

ggplott画图2



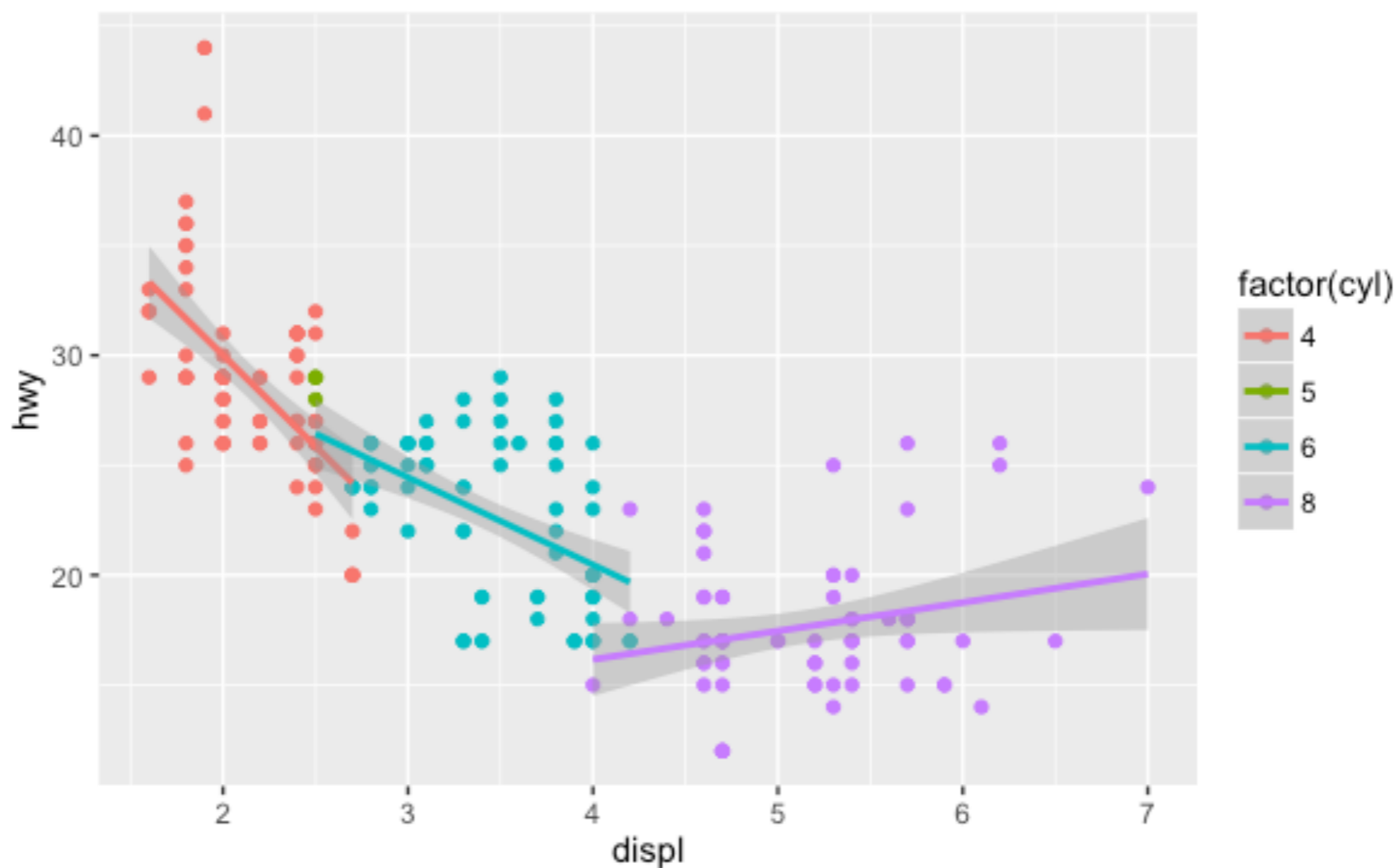
课堂测试时间

- 1、查看数据集mtcars，根据要求作图：
 - (1)分别使用qplot、ggplot函数画出mpg和wt关系的散点图；
 - (2)使用三种方式画出mpg列的直方图,同时在使用qplot和ggplot时指定每个小圆柱体的宽度是4；
 - (3)使用三种绘图函数画出mpg变量的密度曲线。
- 2、使用datasets包中的数据集pressure，查看其数据并按要求画图：
 - (1)请画出pressure和temperature关系的曲线图；
 - (2)分别使用qplot和ggplot画出pressure和temperature关系的散点图和折线图。
- 3、使用datasets中的数据集中ToothGrowth，完成如下的绘图要求：
 - (1)以supp变量作为分类,分别使用三种绘图函数画出len变量的箱型图。
- 4、使用ggplot2包中数据集mpg，完成练习：
 - (1)使用mpg数据集定义一个ggplot对象，表示hwy与cty的关系；
 - (2)画一个散点图，指定颜色有year列来指定，并在上边绘图的基础上画出平滑的拟合曲线；
 - (3)继续使用(1)中定义的ggplot对象画散点图，使用class来指定颜色，displ指定大小，透明度；指定为0.5,position指定为抖动，在散点图的基础上添加拟合曲线；
 - (4)使用qplot画出hwy与cty的关系的散点图，并根据year变量分面，同时添加拟合曲线。

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

工具箱

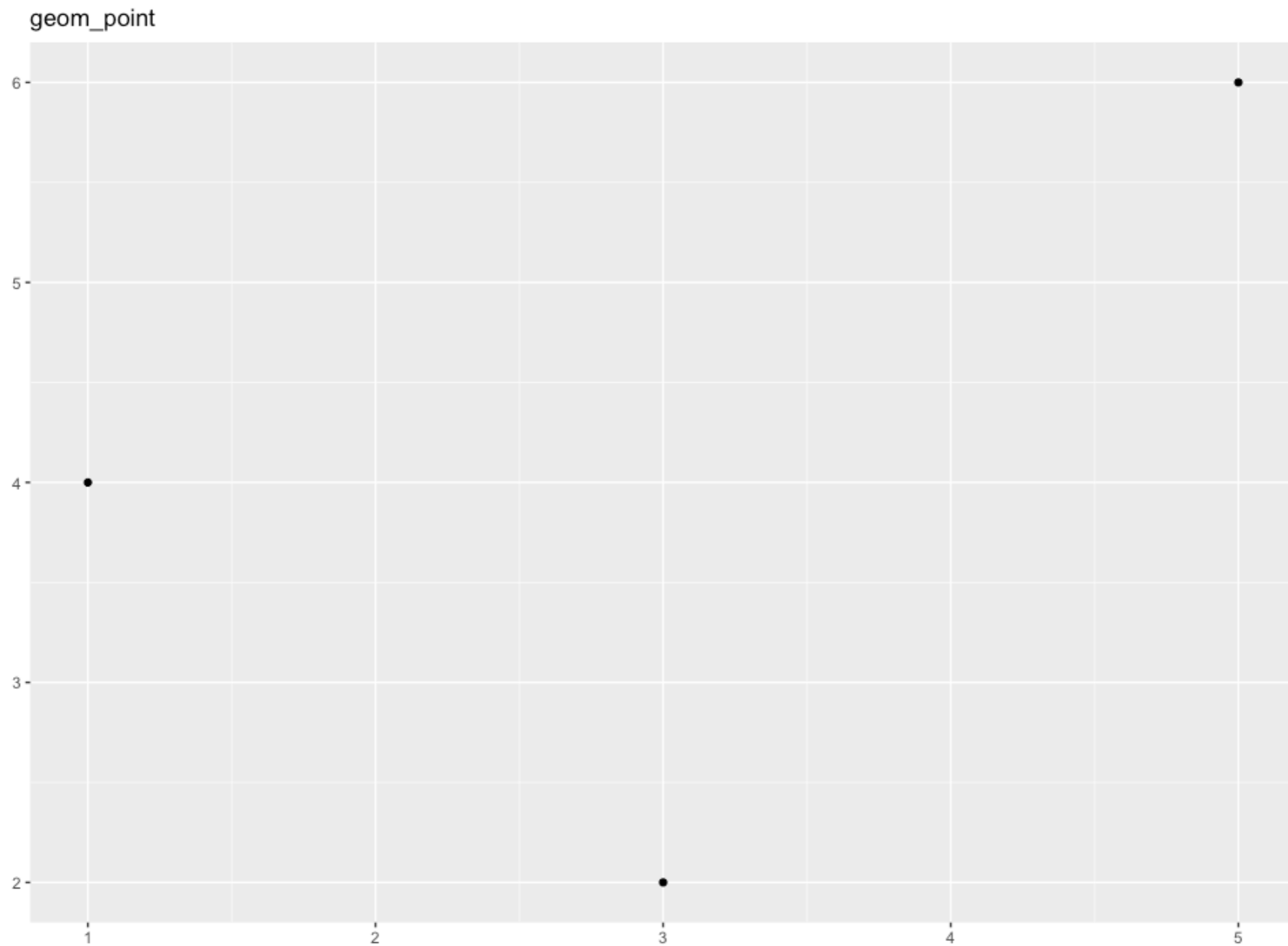
- 展示数据本身
- 展示数据的统计摘要
- 添加额外的元数据、上下文信息和注解



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

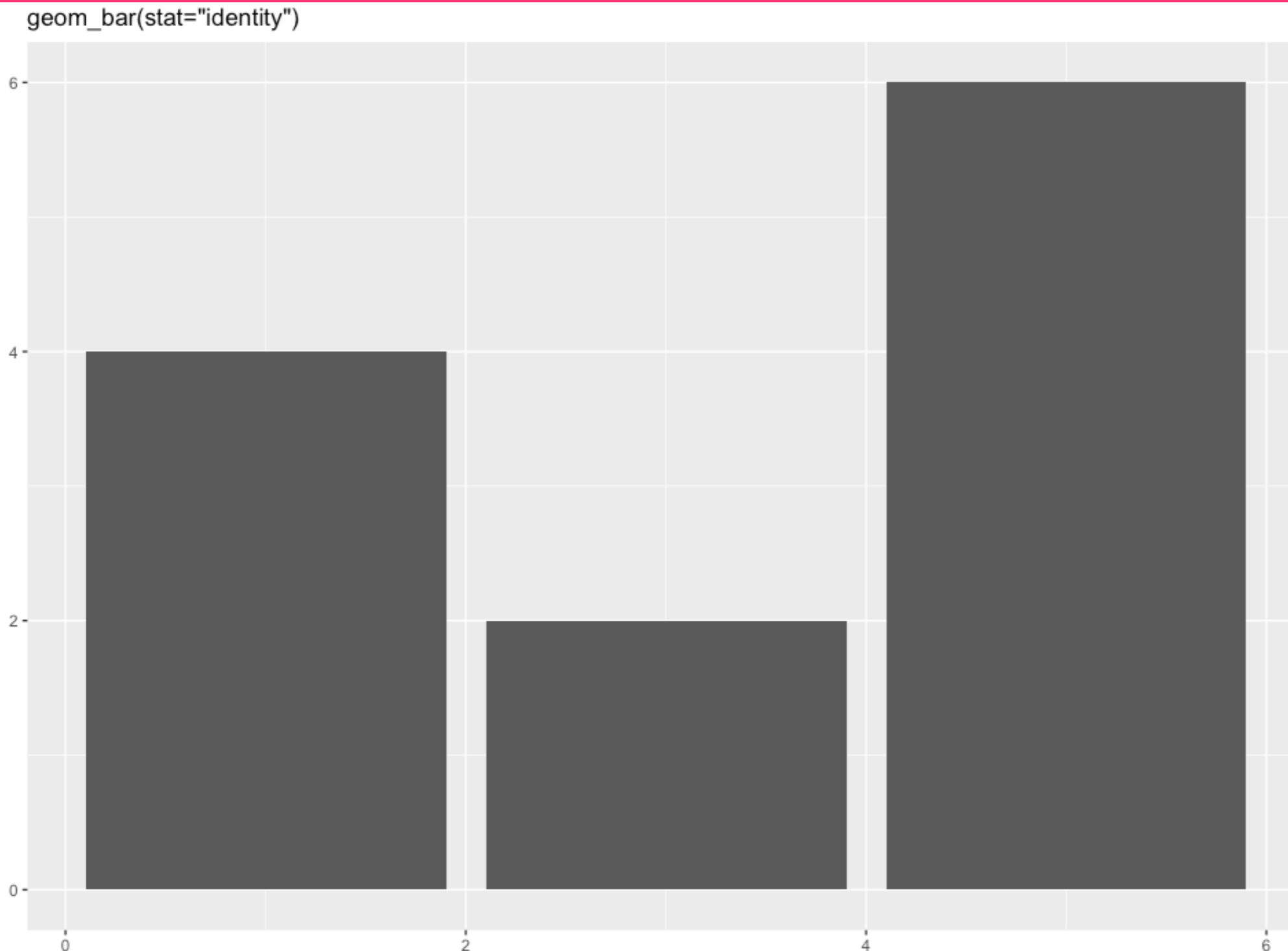
```
> df <- data.frame(  
+   x = c(3, 1, 5),  
+   y = c(2, 4, 6),  
+   label = c("a", "b", "c")  
+ )  
> p <- ggplot(df, aes(x, y, label = label)) +  
+   xlab(NULL) + ylab(NULL)
```

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



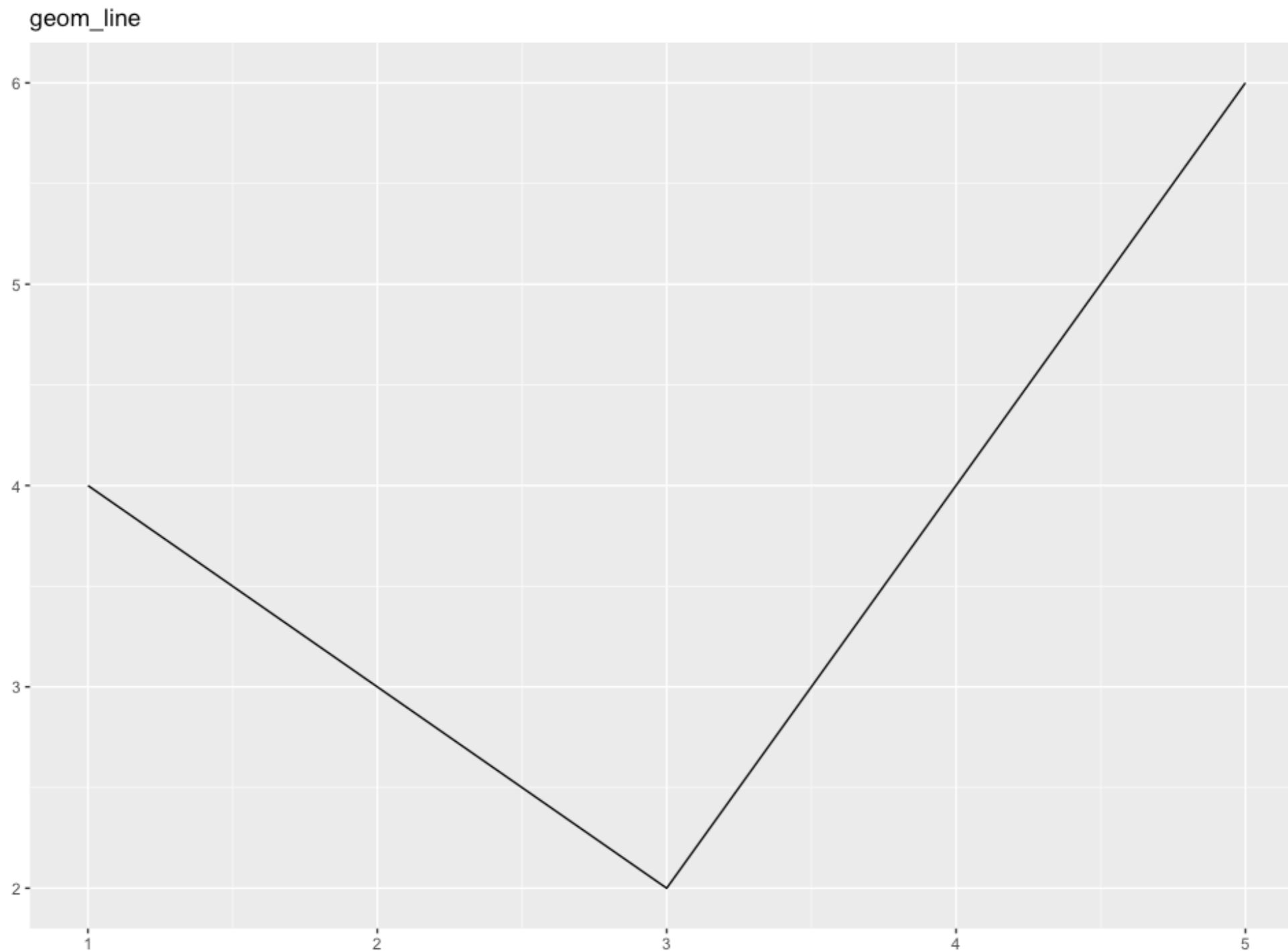

```
> p + geom_bar(stat="identity") +  
+ labs(title = "geom_bar(stat=\\"identity\\)")
```

stat = identity

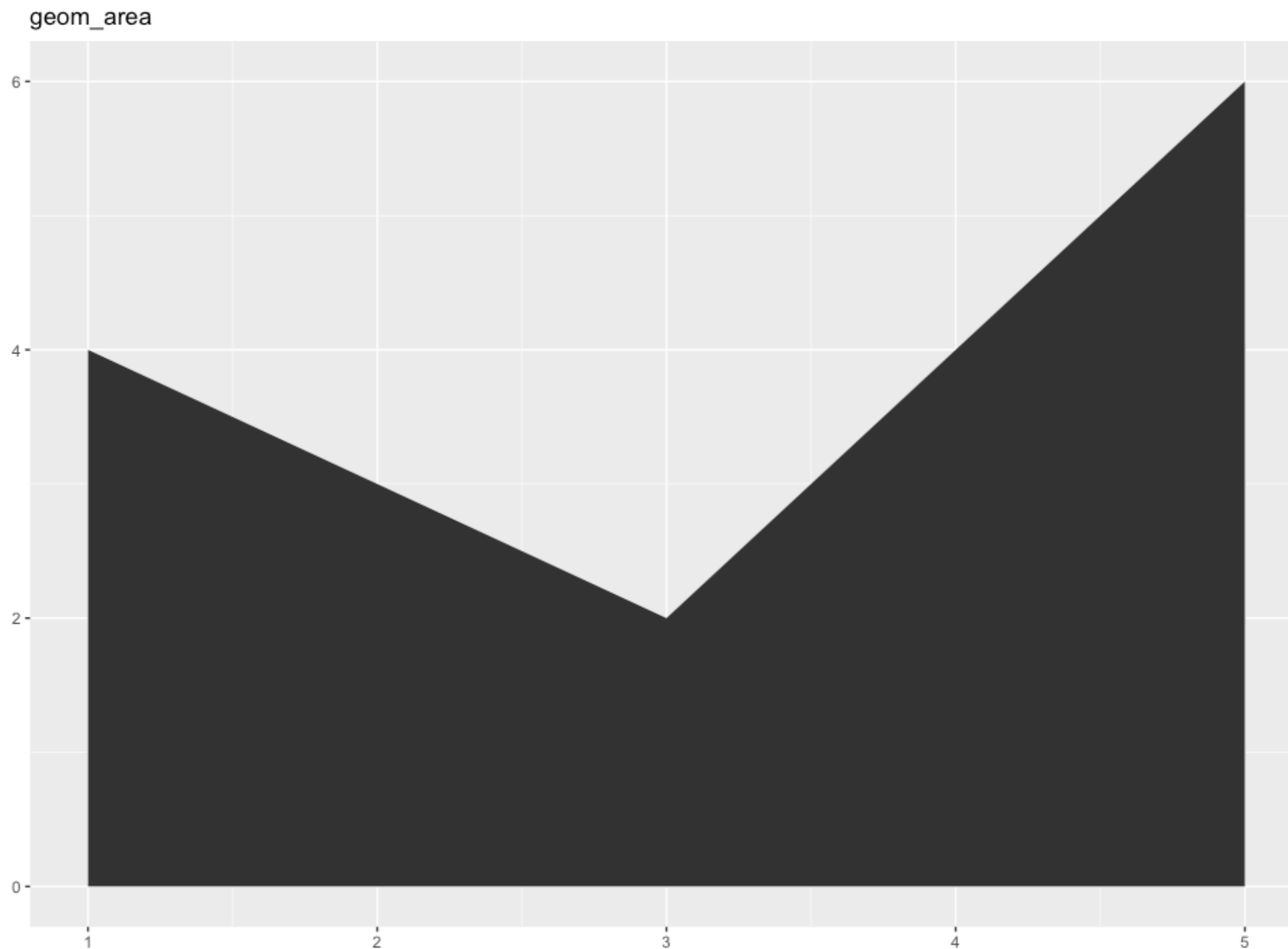


```
> p + geom_line() + labs(title = "geom_line")
```

group

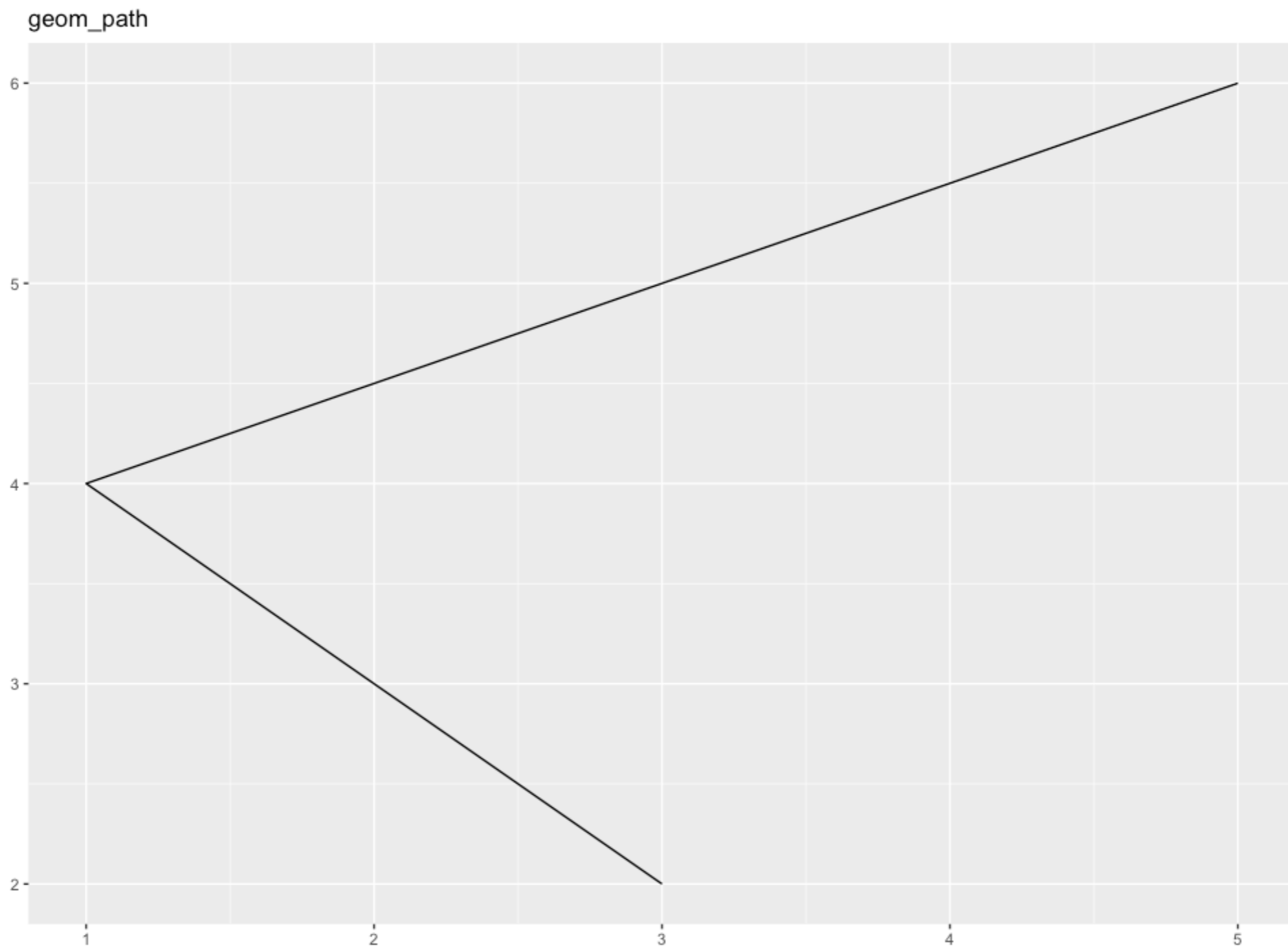


```
> p + geom_area() + labs(title = "geom_area")
```

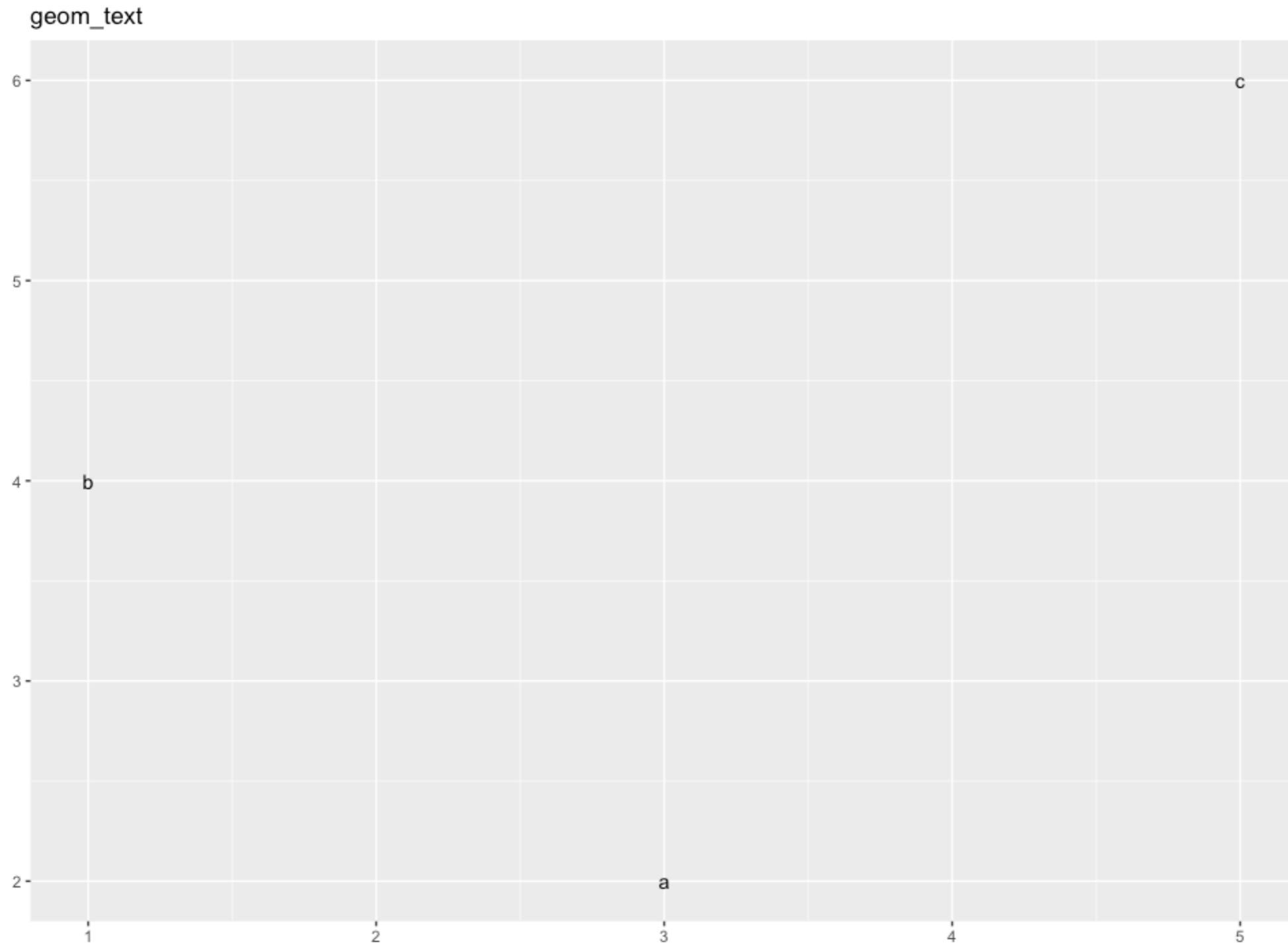


```
> p + geom_path() + labs(title = "geom_path")
```

