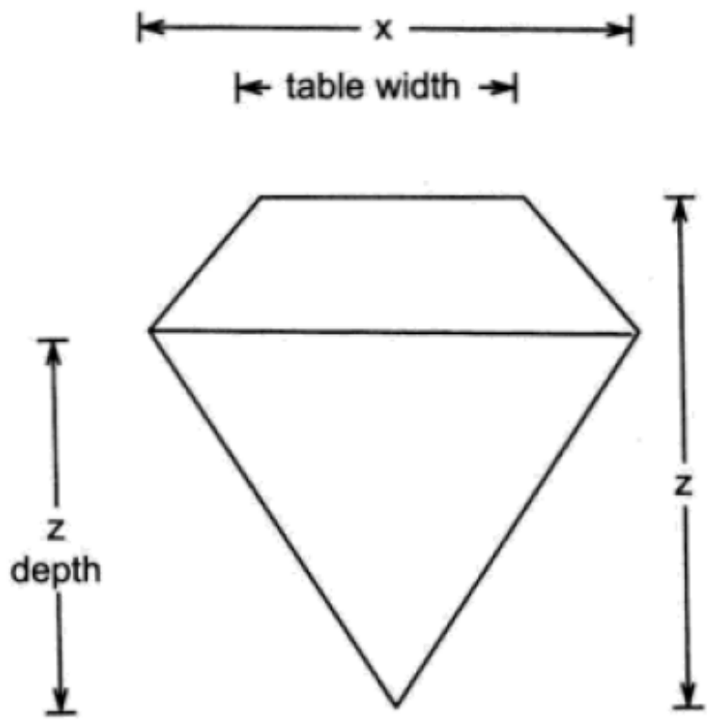
cut：切工

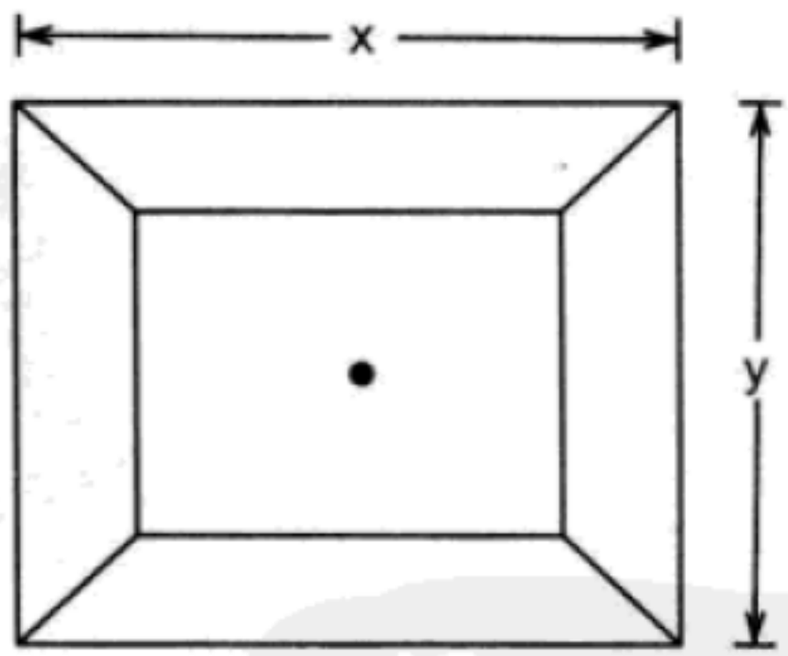
color：颜色

clarity：净度

depty：深度

table：钻面宽度

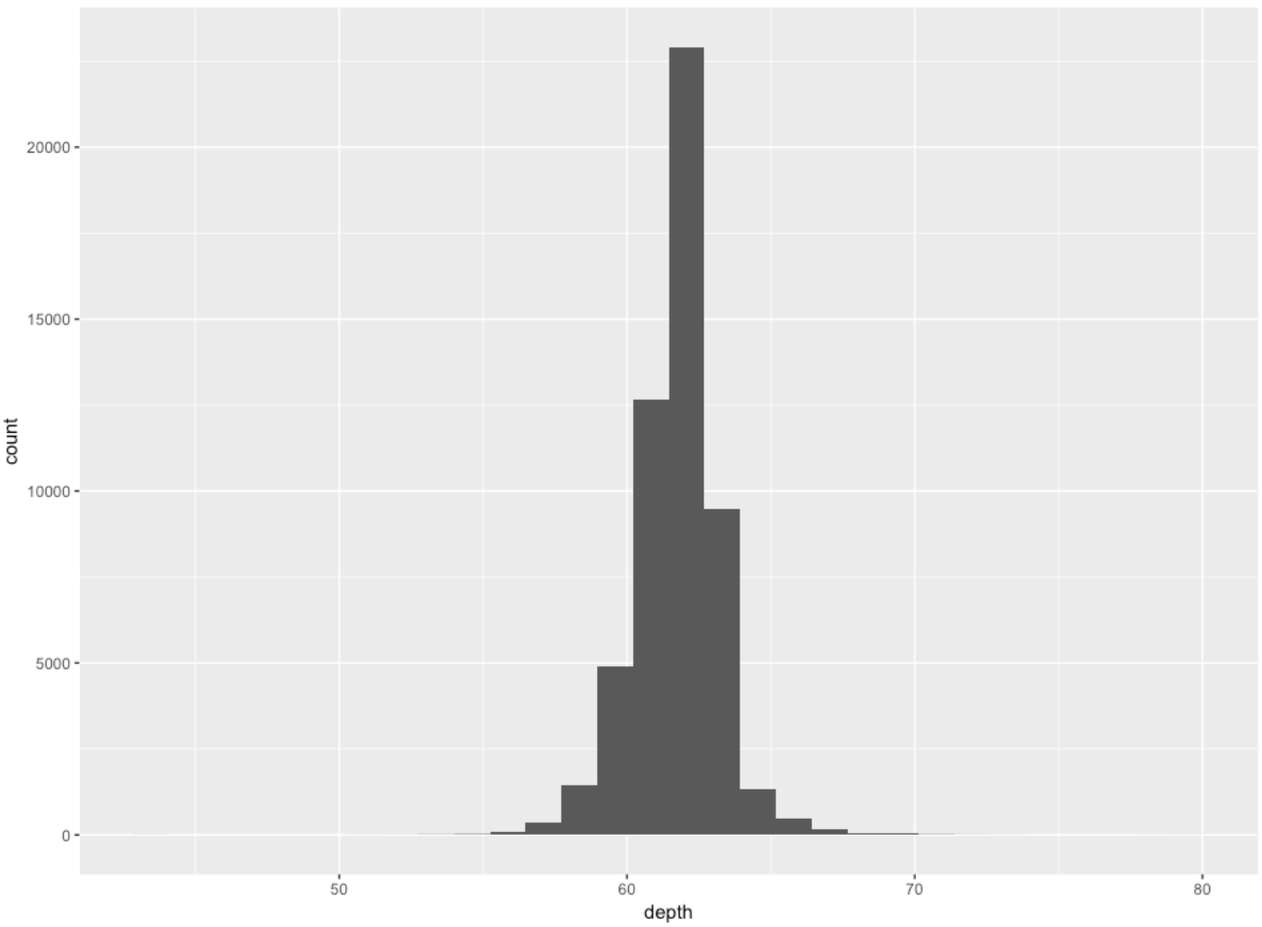


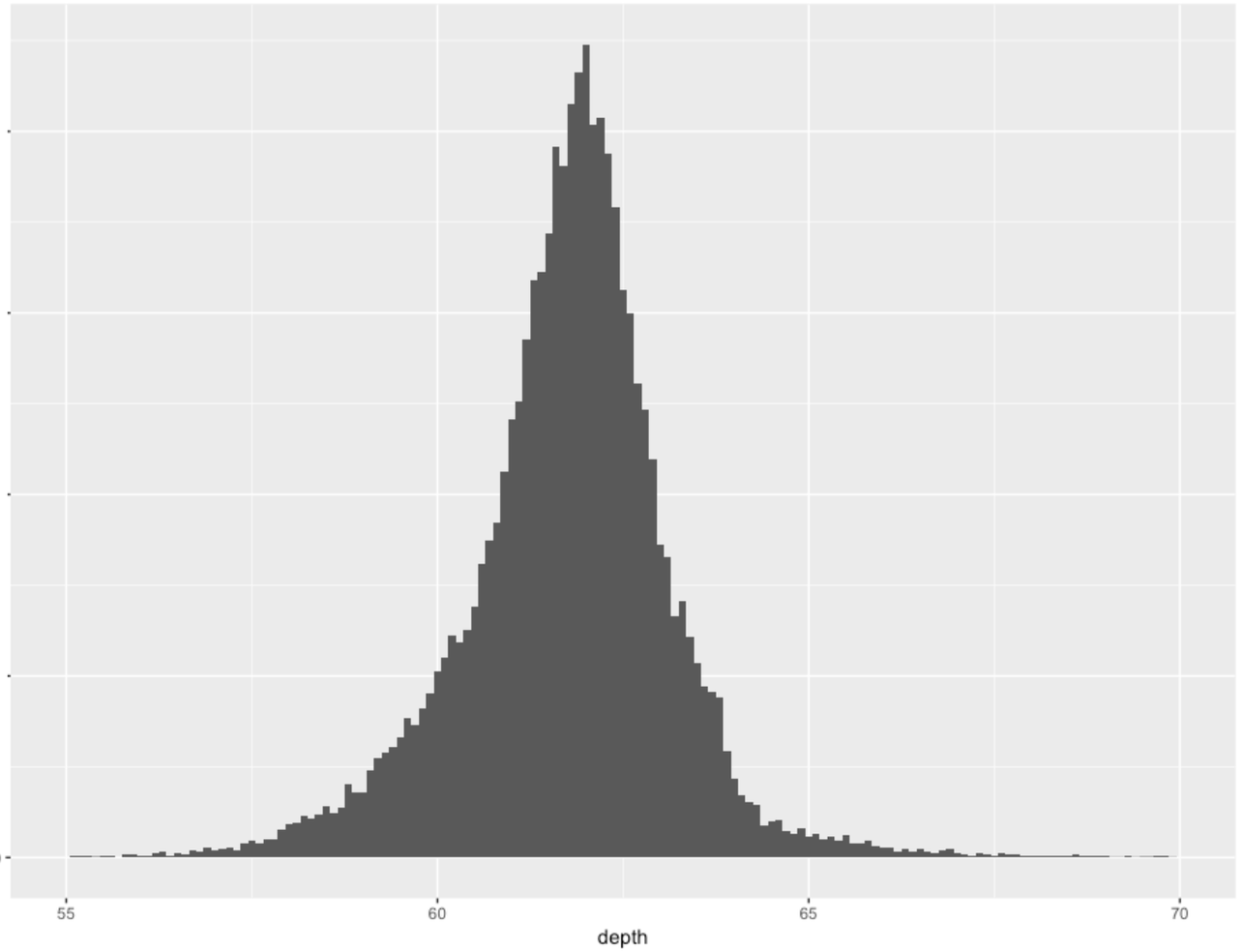


depth = z depth / z * 100
table = table width / x * 100

展示数据分布

```
> qplot(depth, data=diamonds, geom="histogram")
```

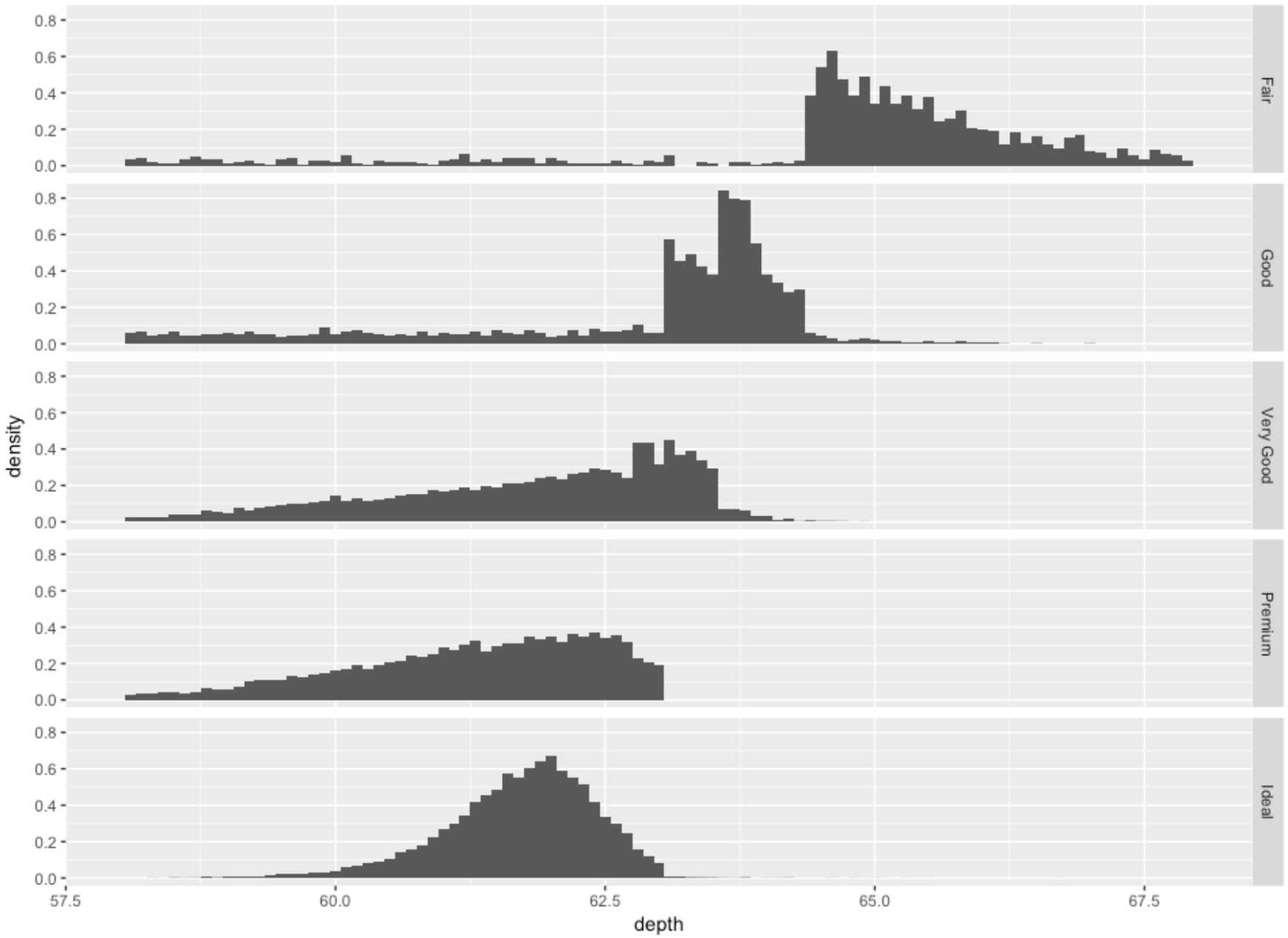





```
> qplot(depth, data=diamonds, geom="histogram", xlim=c(55, 70), binwidth=0.1)
```

分面直方图

```
> depth_dist <- ggplot(diamonds, aes(depth)) + xlim(58, 68)
> depth_dist +
+   geom_histogram(aes(y = ..density..), binwidth = 0.1) +
+   facet_grid(cut ~ .)
```



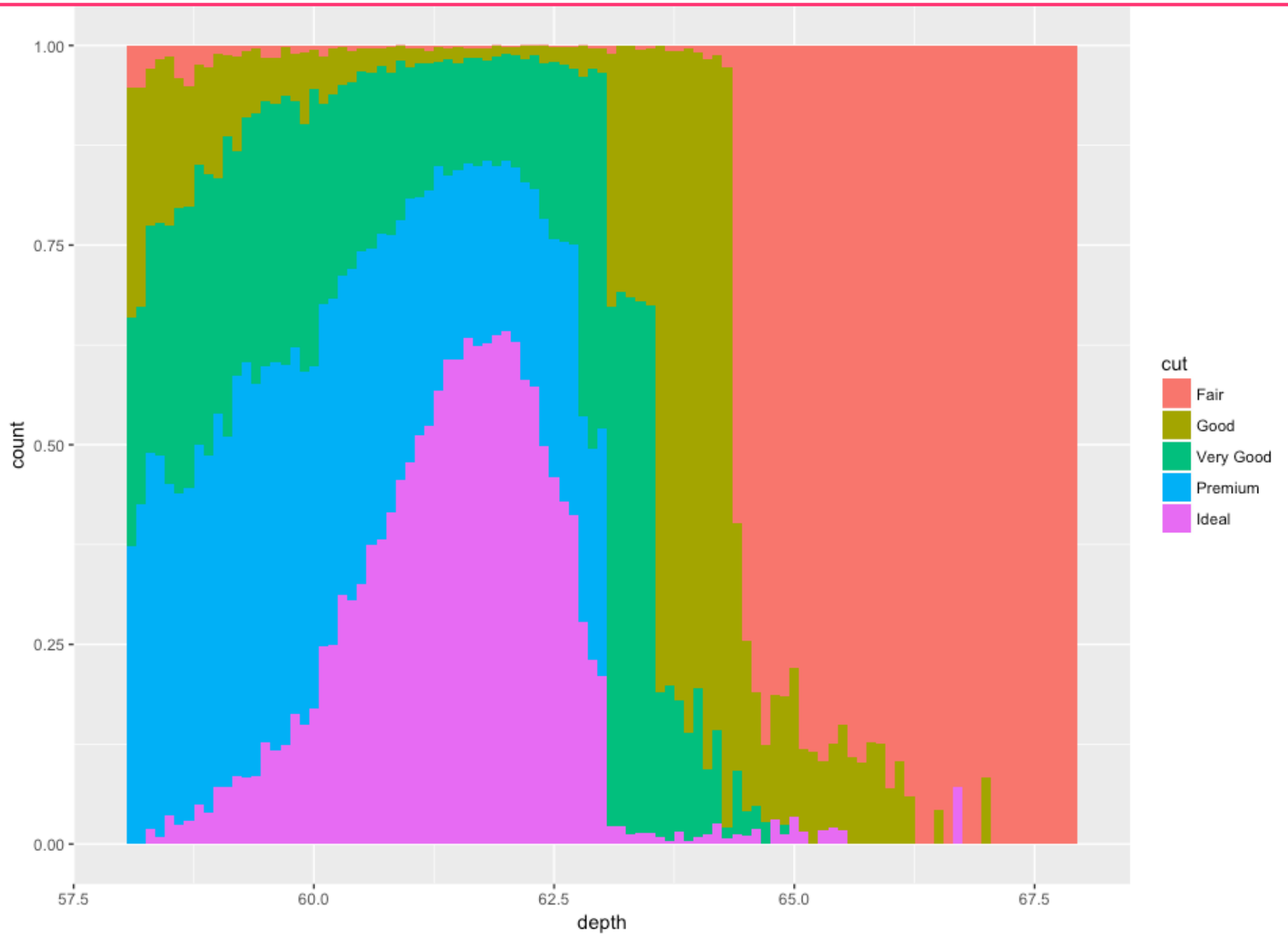
stat_bin

count

density

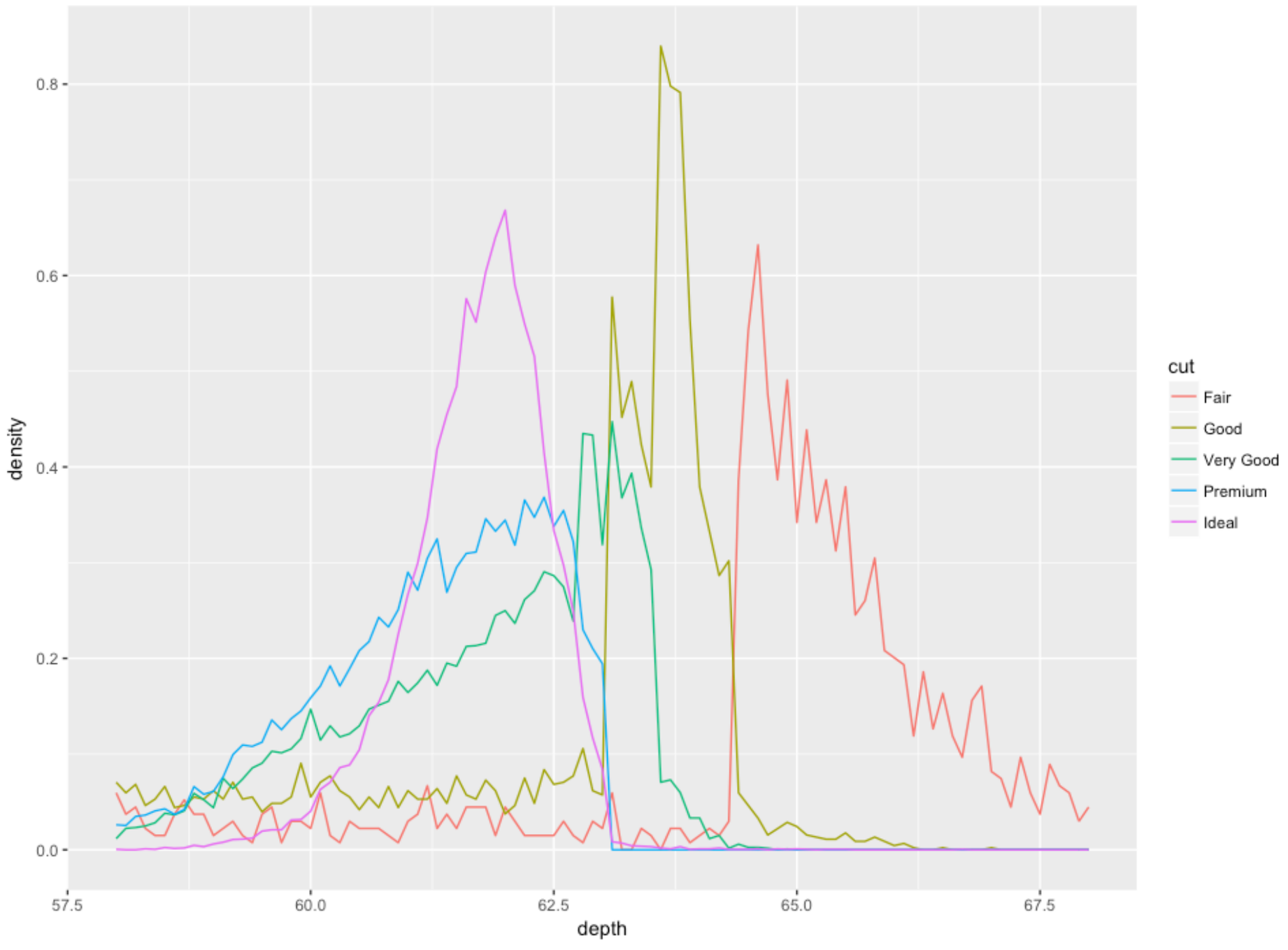
频率多边形图

```
> depth_dist + geom_histogram(aes(fill = cut), binwidth = 0.1,
+    position = "fill")
```



条件密度图

```
> depth_dist + geom_freqpoly(aes(y = ..density.., colour = cut),
+    binwidth = 0.1)
```



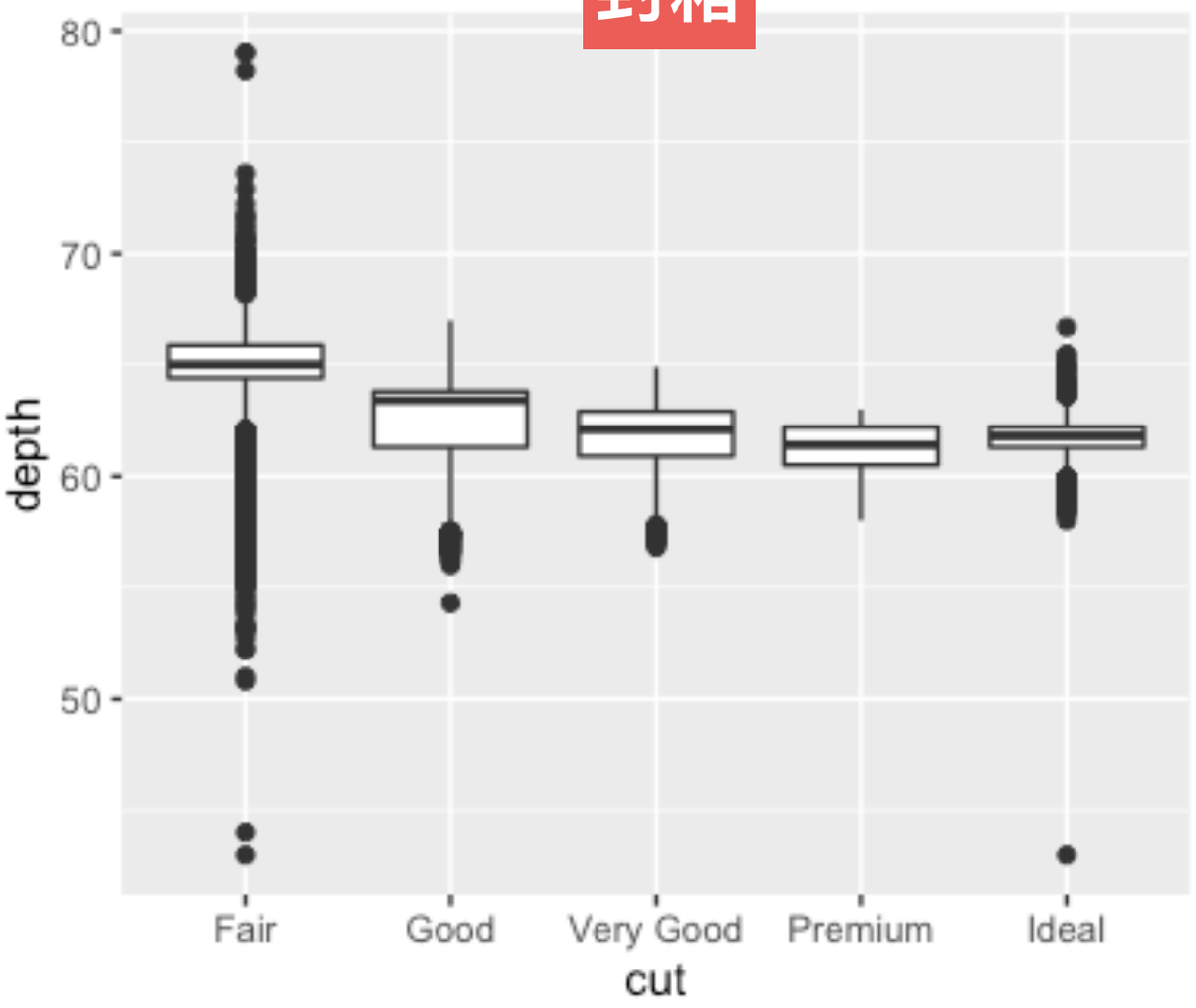
箱线图

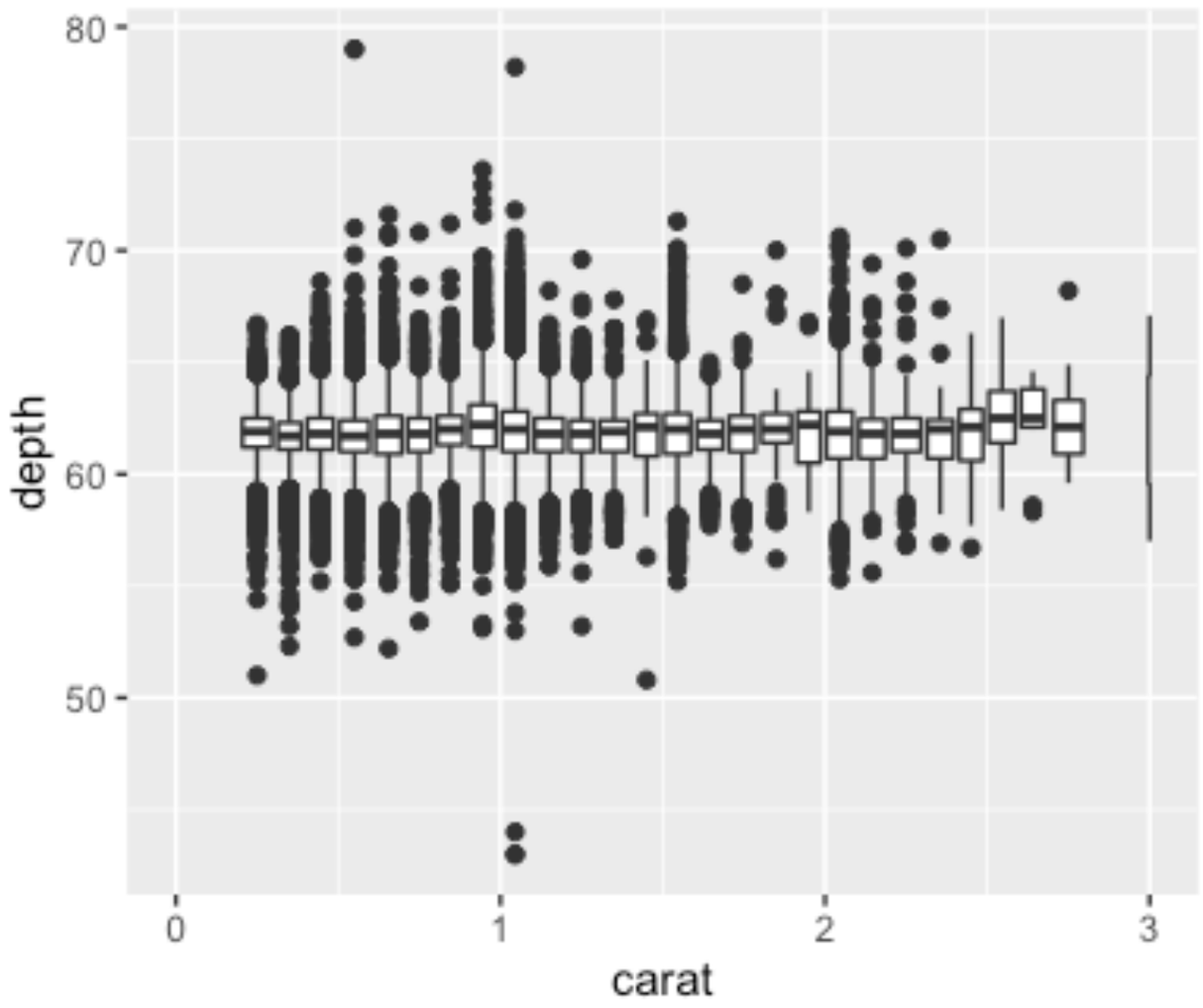
```
library(plyr)

qplot(cut, depth, data=diamonds, geom="boxplot")
qplot(carat, depth, data=diamonds, geom="boxplot",
      group = round_any(carat, 0.1, floor), xlim = c(0, 3))
```

连续型变量

类别型变量

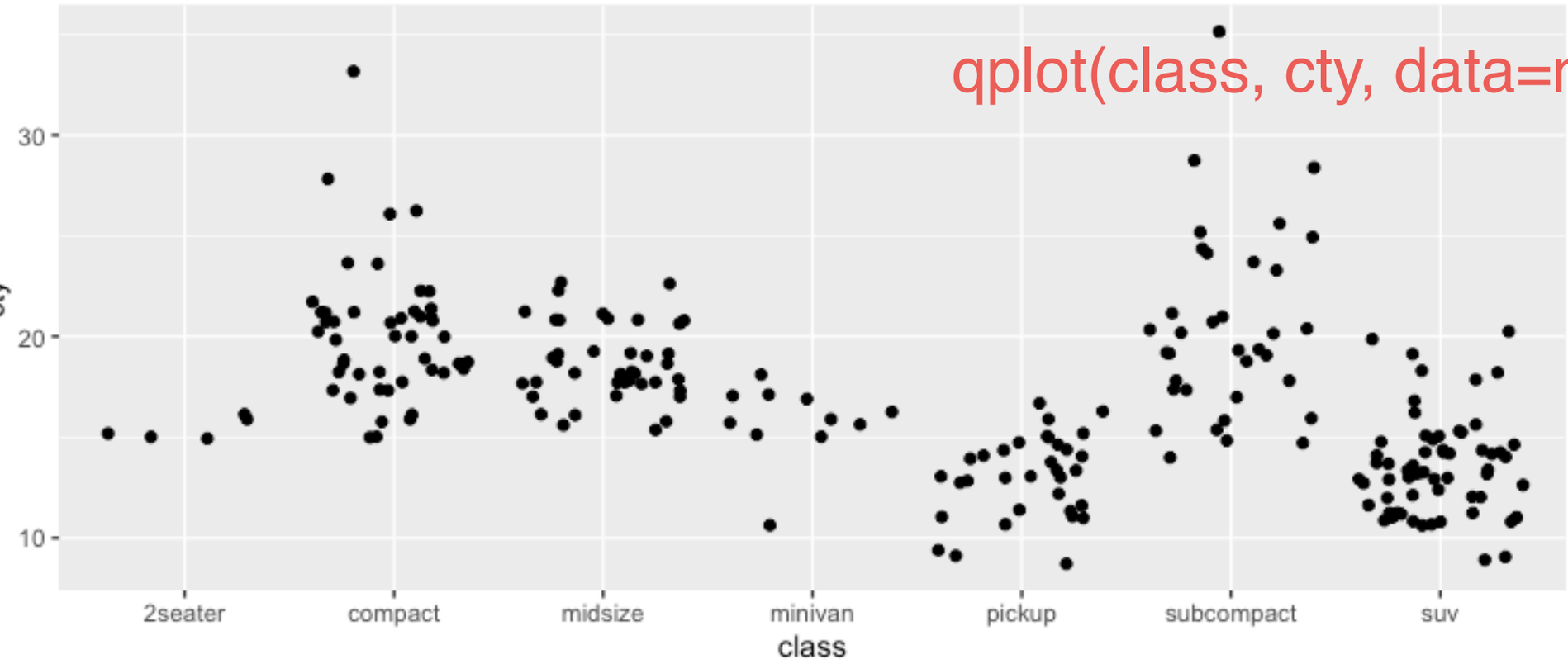
封箱





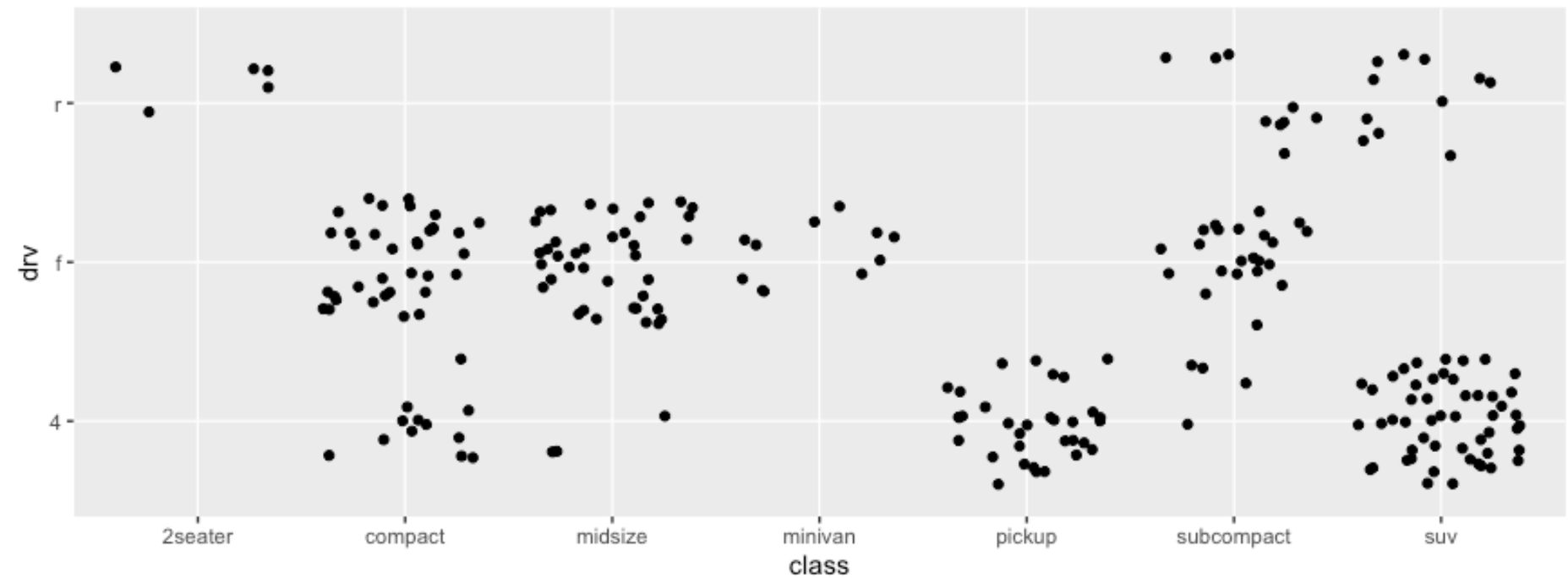
抖动

qplot(class, cty, data=mpg, geom="jitter")



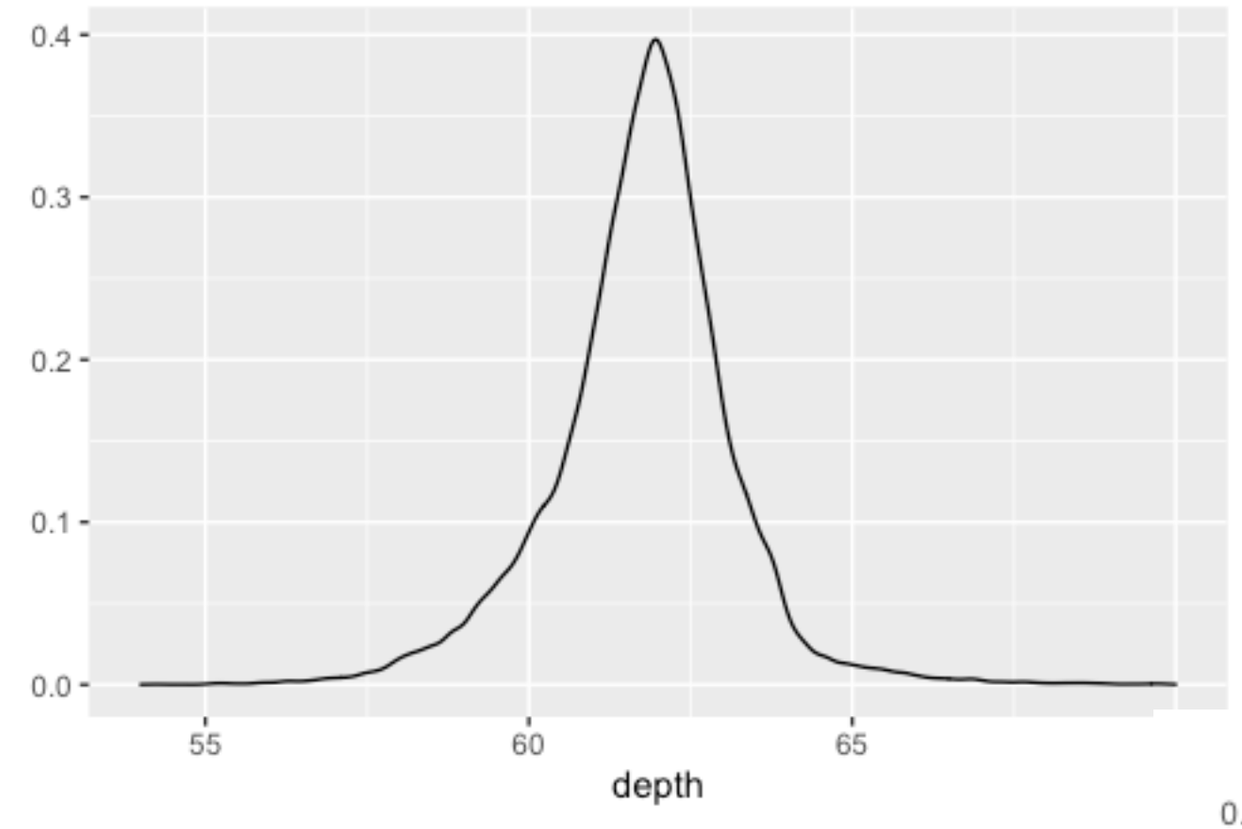
连续型变量

qplot(class, drv, data=mpg, geom="jitter")



离散型变量

密度图

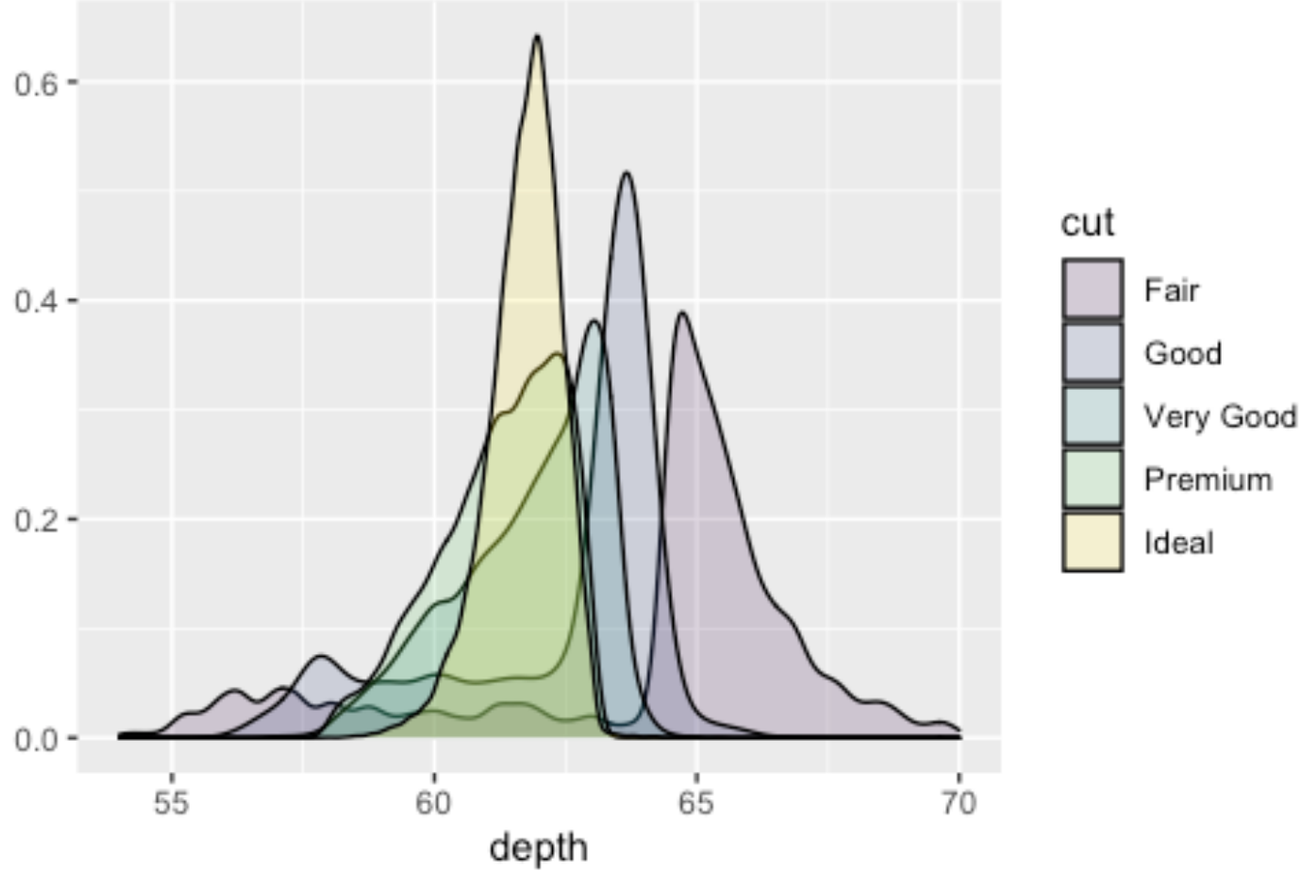


qplot(depth, data=diamonds, geom="density", xlim = c(54, 70))

核平滑方法

qplot(depth, data=diamonds, geom="density", xlim = c(54, 70), fill = cut, alpha = I(0.2))

难于回溯到数据

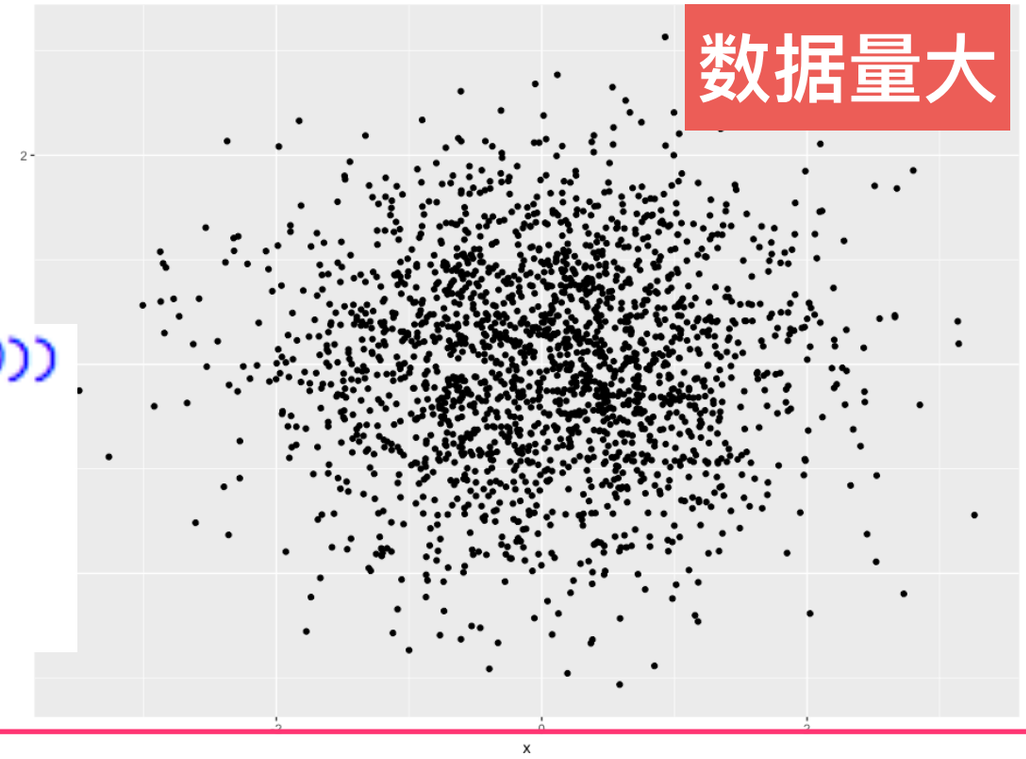


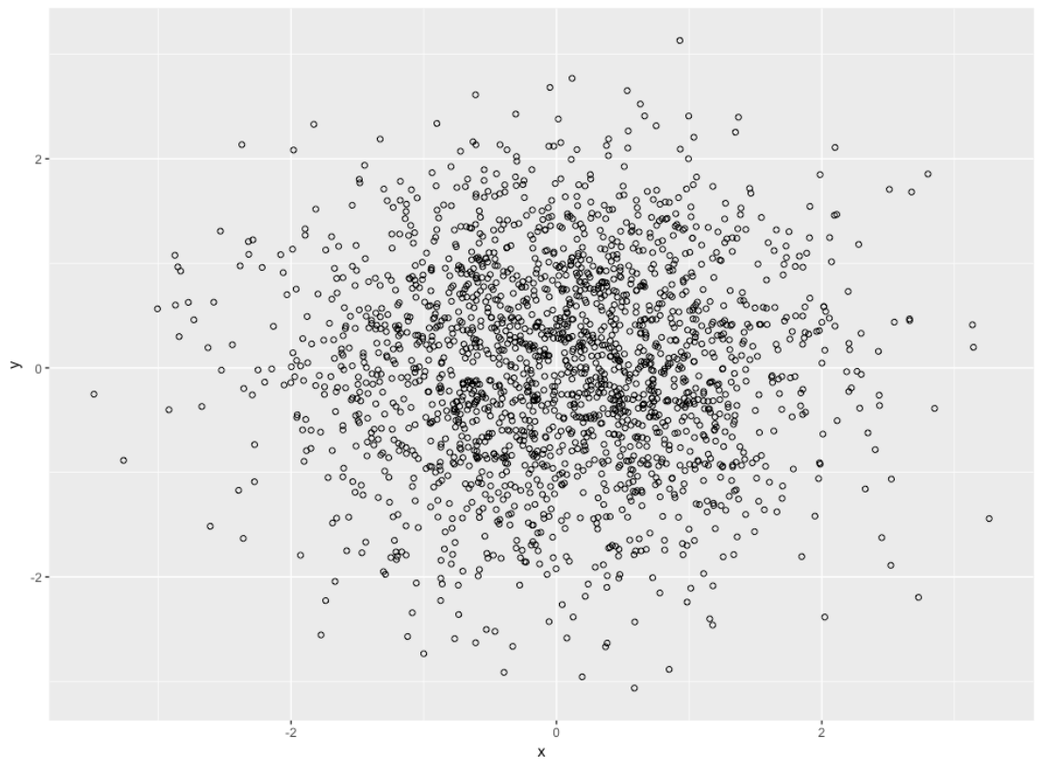
遮盖绘制

使用点的大小

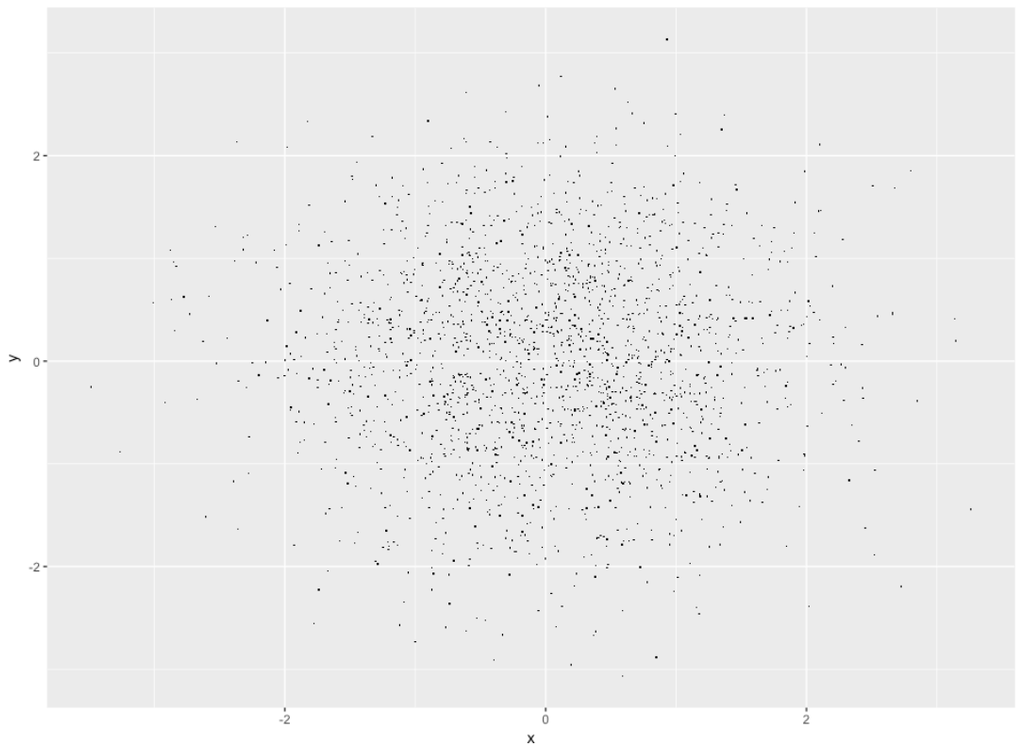
```
> df <- data.frame(x = rnorm(2000), y = rnorm(2000))
> norm <- ggplot(df, aes(x, y))
> norm + geom_point()
> norm + geom_point(shape = 1)
> norm + geom_point(shape = ".") # Pixel sized
```