数据出现顺序



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



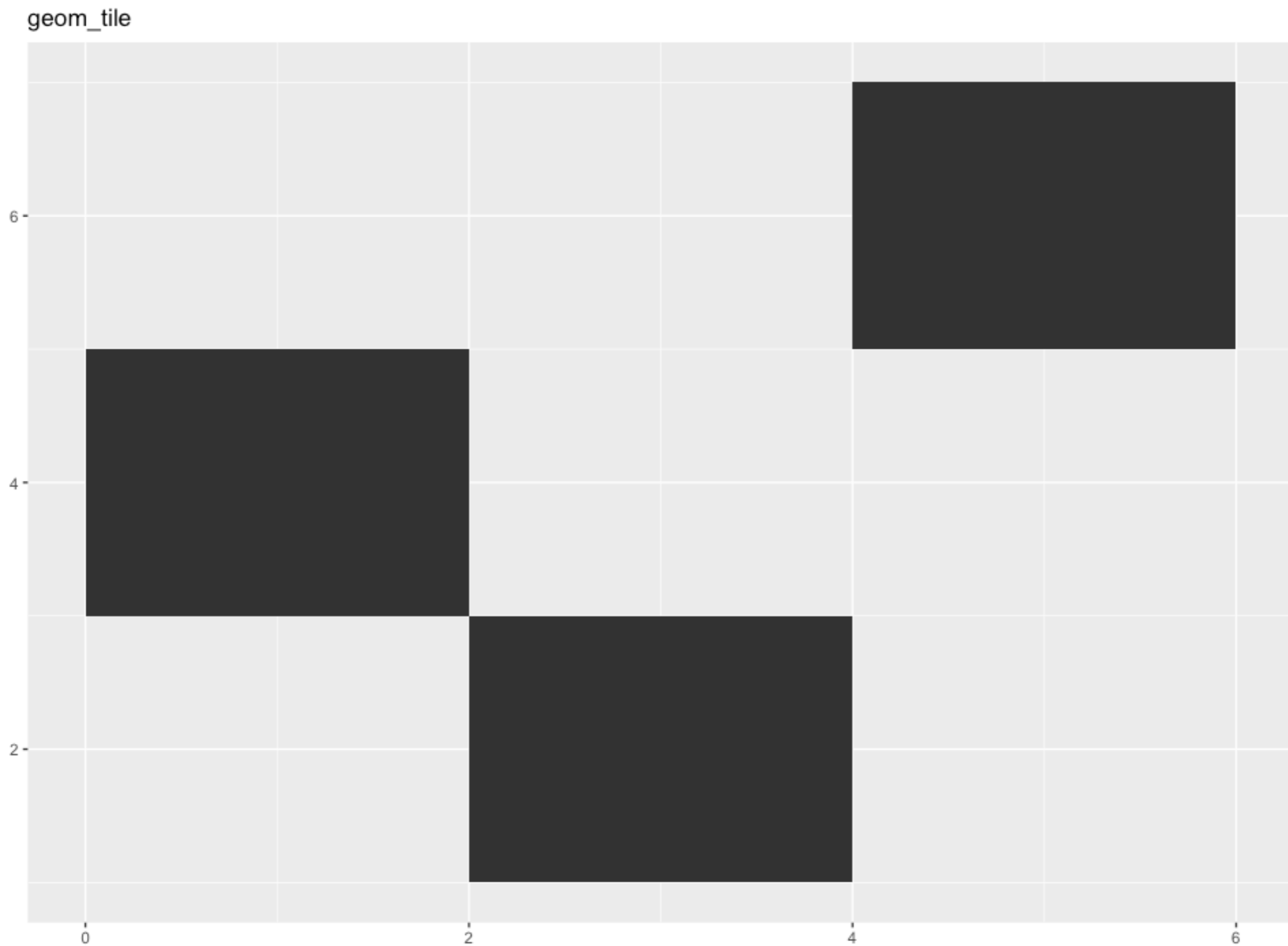
label

hjust

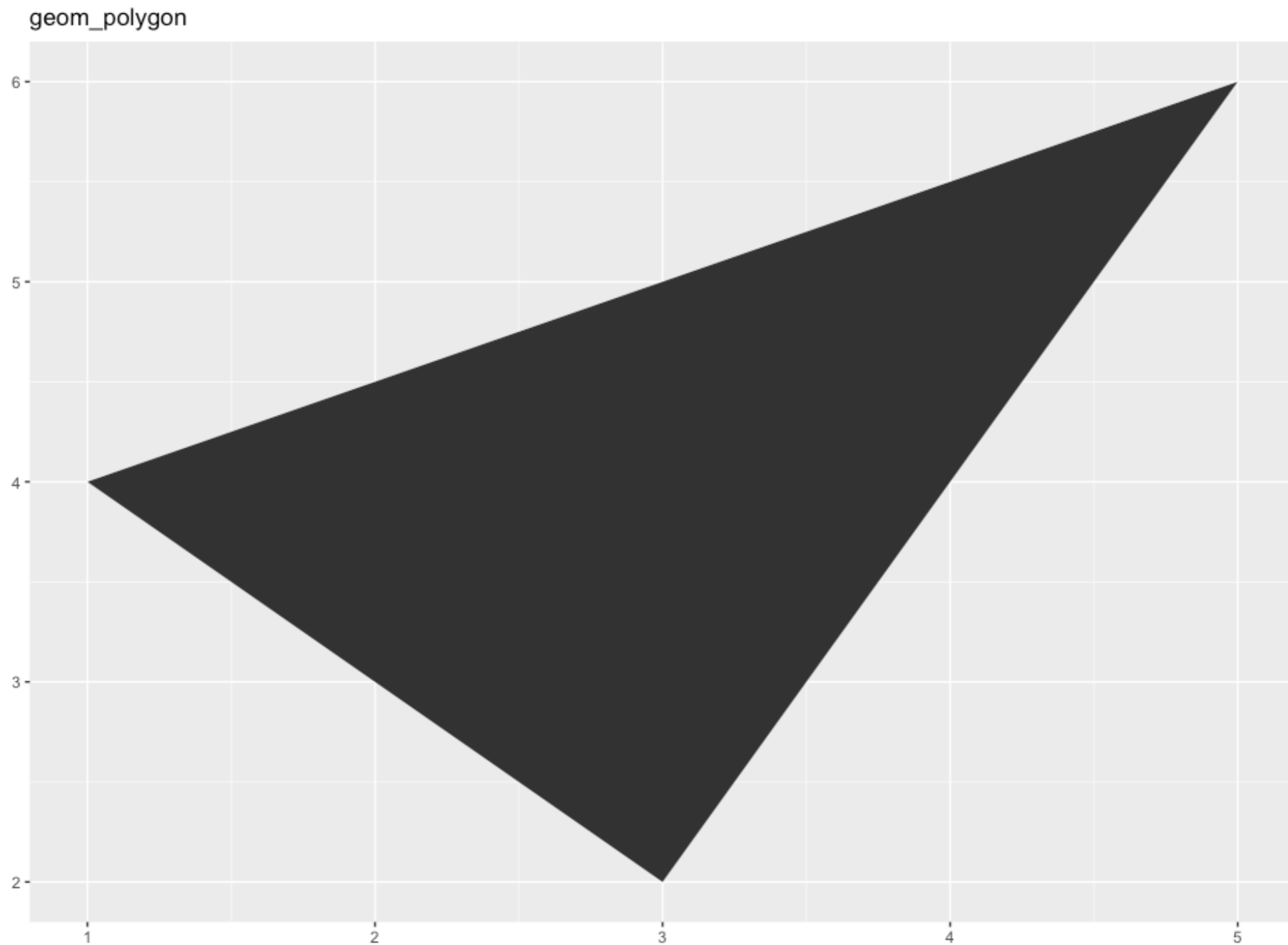
vjust

angle

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



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



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

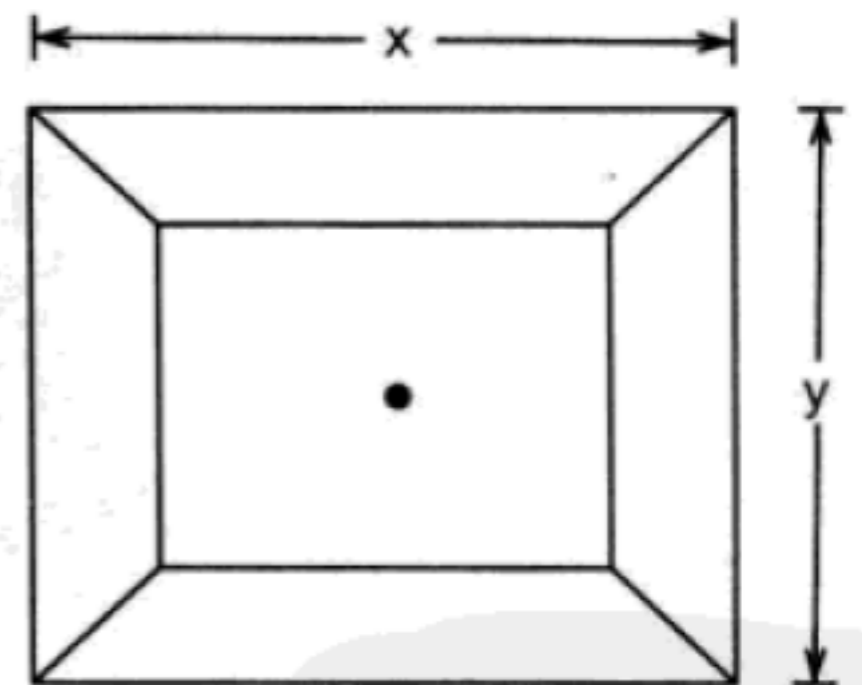
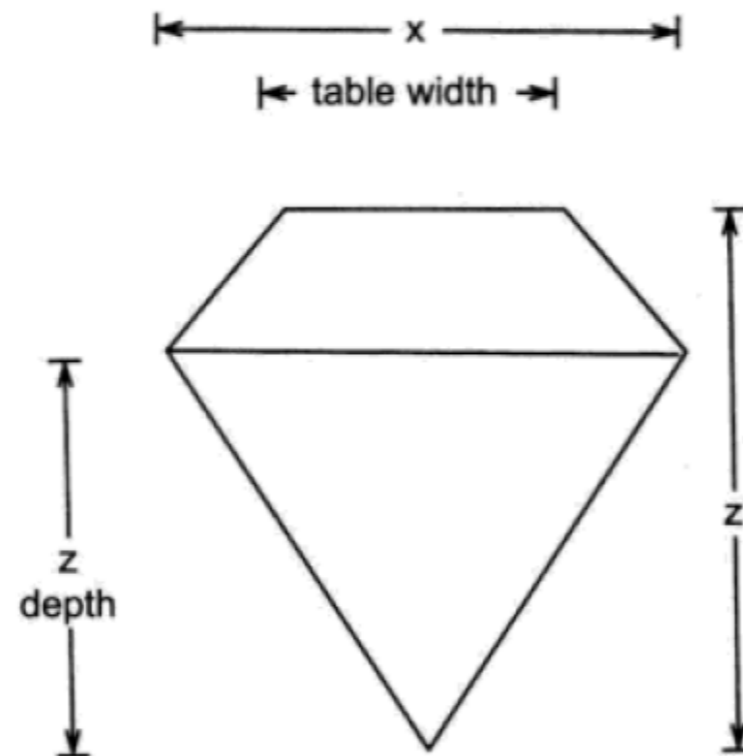
cut: 切工

color: 颜色

clarity: 净度

depty: 深度

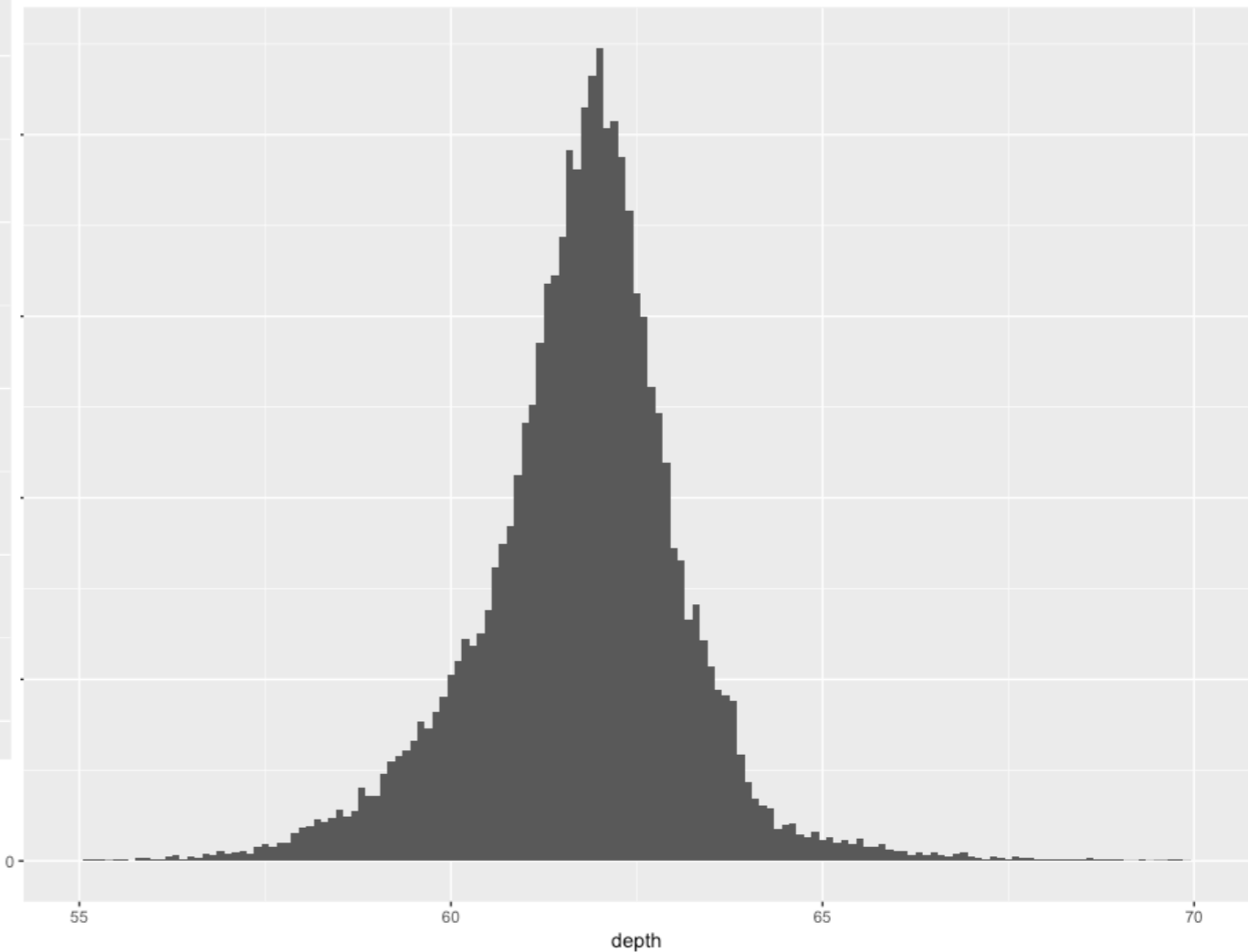
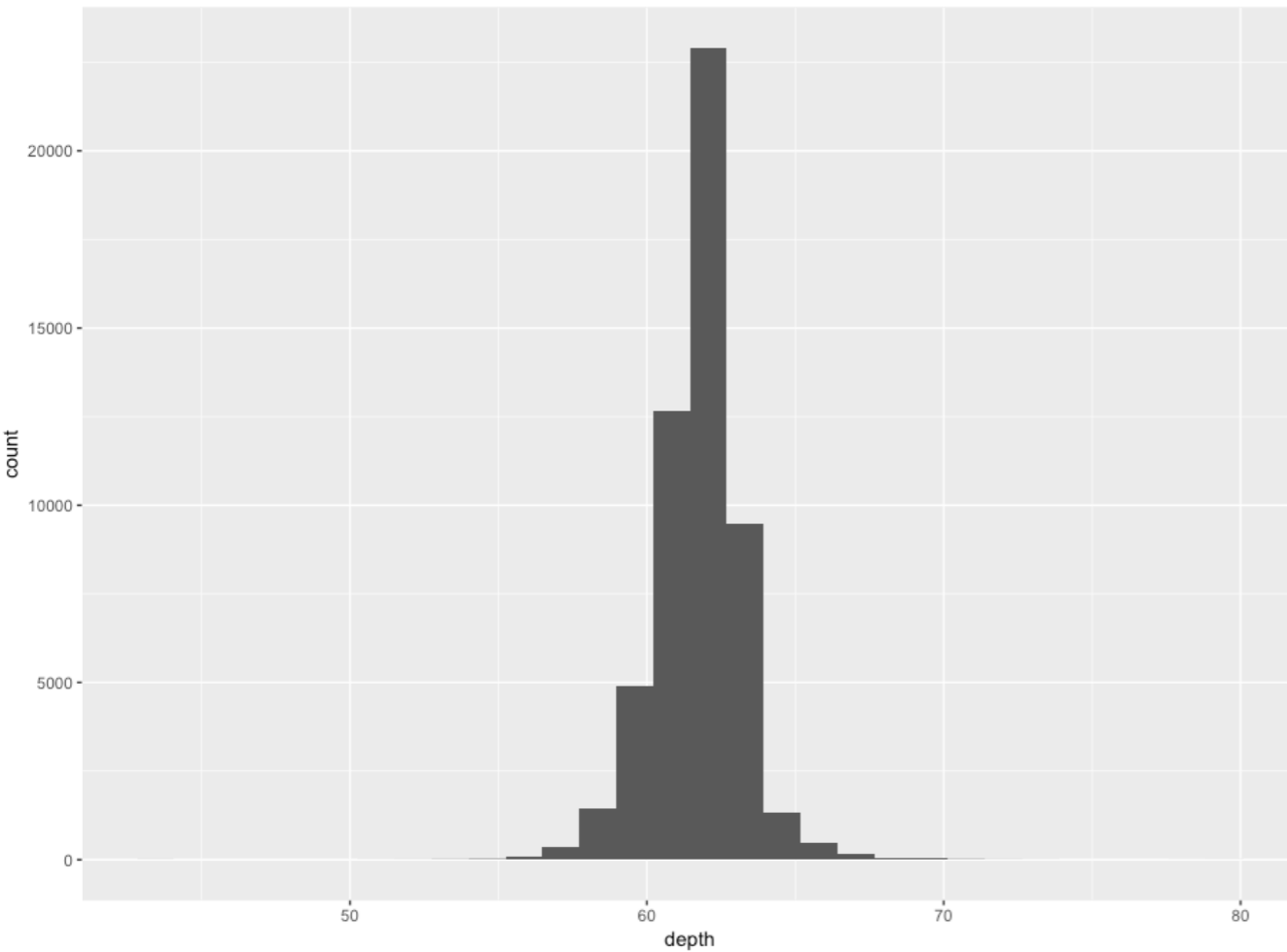
table: 钻面宽度



$$\text{depth} = z \text{ depth} / z * 100$$

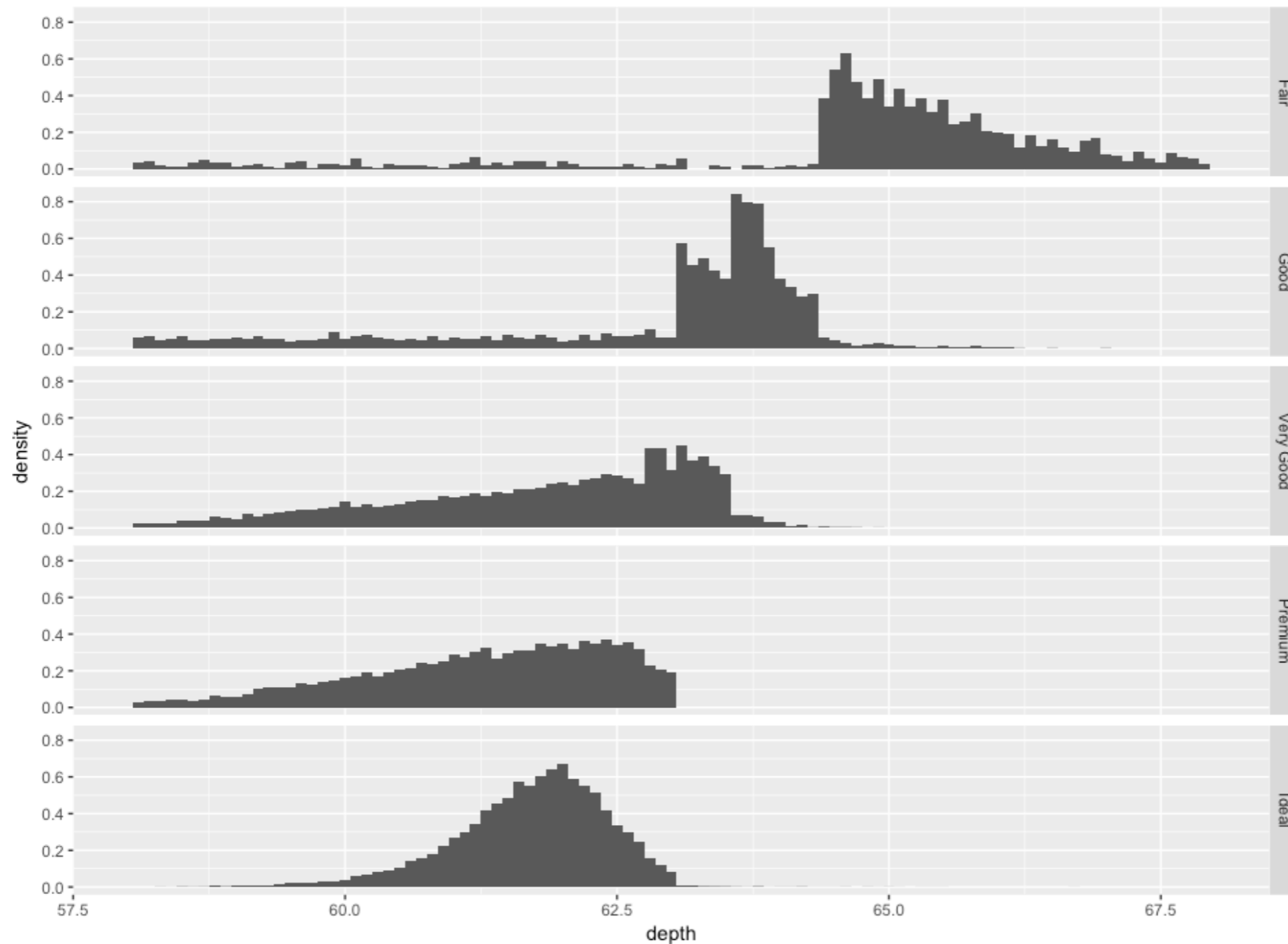
$$\text{table} = \text{table width} / x * 100$$


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



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

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

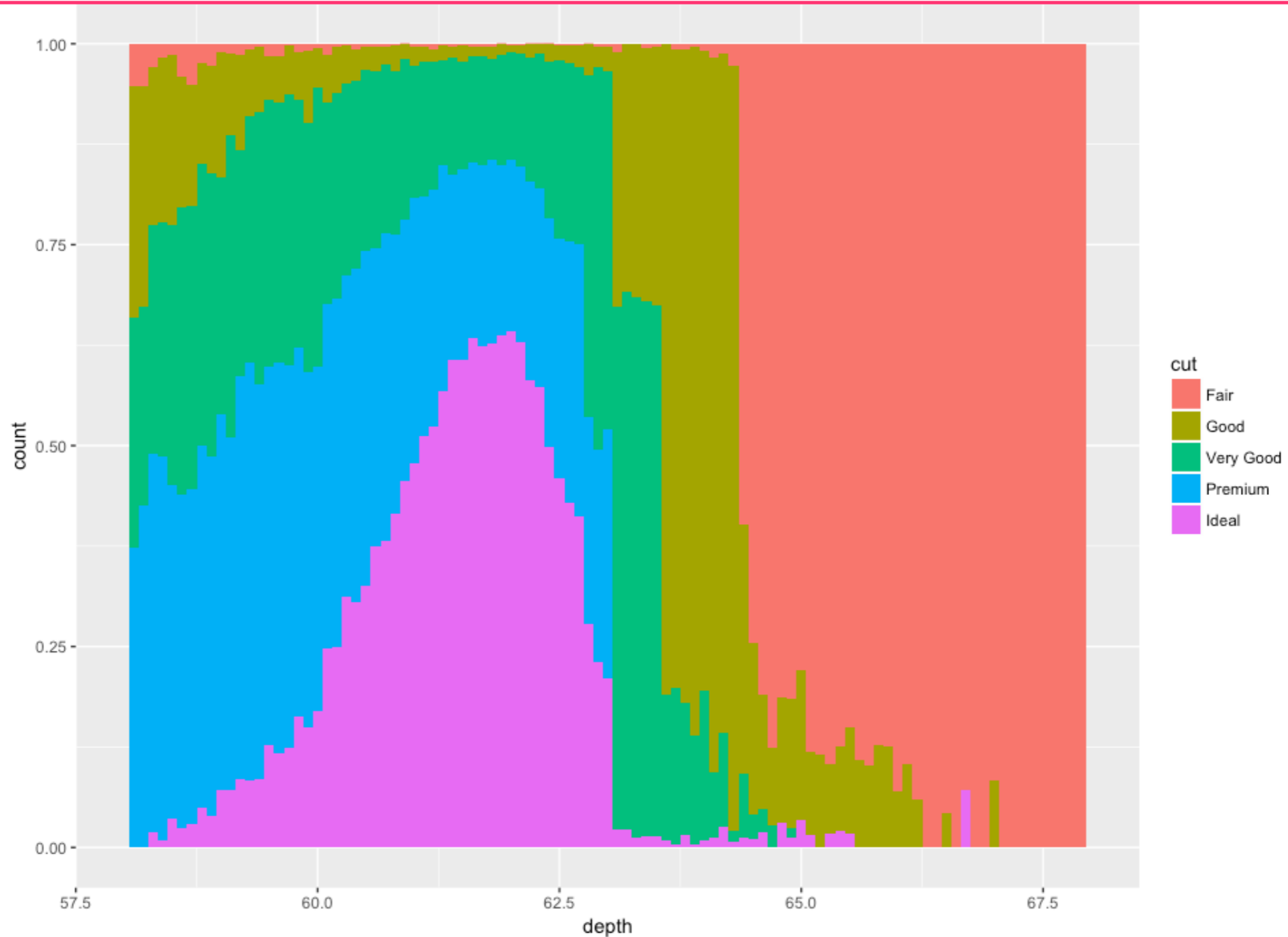


stat_bin

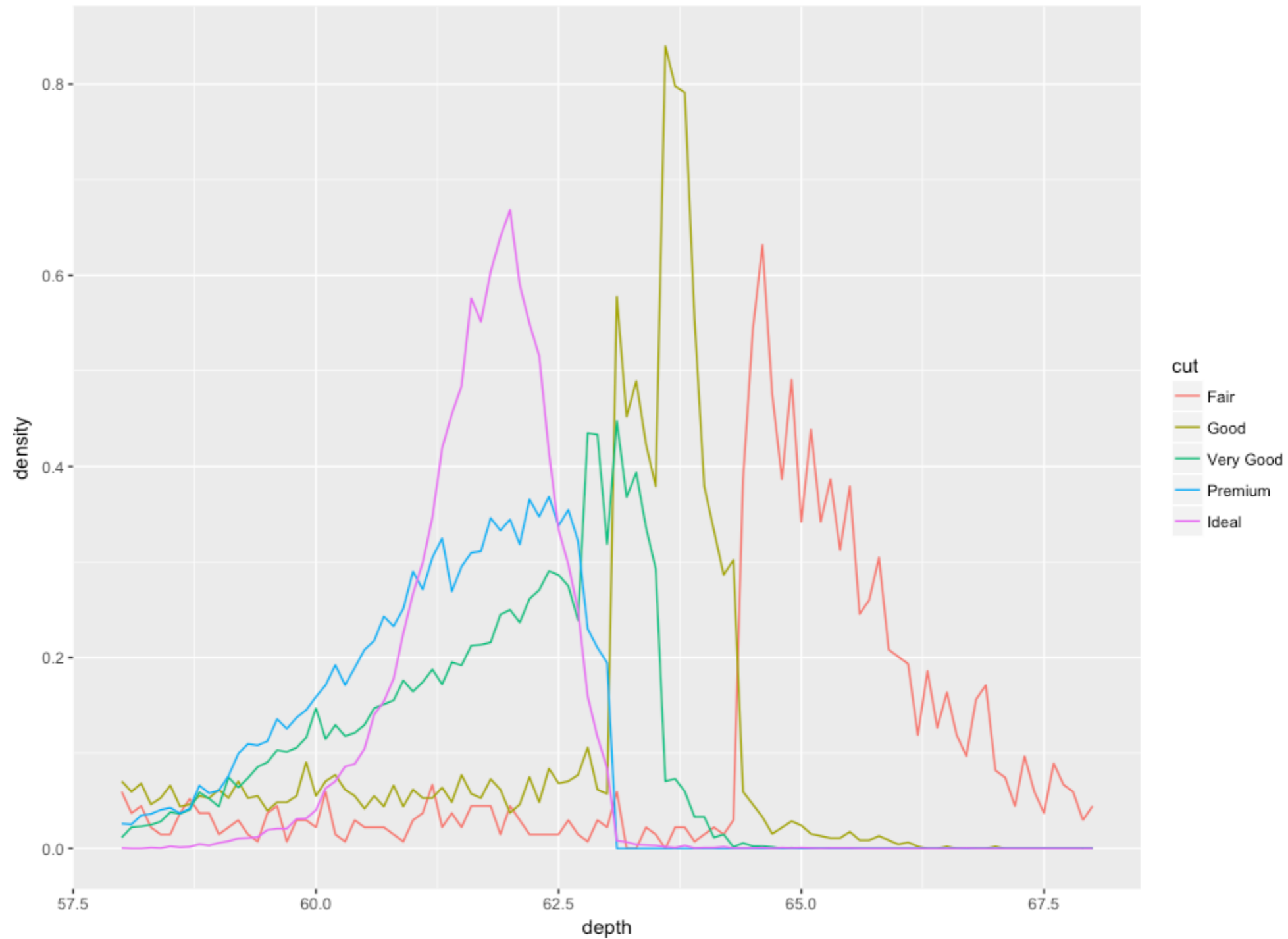
count

density

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



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



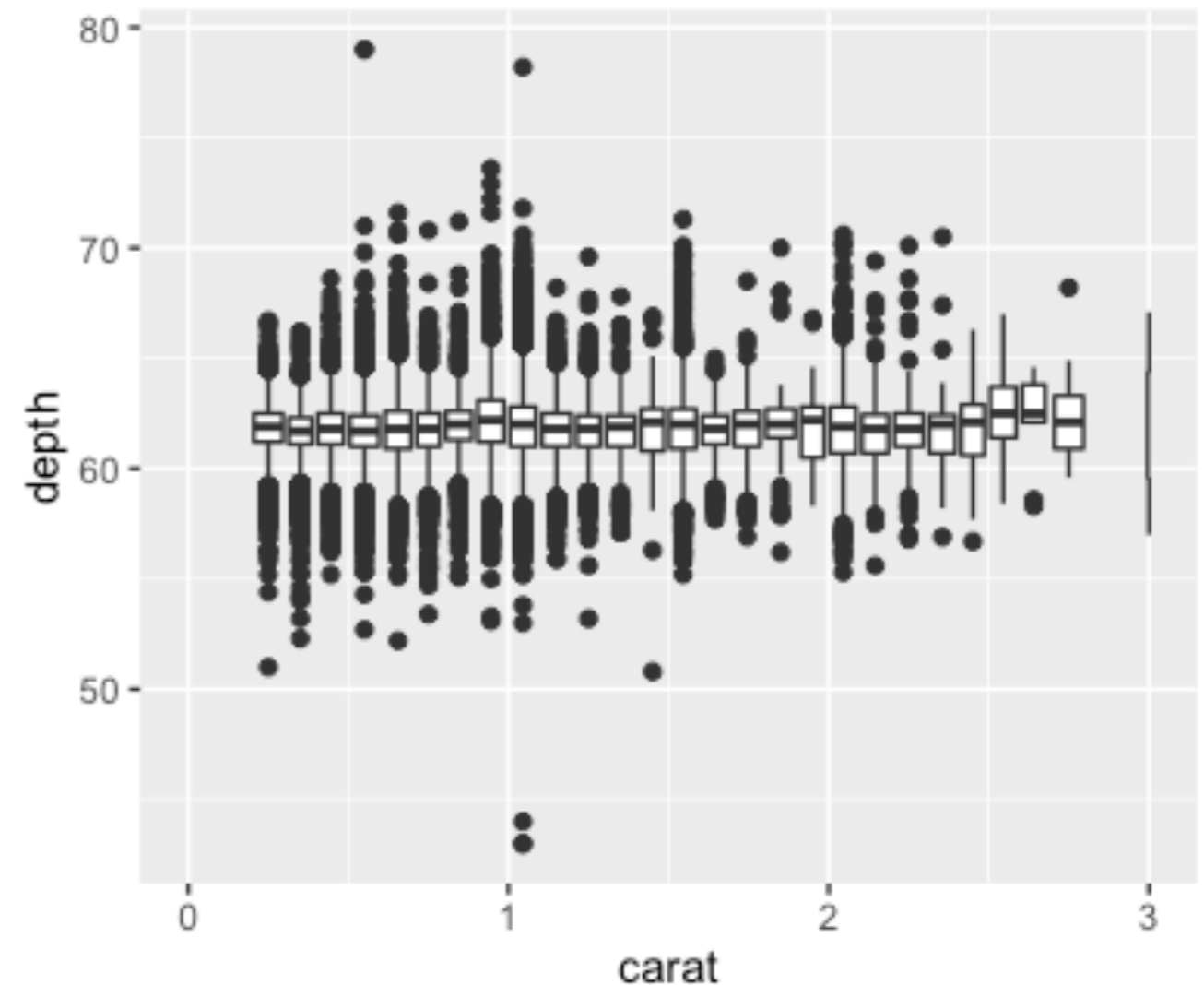
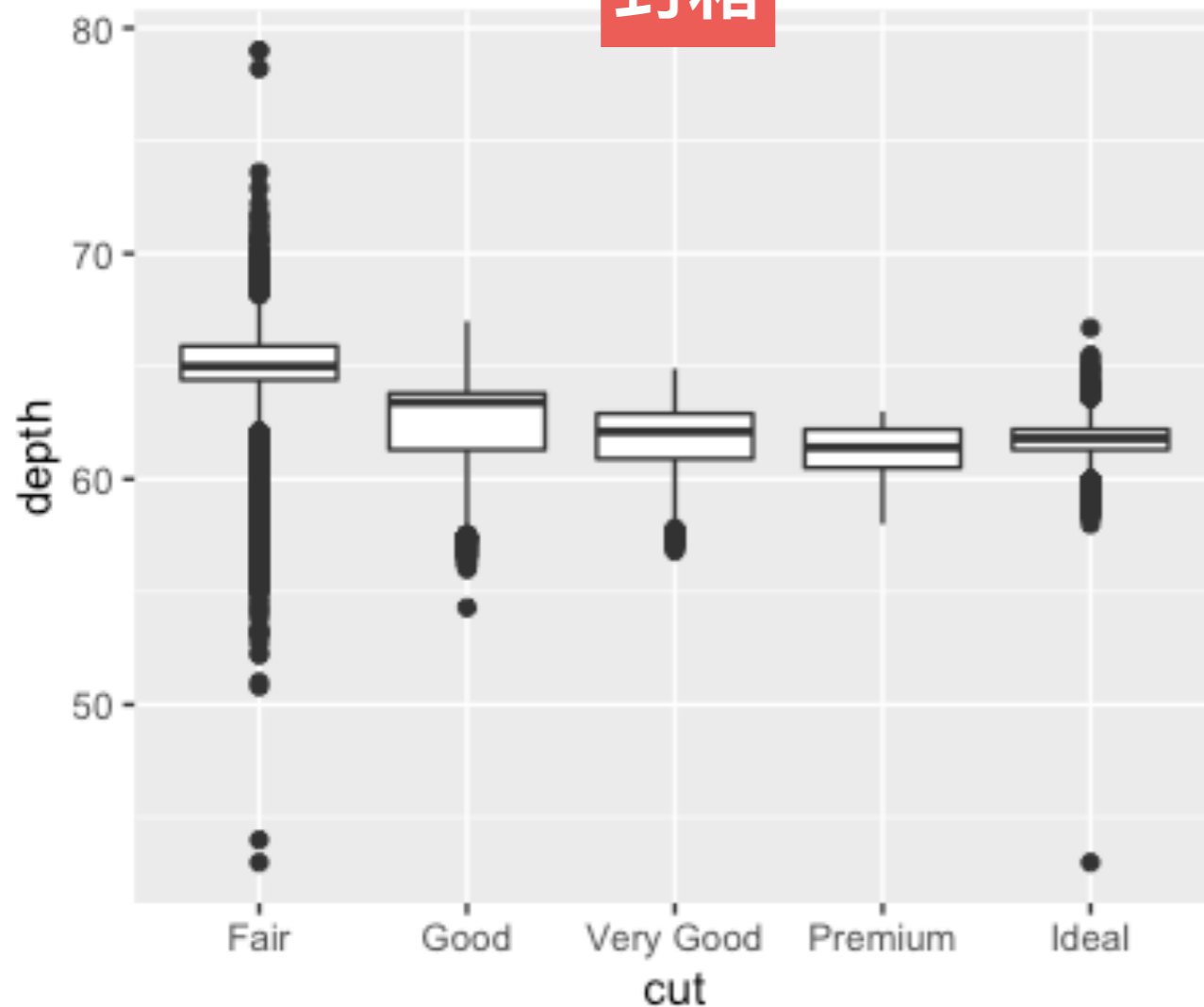
library(plyr)

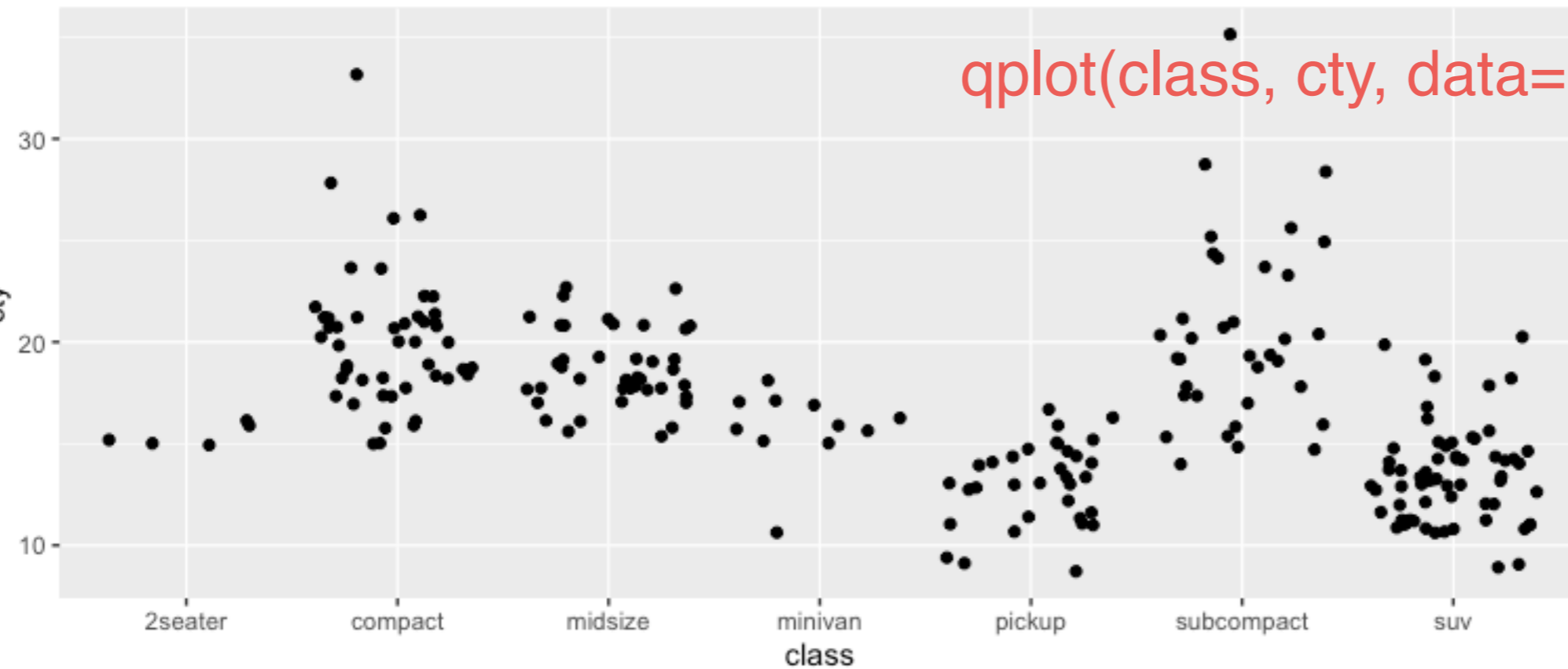
```
qplot(cut, depth, data=diamonds, geom="boxplot")  
qplot(carat, depth, data=diamonds, geom="boxplot",  
      group = round_any(carat, 0.1, floor), xlim = c(0, 3))
```

连续型变量

类别型变量

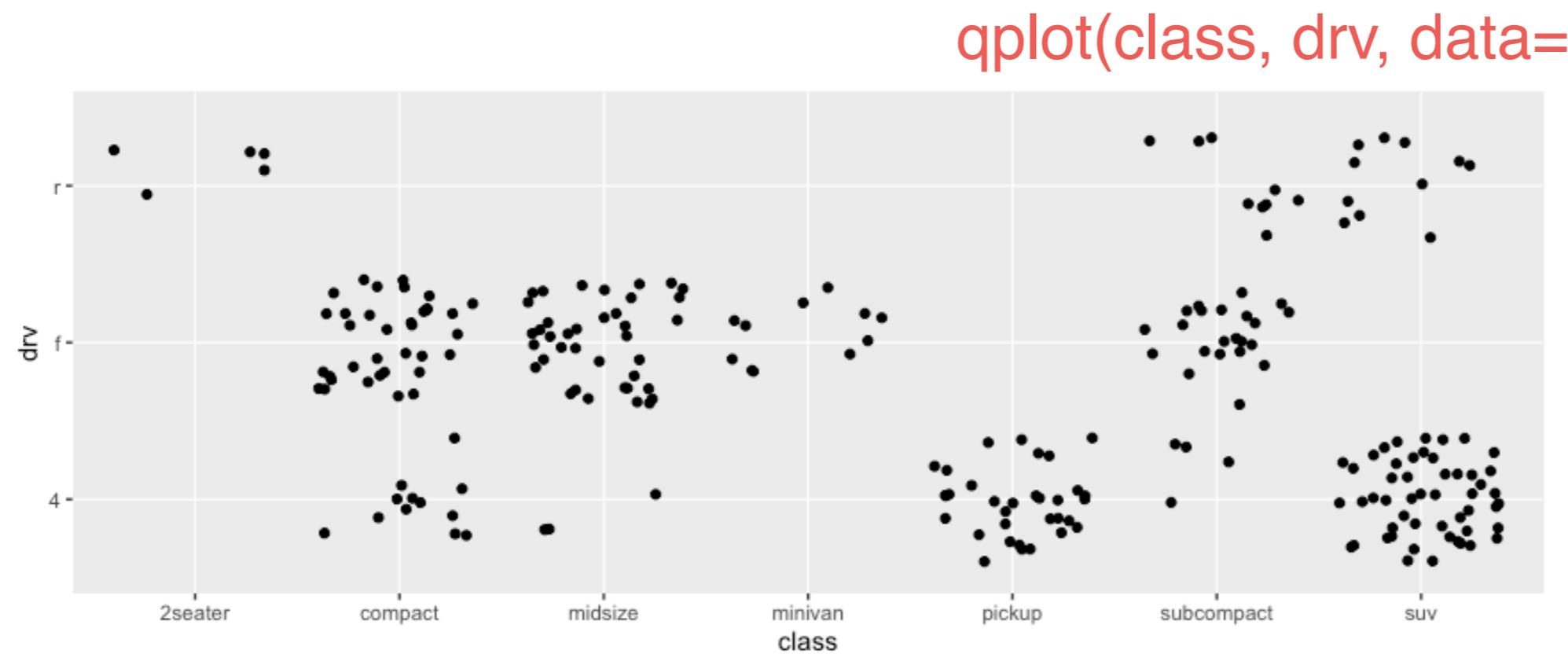
封箱





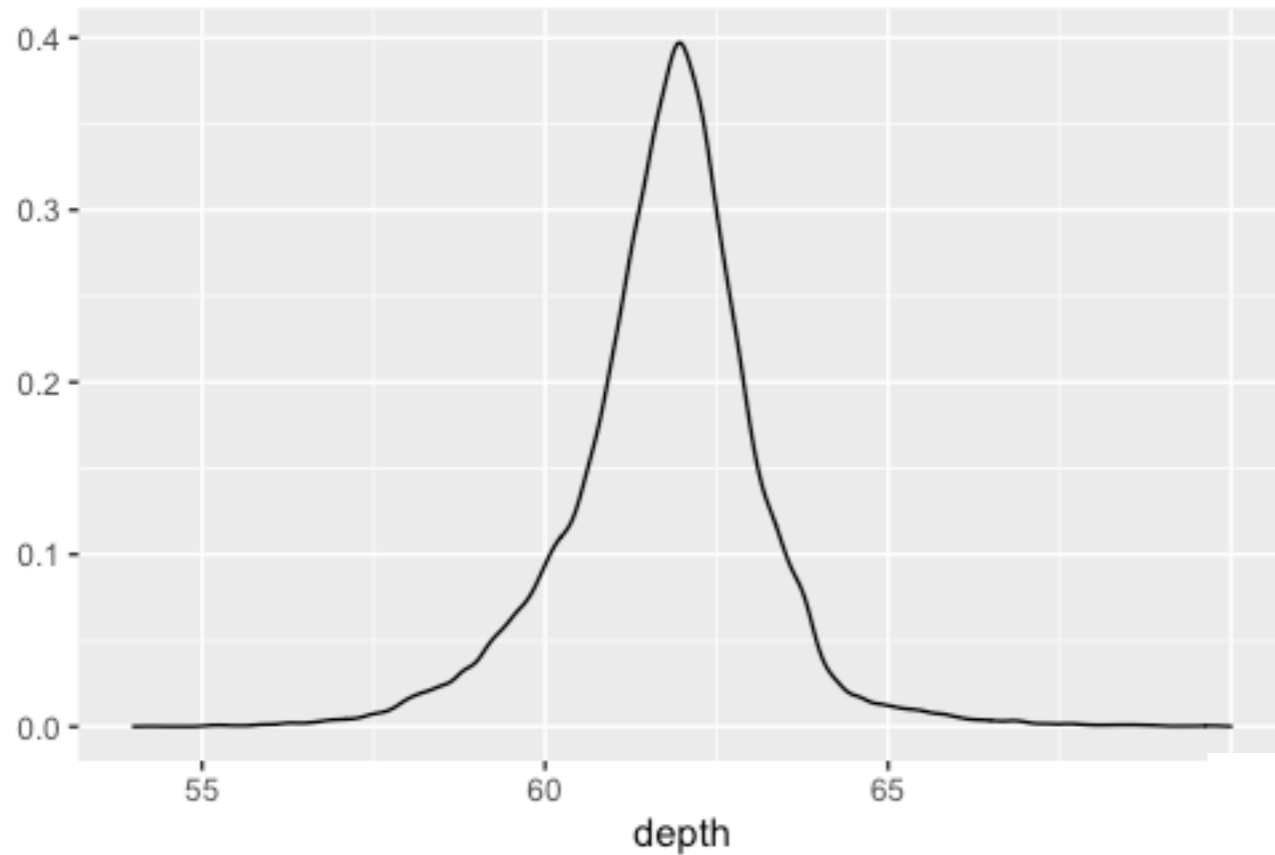
```
qplot(class, cty, data=mpg, geom="jitter")
```

连续型变量



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

离散型变量

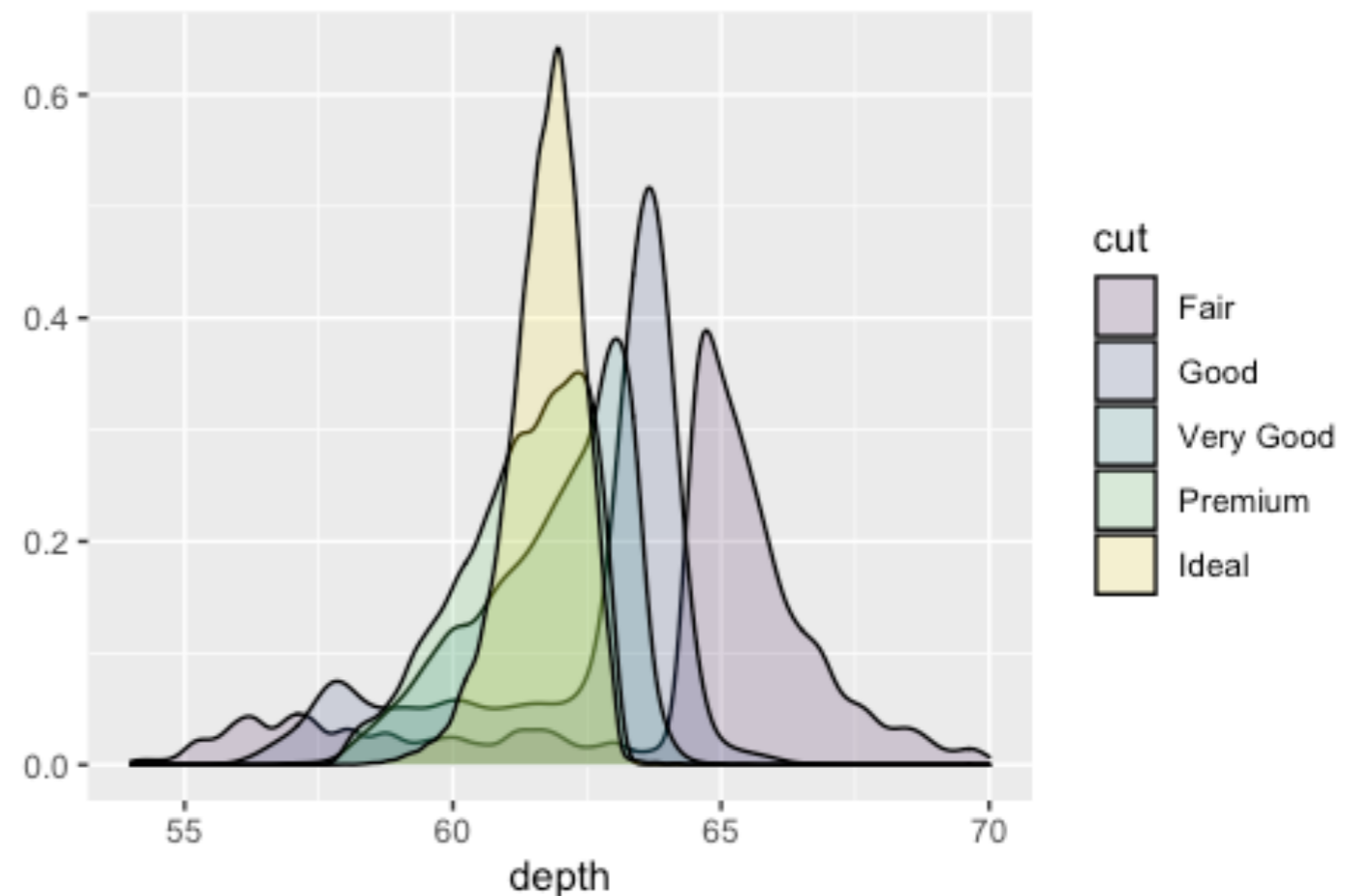


```
qplot(depth, data=diamonds,  
geom="density", xlim = c(54,  
70))
```