数据量大





shape

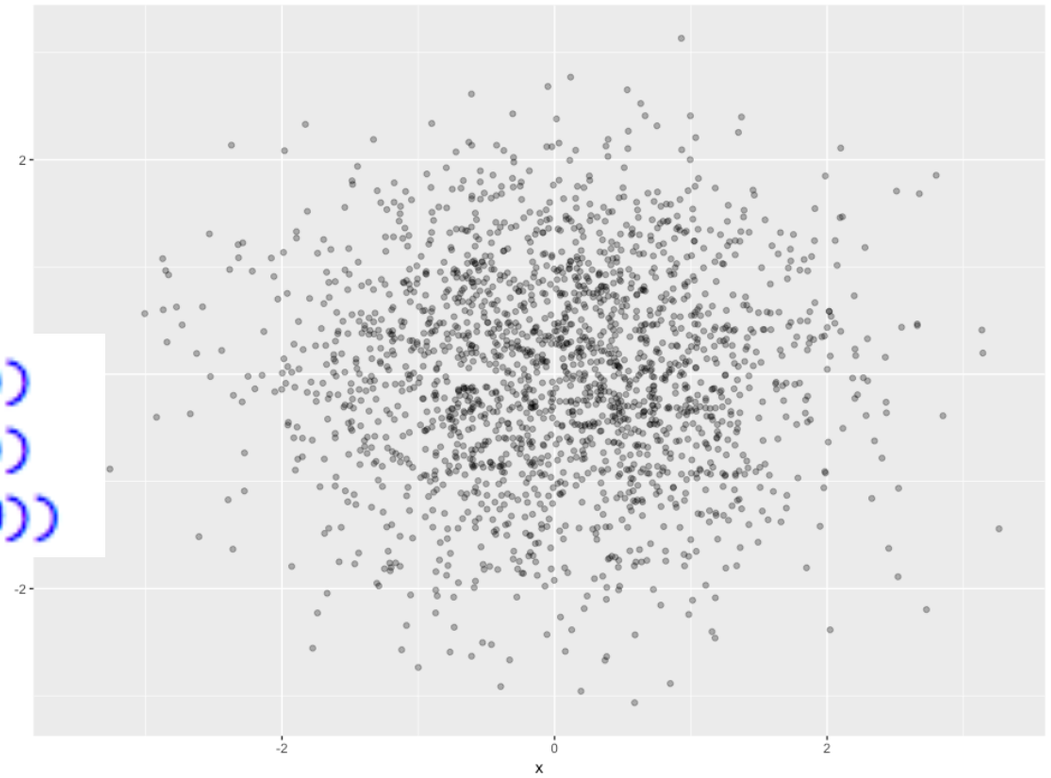


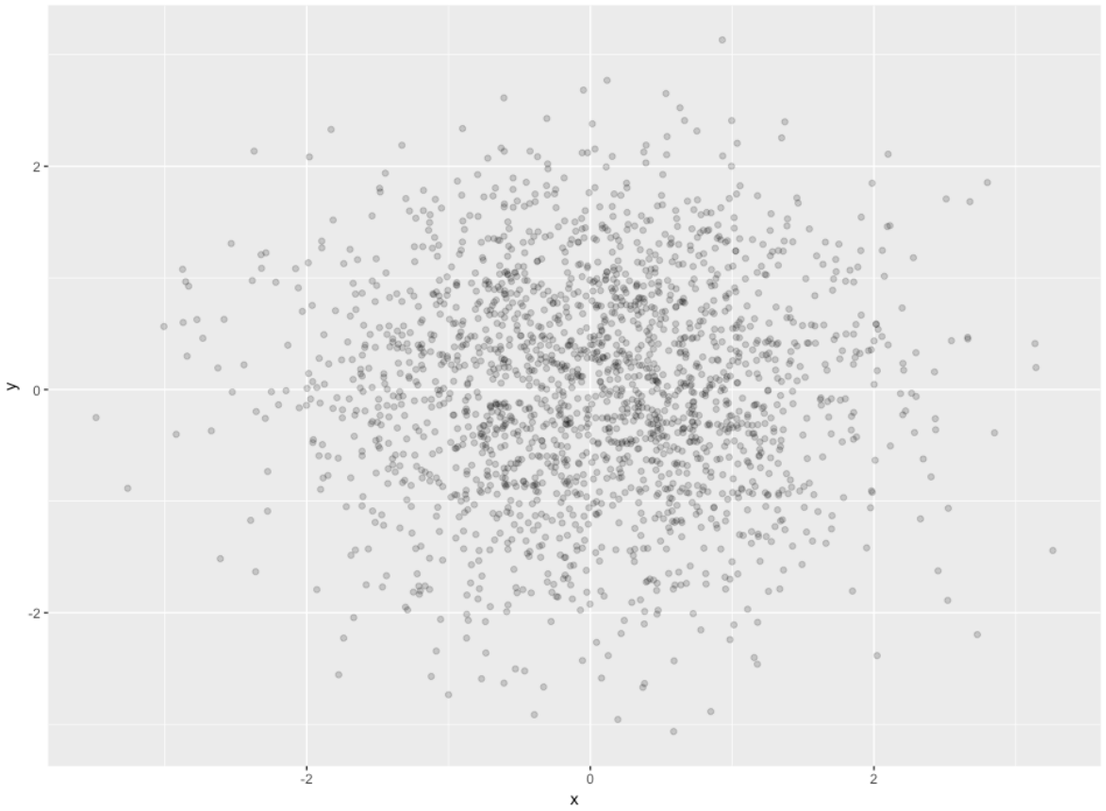
遮盖绘制

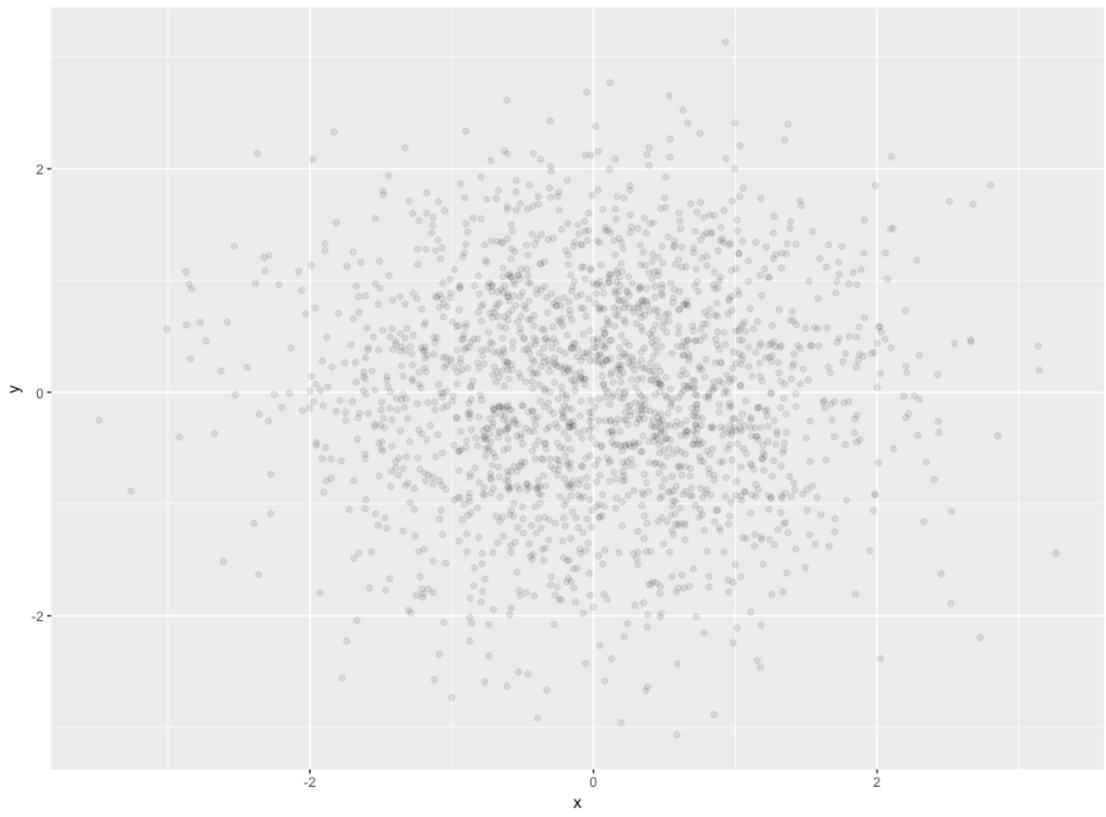
使用点的透明度

alpha

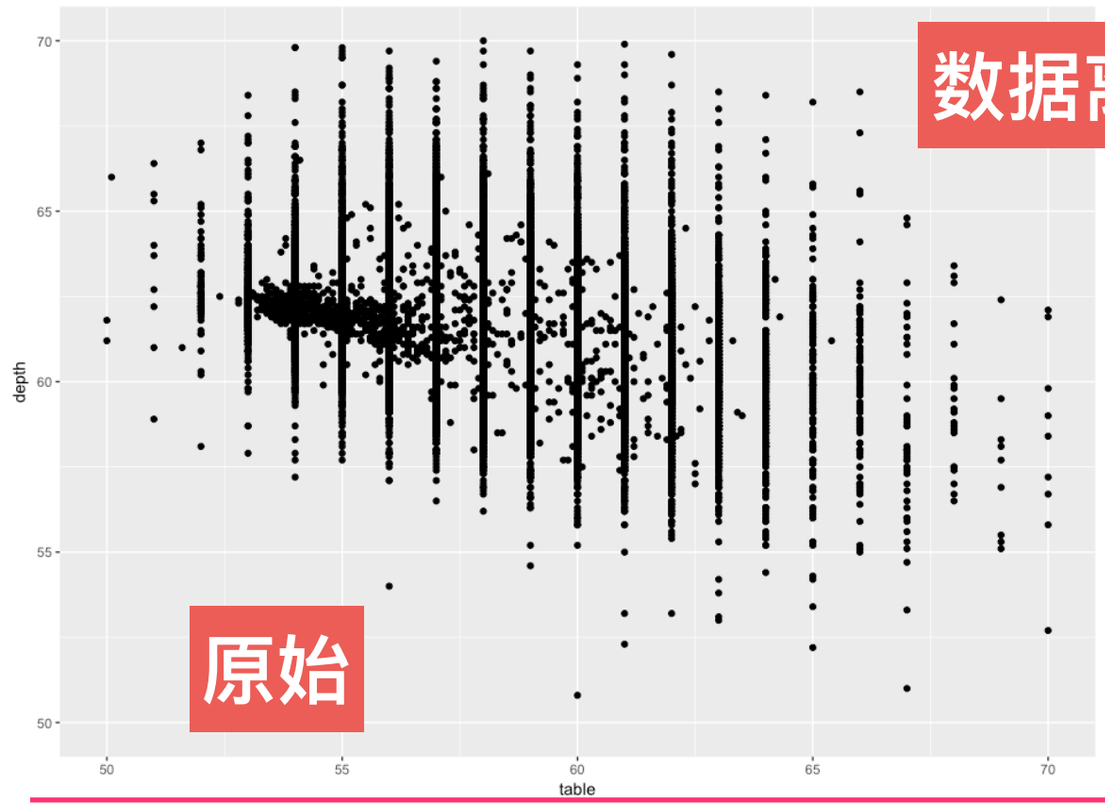
```
> norm + geom_point(colour = alpha("black", 1/3))
> norm + geom_point(colour = alpha("black", 1/5))
> norm + geom_point(colour = alpha("black", 1/10))
```







遮盖绘制



数据离散性

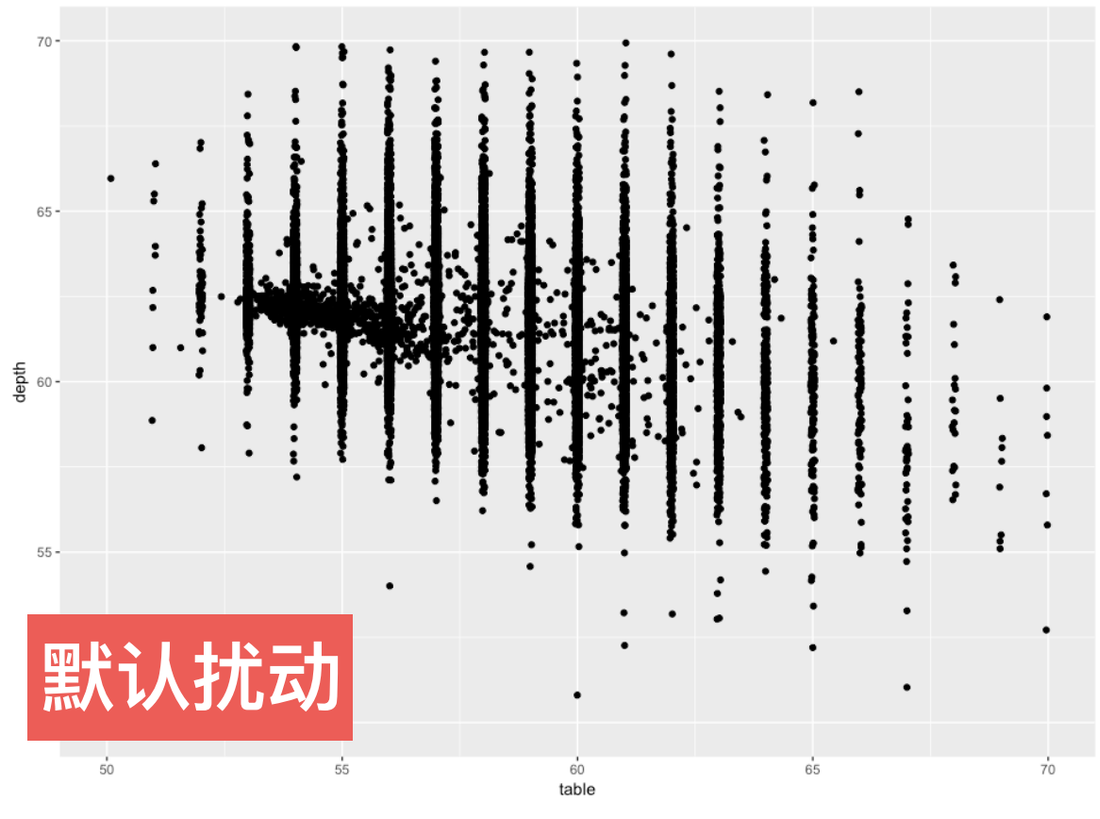
原始

使用随机扰动

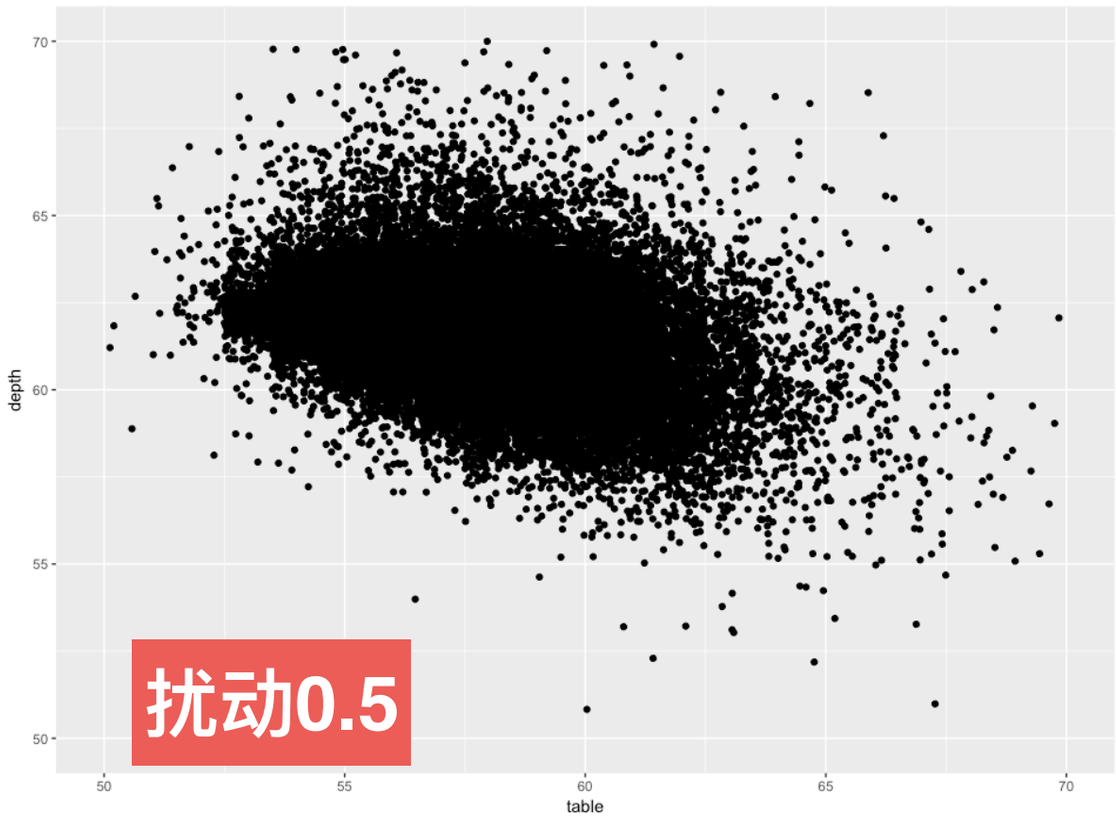
```
> td <- ggplot(diamonds, aes(table, depth)) +
+   xlim(50, 70) + ylim(50, 70)
> td + geom_point()
```

```
> td + geom_jitter()
```

```
> jit <- position_jitter(width = 0.5)
> td + geom_jitter(position = jit)
```

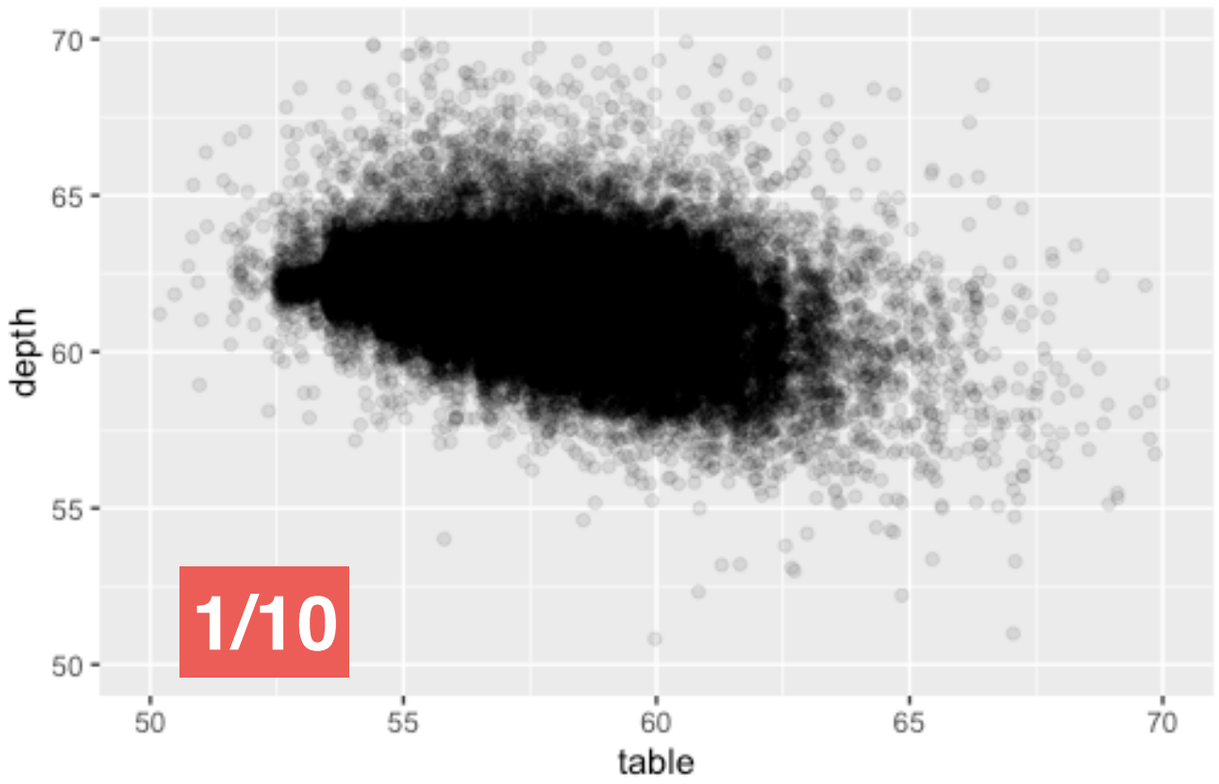


默认扰动



扰动0.5

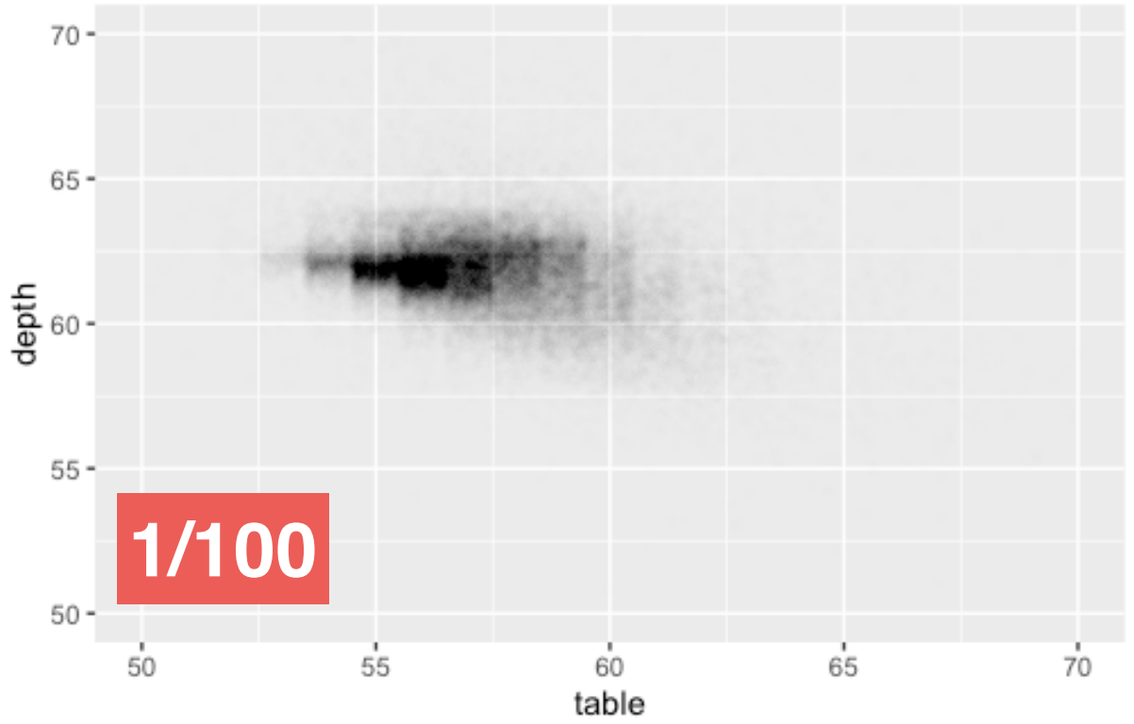
遮盖绘制



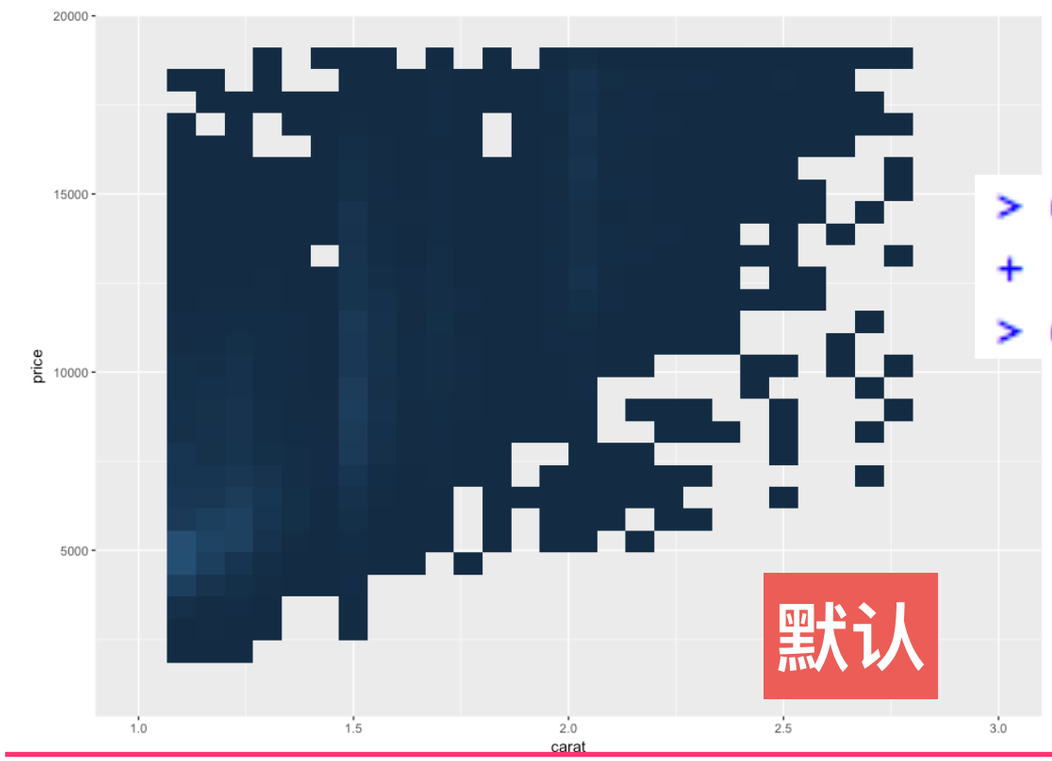

```
td + geom_jitter(position = jit,
colour = alpha("black", 1/10))
```

```
td + geom_jitter(position = jit,
colour = alpha("black", 1/50))
```

```
td + geom_jitter(position = jit,
colour = alpha("black", 1/200))
```