核平滑方法

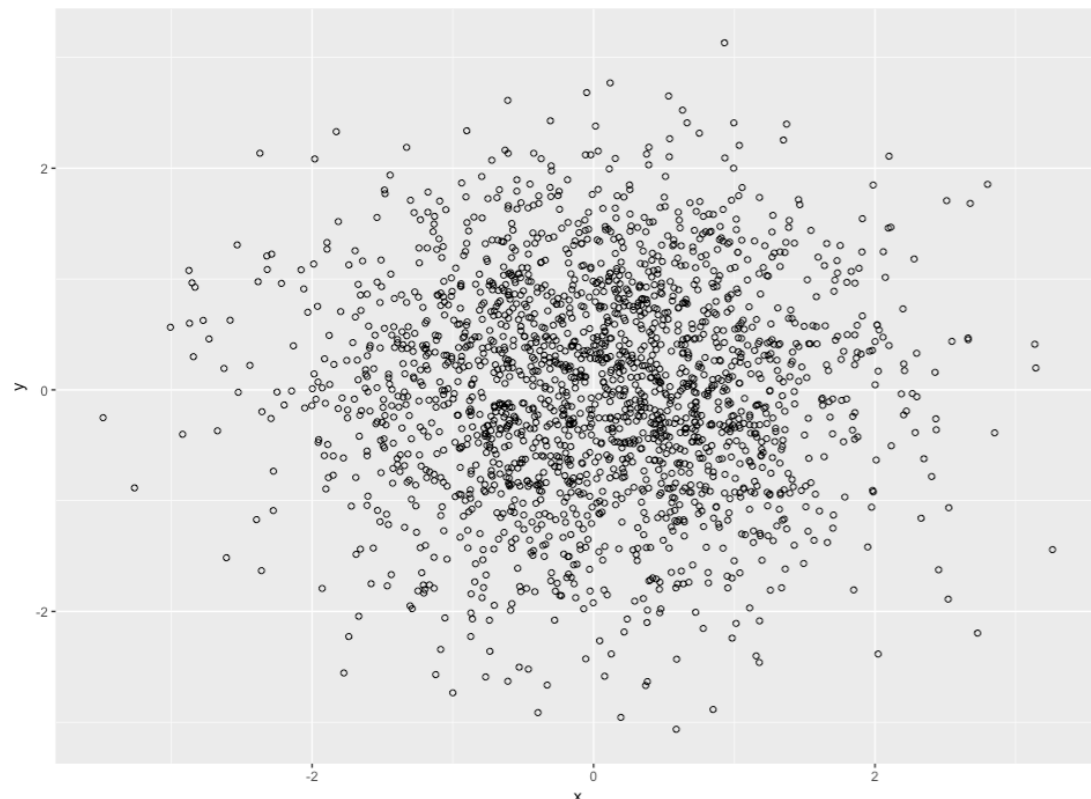
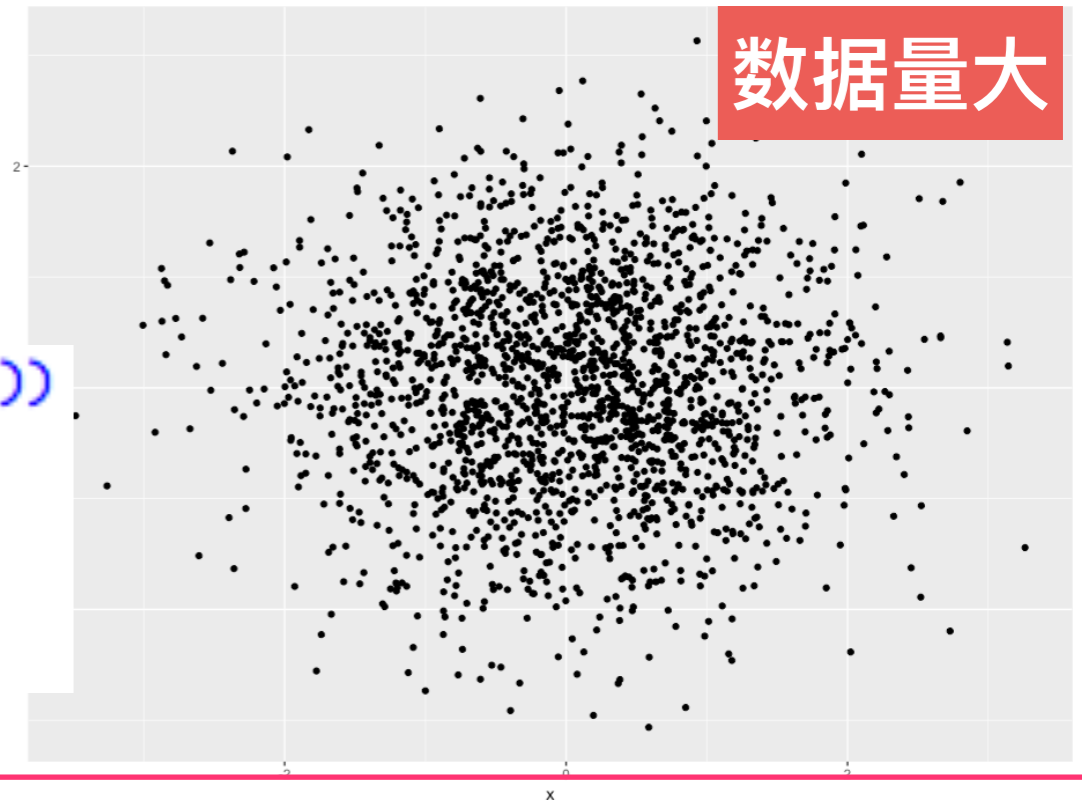
```
qplot(depth, data=diamonds,  
geom="density", xlim = c(54,  
70), fill = cut, alpha = 1(0.2))
```

难于回溯到数据

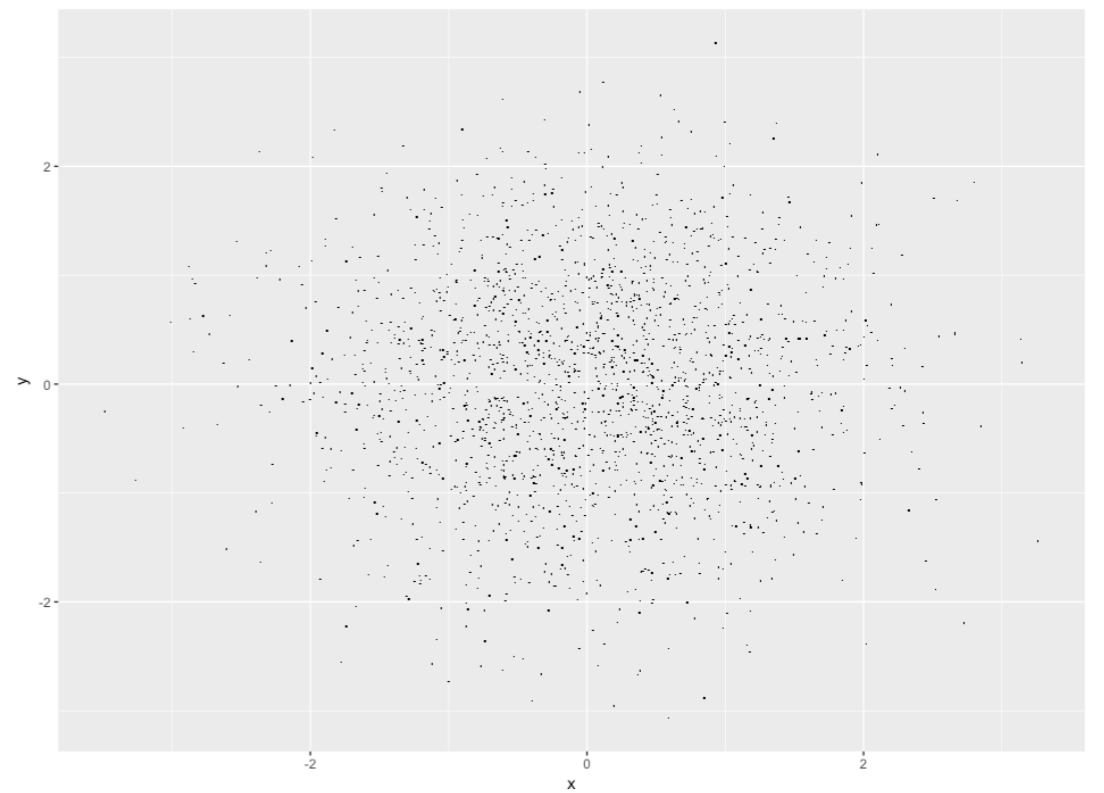


使用点的大小

```
> 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
```



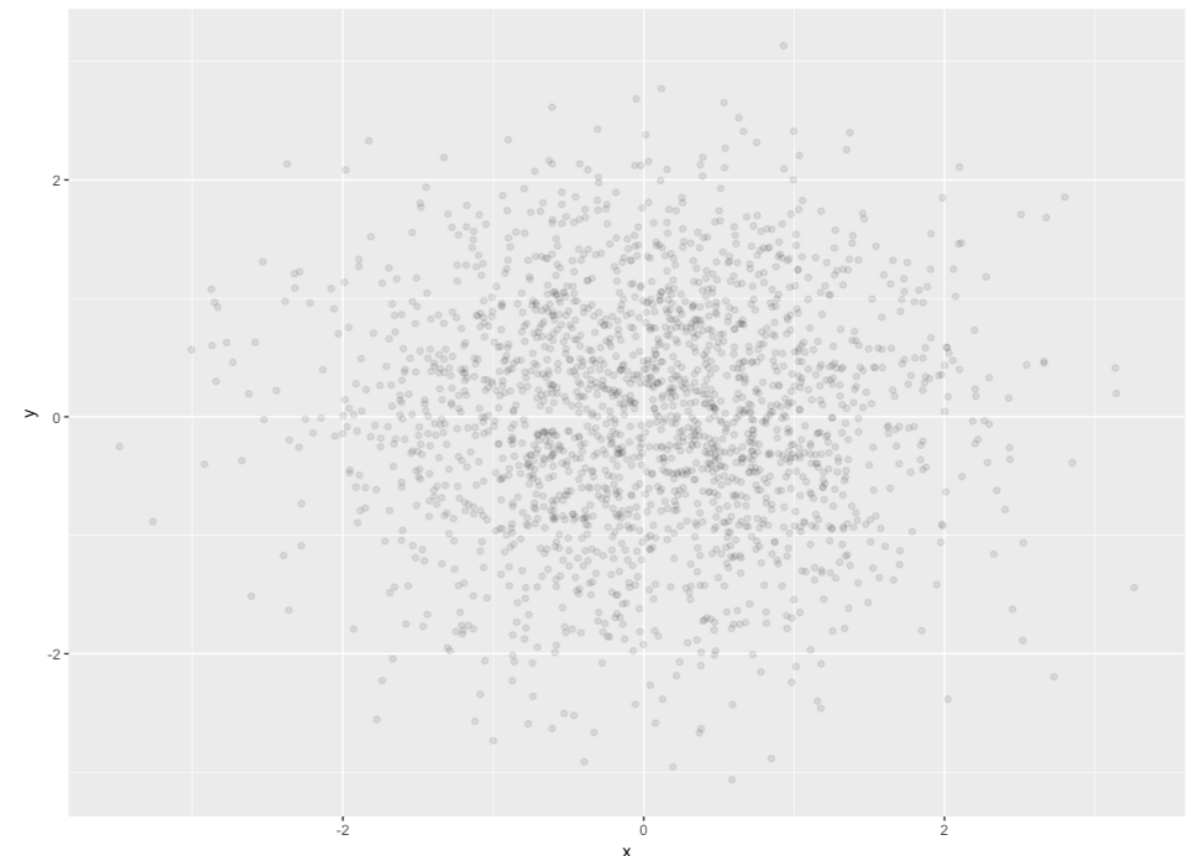
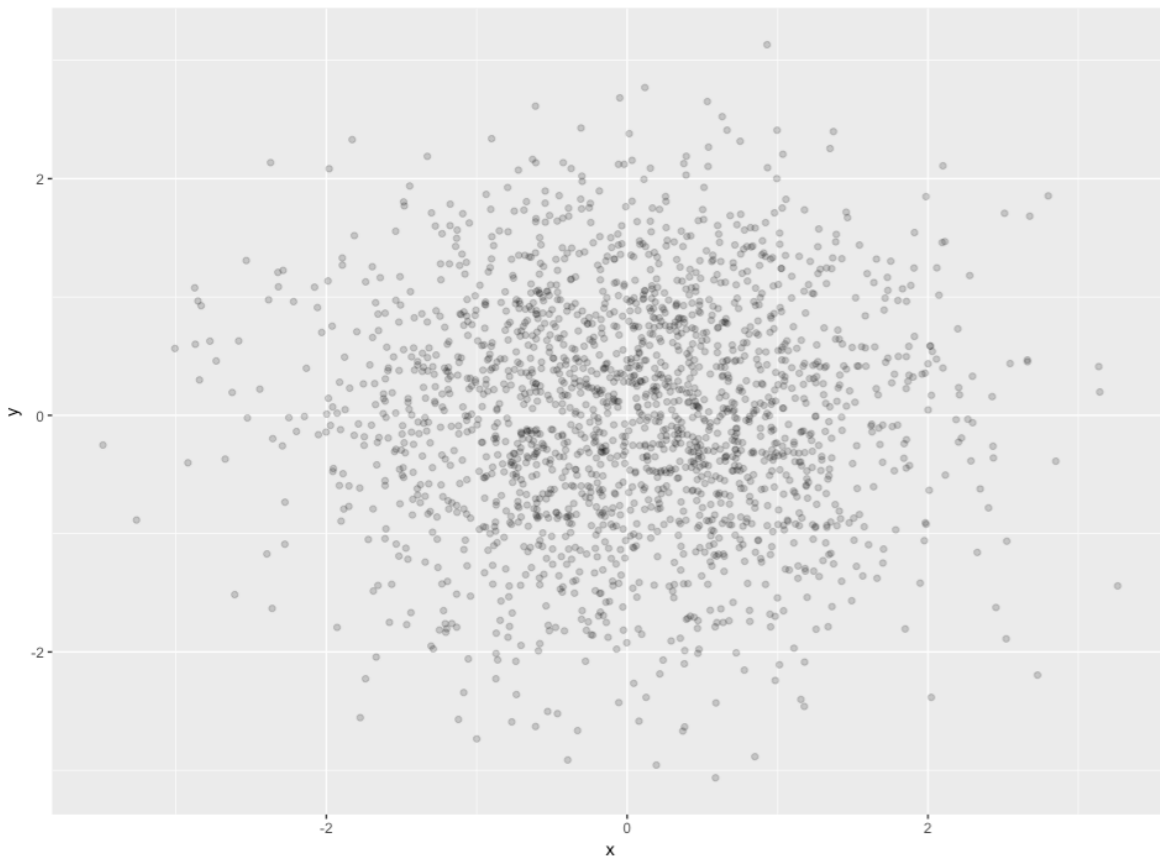
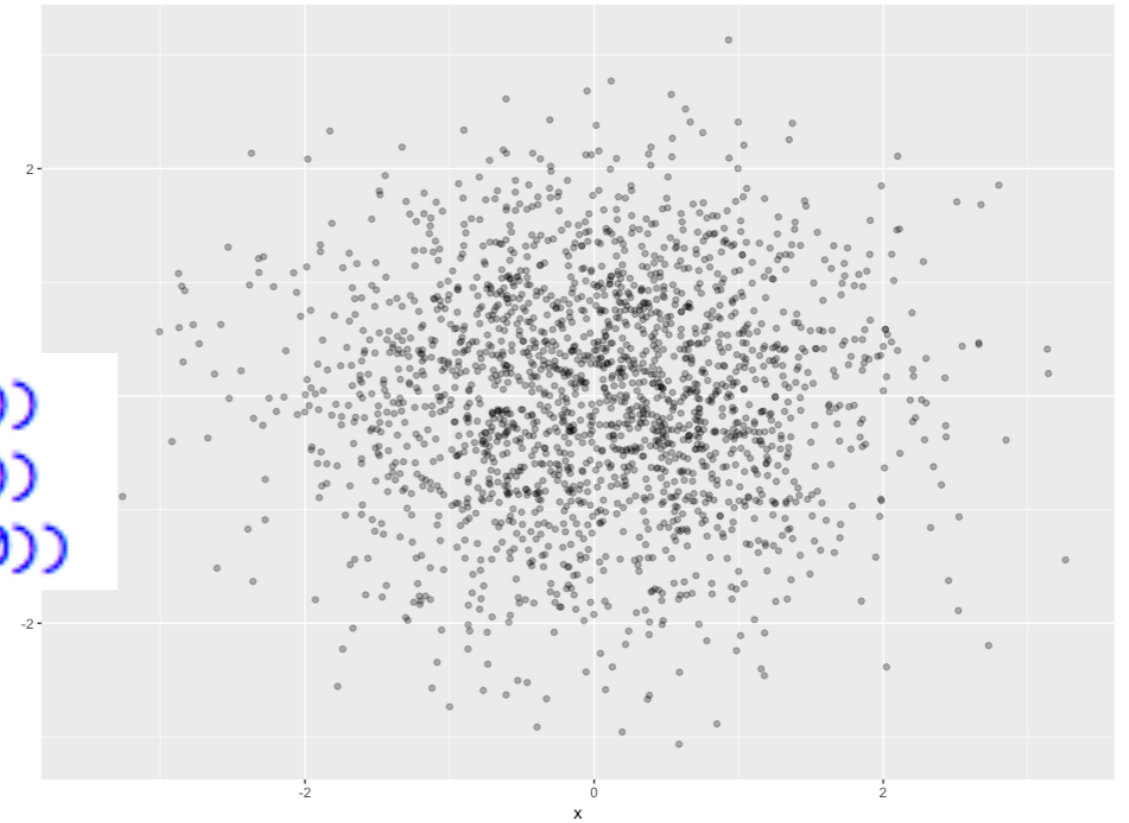
shape

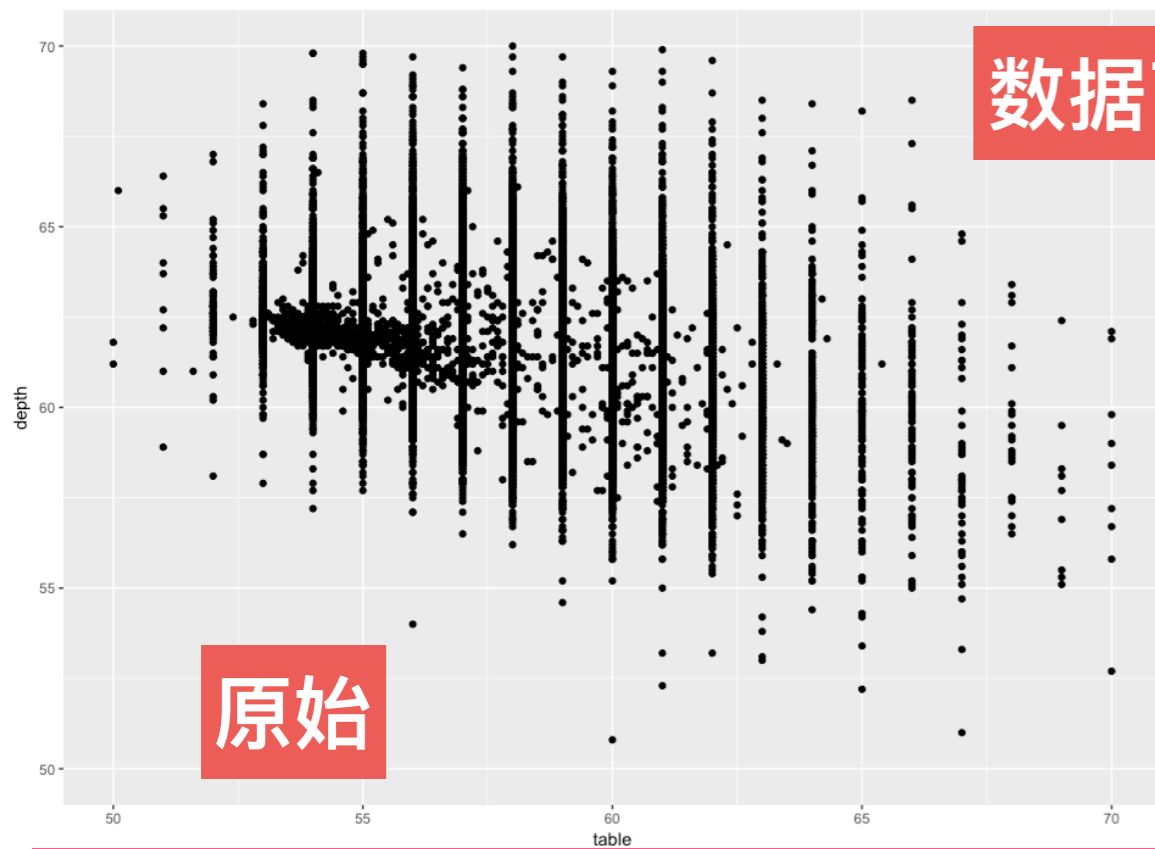


使用点的透明度

alpha

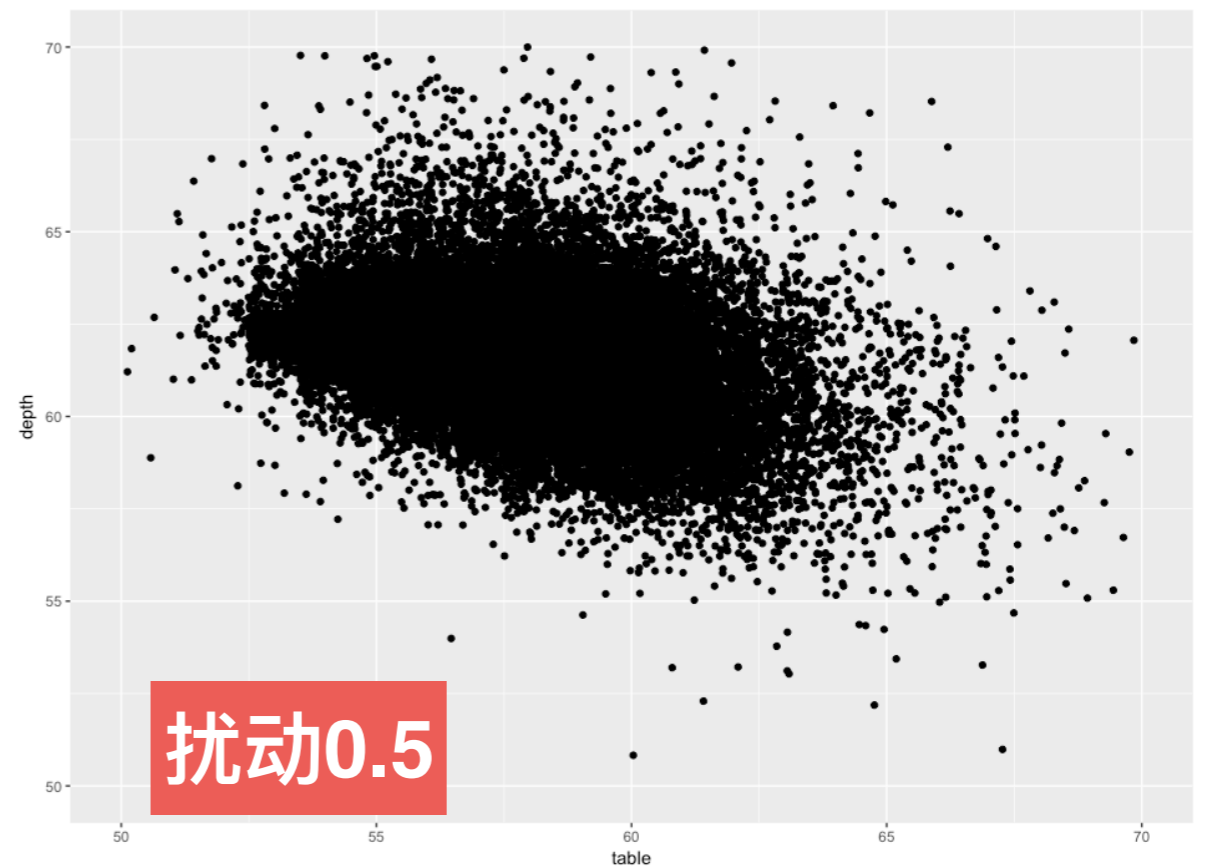
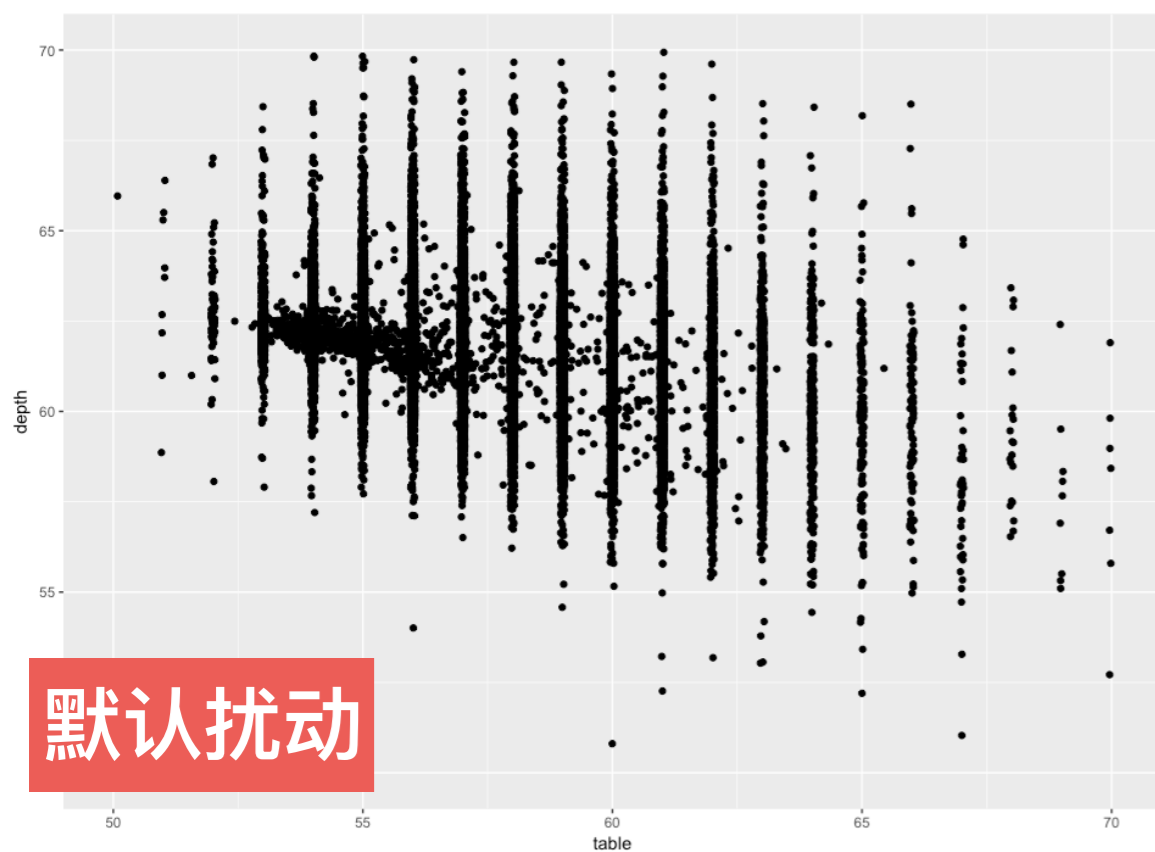
```
> norm + geom_point(colour = alpha("black", 1/3))  
> norm + geom_point(colour = alpha("black", 1/5))  
> norm + geom_point(colour = alpha("black", 1/10))
```