遮盖绘制



使用分箱计数

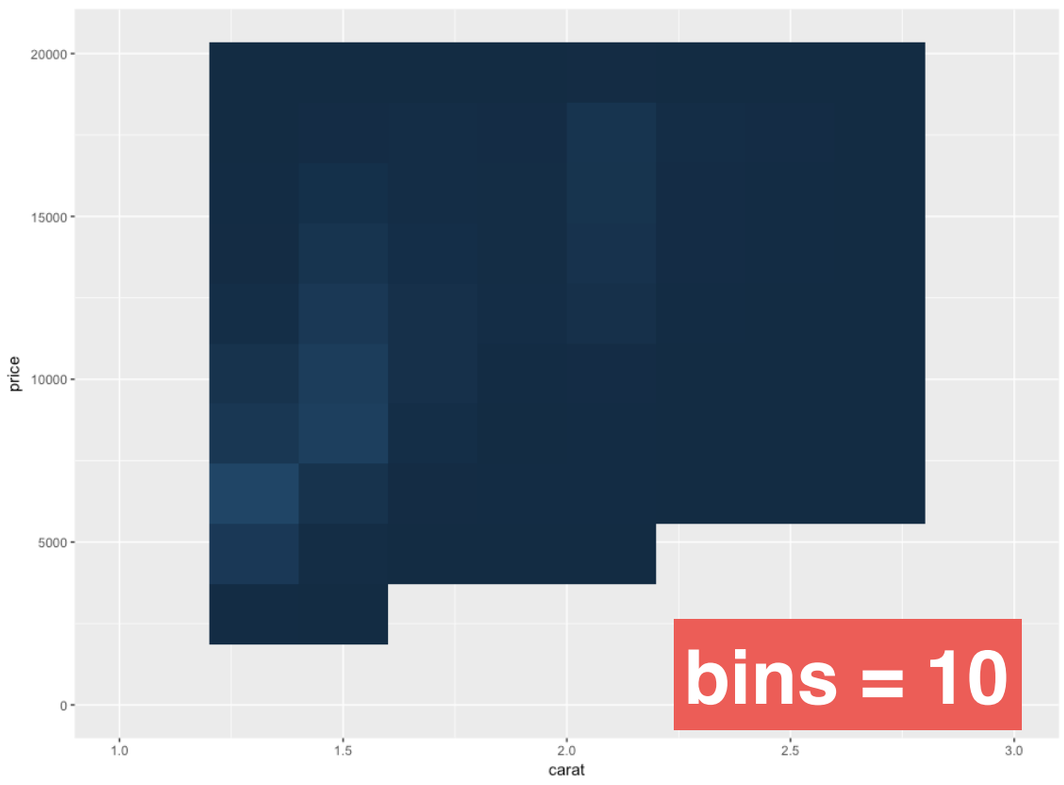
二维直方图

```
> d <- ggplot(diamonds, aes(carat, price)) + xlim(1,3) +
+   theme(legend.position = "none")
> d + stat_bin2d()
```

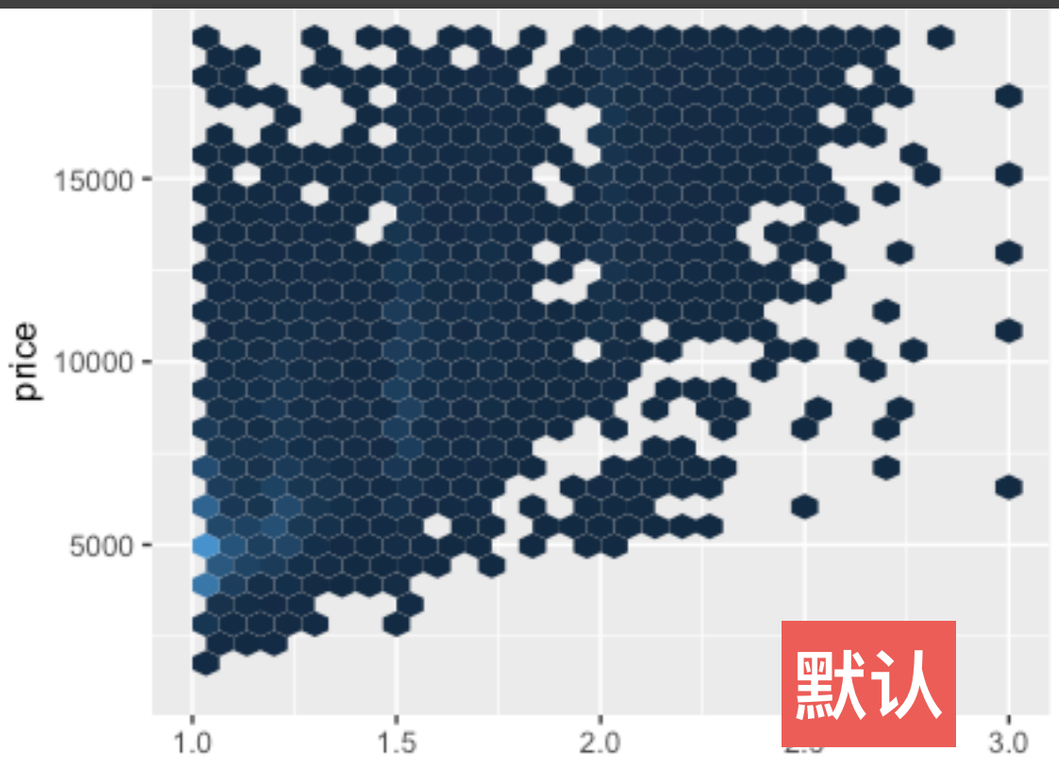
stat_bin2d

```
> d + stat_bin2d(bins = 10)
```

```
> d + stat_bin2d(binwidth=c(0.02, 200))
```



遮盖绘制



默认

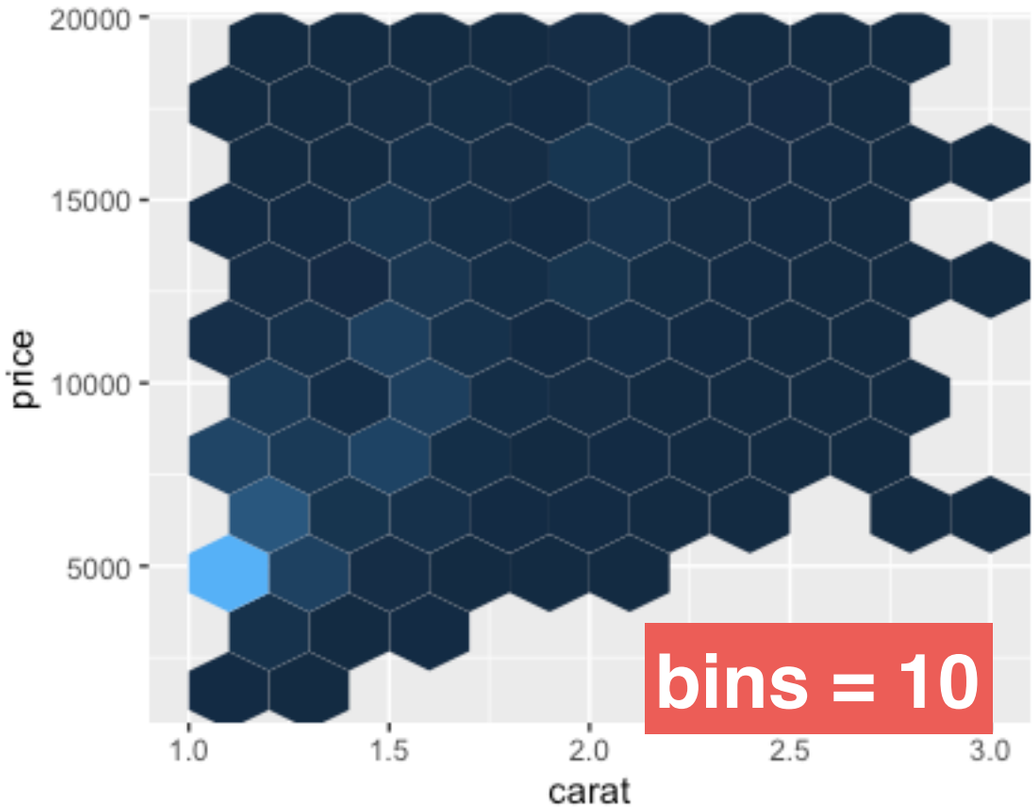
六边形

hexbin包

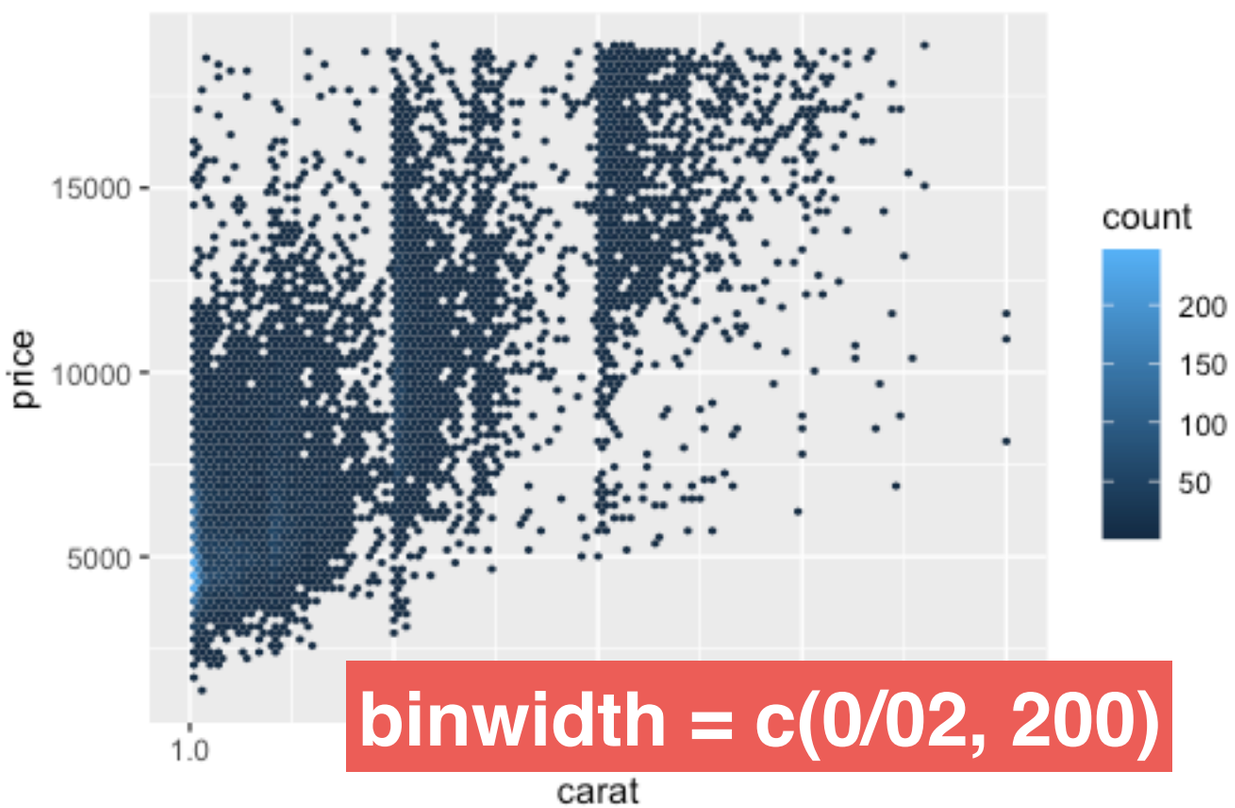
d + stat_binhex()

d + stat_binhex(bins = 10)

d + stat_binhex(binwidth=c(0.02, 200))



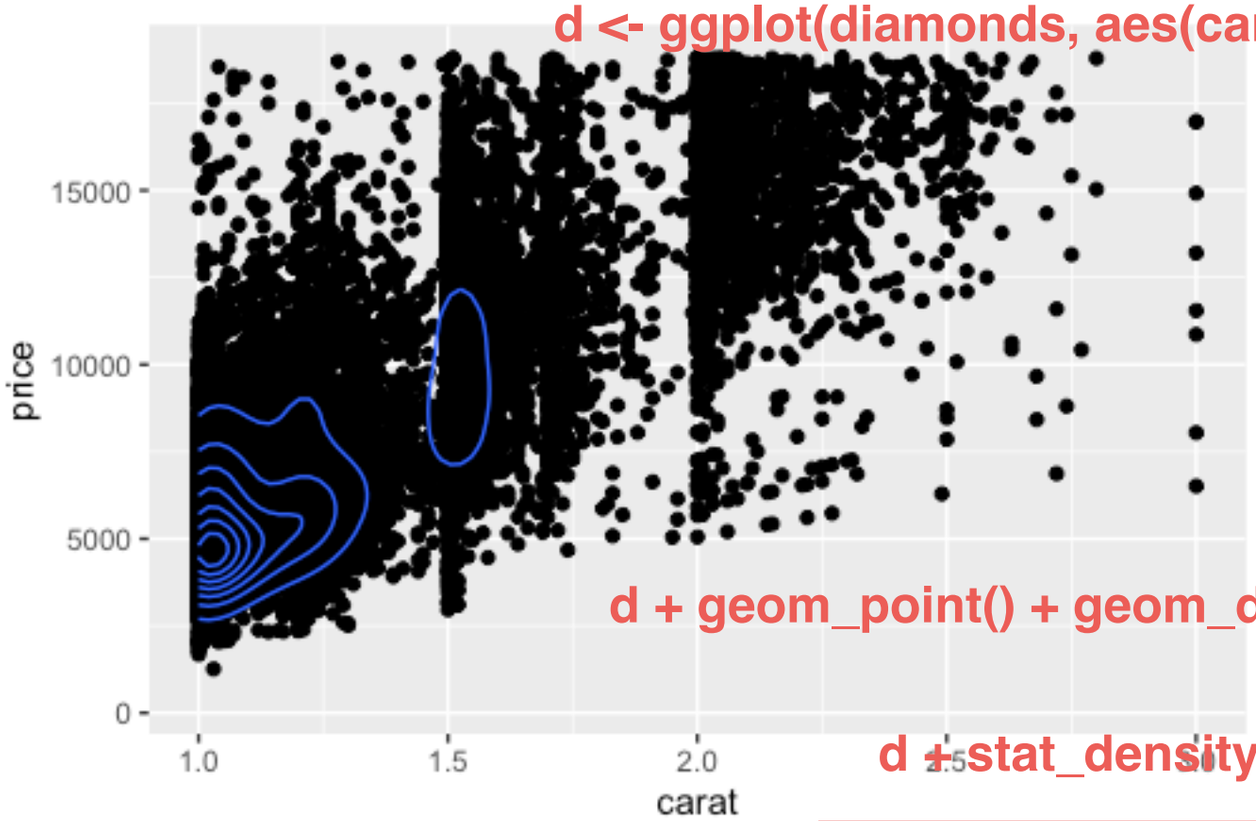
bins = 10



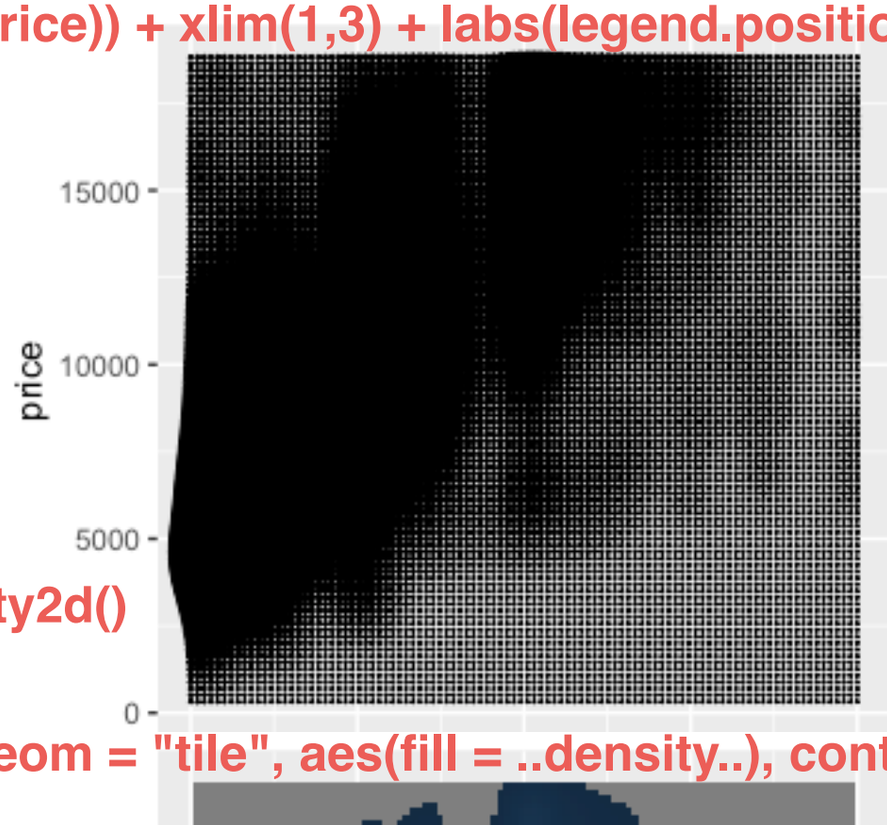
binwidth = c(0/02, 200)

遮盖绘制

d <- ggplot(diamonds, aes(carat, price)) + xlim(1,3) + labs(legend.position = "none")



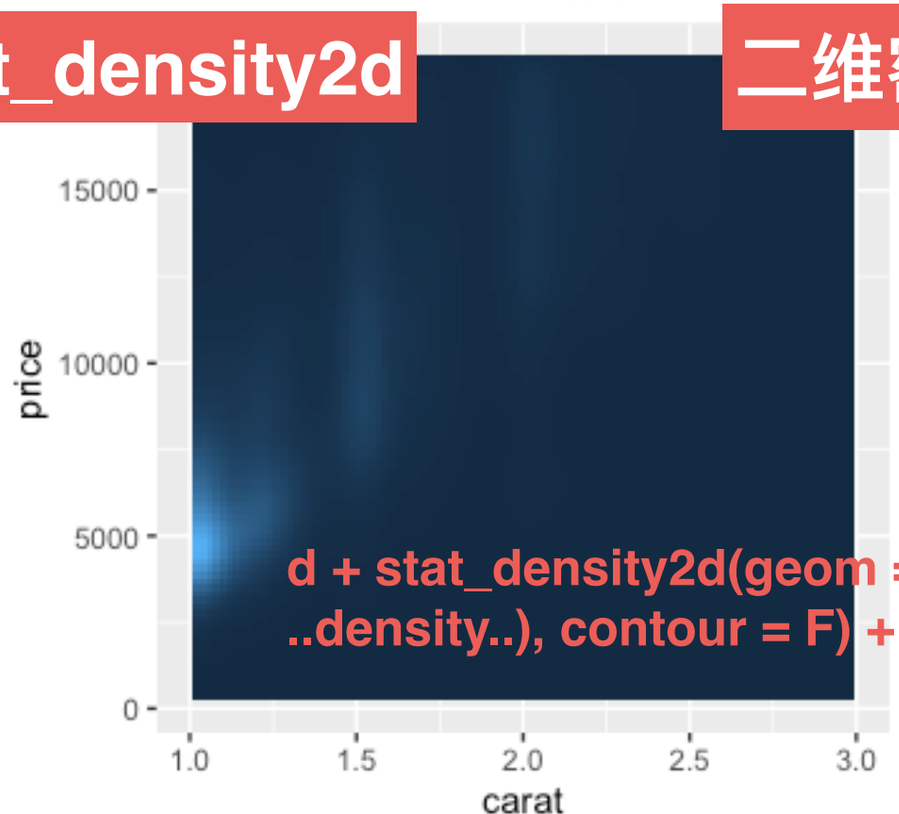
d + geom_point() + geom_density2d()



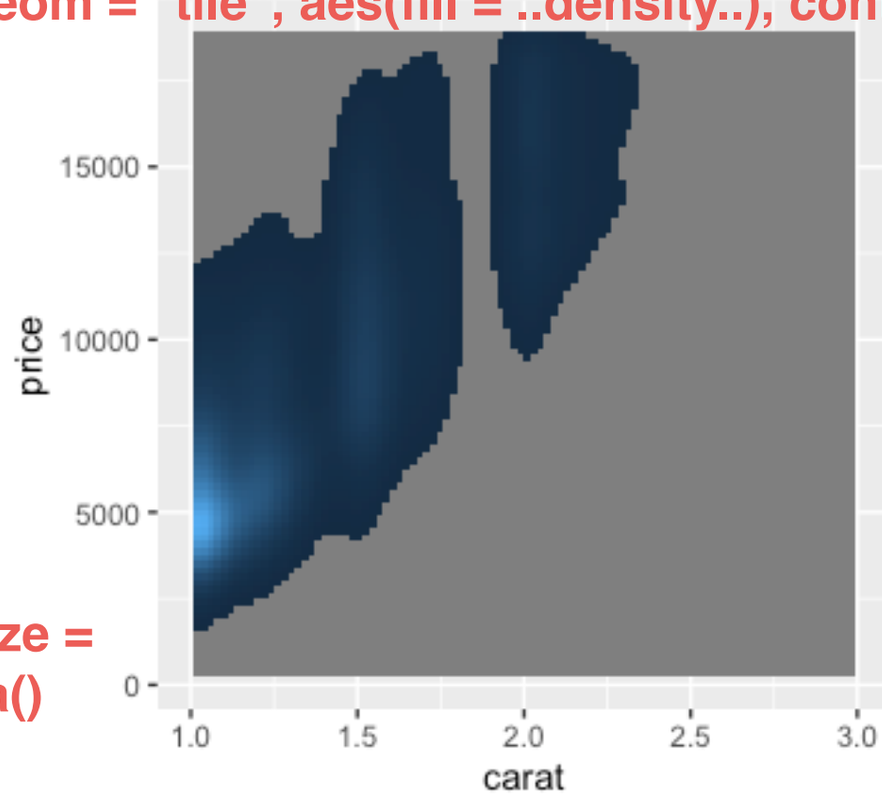
stat_density2d

二维密度估计

d + stat_density2d(geom = "tile", aes(fill = ..density..), contour = F)



d + stat_density2d(geom = "point", aes(size = ..density..), contour = F) + scale_size_area()



last_plot() + scale_fill_gradient(limits = c(1e-5,8e-4))

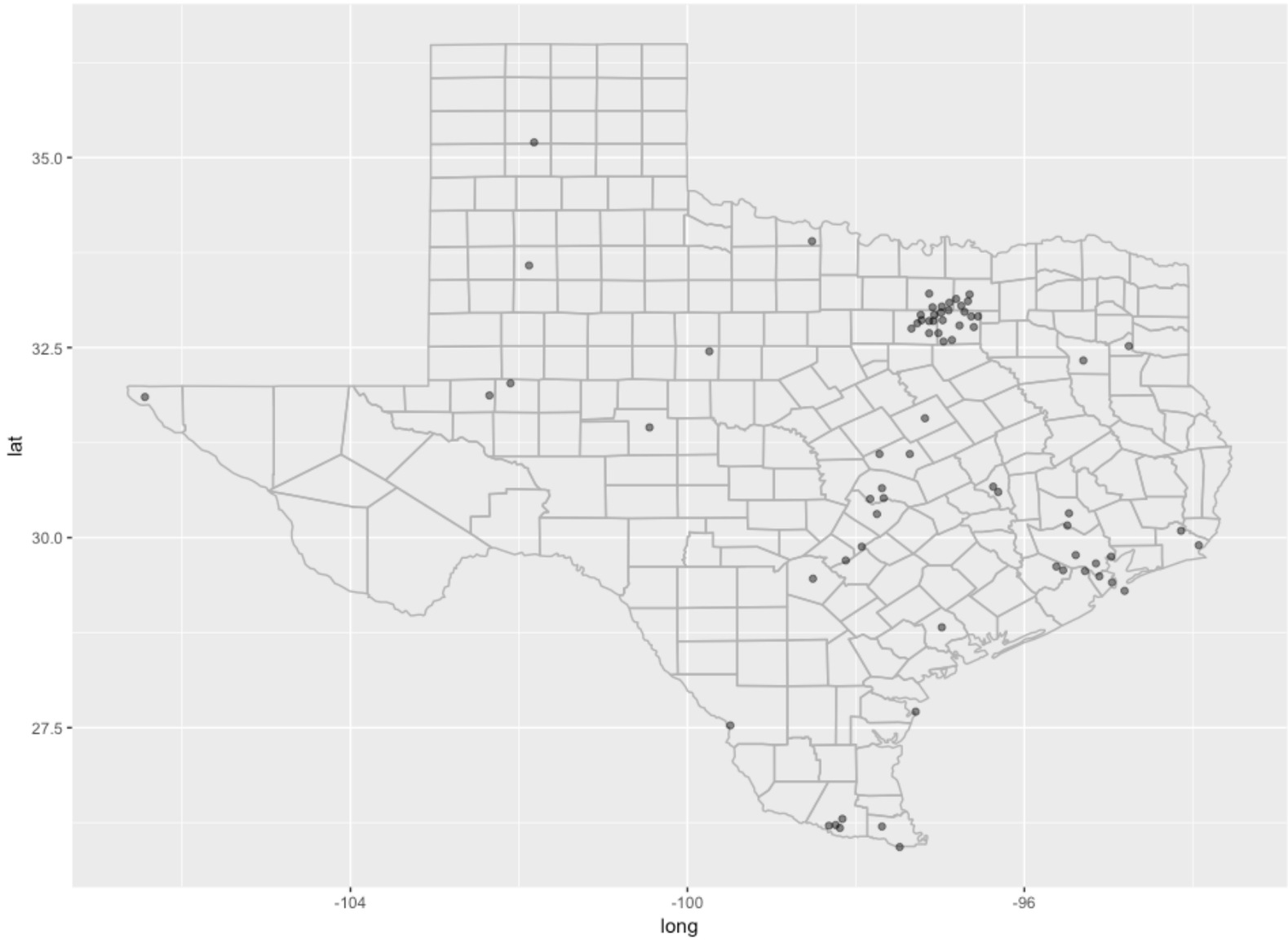
地图

```
> library(maps)
> data(us.cities)
> big_cities <- subset(us.cities, pop > 500000)
> qplot(long, lat, data = big_cities) + borders("state", size = 0.5)
```

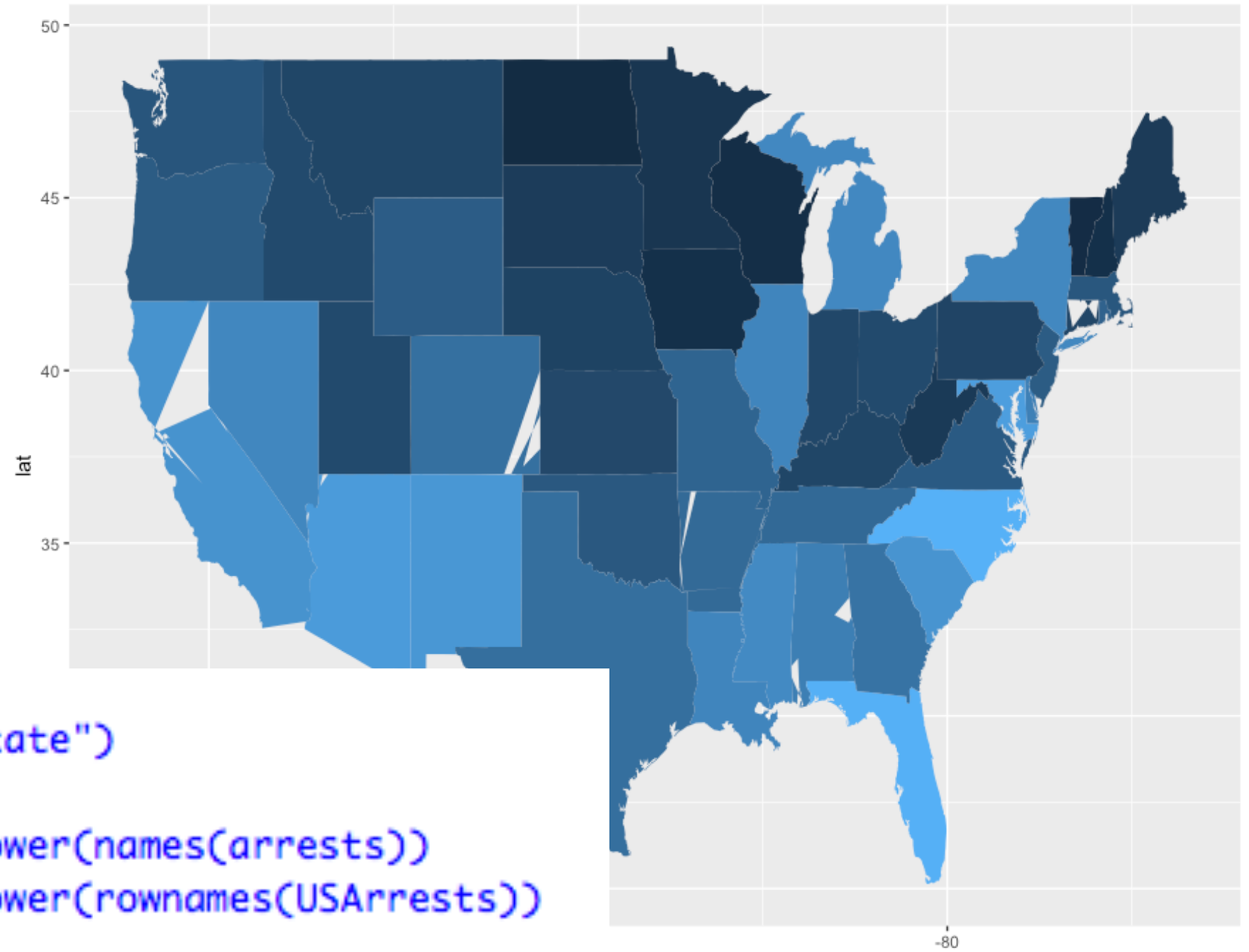
borders(): 地图边界

地图

```
> tx_cities <- subset(us.cities, country.etc == "TX")
> ggplot(tx_cities, aes(long, lat)) +
+   borders("county", "texas", colour = "grey70") +
+   geom_point(colour = alpha("black", 0.5))
```



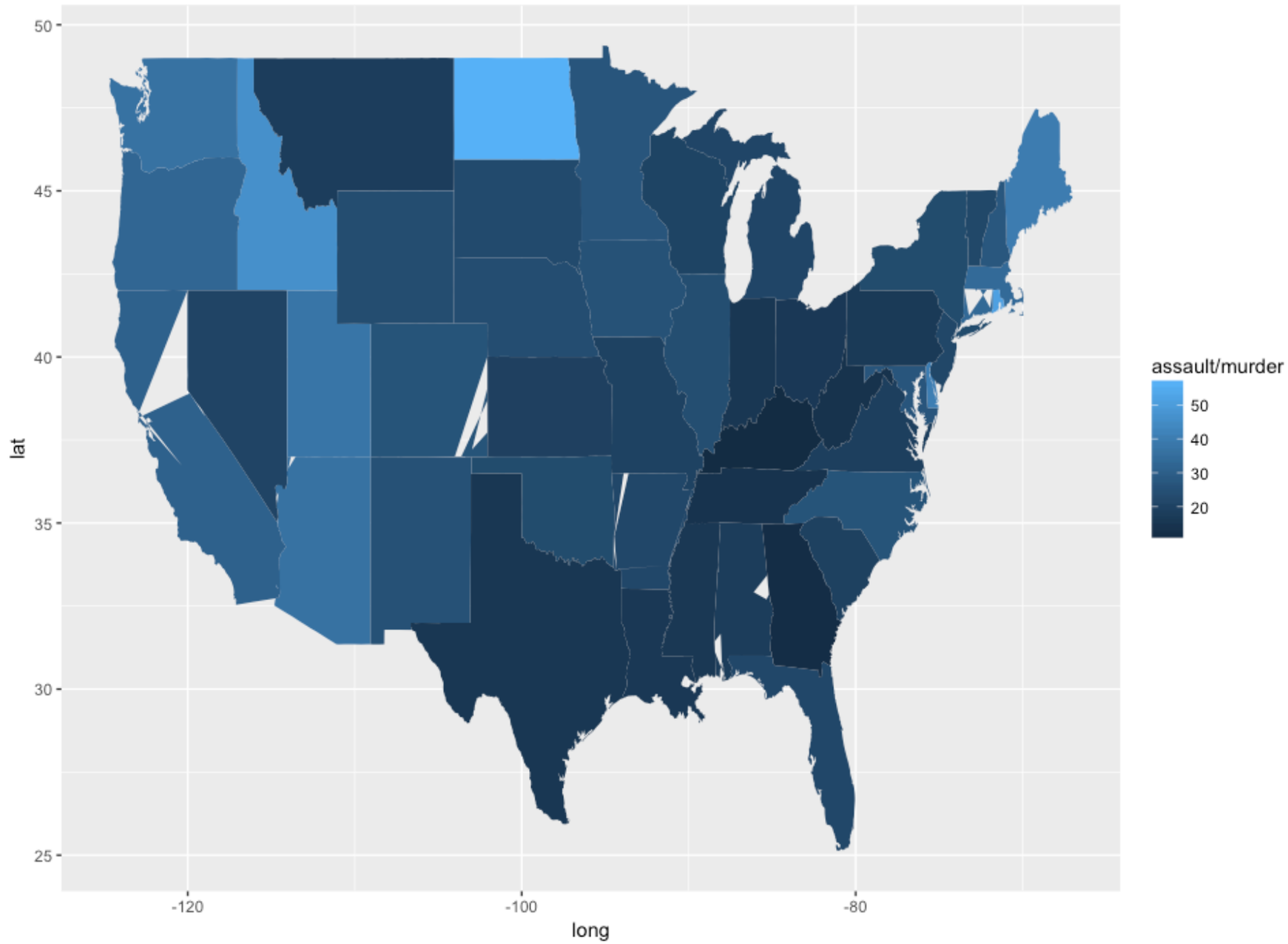
地图




```
> library(maps)
> states <- map_data("state")
> arrests <- USArrests
> names(arrests) <- tolower(names(arrests))
> arrests$region <- tolower(rownames(USArrests))
>
> choro <- merge(states, arrests, by = "region")
> choro <- choro[order(choro$order), ]
> qplot(long, lat, data = choro, group = group,
+   fill = assault, geom = "polygon")
```

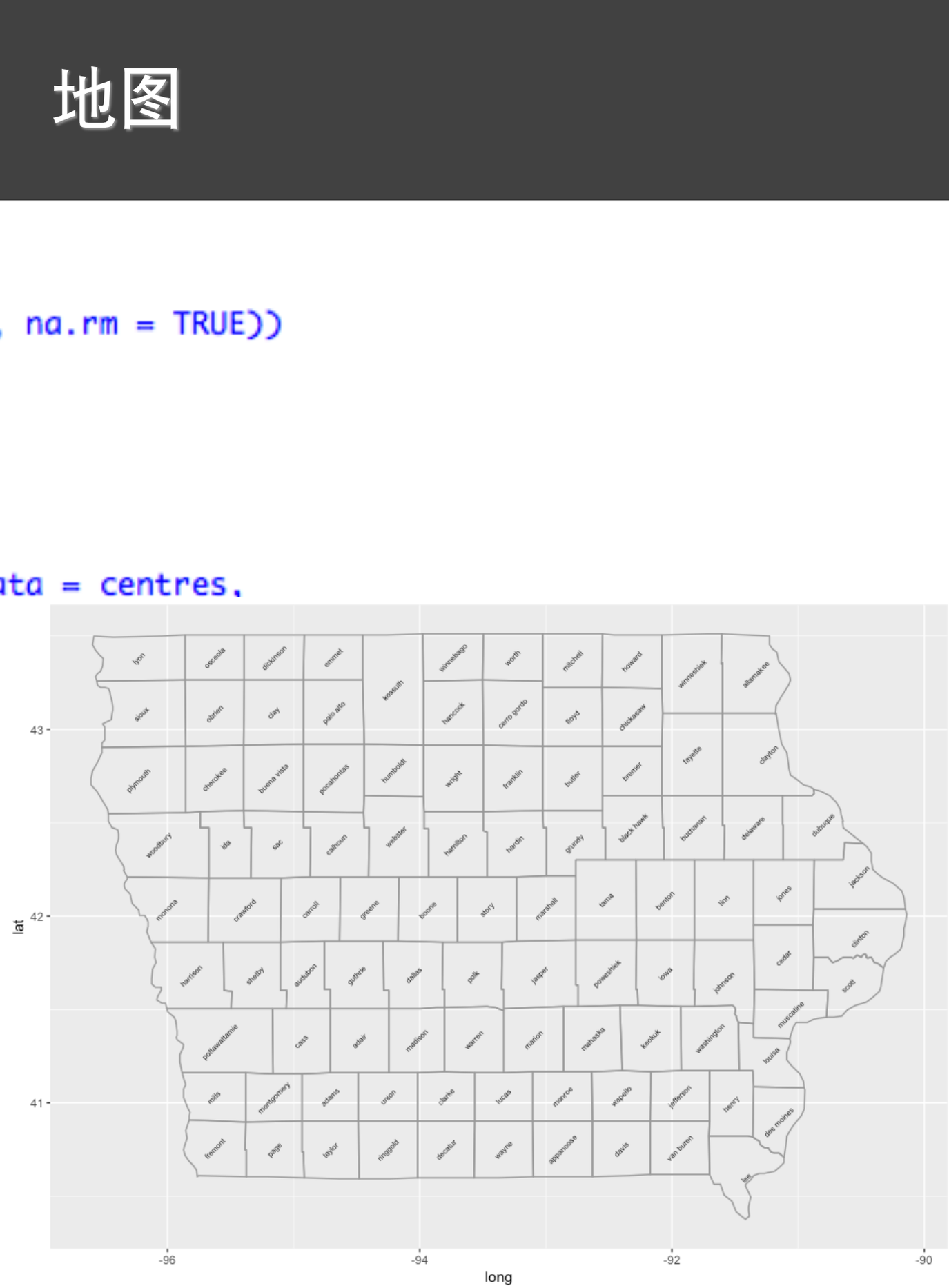
地图

```
> qplot(long, lat, data = choro, group = group,
+   fill = assault / murder, geom = "polygon")
```



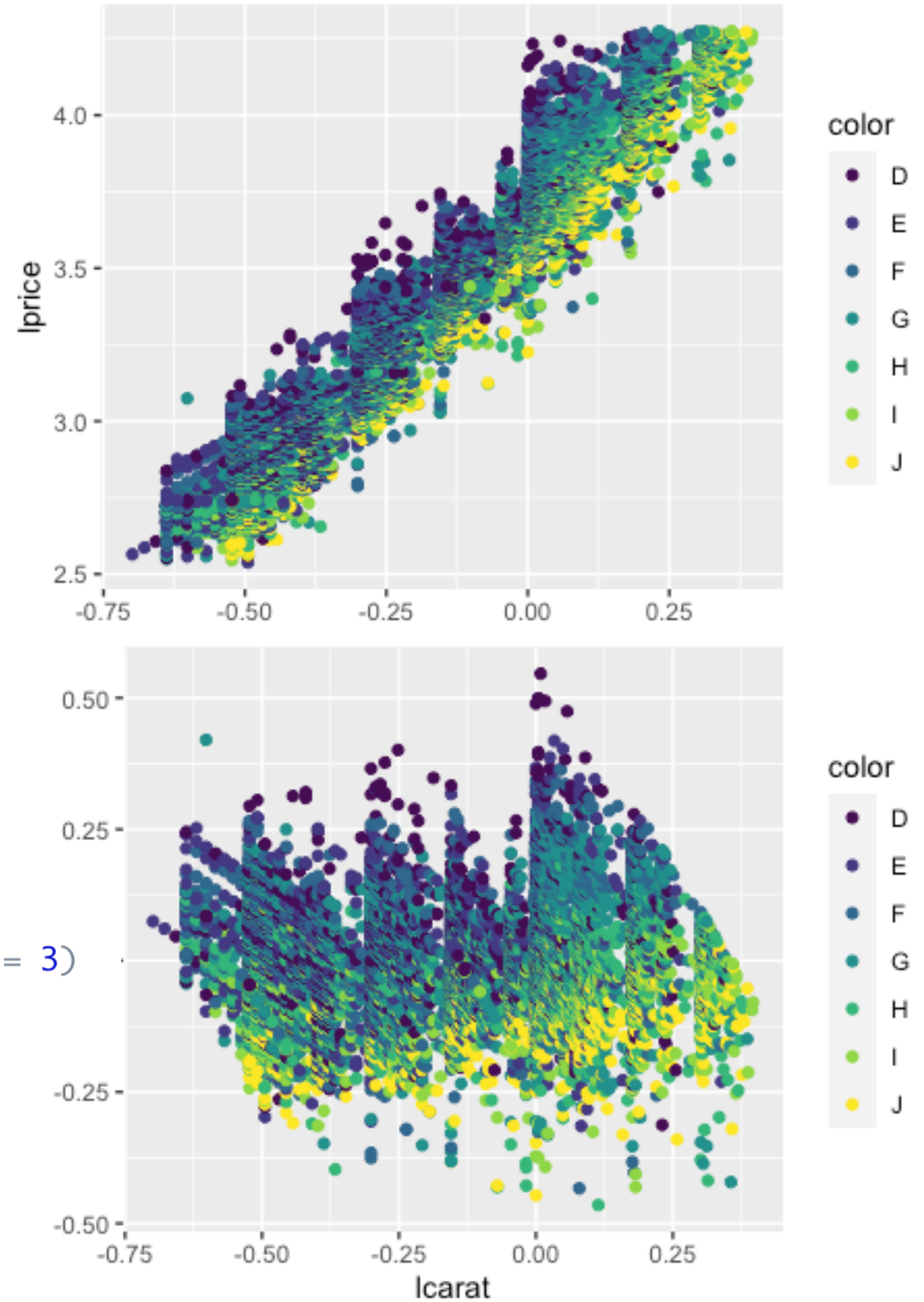
地图

```
> library(plyr)
> ia <- map_data("county", "iowa")
> mid_range <- function(x) mean(range(x, na.rm = TRUE))
> centres <- ddply(ia, .(subregion),
+   colwise(mid_range, .(lat, long)))
> ggplot(ia, aes(long, lat)) +
+   geom_polygon(aes(group = group),
+     fill = NA, colour = "grey60") +
+   geom_text(aes(label = subregion), data = centres,
+     size = 2, angle = 45)
```



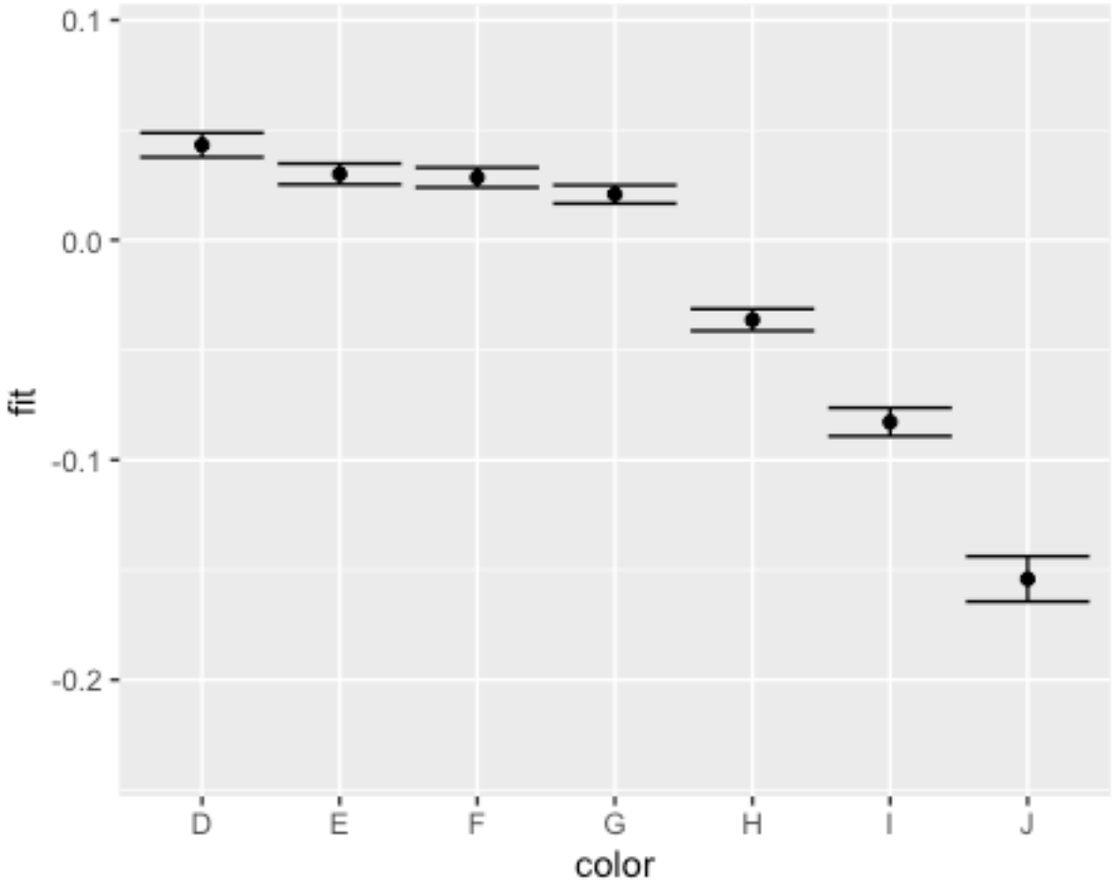
不确定性

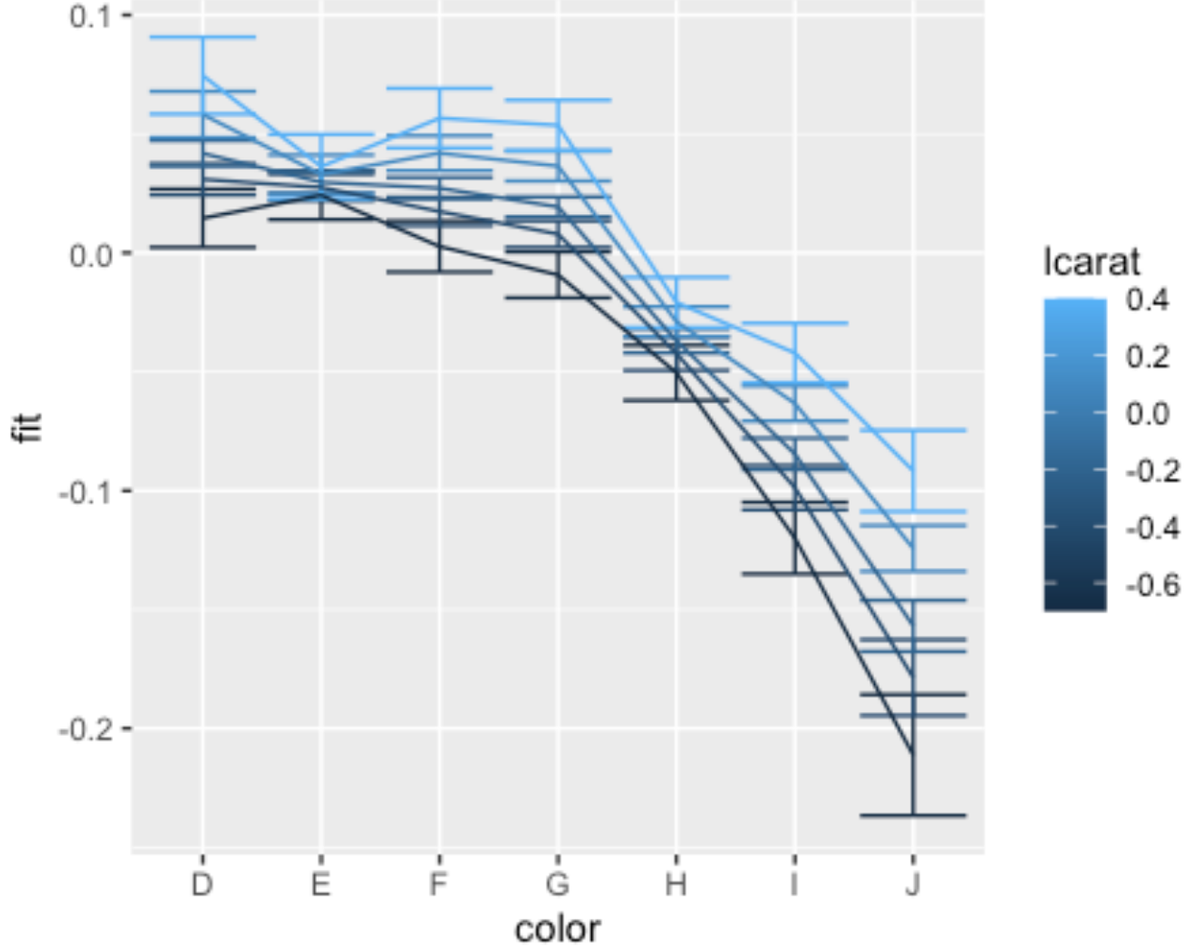
```
d <- subset(diamonds, carat < 2.5 &
              rbinom(nrow(diamonds), 1, 0.2) == 1)
d$lcarat <- log10(d$carat)
d$lprice <- log10(d$price)

detrend <- lm(lprice ~ lcarat, data = d)
d$lprice2 <- resid(detrend)

mod <- lm(lprice2 ~ lcarat * color, data = d)

install.packages("effects")
library(effects)

effectdf <- function(...) {
  suppressWarnings(as.data.frame(effect(...)))
}
color <- effectdf("color", mod)
both1 <- effectdf("lcarat:color", mod)

carat <- effectdf("lcarat", mod, default.levels = 50)
both2 <- effectdf("lcarat:color", mod, default.levels = 3)

qplot(lcarat, lprice, data=d, colour = color)
qplot(lcarat, lprice2, data=d, colour = color)
```