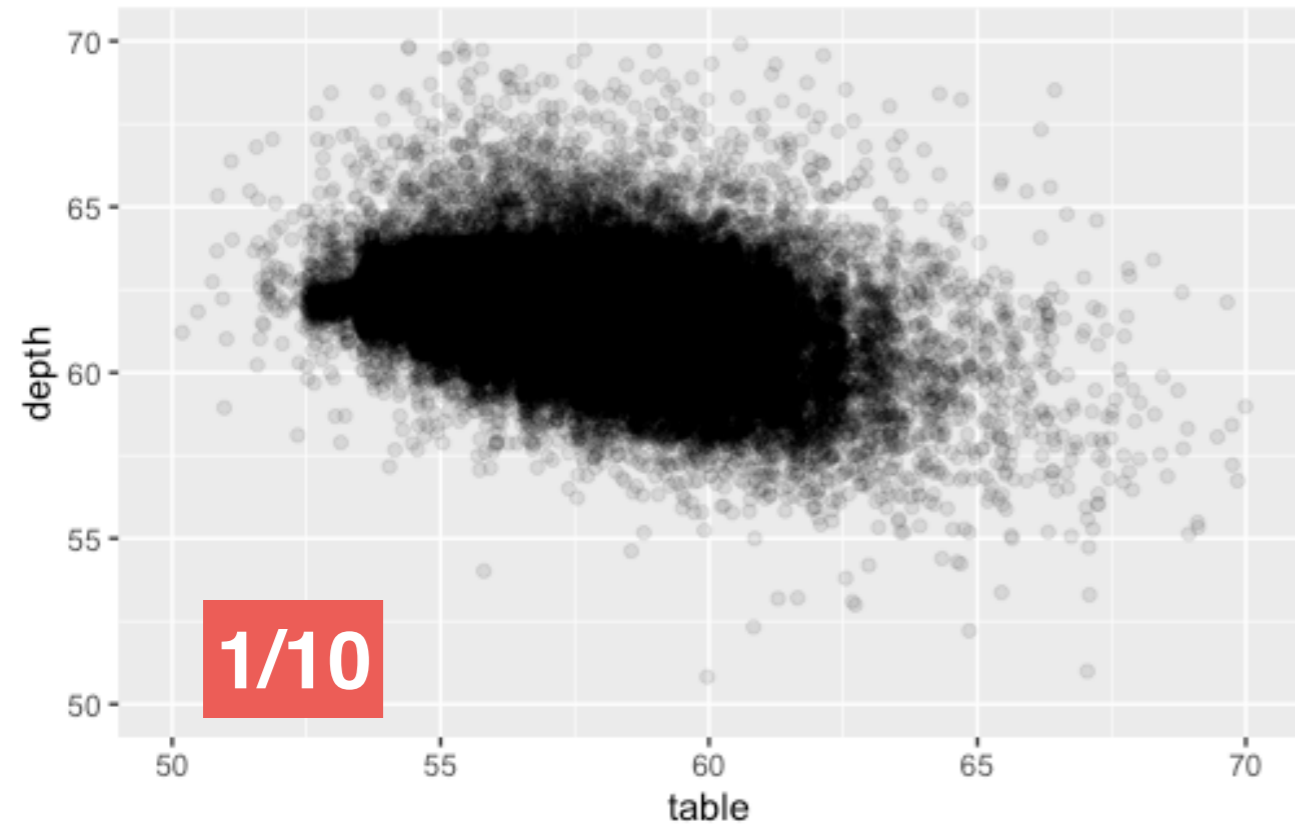




使用随机扰动

```
> 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)
```

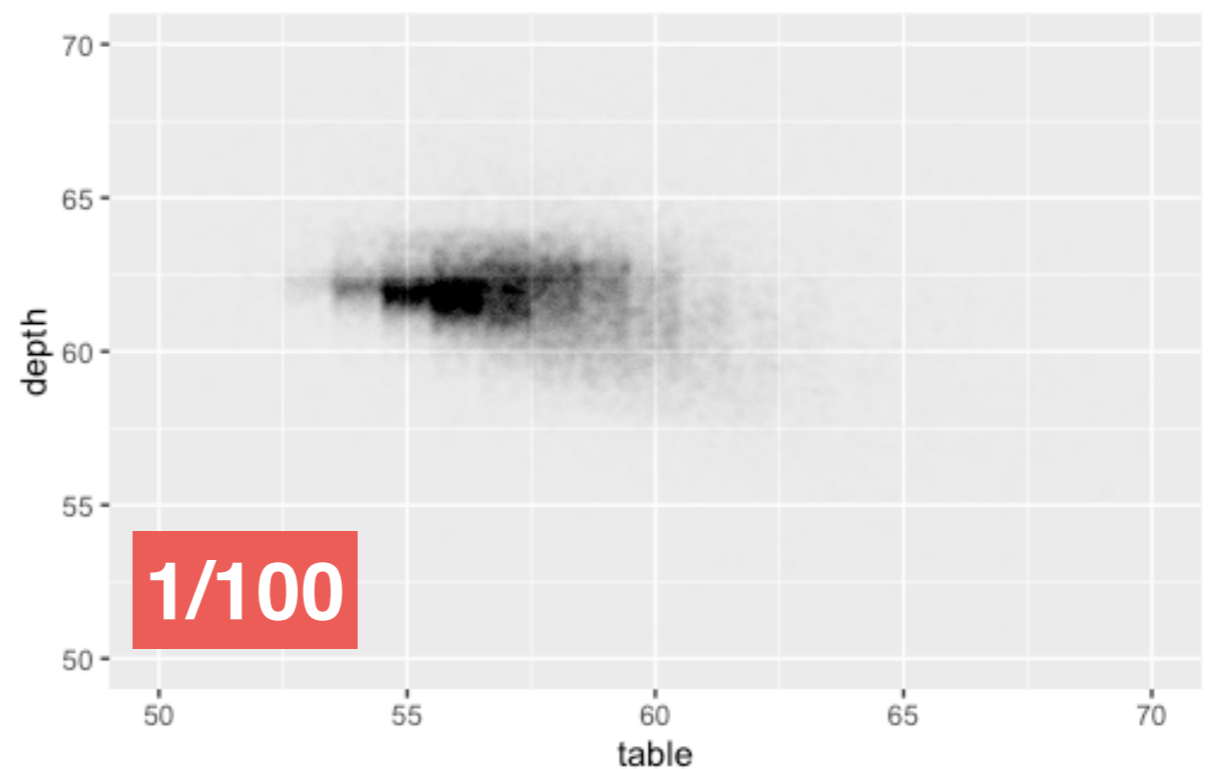
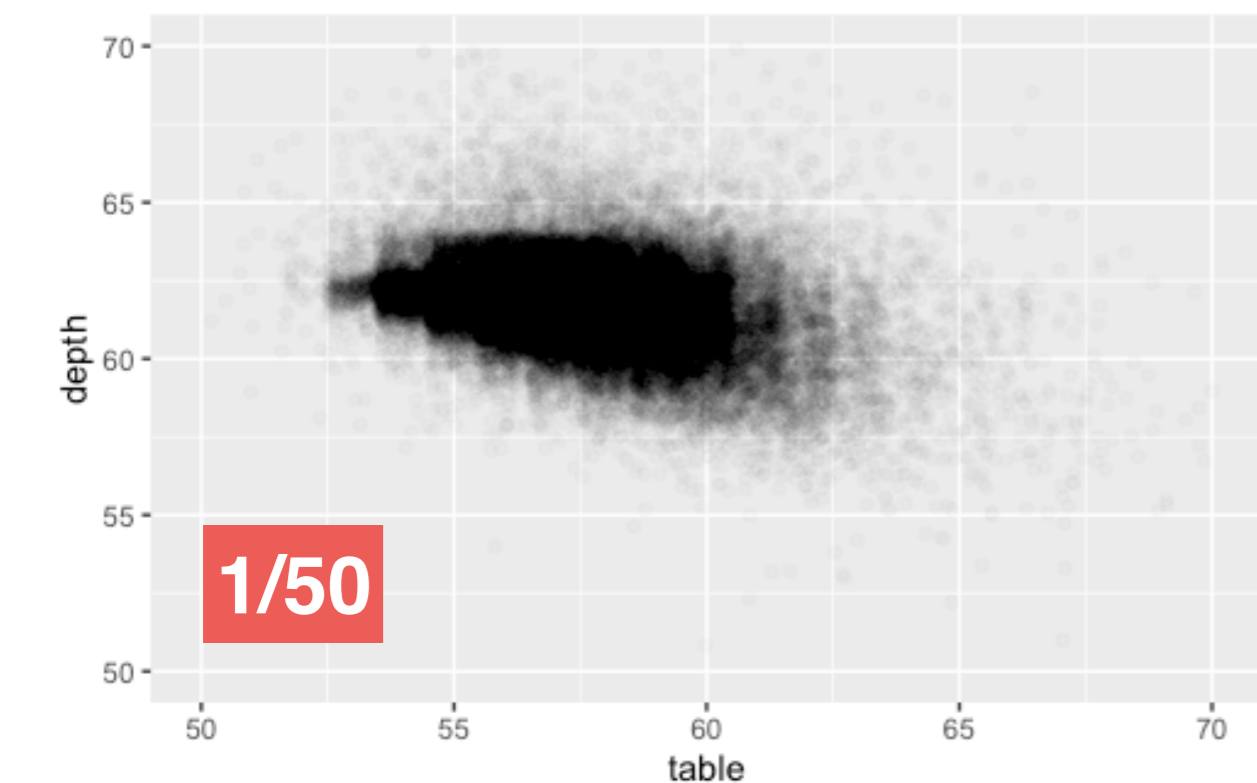


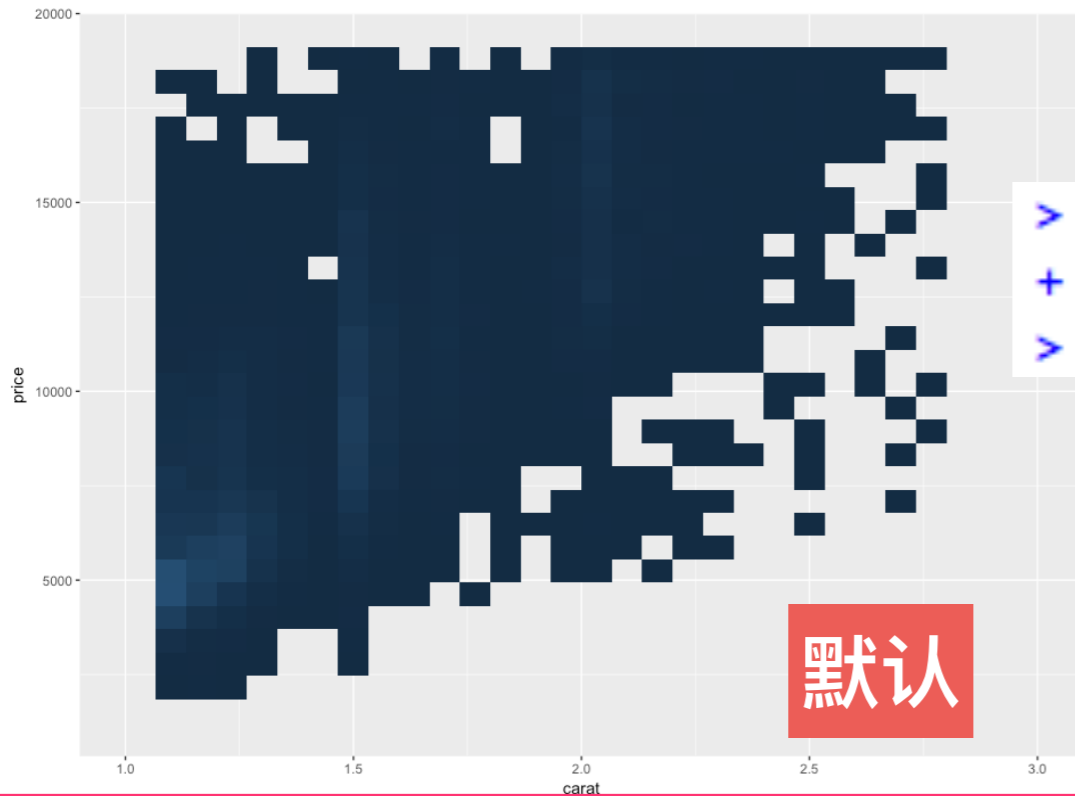


`td + geom_jitter(position = jit,
colour = alpha("black", 1/10))`

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

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





使用分箱计数

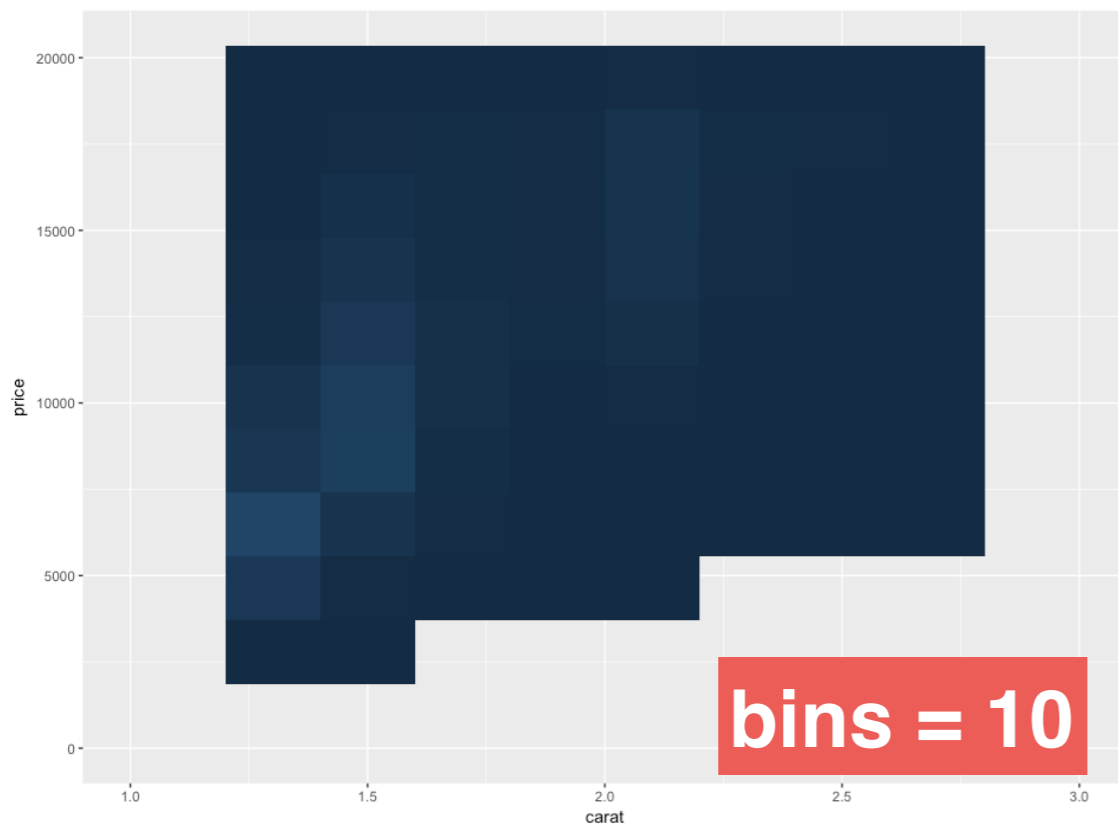
二维直方图

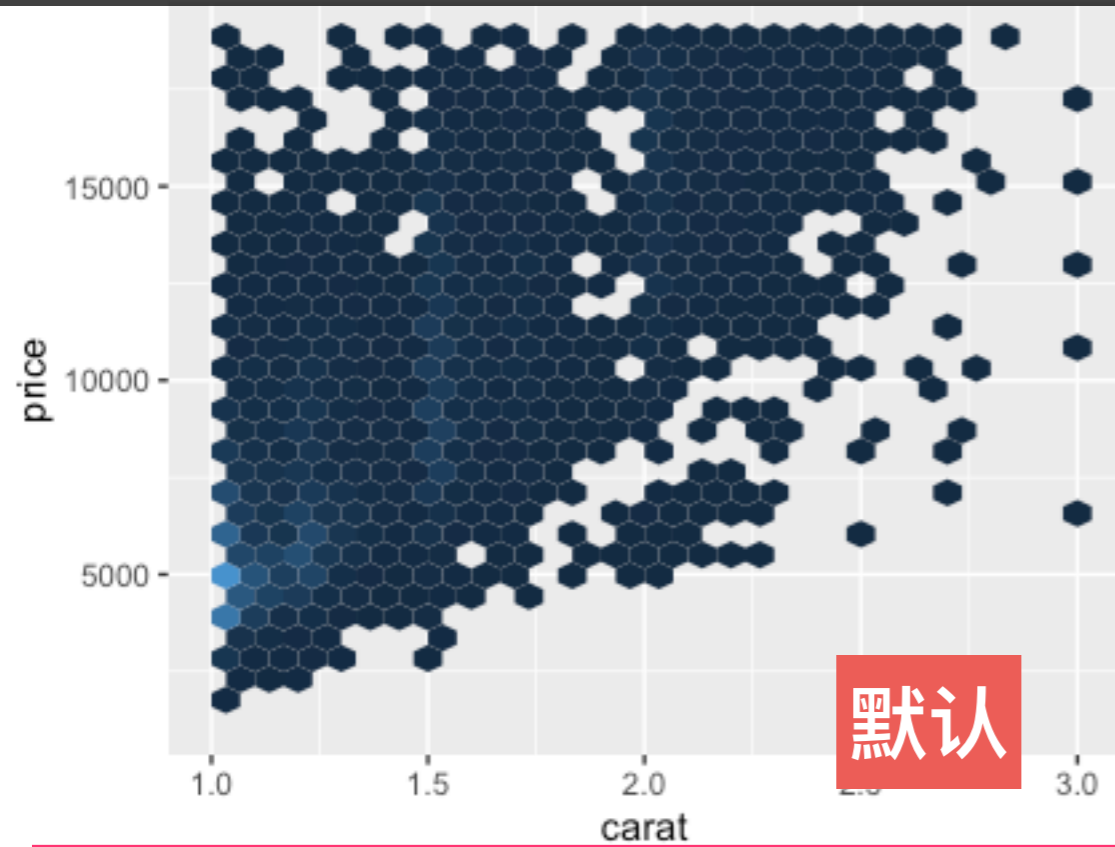
```
> d <- ggplot(diamonds, aes(carat, price)) + xlim(1,3) +  
+ theme(legend.position = "none")  
> d + stat_bin2d()
```

stat_bin2d

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

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





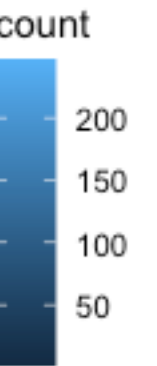
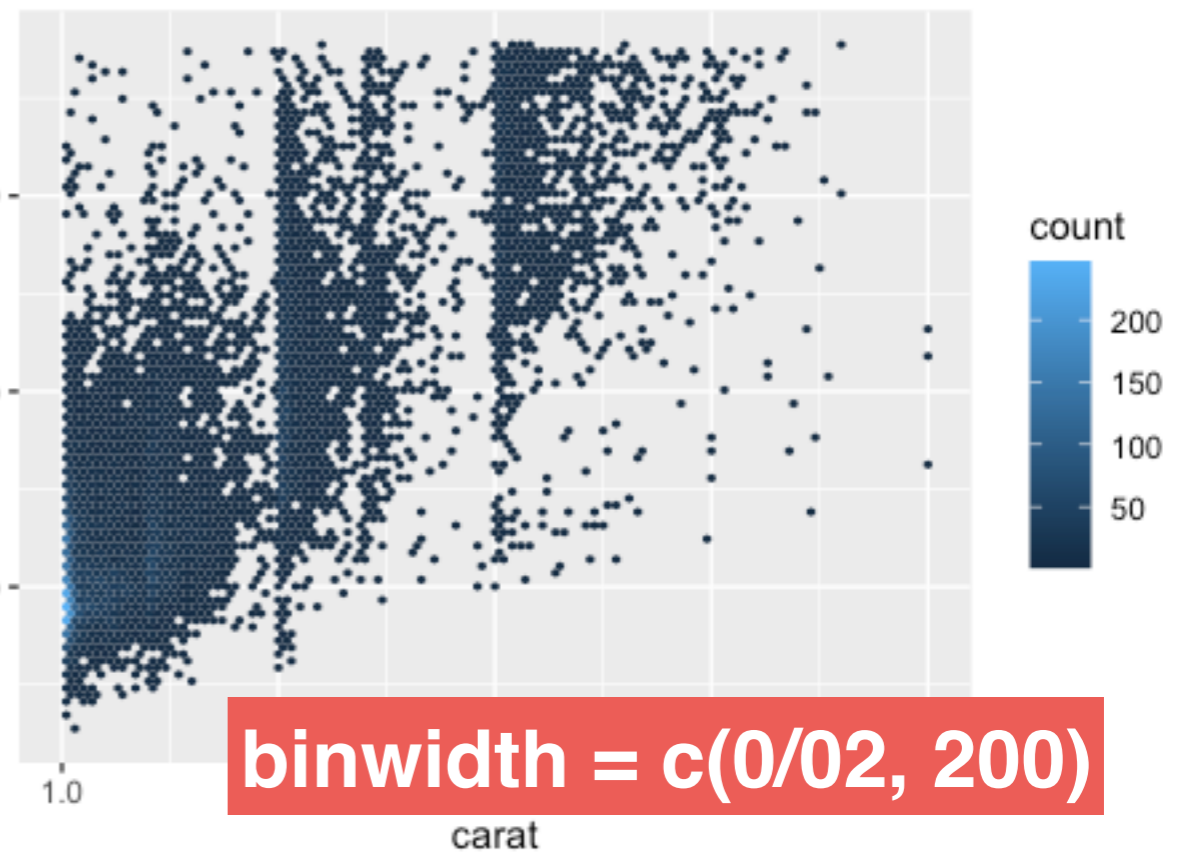
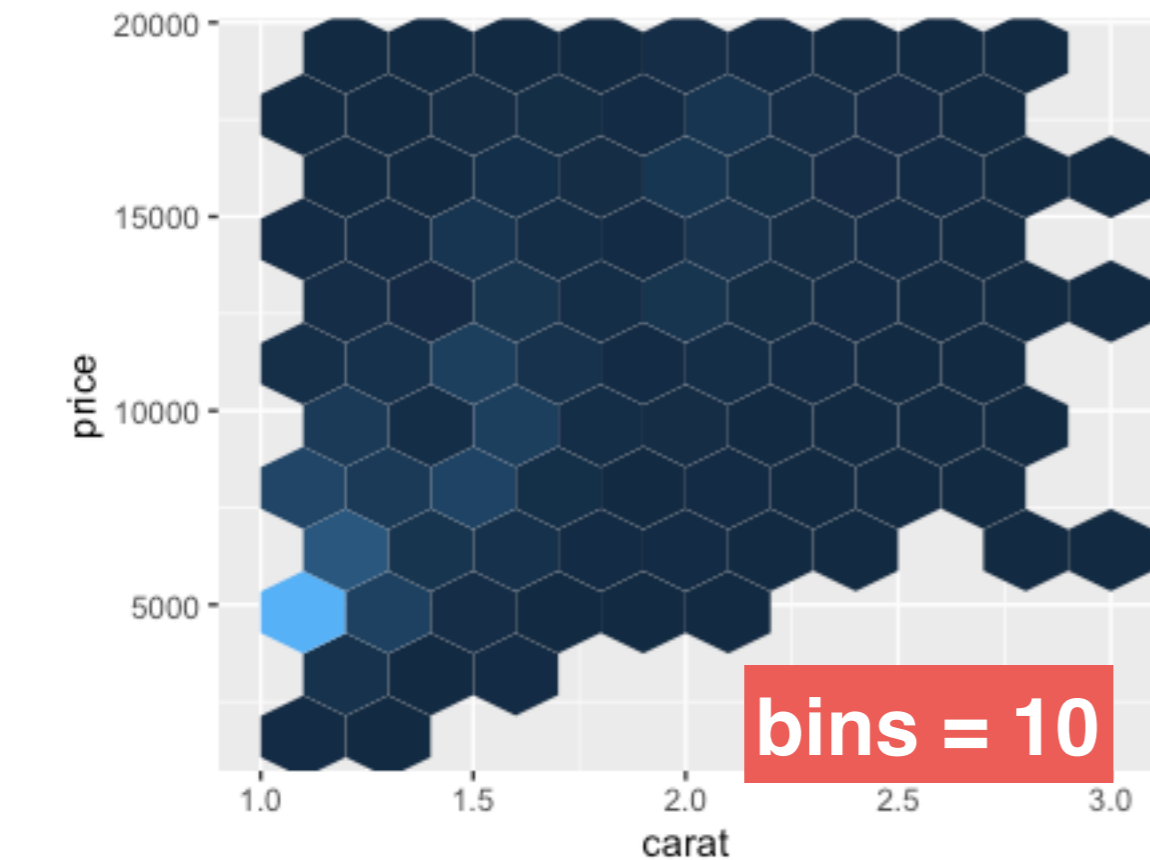
六边形

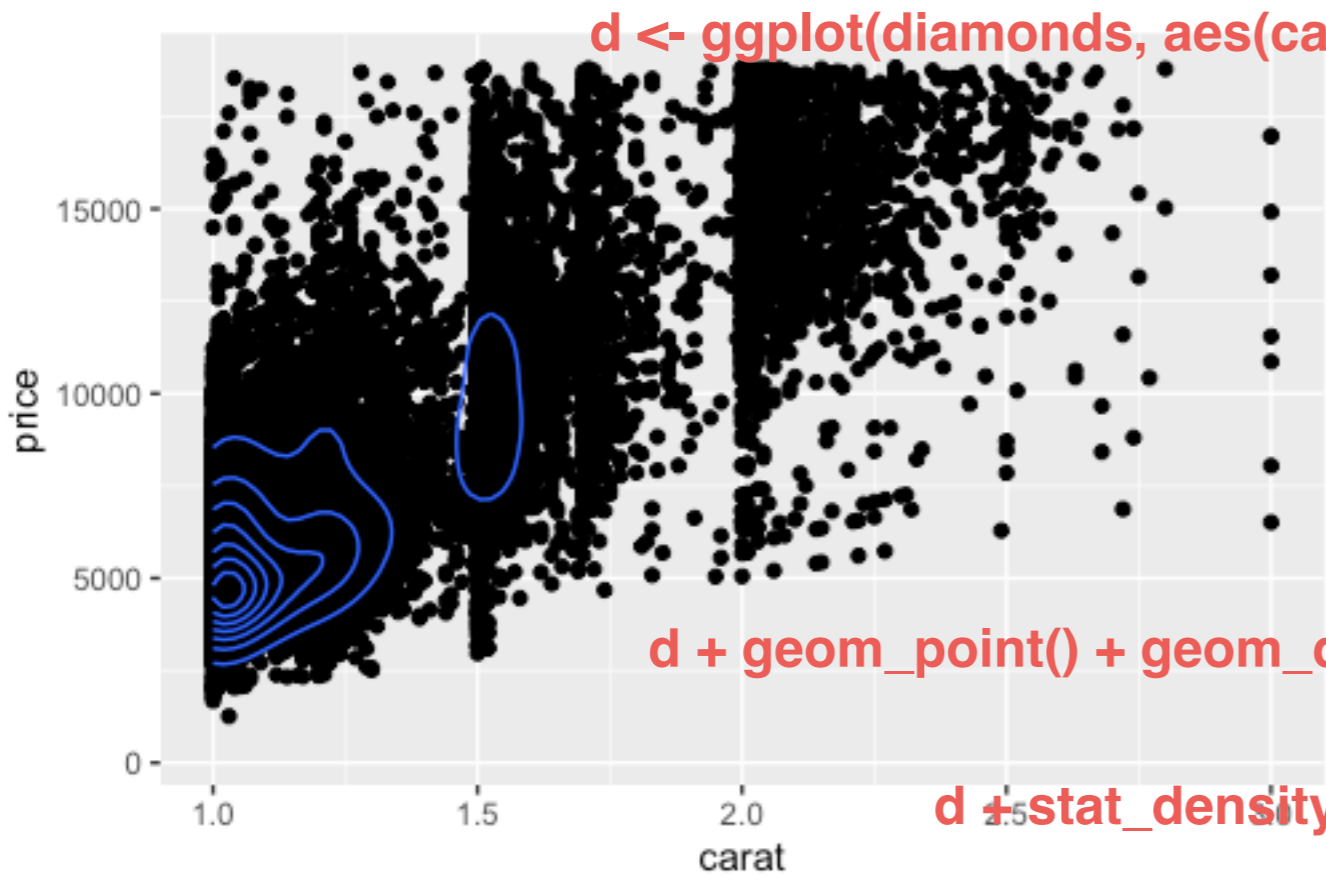
hexbin包

`d + stat_binhex()`

`d + stat_binhex(bins = 10)`

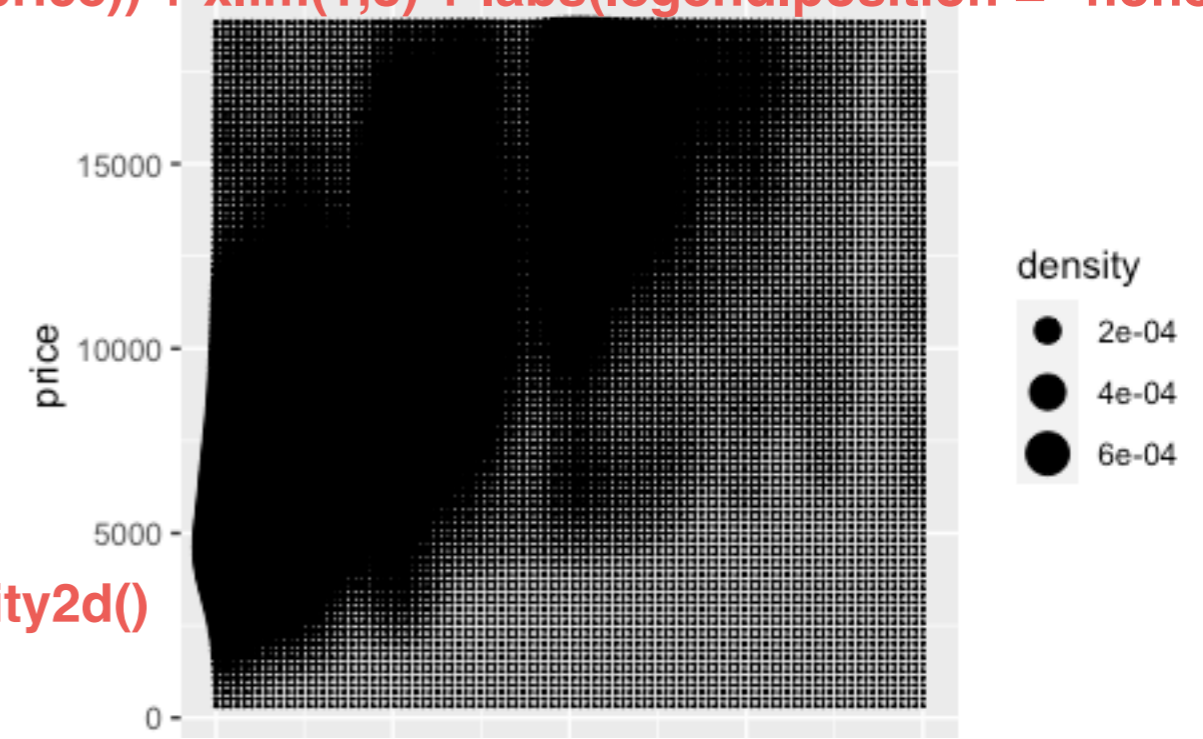
`d + stat_binhex(binwidth=c(0.02, 200))`