不確定性

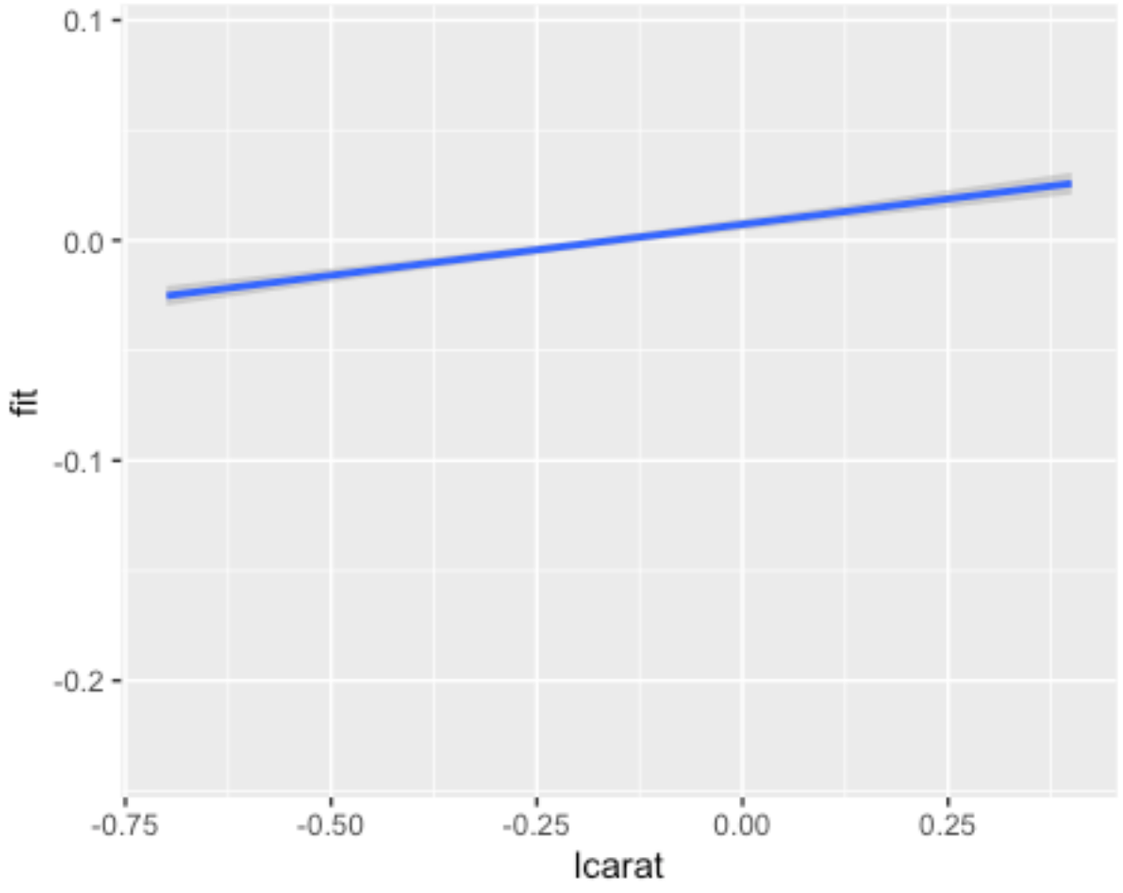
```
fplot <- ggplot(mapping = aes(y = fit, ymin = lower, ymax = upper)) +
  ylim(range(both2$lower, both2$upper))
fplot %+% color + aes(x = color) + geom_point() + geom_errorbar()
fplot %+% both2 +
  aes(x = color, colour = lcarat, group = interaction(color, lcarat)) +
  geom_errorbar() + geom_line(aes(group=lcarat)) +
  scale_colour_gradient()
```

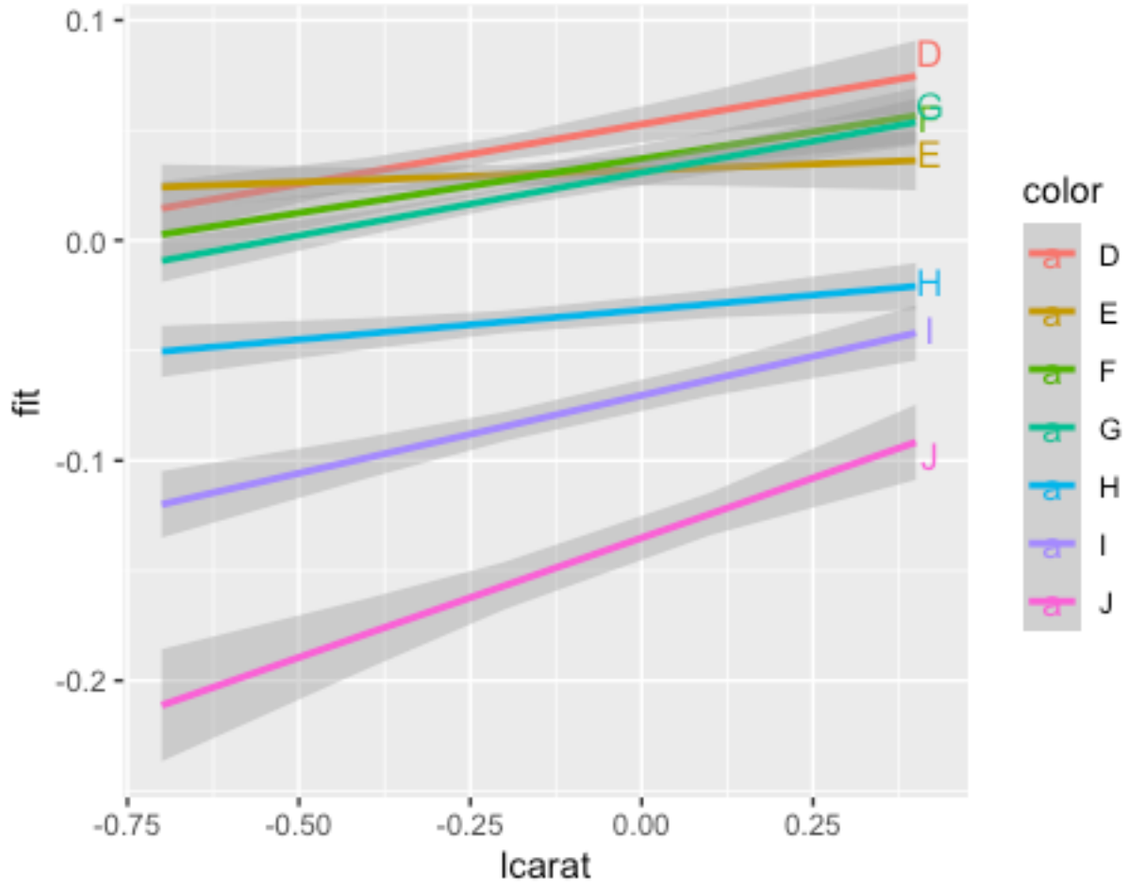




不确定性

```
fplot %+% carat + aes(x = lcarat) + geom_smooth(stat="identity")

ends <- subset(both1, lcarat == max(lcarat))
fplot %+% both1 + aes(x = lcarat, colour = color) +
  geom_smooth(stat="identity") +
  scale_colour_hue() + labs(legend.position = "none") +
  geom_text(aes(label = color, x = lcarat + 0.02), ends)
```

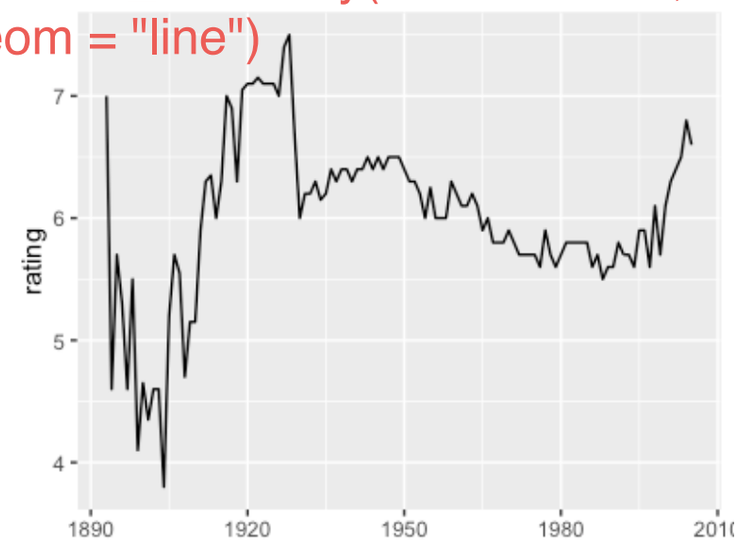




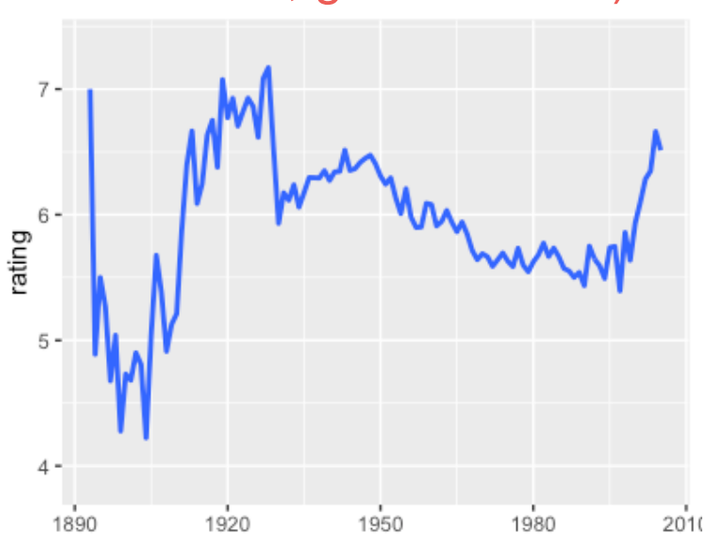
统计摘要

m <- ggplot(movies, aes(year, rating))

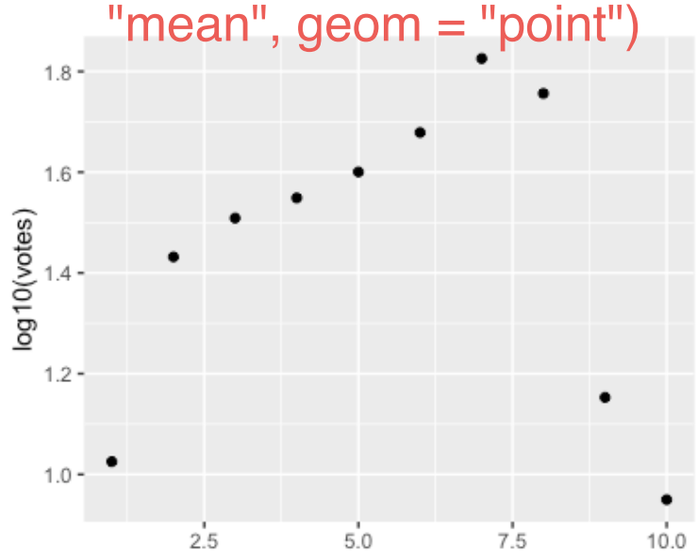
m + stat_summary(fun = median, geom = "line")



m + stat_summary(fun = "mean", geom = "line")

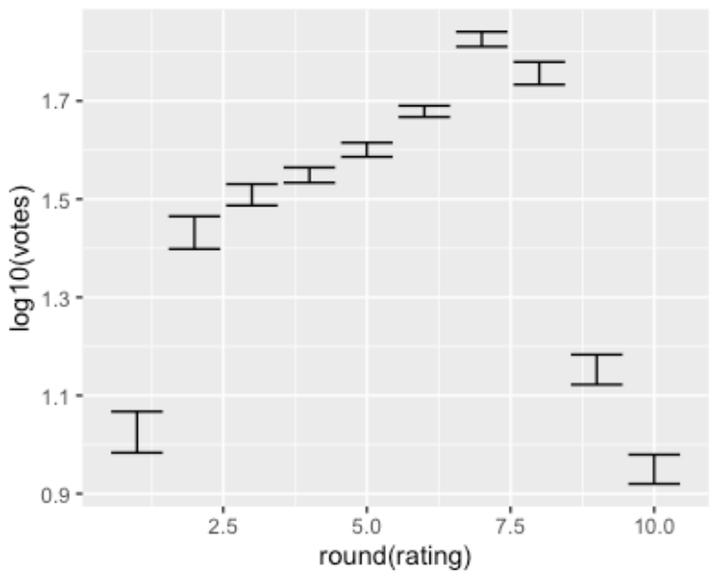


m2 + stat_summary(fun = "mean", geom = "point")

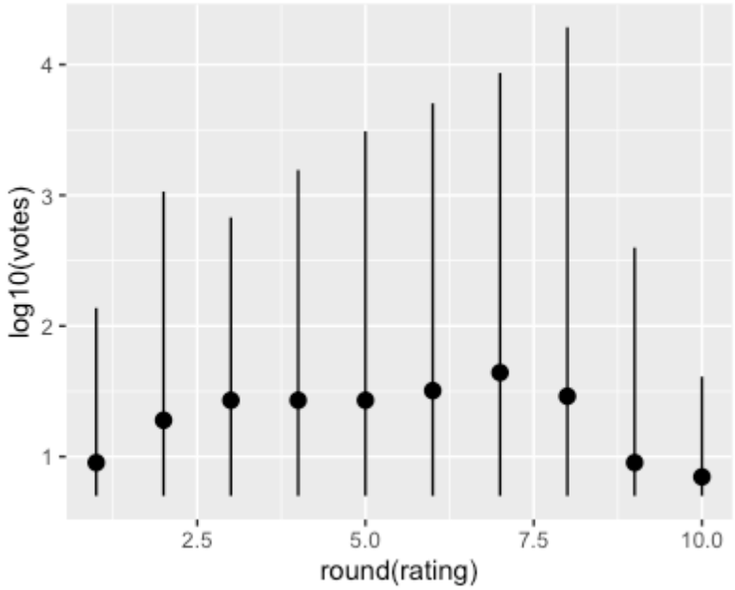


m2 <- ggplot(movies, aes(round(rating), log10(votes)))

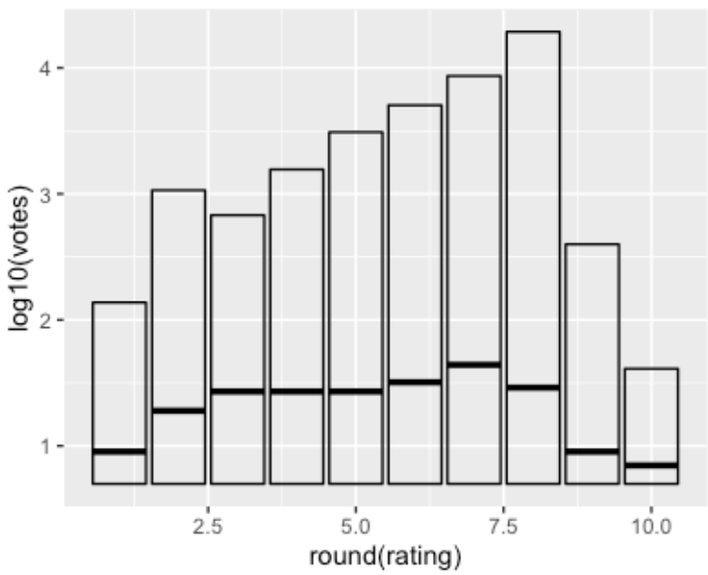
m2 + stat_summary(fun.data = "mean_cl_normal", geom = "errorbar")



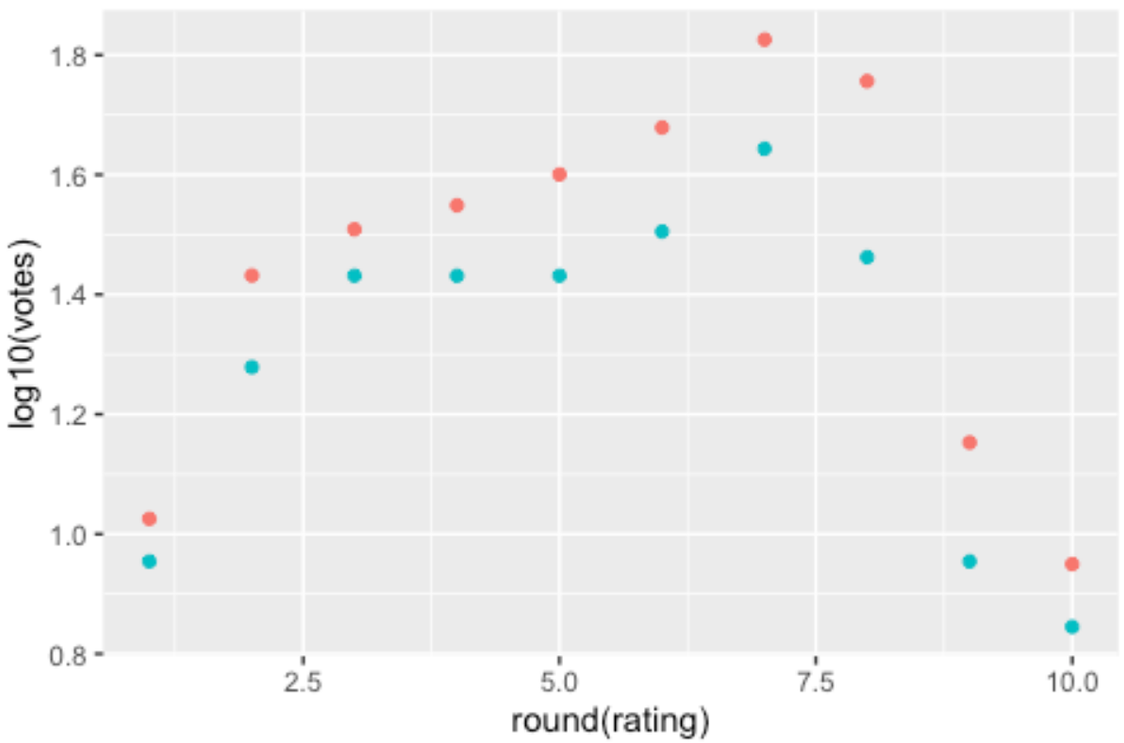
m2 + stat_summary(fun.data = "median_hilow", geom = "pointrange")



m2 + stat_summary(fun.data = "median_hilow", geom = "crossbar"

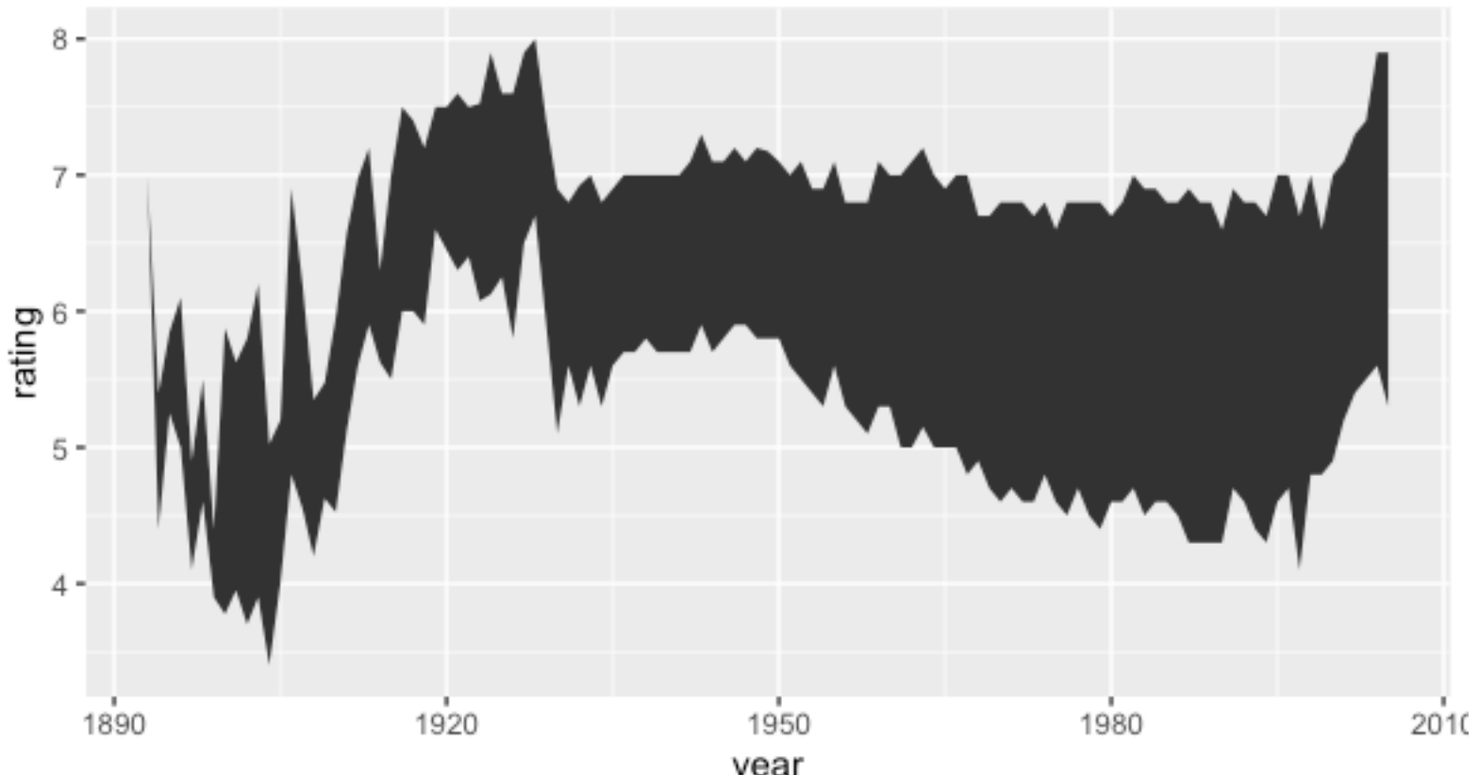


摘要计算函数: 单独 vs. 统一

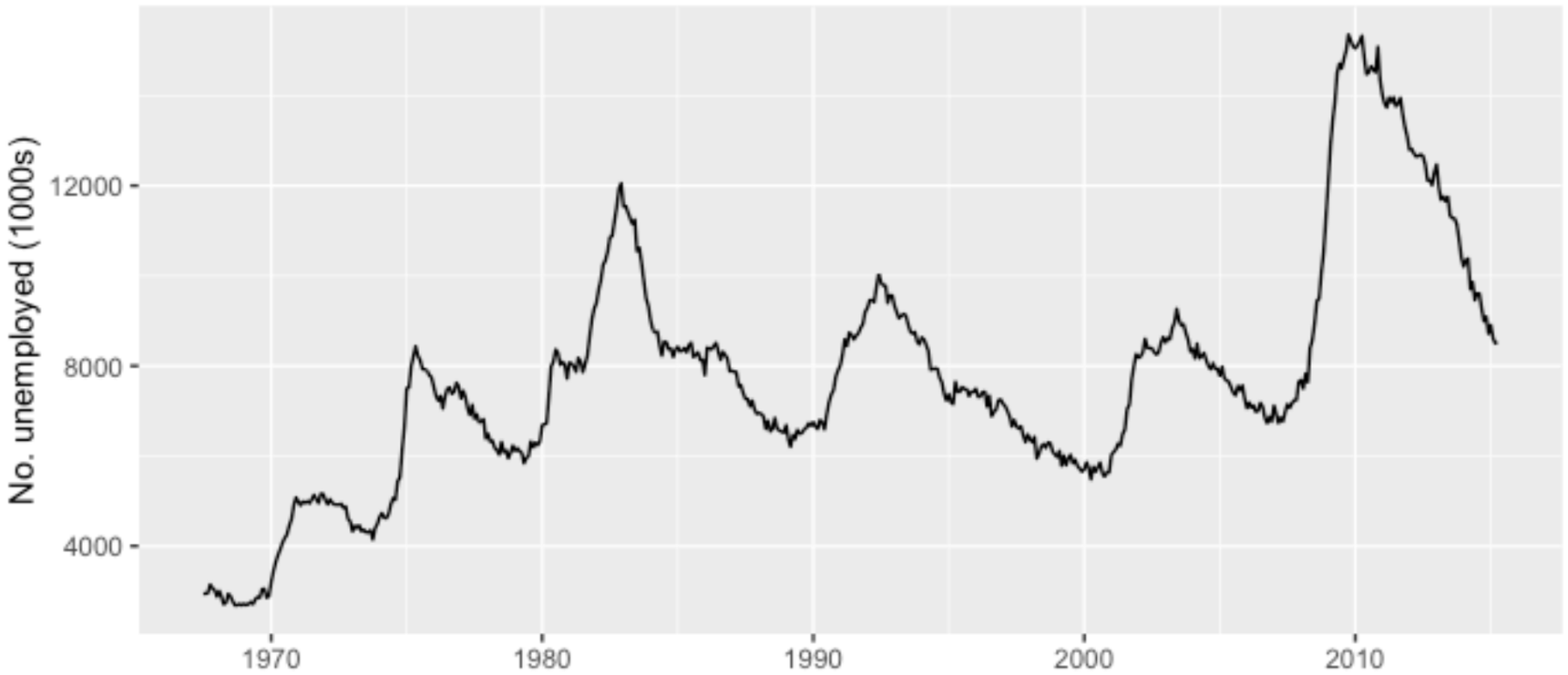



```
midm <- function(x) mean(x, trim = 0.5)
m2 +
  stat_summary(aes(colour = "trimmed"), fun.y =
midm, geom = "point") +
  stat_summary(aes(colour = "raw"), fun.y =
mean, geom = "point") +
  scale_colour_hue("Mean")
```

```
iqr <- function(x, ...) {
  qs <- quantile(as.numeric(x), c(0.25,
0.75), na.rm = T)
  names(qs) <- c("ymin", "ymax")
  qs
}
m + stat_summary(fun.data = "iqr",
geom="ribbon")
```