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

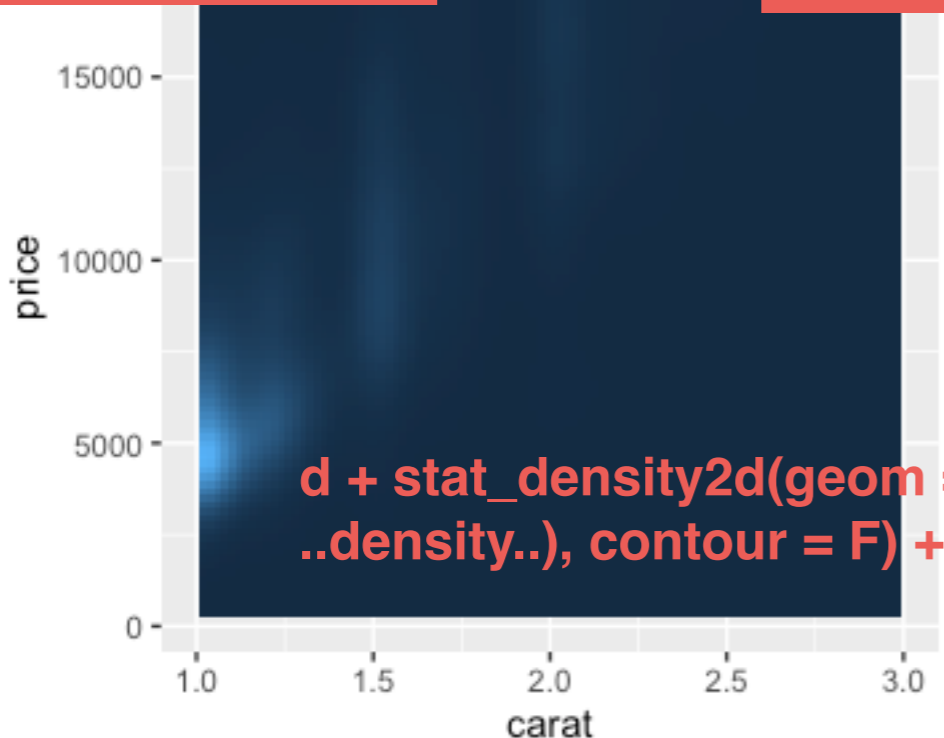
```
d + geom_point() + geom_density2d()
```



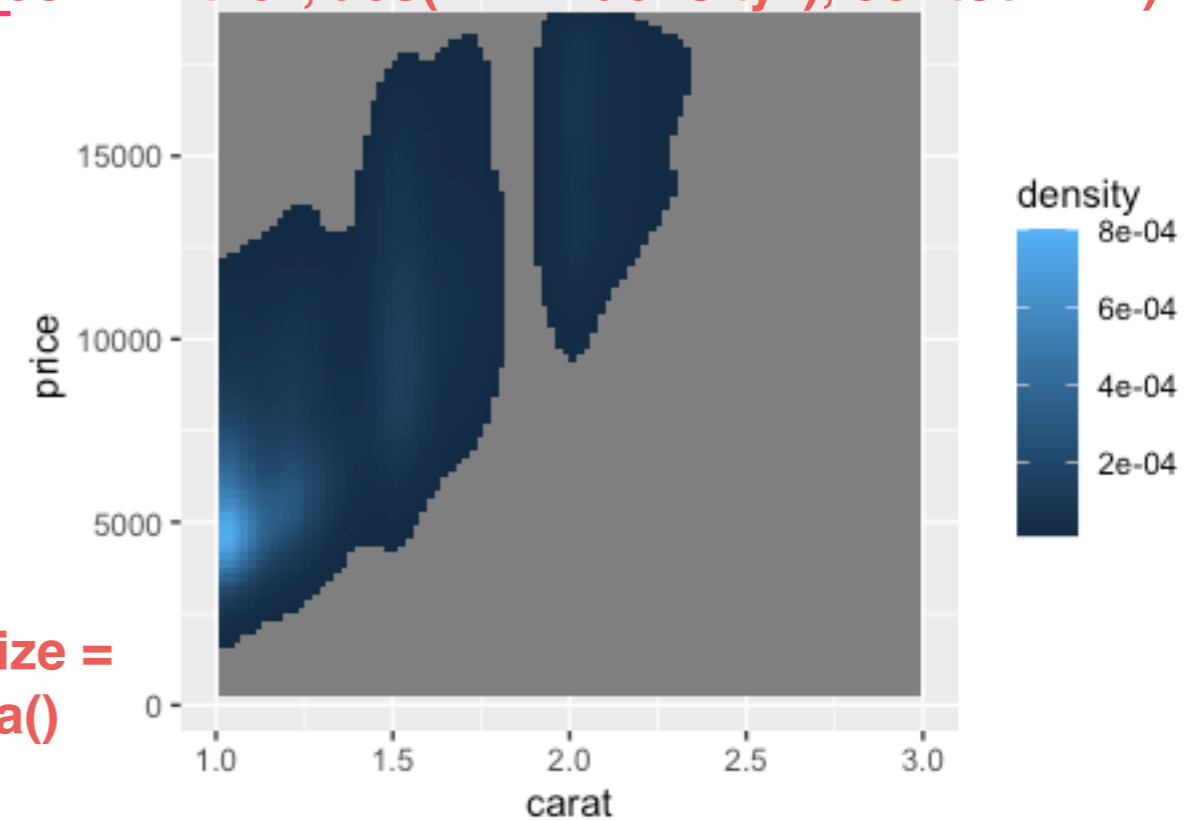
```
d + stat_density2d(geom = "tile", aes(fill = ..density..), contour = F)
```

stat_density2d

二维密度估计



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



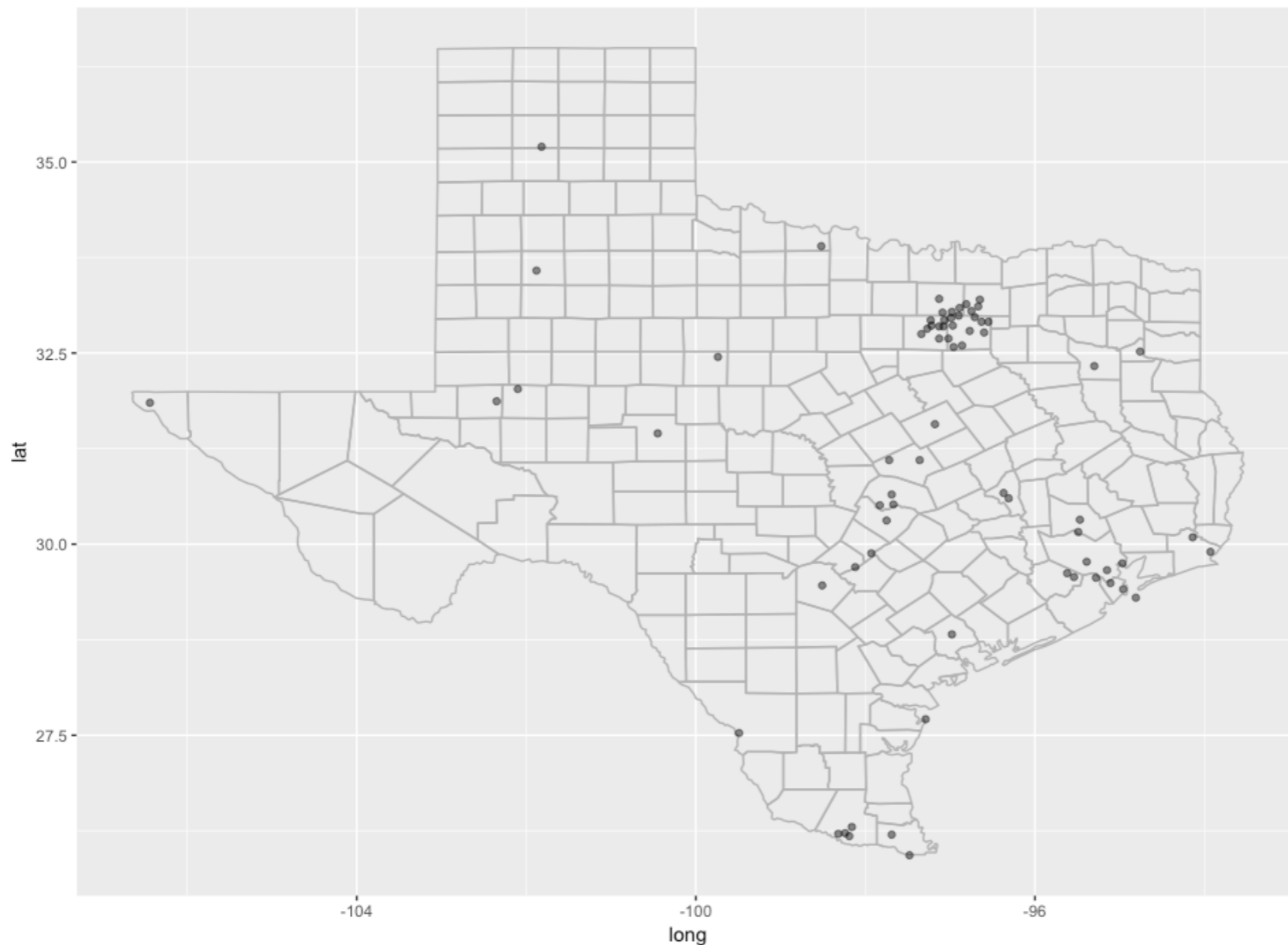
```
last_plot() + scale_fill_gradient(limits = c(1e-5,8e-4))
```

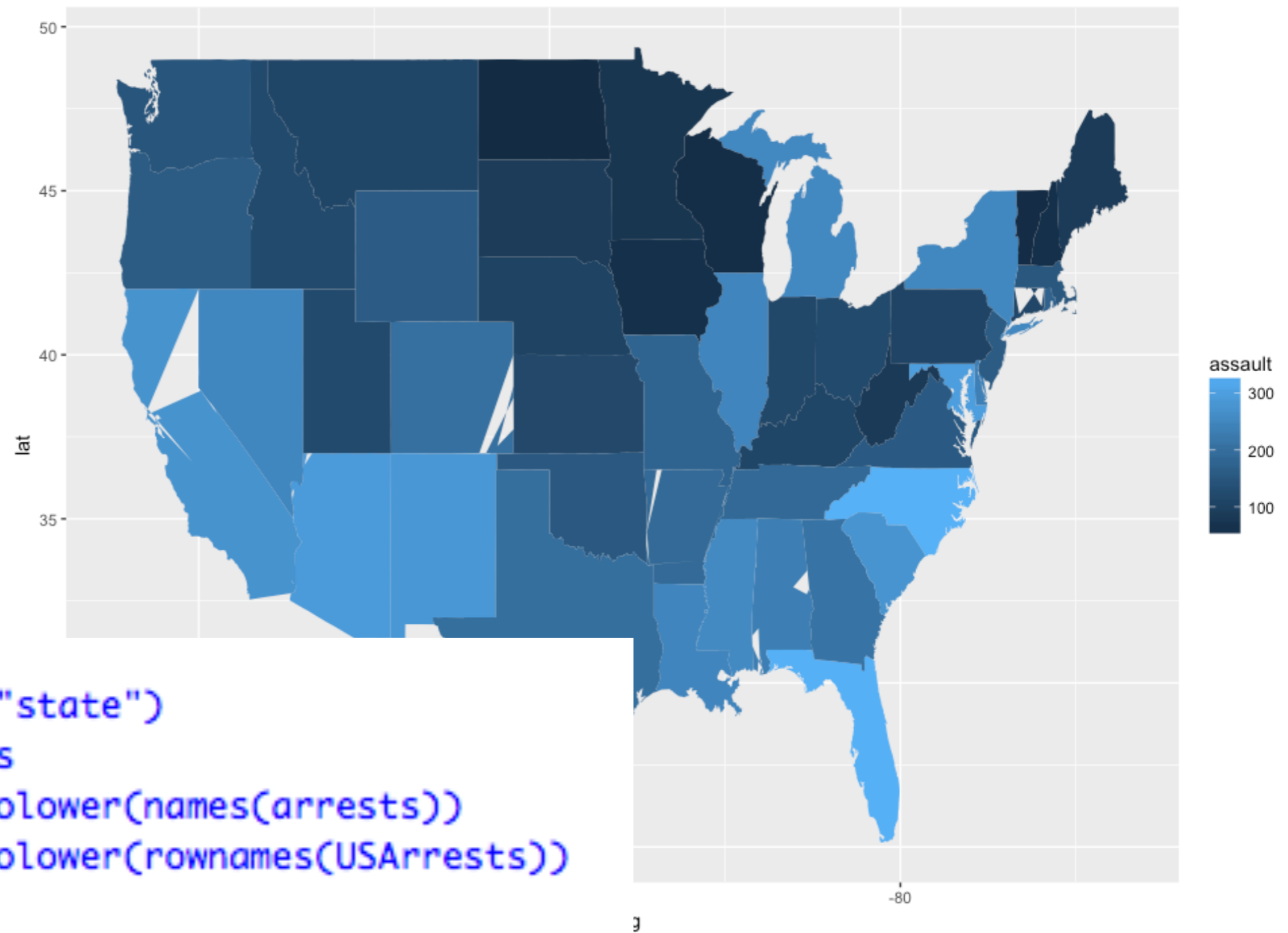
```
> library(maps)
> data(us.cities)
> big_cities <- subset(us.cities, pop > 500000)
> qplot(long, lat, data = big_cities) + borders("state", size = 0.5)
```

borders(): 地图边界



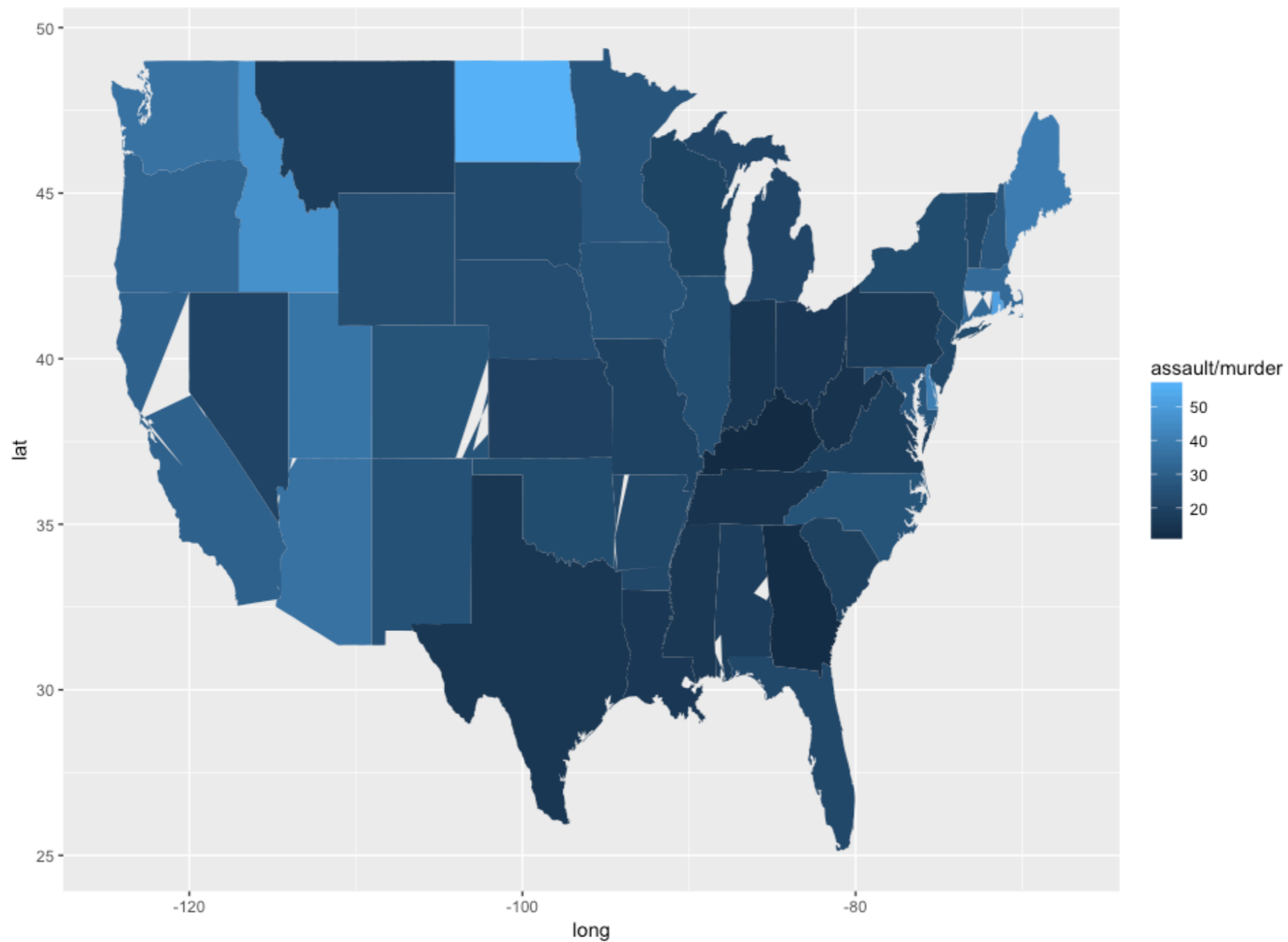
```
> tx_cities <- subset(us.cities, country.etc == "TX")  
> ggplot(tx_cities, aes(long, lat)) +  
+   borders("county", "texas", colour = "grey70") +  
+   geom_point(colour = alpha("black", 0.5))
```



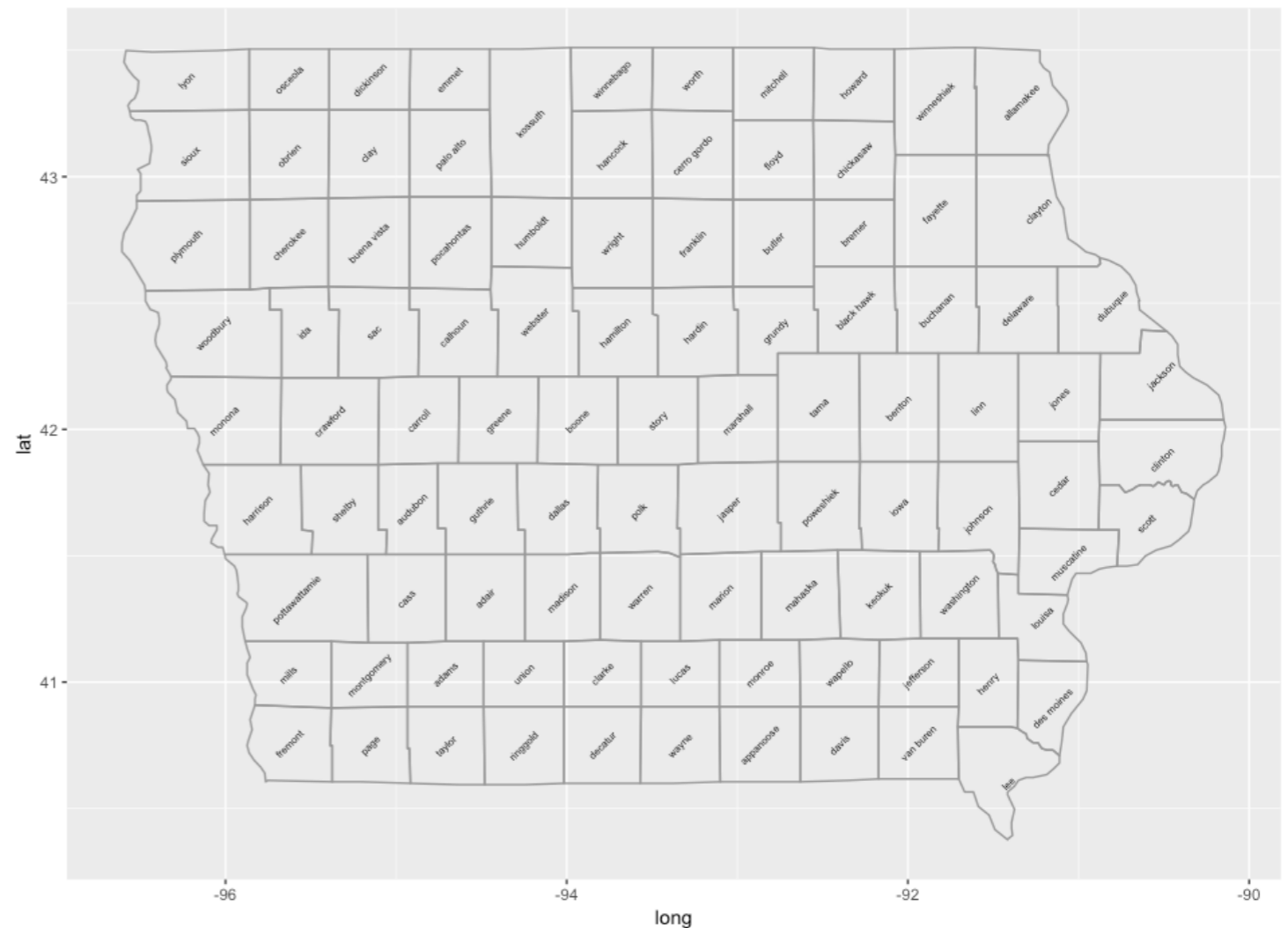


```
> library(maps)
> states <- map_data("state")
> arrests <- USArrests
> names(arrests) <- tolower(names(arrests))
> arrests$region <- tolower(rownames(USArrests))
>
> choro <- merge(states, arrests, by = "region")
> choro <- choro[order(choro$order), ]
> qplot(long, lat, data = choro, group = group,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

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



```
> 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)
```



```

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)

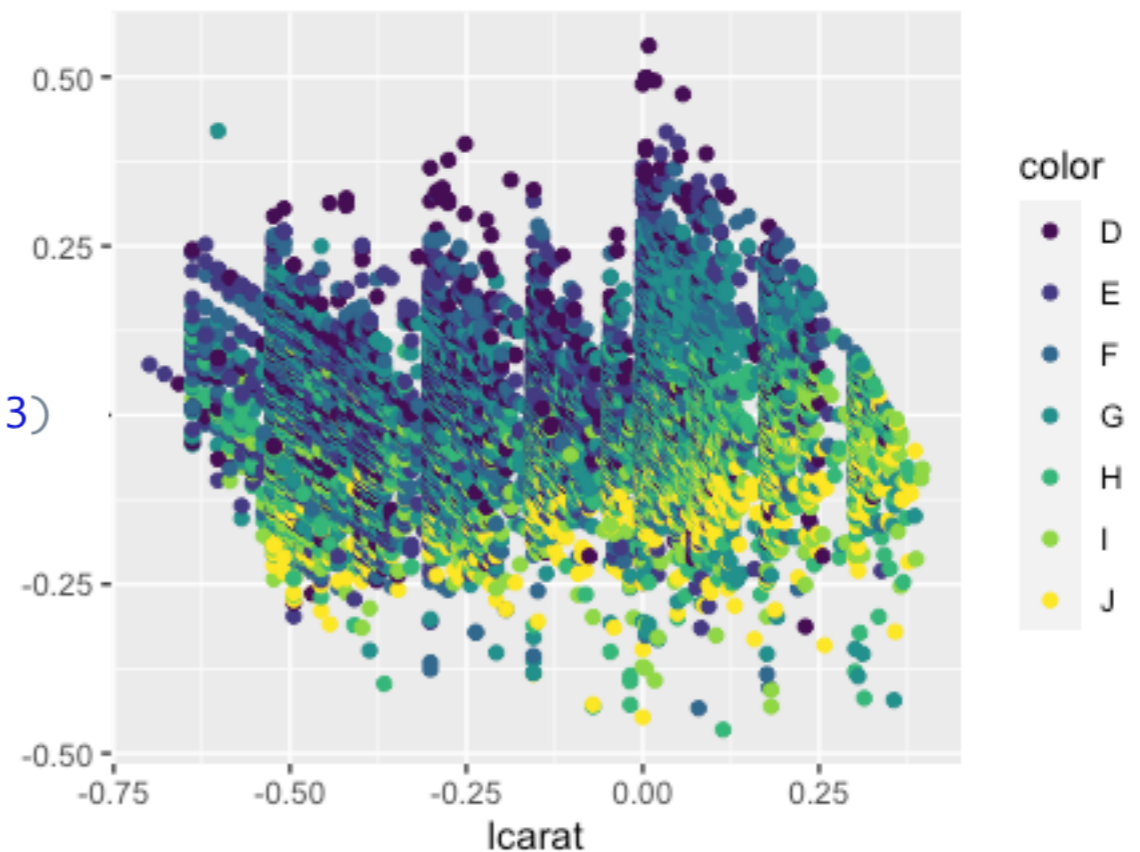
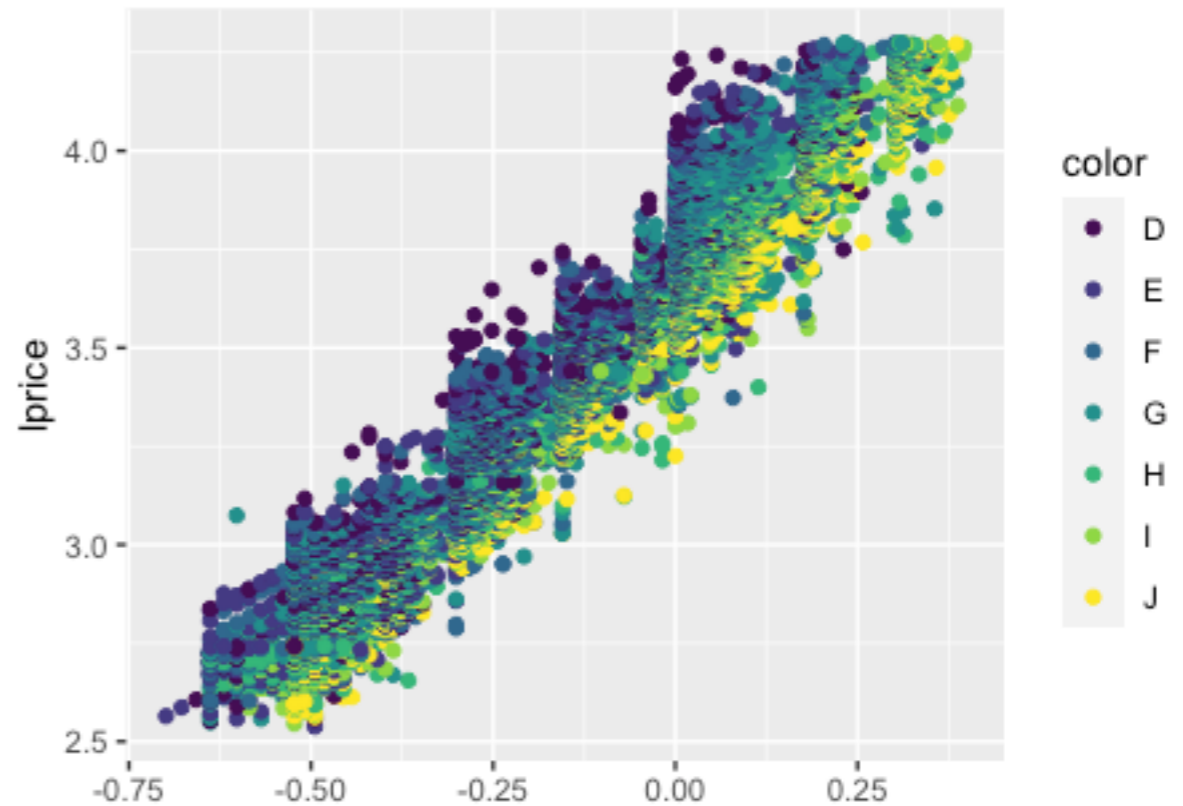
effectdf <- function(...) {
  suppressWarnings(as.data.frame(effect(...)))
}

color <- effectdf("color", mod)
both1 <- effectdf("lcarat:color", mod)

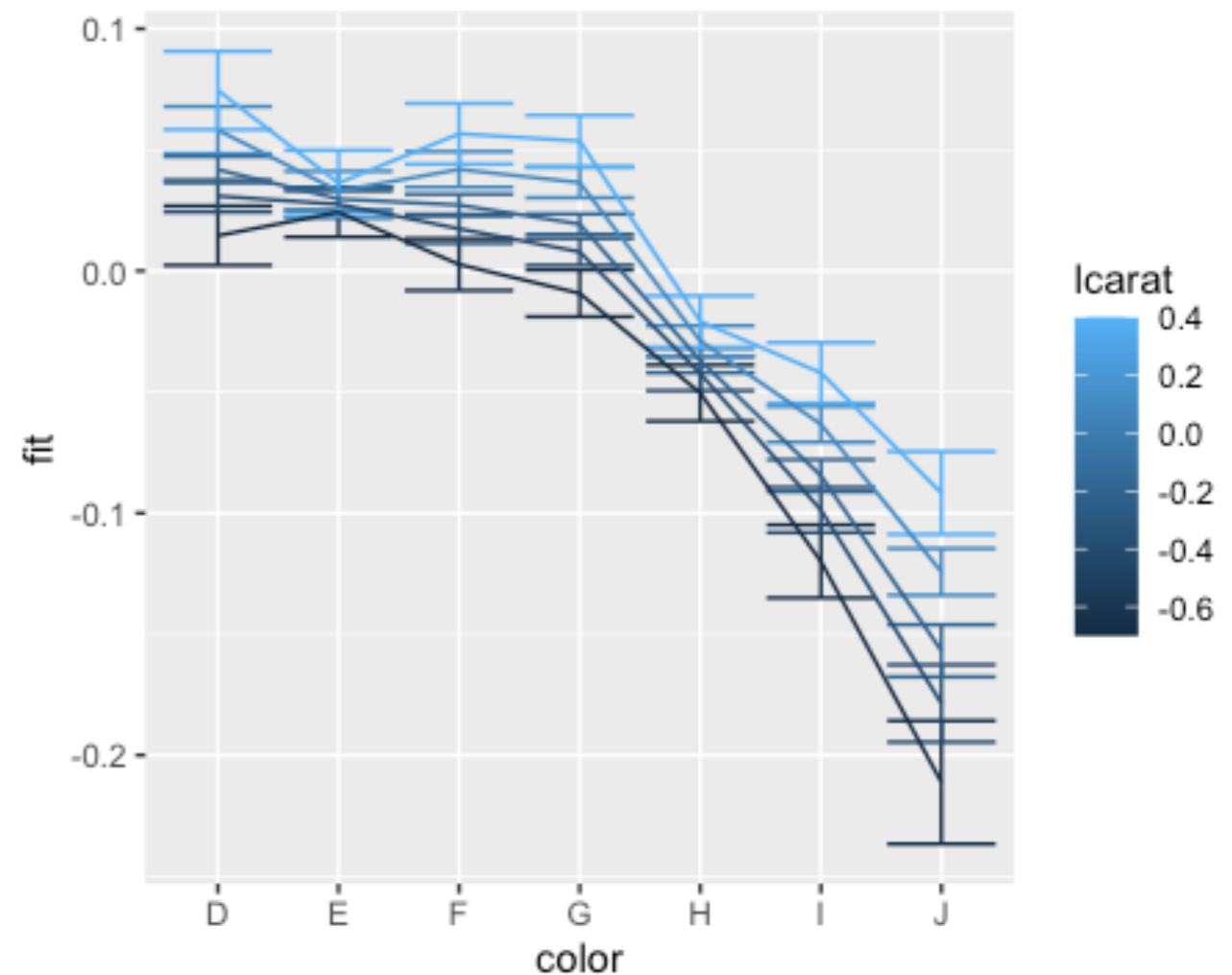
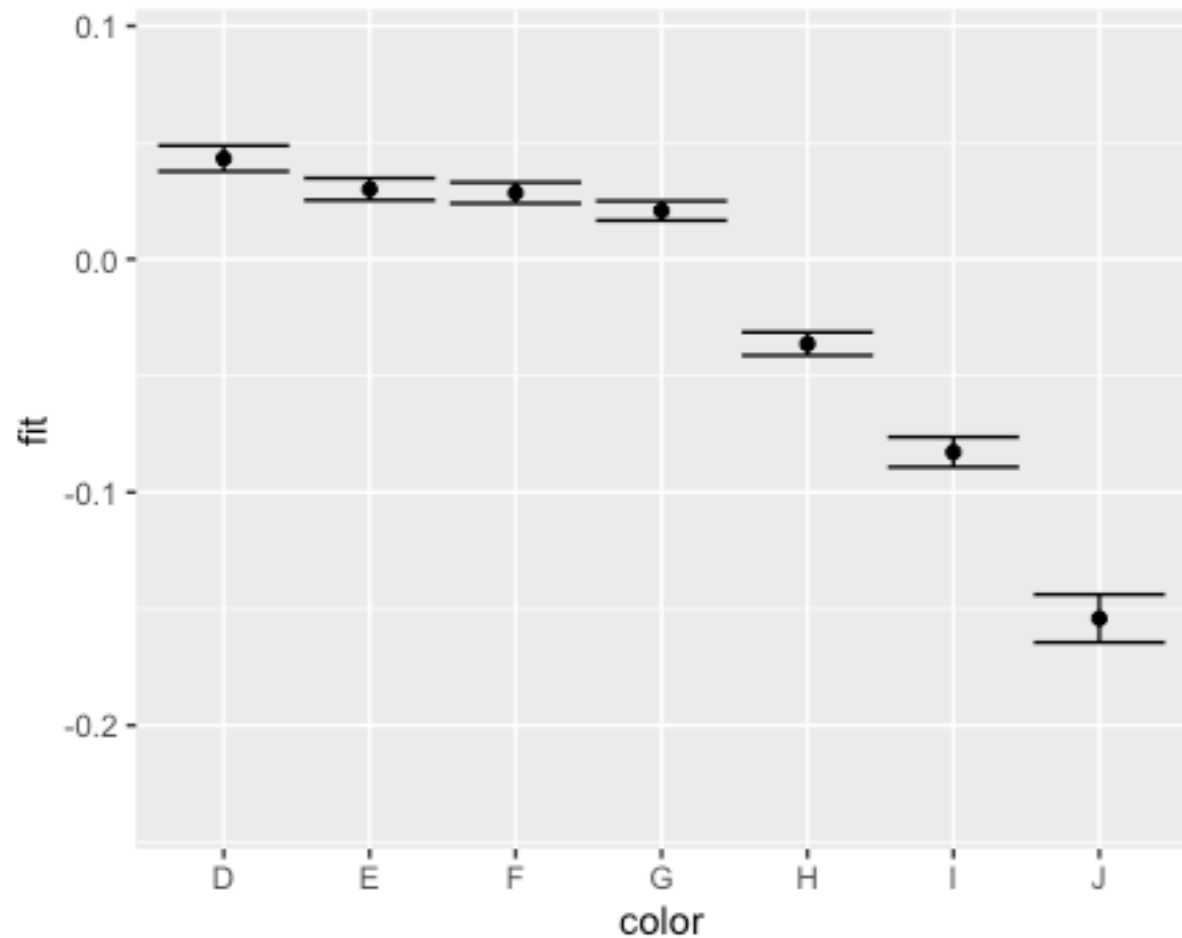
carat <- effectdf("lcarat", mod, default.levels = 50)
both2 <- effectdf("lcarat:color", mod, default.levels = 3)

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

```



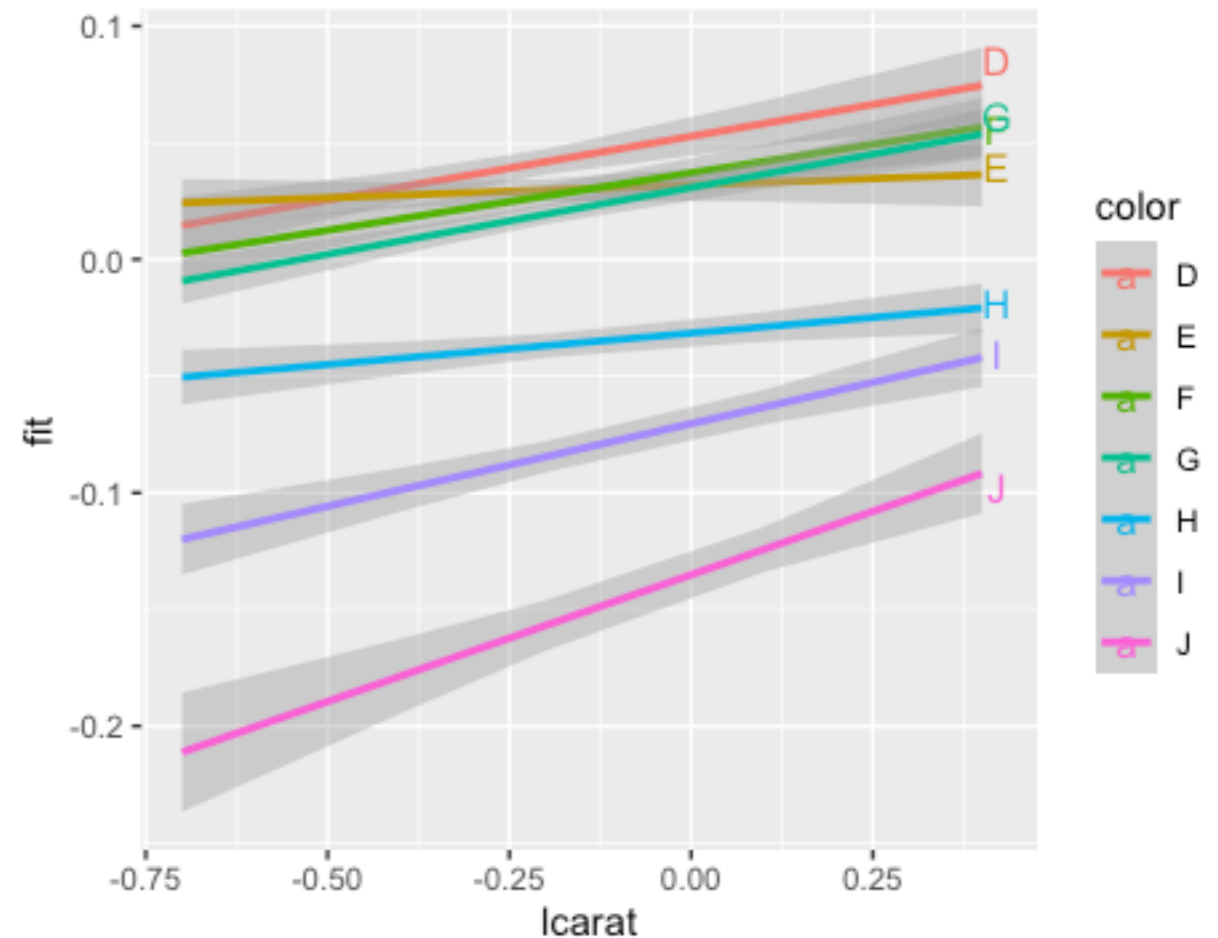
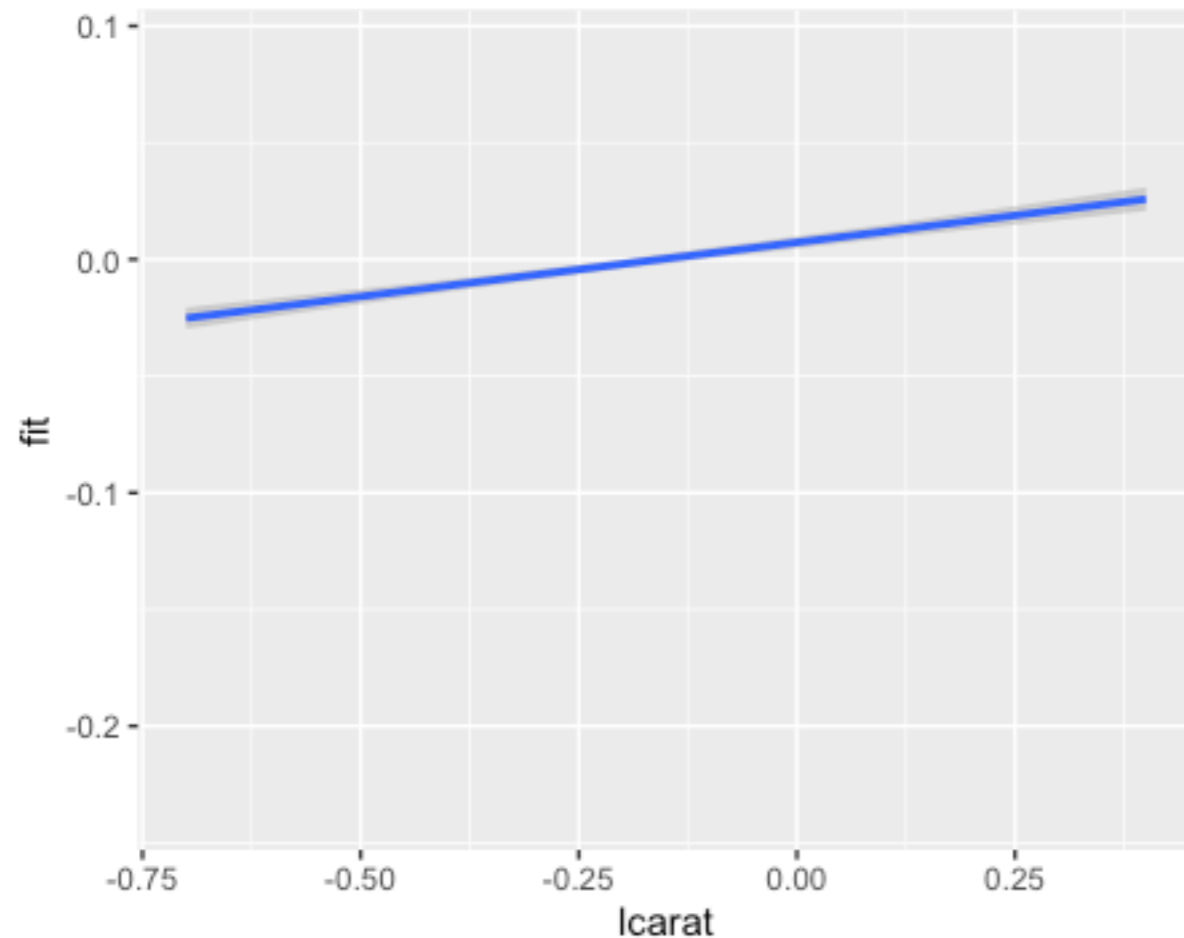
```
fplot <- ggplot(mapping = aes(y = fit, ymin = lower, ymax = upper)) +  
  ylim(range(both2$lower, both2$upper))  
fplot %>% color + aes(x = color) + geom_point() + geom_errorbar()  
fplot %>% both2 +  
  aes(x = color, colour = lcarat, group = interaction(color, lcarat)) +  
  geom_errorbar() + geom_line(aes(group=lcarat))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



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

```
ends <- subset(both1, lcarat == max(lcarat))
```

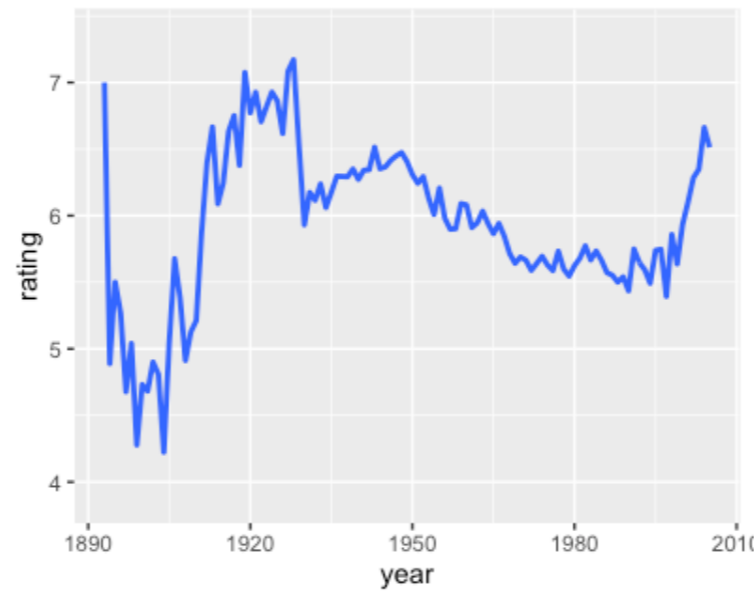
```
fplot %>% both1 + aes(x = lcarat, colour = color) +  
  geom_smooth(stat="identity") +  
  scale_colour_hue() + labs(legend.position = "none") +  
  geom_text(aes(label = color, x = lcarat + 0.02), ends)
```



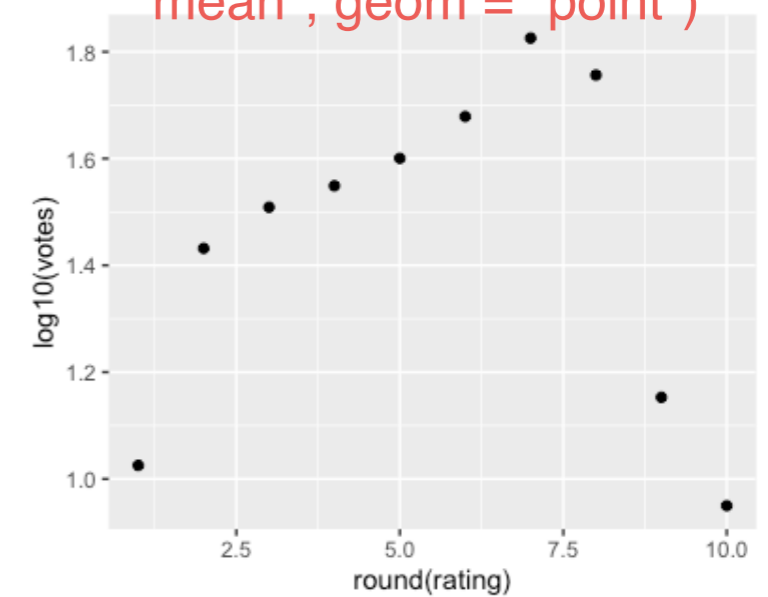
```
m <- ggplot(movies, aes(year, rating))  
m + stat_summary(fun = median,  
geom = "line")
```



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

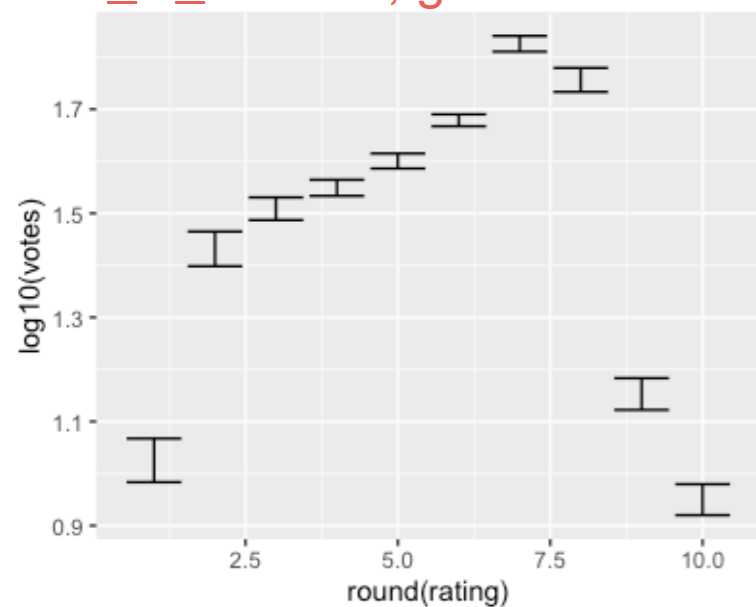


```
m2 + stat_summary(fun = "mean", geom = "point")
```

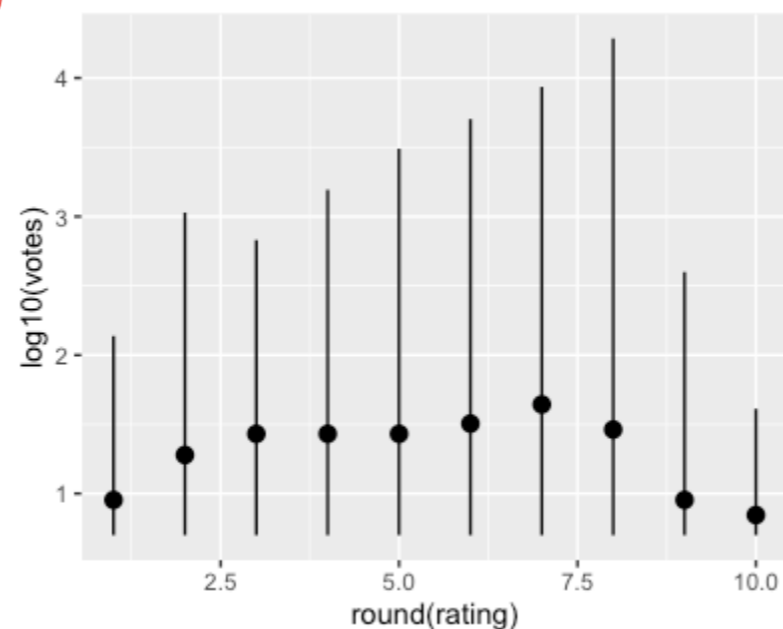


```
m2 <- ggplot(movies, aes(round(rating), log10(votes)))
```

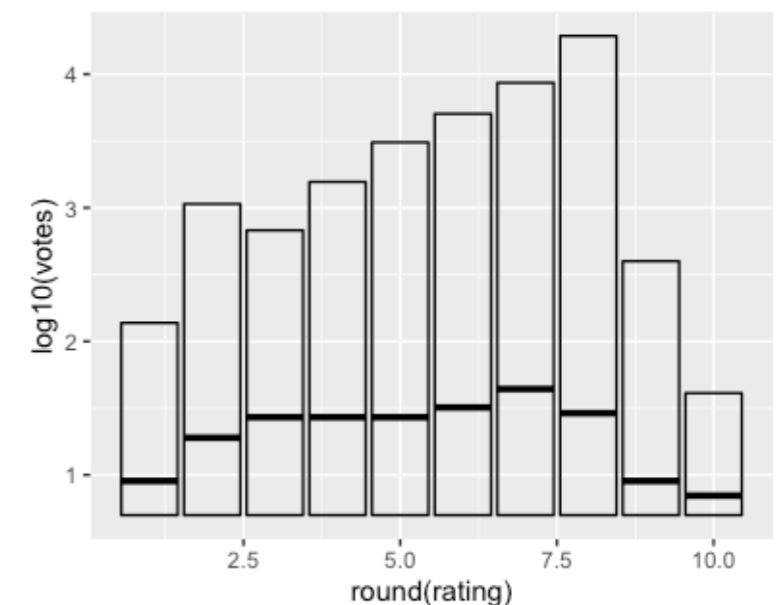
```
m2 + stat_summary(fun.data = "mean_cl_normal", geom = "errorbar")
```

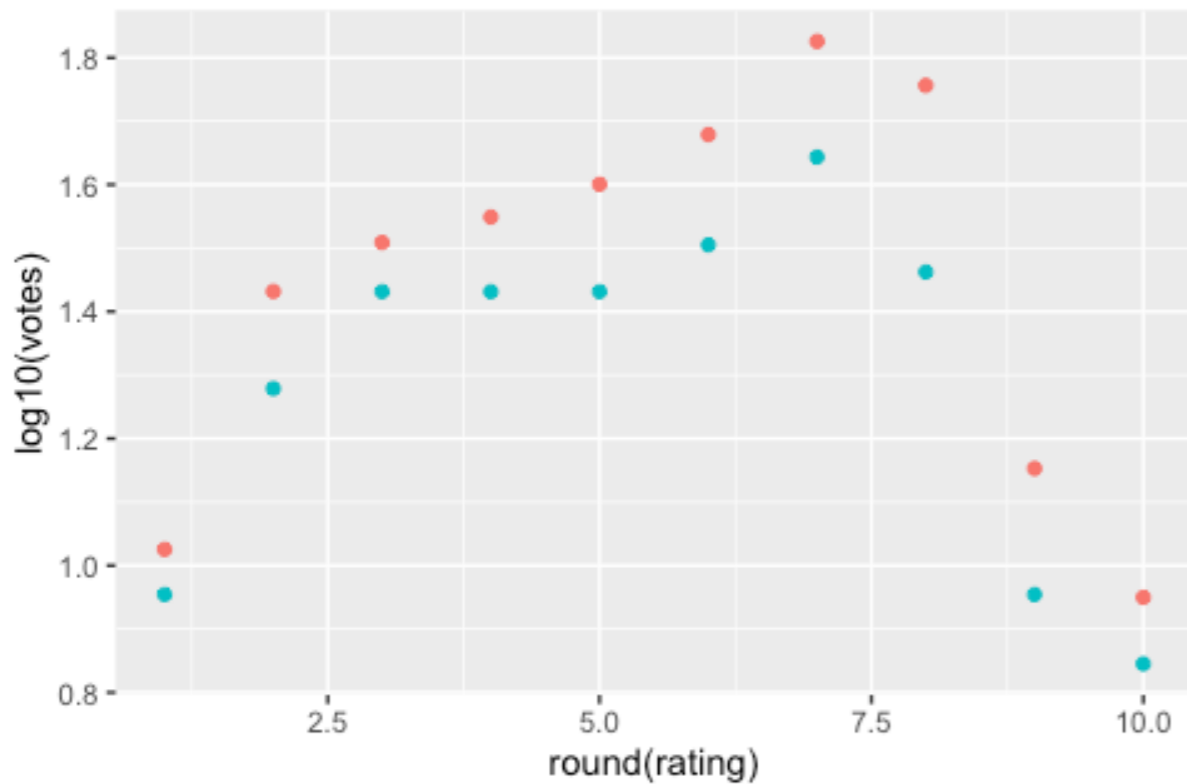


```
m2 + stat_summary(fun.data = "median_hilow", geom = "pointrange")
```