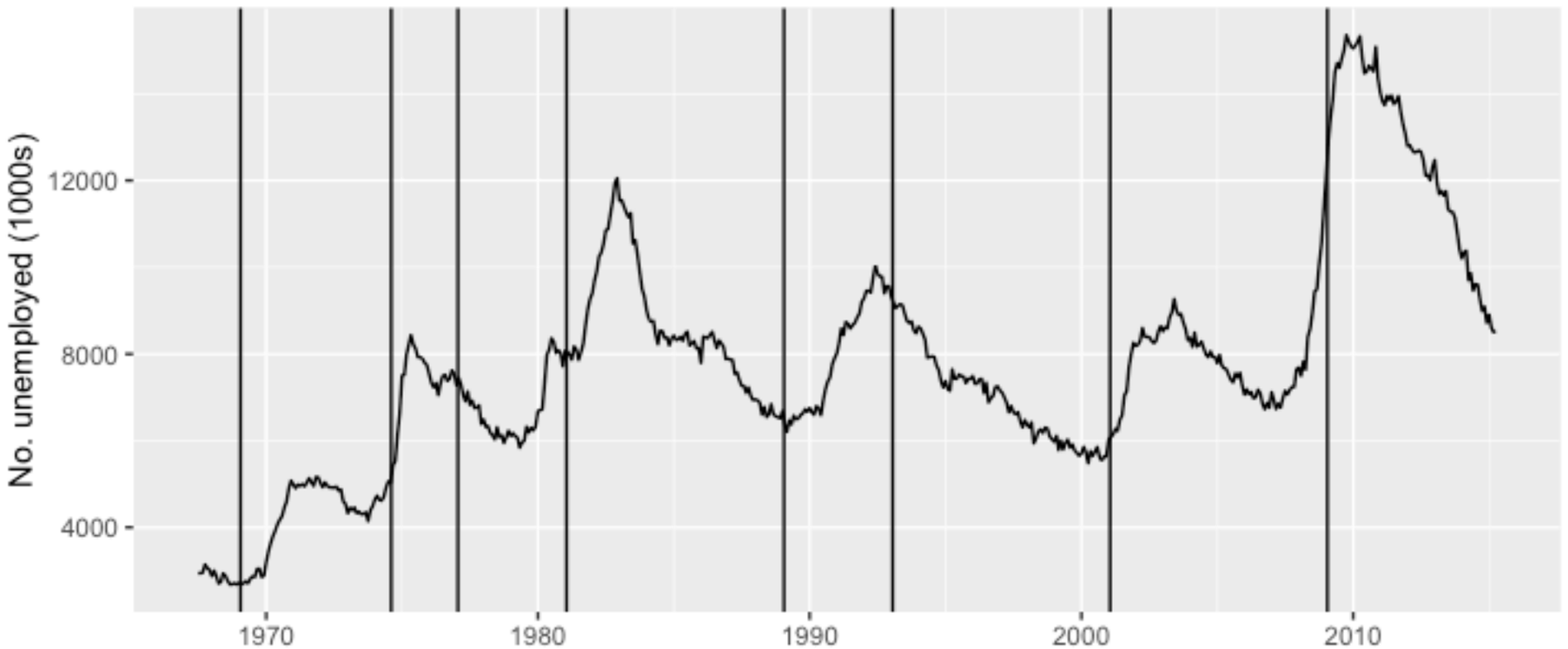


添加图形注解



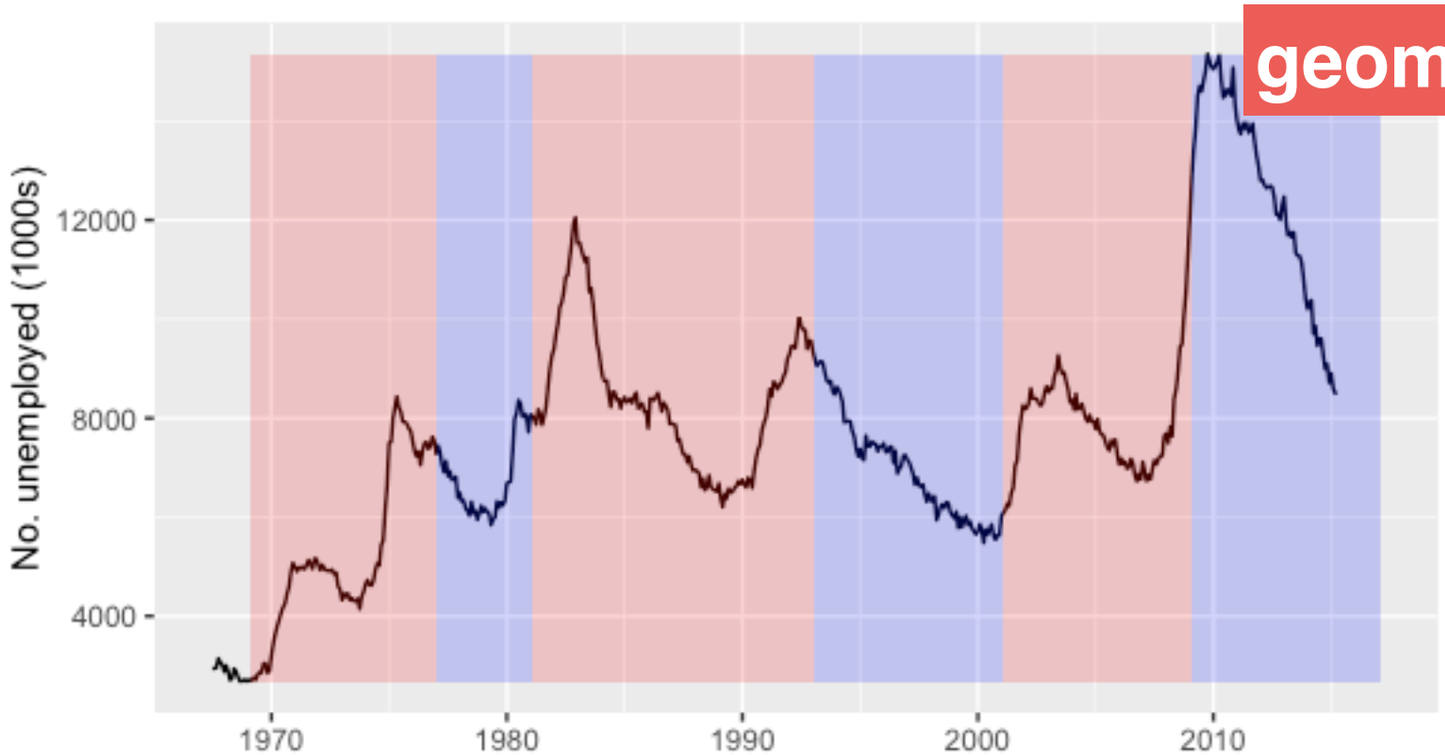

```
(unemp <- qplot(date,
unemploy,
data=economics,
geom="line",
 xlab = "",
ylab = "No. unemployed
(1000s)"))
```

geom_vline



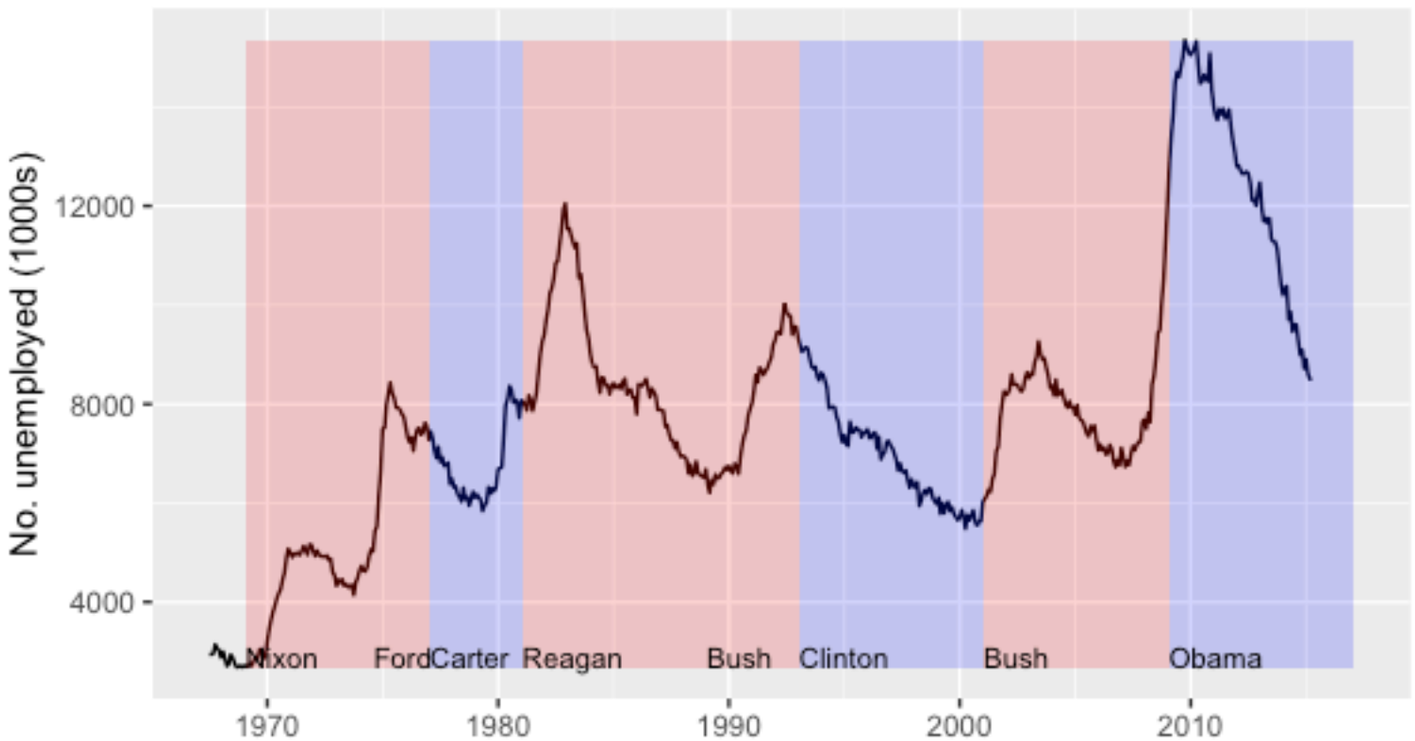

```
presidential <- presidential[-
(1:3), ]

yrng <-
range(economics$unemploy)
xrng <-
range(economics$date)
unemp +
geom_vline(aes(xintercept =
start), data = presidential)
```

添加图形注解



geom_rect

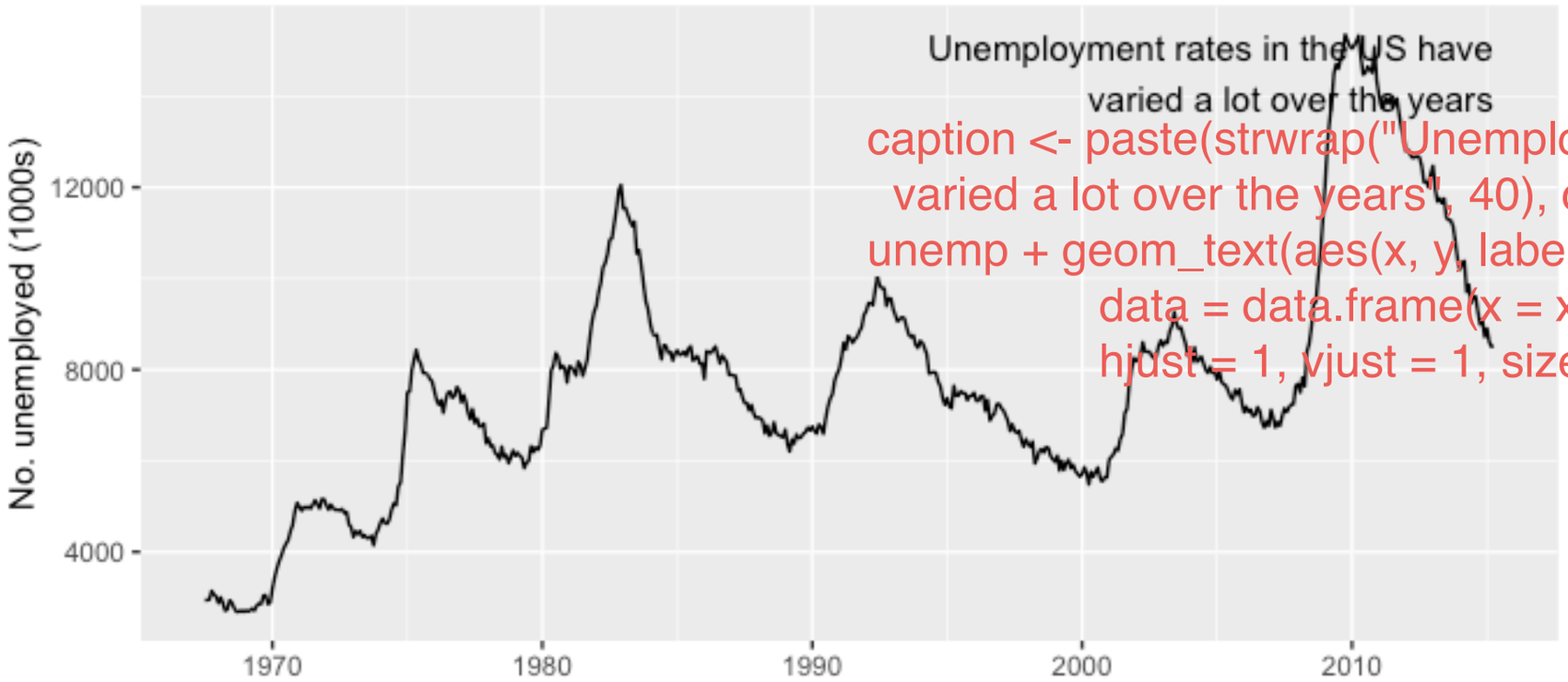
```
unemp + geom_rect(aes(NULL, NULL, xmin = start, xmax = end, fill = party),
ymin = yrng[1], ymax = yrng[2], data = presidential) +
scale_fill_manual(values = alpha(c("blue", "red"), 0.2))
```



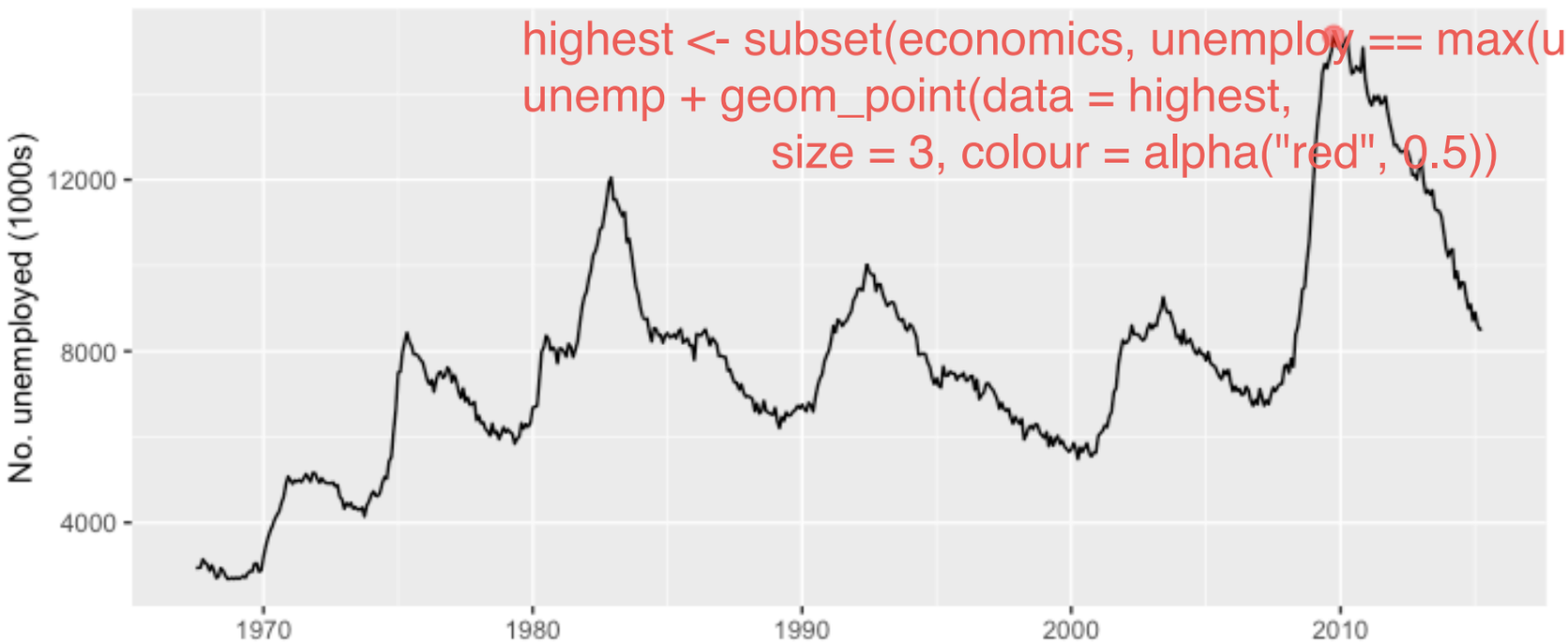

```
last_plot() +
geom_text(aes(x = start, y = yrng[1], label = name),
data = presidential, size = 3,
hjust = 0, vjust = 0)
```

geom_text

添加图形注解




```
caption <- paste(strwrap("Unemployment rates in the US have
  varied a lot over the years", 40), collapse="\n")
unemp + geom_text(aes(x, y, label = caption),
                  data = data.frame(x = xrng[2], y = yrng[2]),
                  hjust = 1, vjust = 1, size = 4)
```

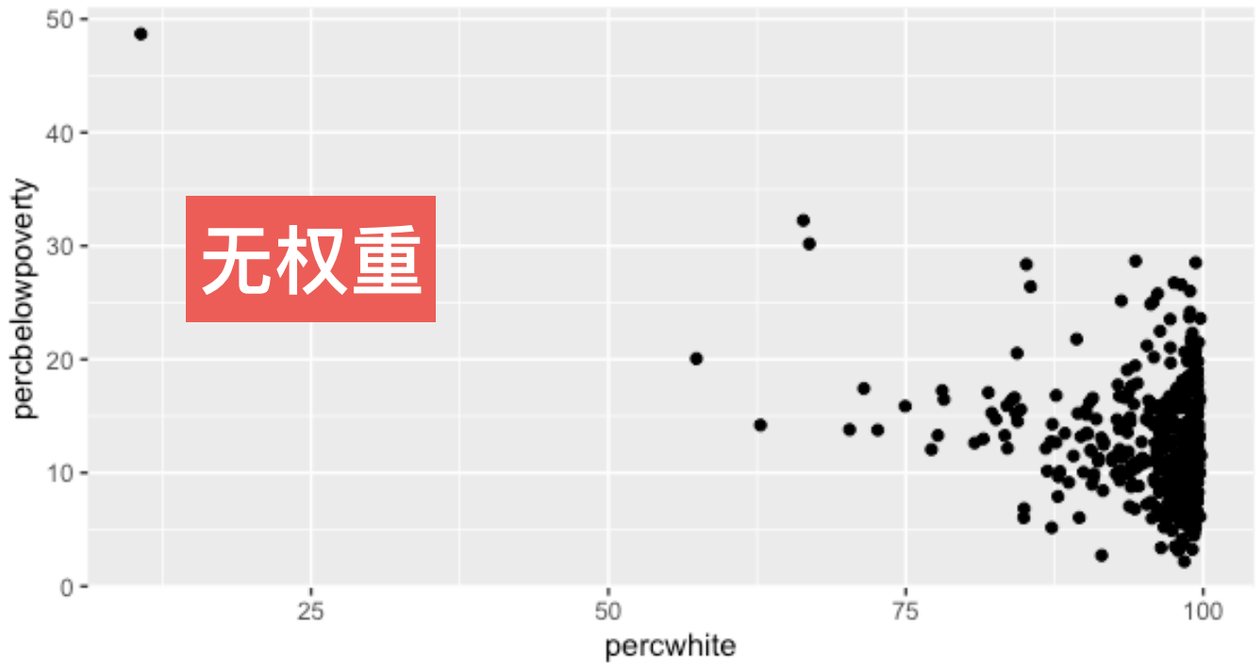


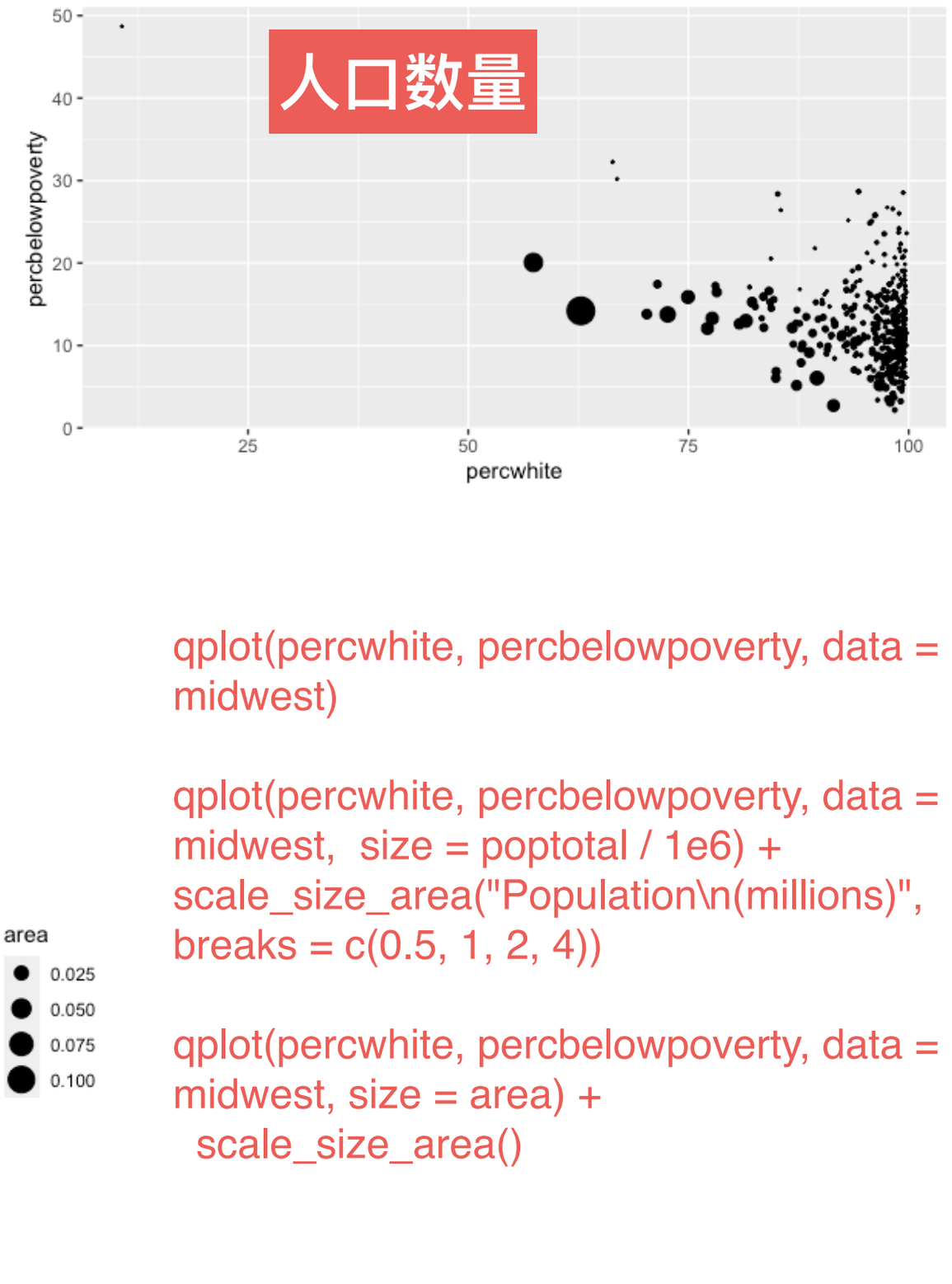

```
highest <- subset(economics, unemploy == max(unemploy))
unemp + geom_point(data = highest,
                   size = 3, colour = alpha("red", 0.5))
```

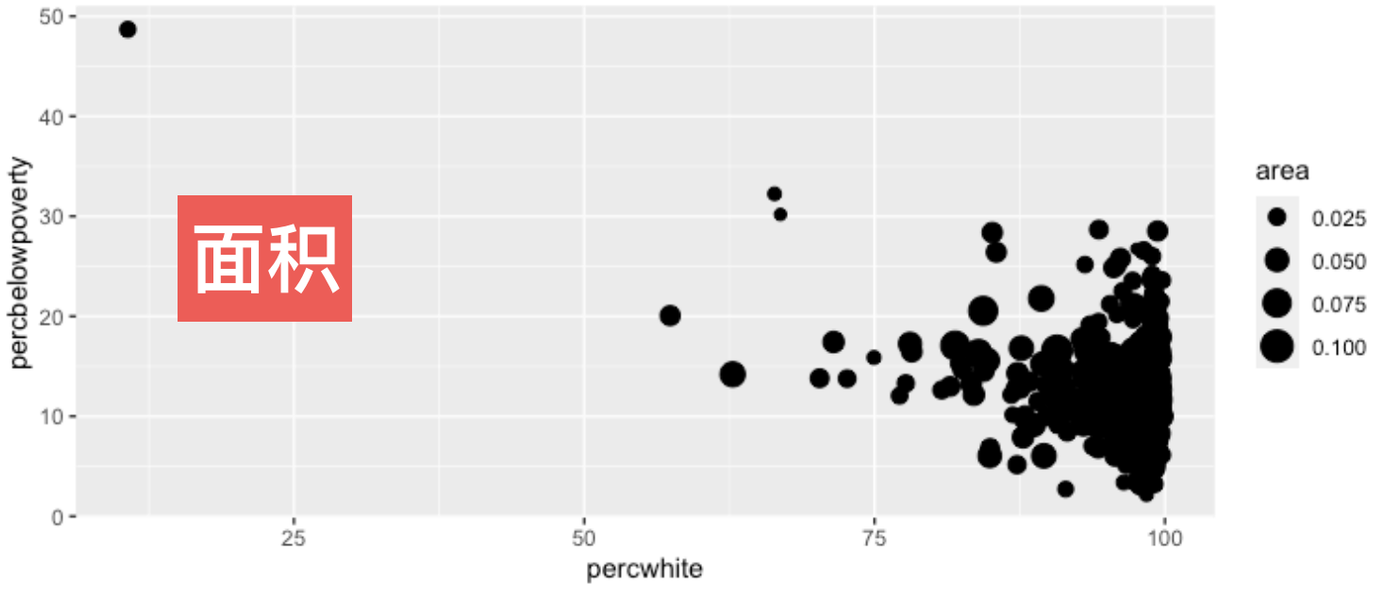
注解

- geom_text()
- geom_vline()、geom_hline()
- geom_abline()
- geom_rect()
- geom_line()、geom_path()、geom_segment()
- arrow()
- … …

含权数据

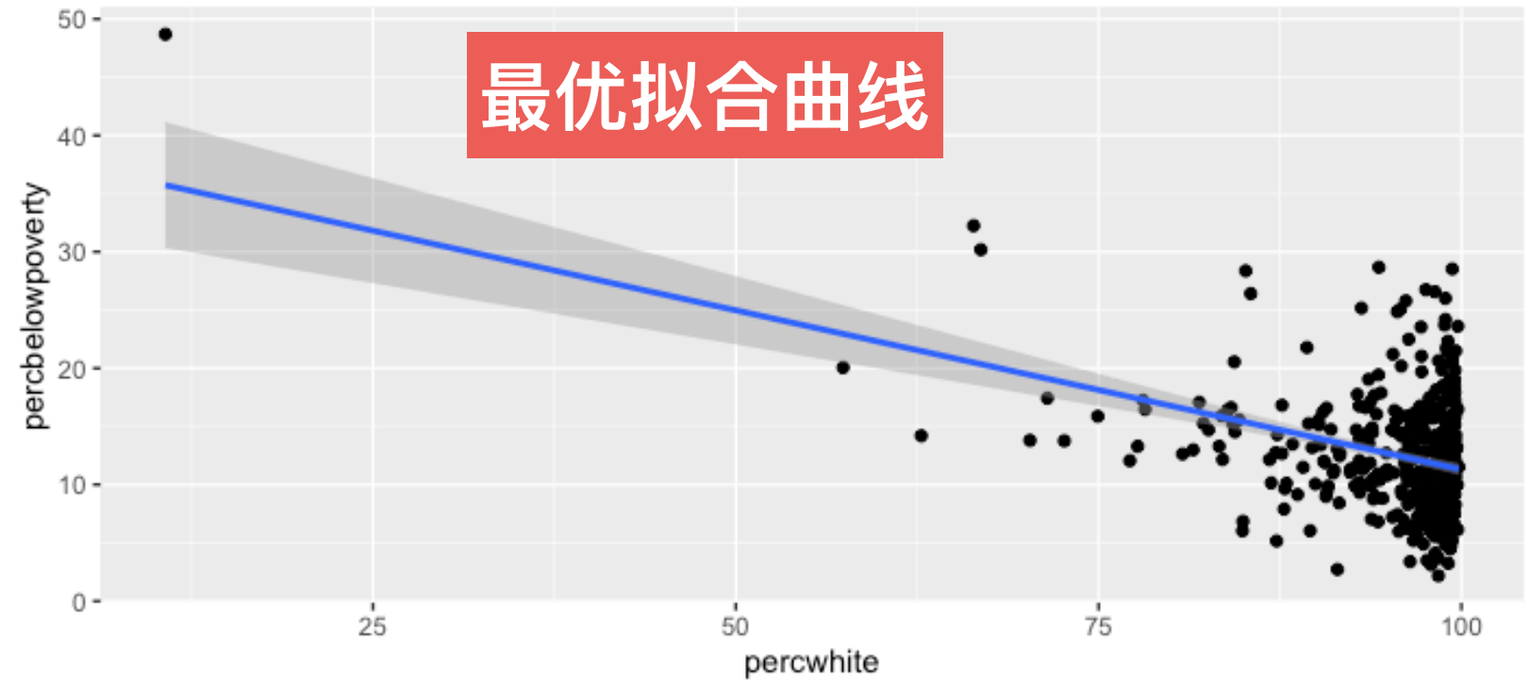






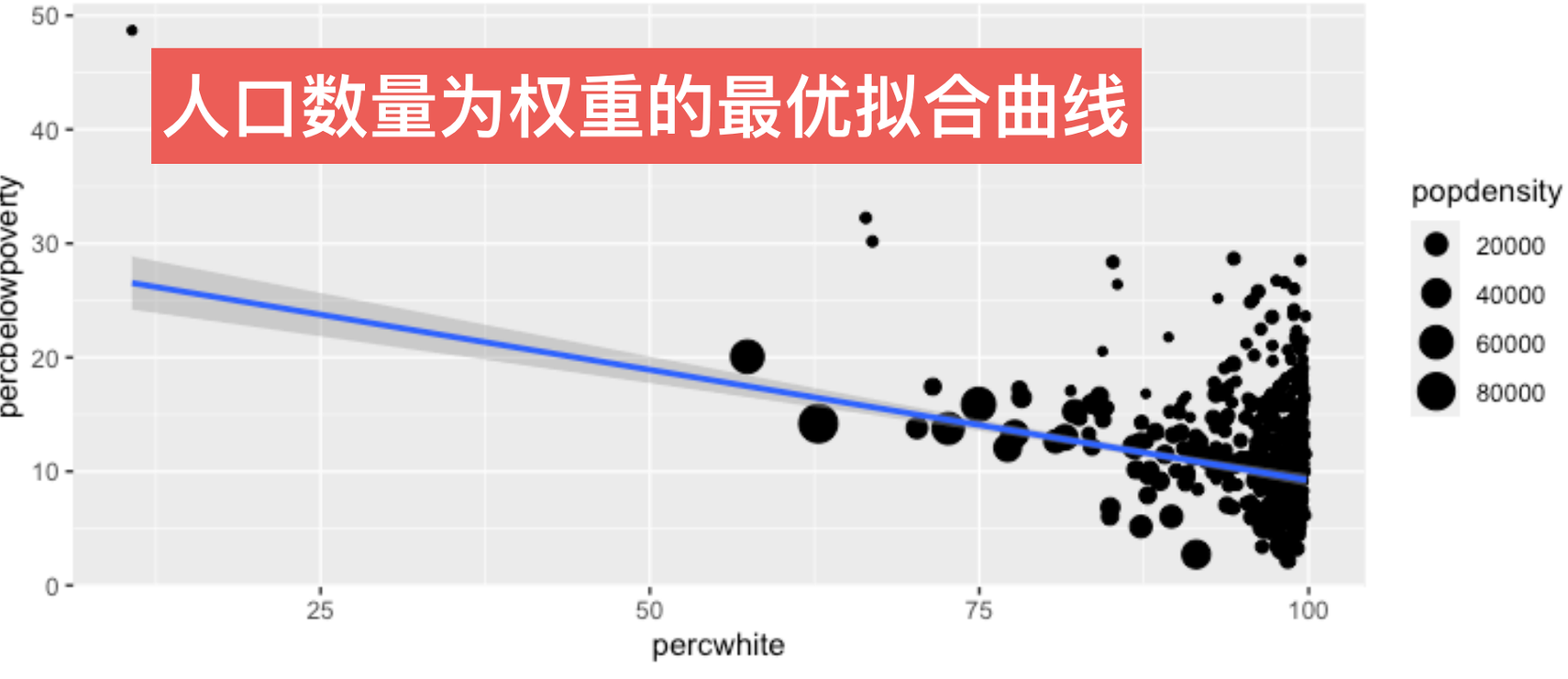

```
qplot(percwhite, percbelowpoverty, data = midwest)

qplot(percwhite, percbelowpoverty, data = midwest,  size = poptotal / 1e6) + scale_size_area("Population\n(millions)", breaks = c(0.5, 1, 2, 4))

qplot(percwhite, percbelowpoverty, data = midwest, size = area) +
  scale_size_area()
```

含权数据



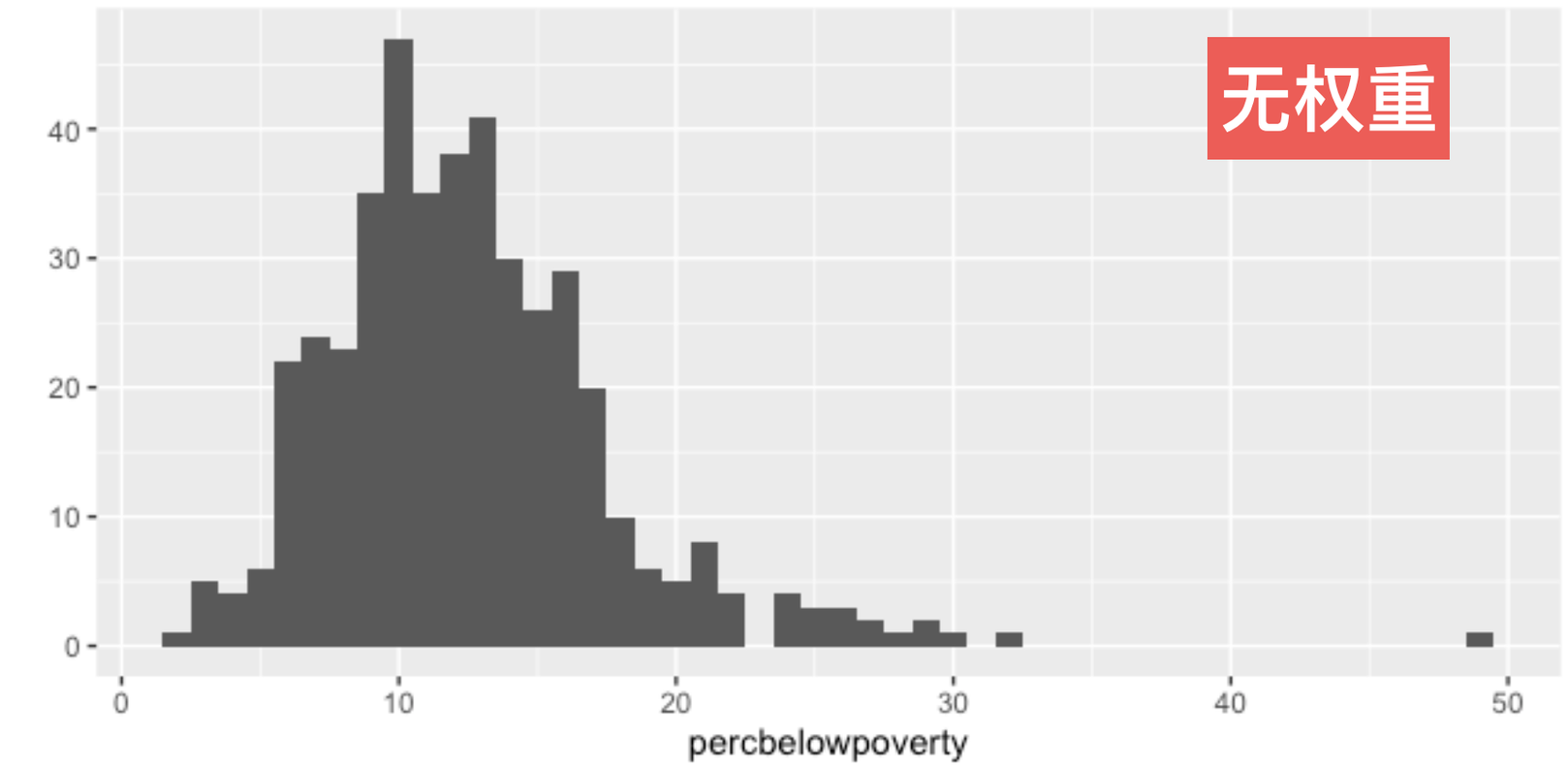

```
lm_smooth <-
geom_smooth(method = lm, size =
1)
```

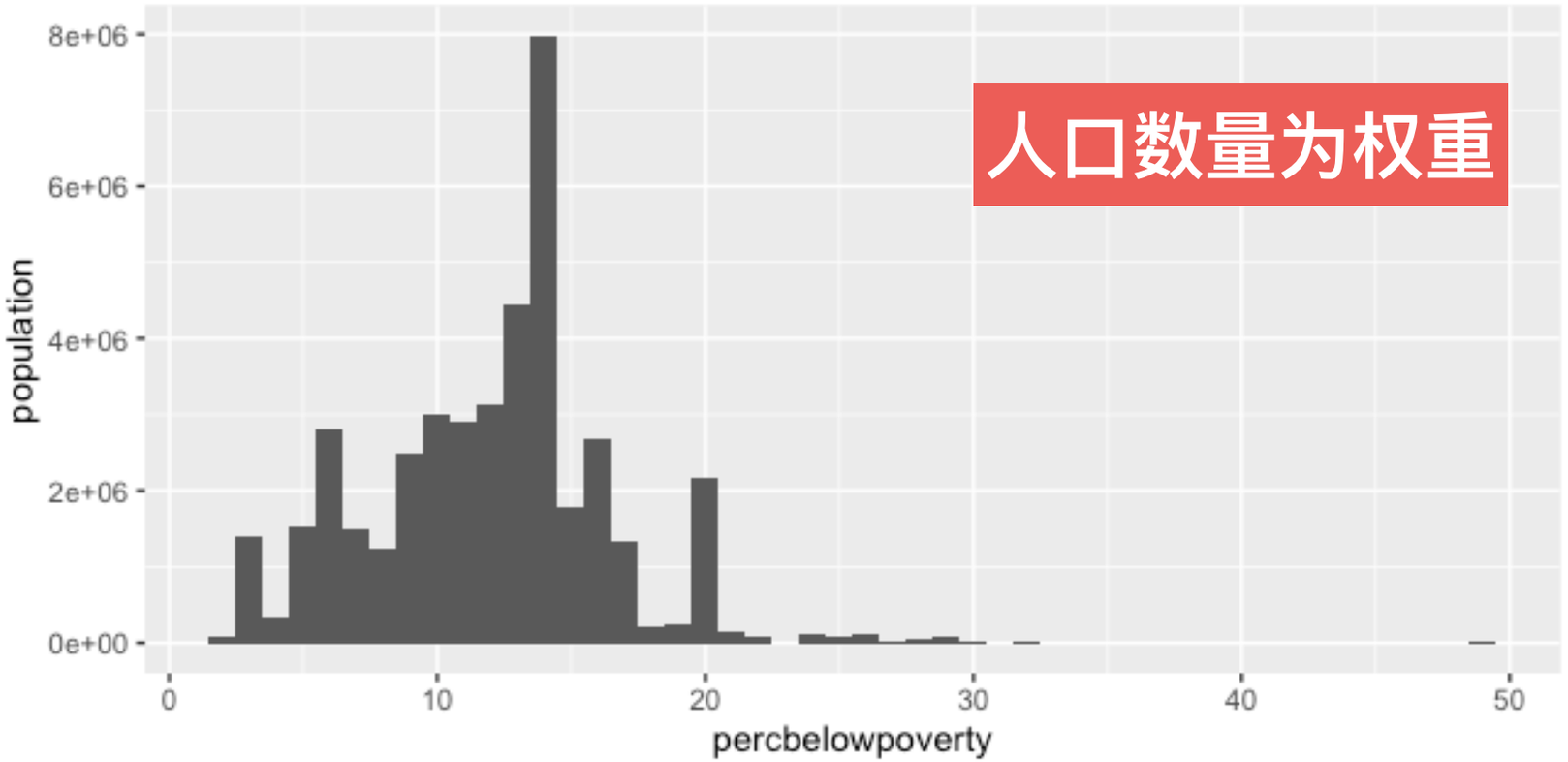
```
qplot(percwhite, percbelowpoverty,
data = midwest) + lm_smooth
```




```
qplot(percwhite, percbelowpoverty,
data = midwest,
      weight = popdensity, size =
popdensity) + lm_smooth
```

含权数据






```
qplot(percbelowpoverty, data = midwest, binwidth = 1)
```

```
qplot(percbelowpoverty, data = midwest, weight = poptotal, binwidth = 1) + ylab("population")
```

标度、坐标系和图例

CH6

定位

CH7

提问时间！

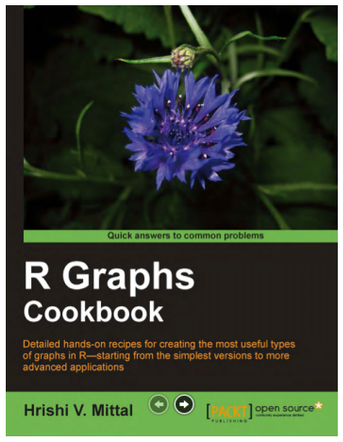
孙惠平

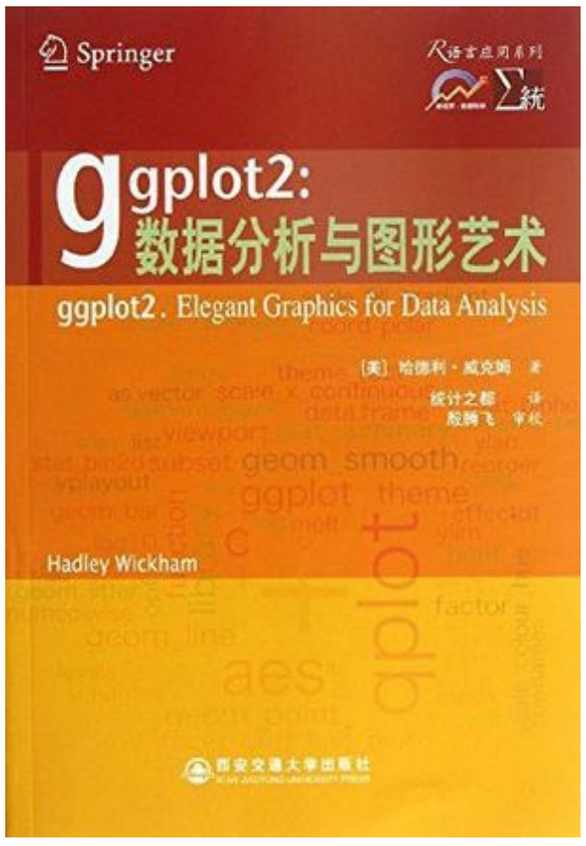
sunhp@ss.pku.edu.cn

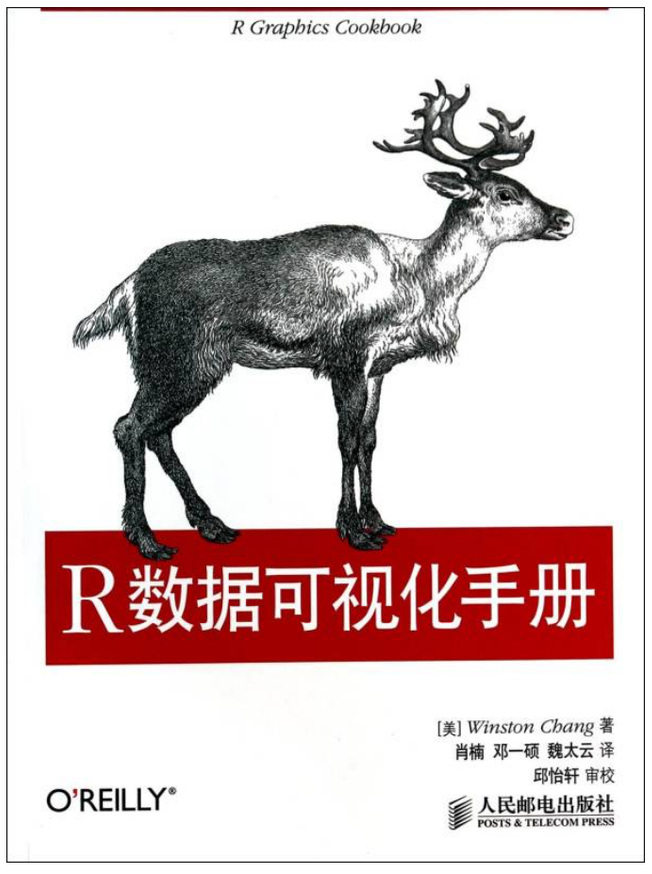
练习

练习 - 0028

- ggplot2的4-7章，熟悉所有例子。
- R数据可视化手册的6-13章，熟悉所有例子。
- 教材RIA（第二版）的第19章，熟悉所有例子。
- 看R Graphs Cookbook所有章节



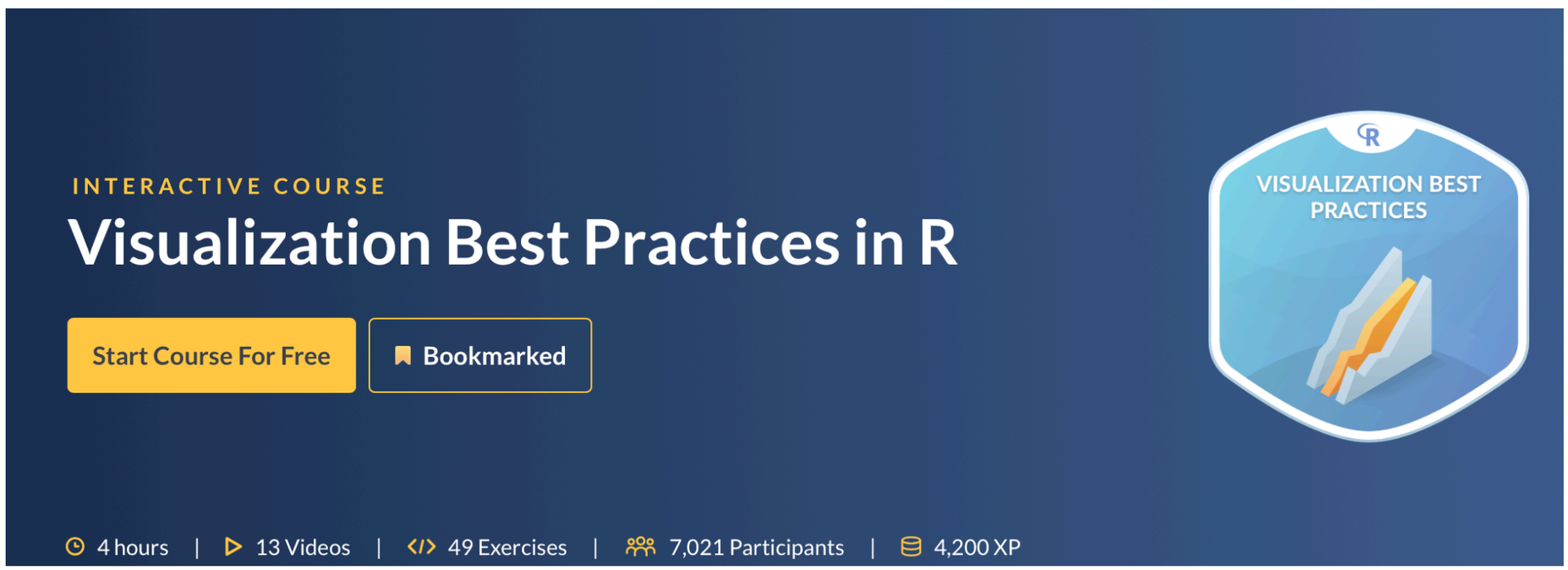




练习 – 0029

提交方式和上节课一样！

https://www.datacamp.com/courses



谢谢！

孙惠平

sunhp@ss.pku.edu.cn