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



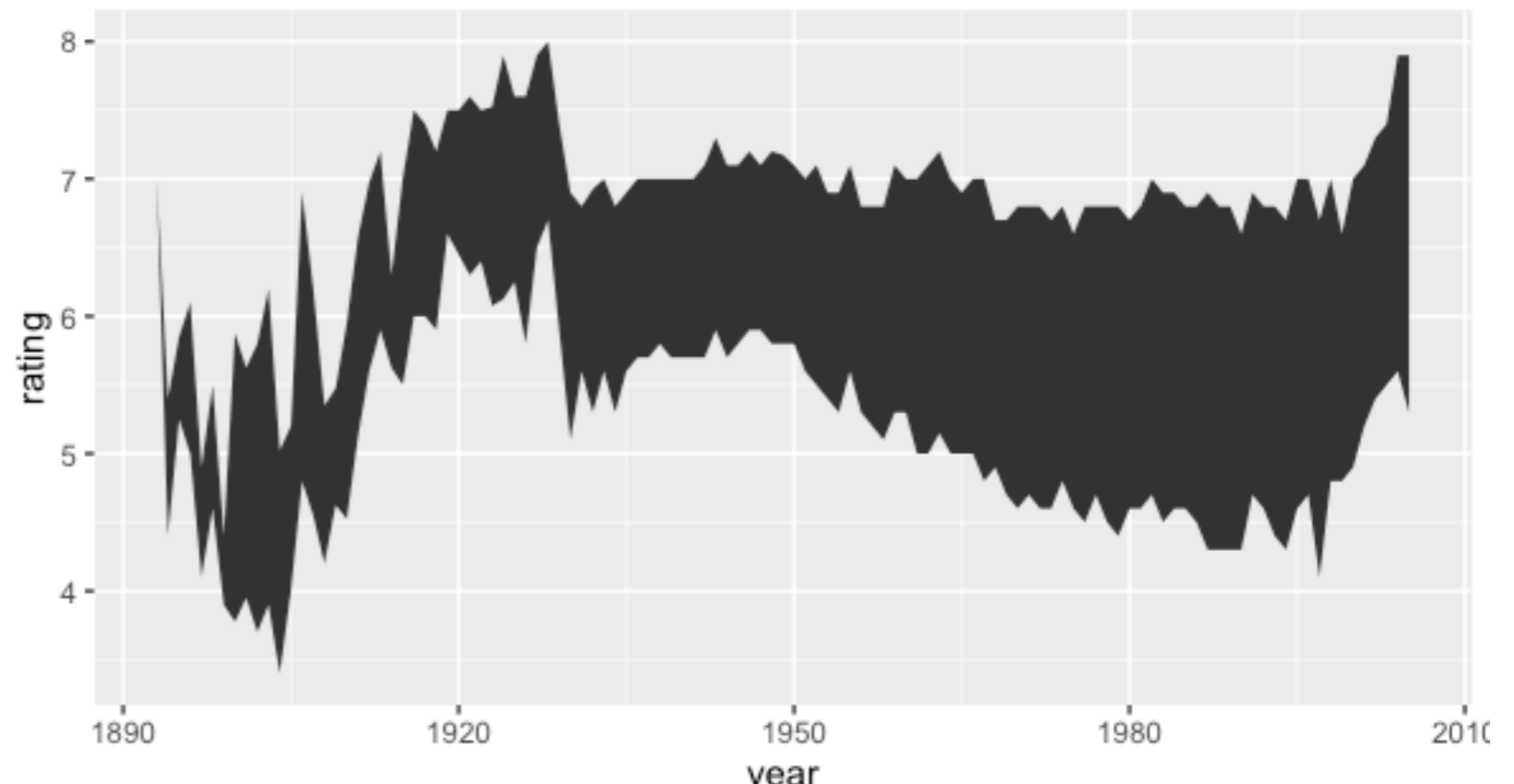


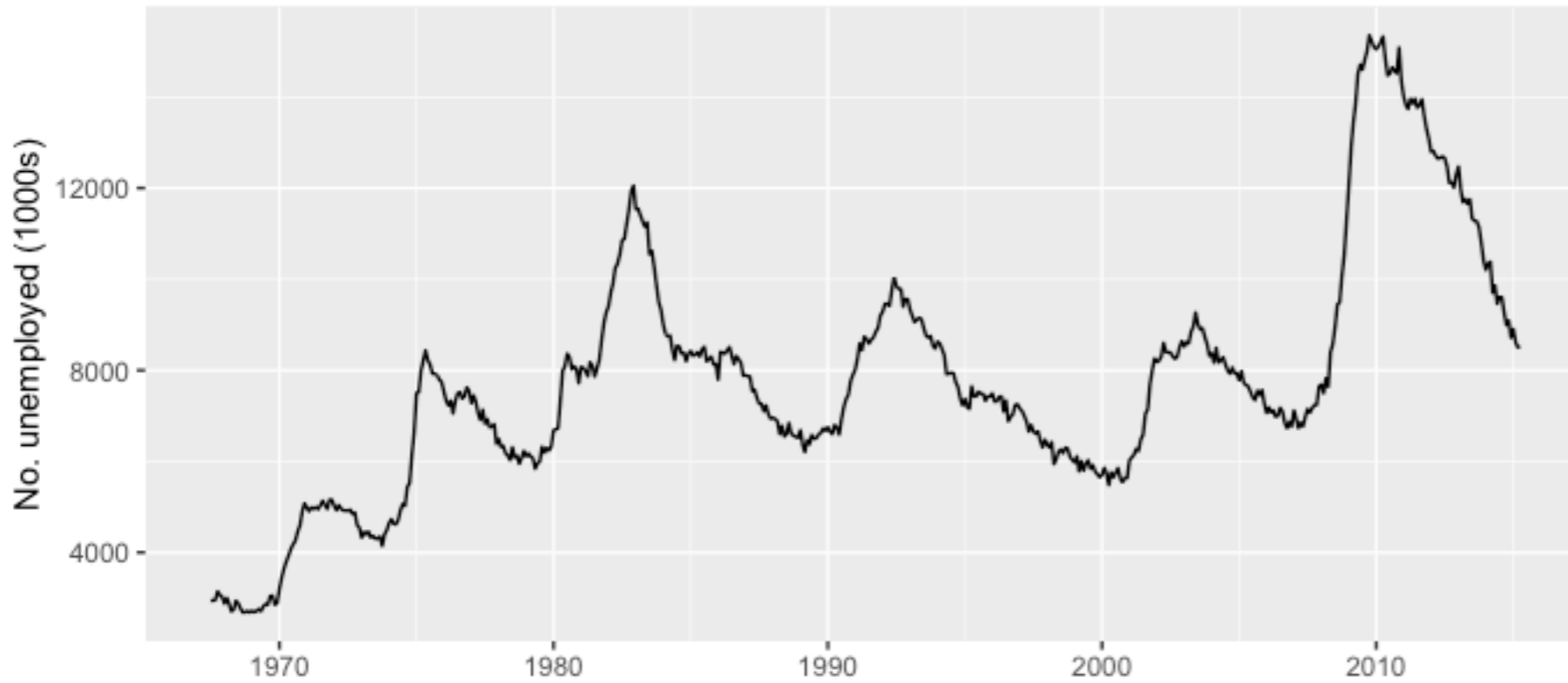
Mean

- raw
- trimmed

```
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")
```

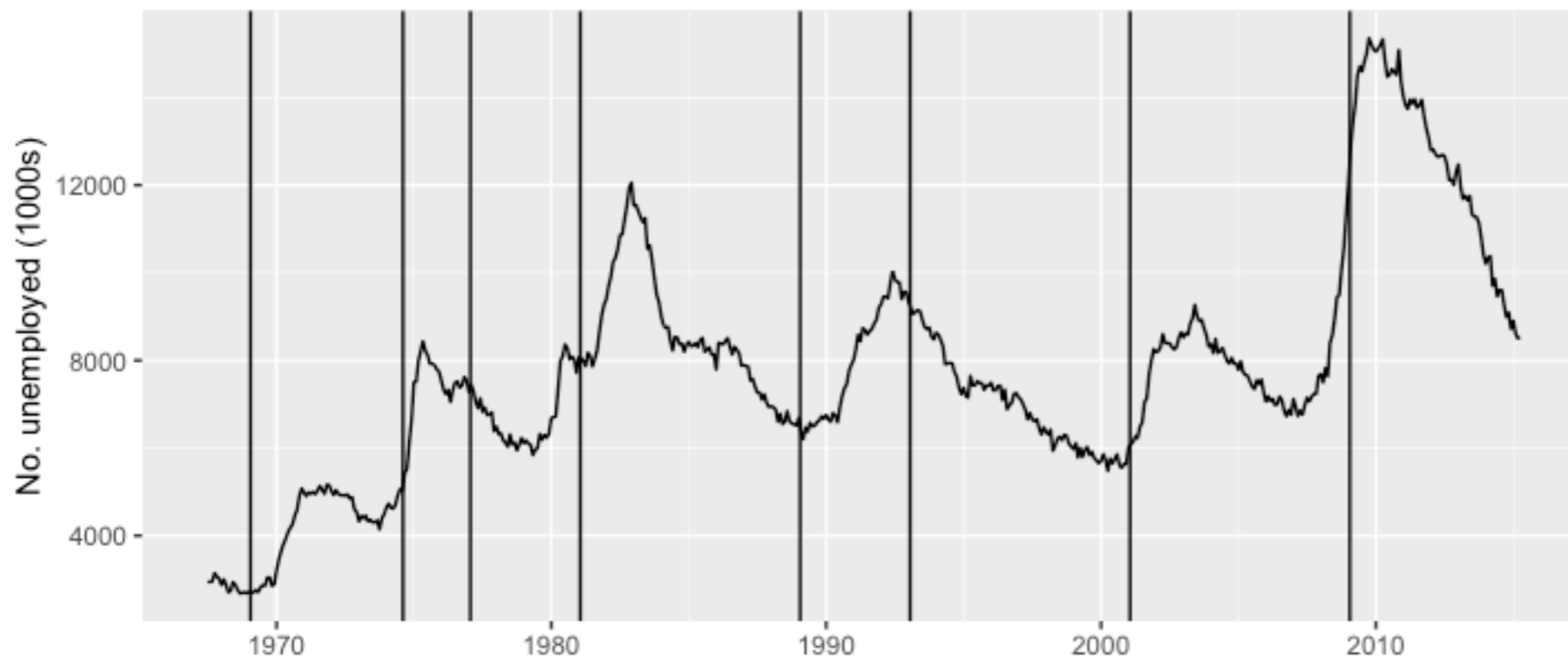
```
iqr <- function(x, ...) {
  qs <- quantile(as.numeric(x), c(0.25,
0.75), na.rm = T)
  names(qs) <- c("ymin", "ymax")
  qs
}
m + stat_summary(fun.data = "iqr",
geom="ribbon")
```





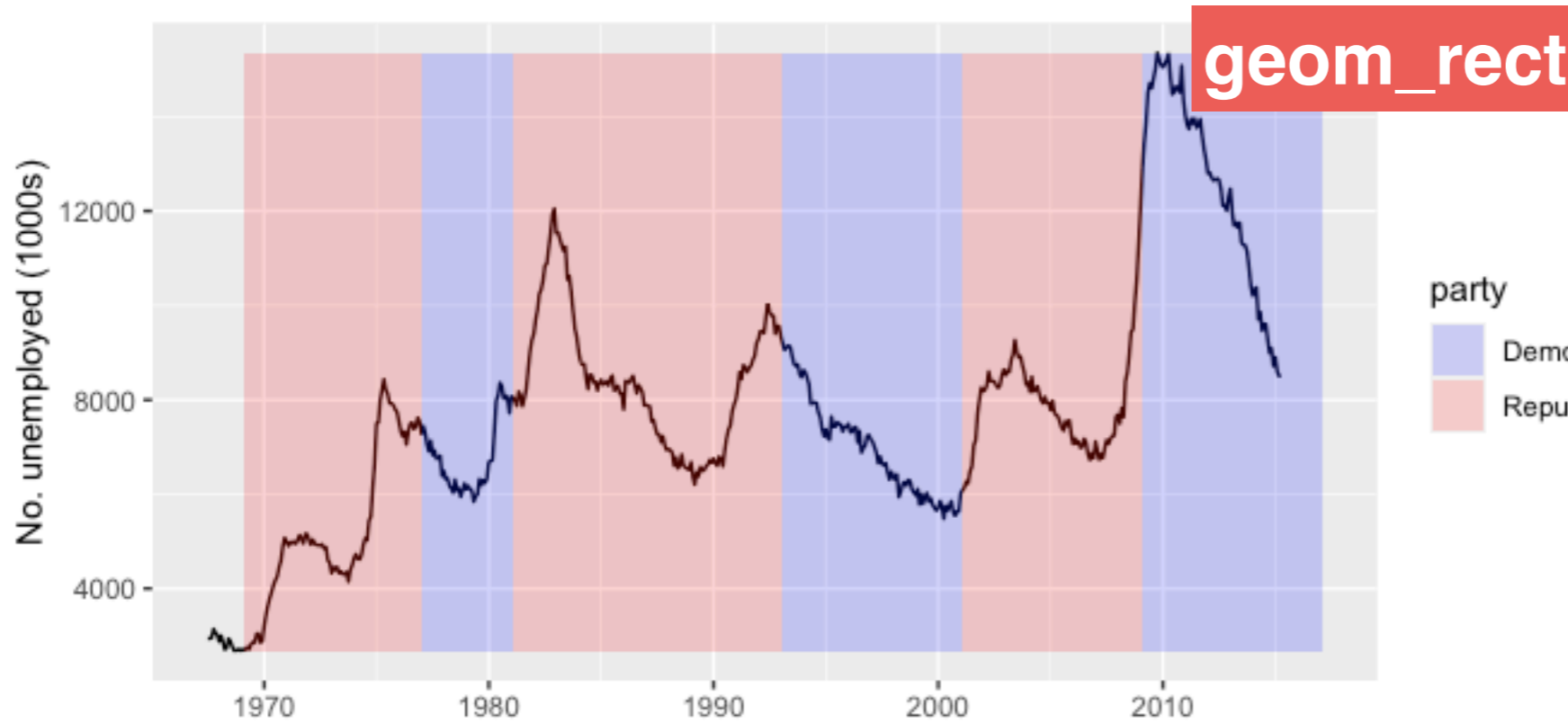
```
(unemp <- qplot(date,  
unemploy,  
data=economics,  
geom="line",  
xlab = "",  
ylab = "No. unemployed  
(1000s)"))
```

geom_vline

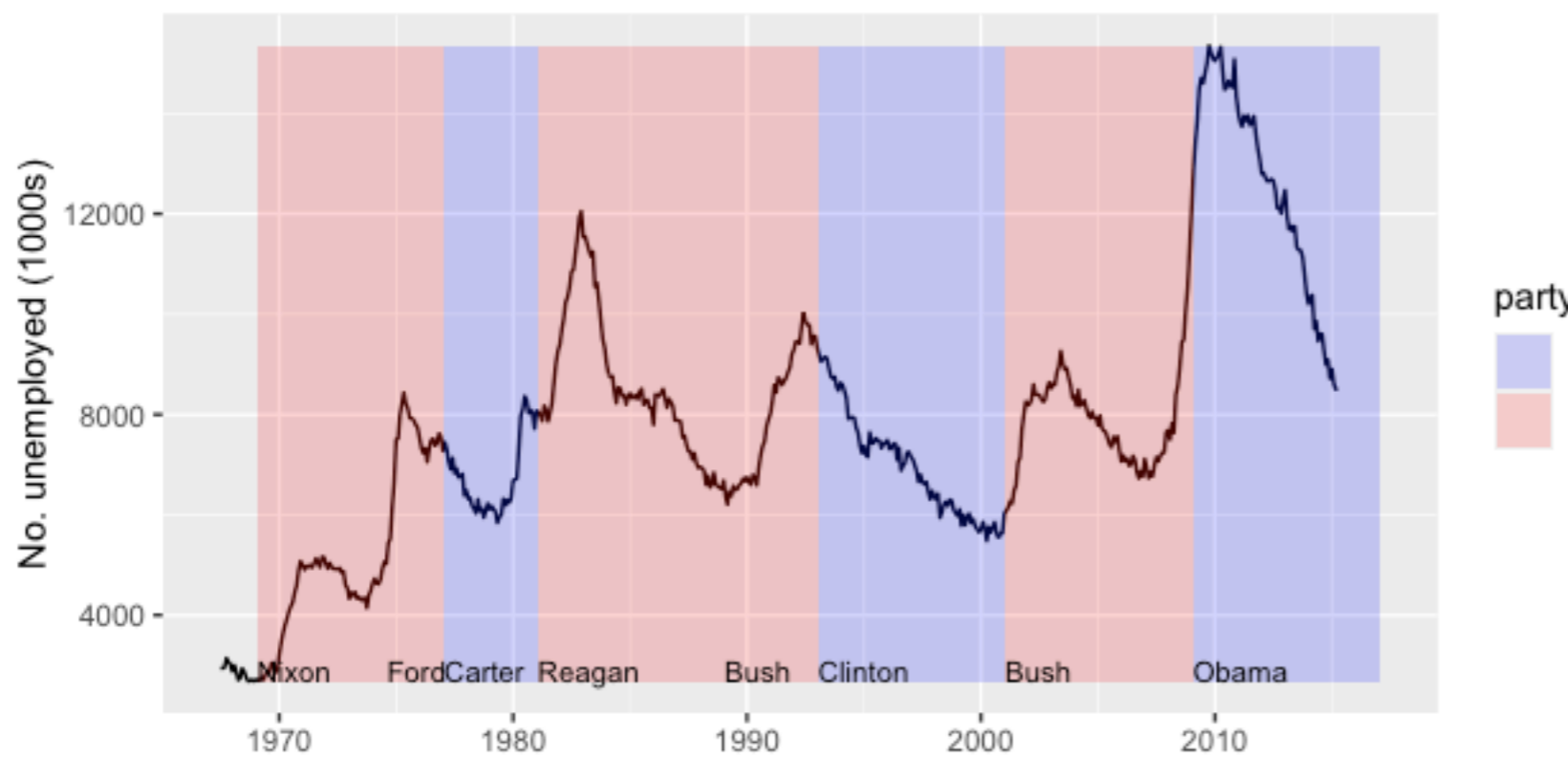


```
presidential <- presidential[-  
(1:3), ]
```

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



```
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))
```

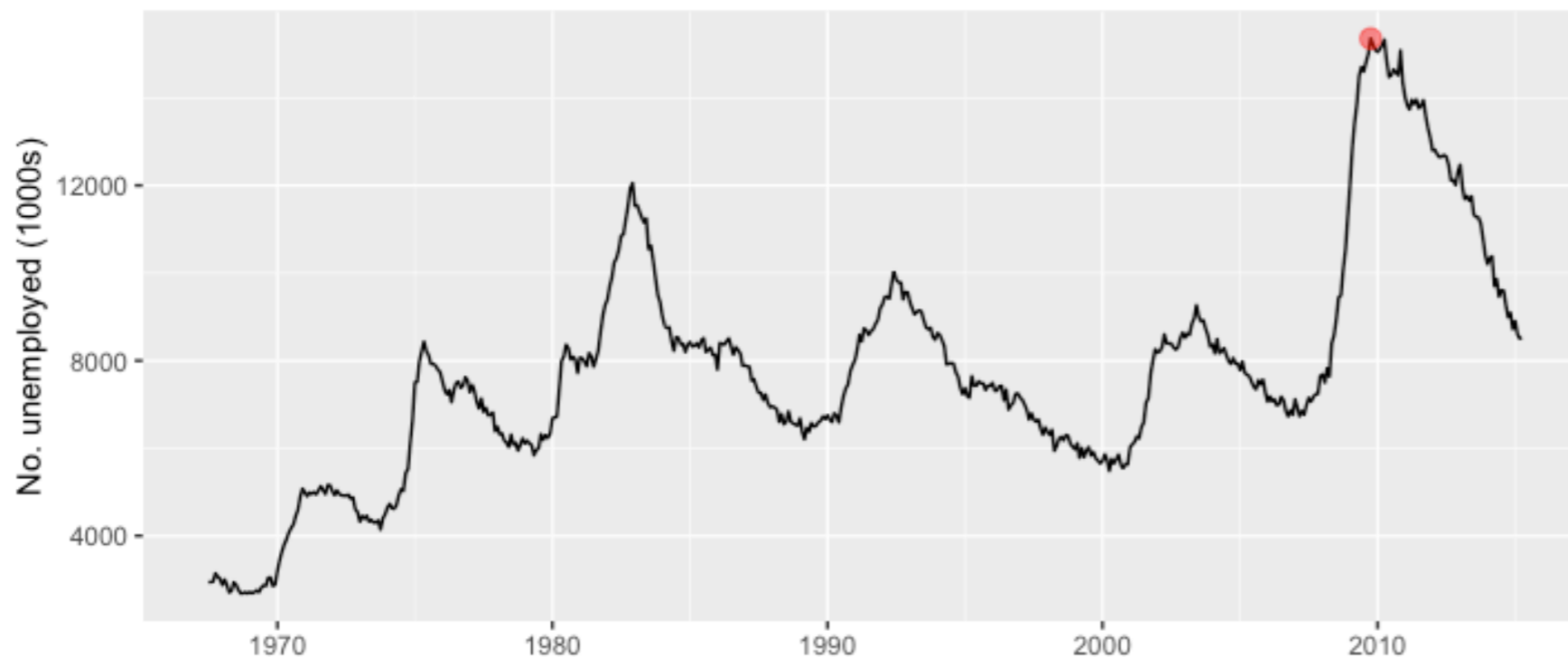


```
last_plot() +
  geom_text(aes(x = start, y =
  yrng[1], label = name),
  data = presidential, size = 3,
  hjust = 0, vjust = 0)
```

geom_text

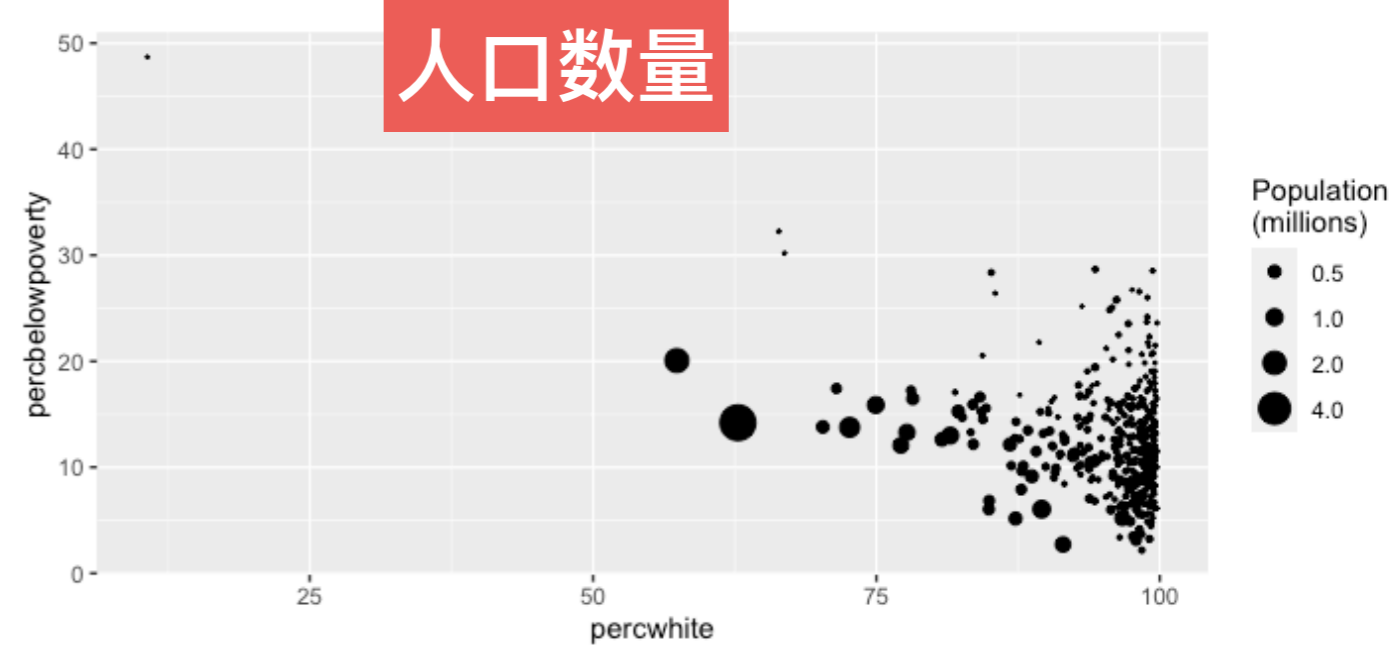
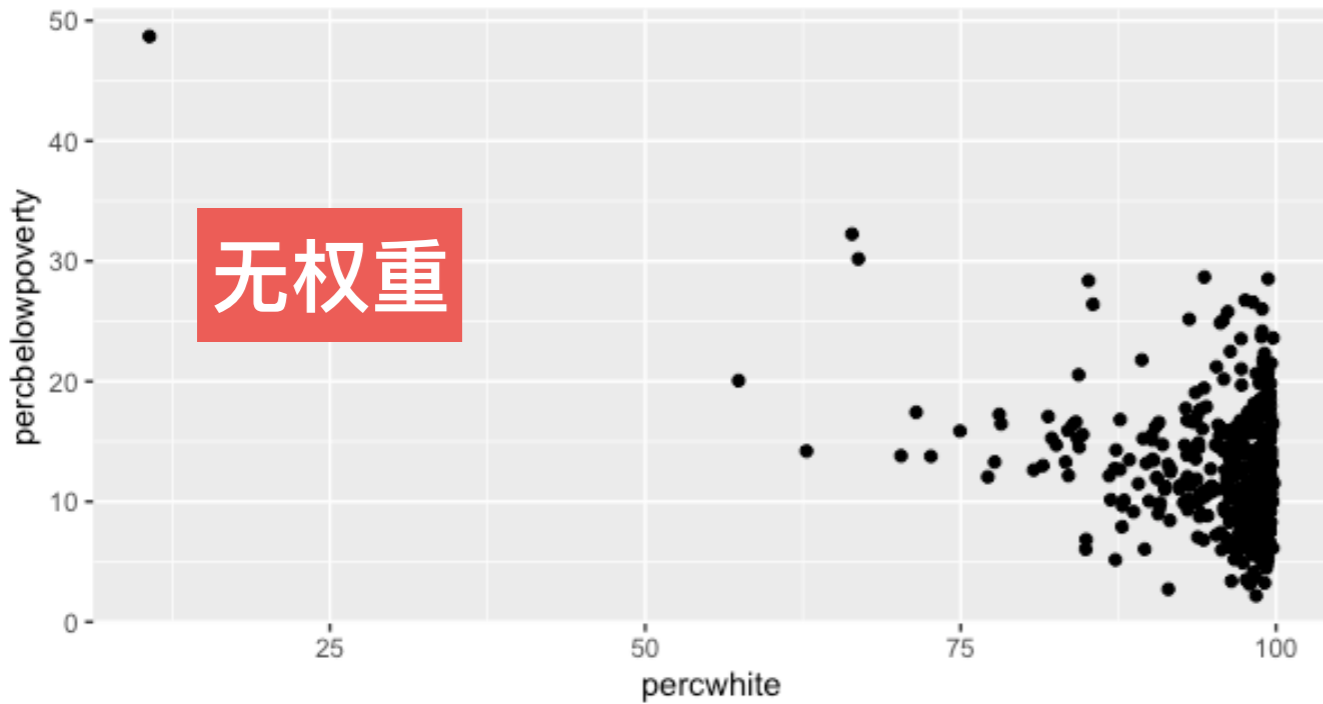


```
caption <-  
paste(strwrap("Unemployment rates in the US have  
varied a lot over the  
years", 40), collapse="\n")  
unemp + geom_text(aes(x,  
y, label = caption),  
data = data.frame(x =  
xrng[2], y = yrng[2]),  
hjust = 1, vjust = 1, size = 4)
```



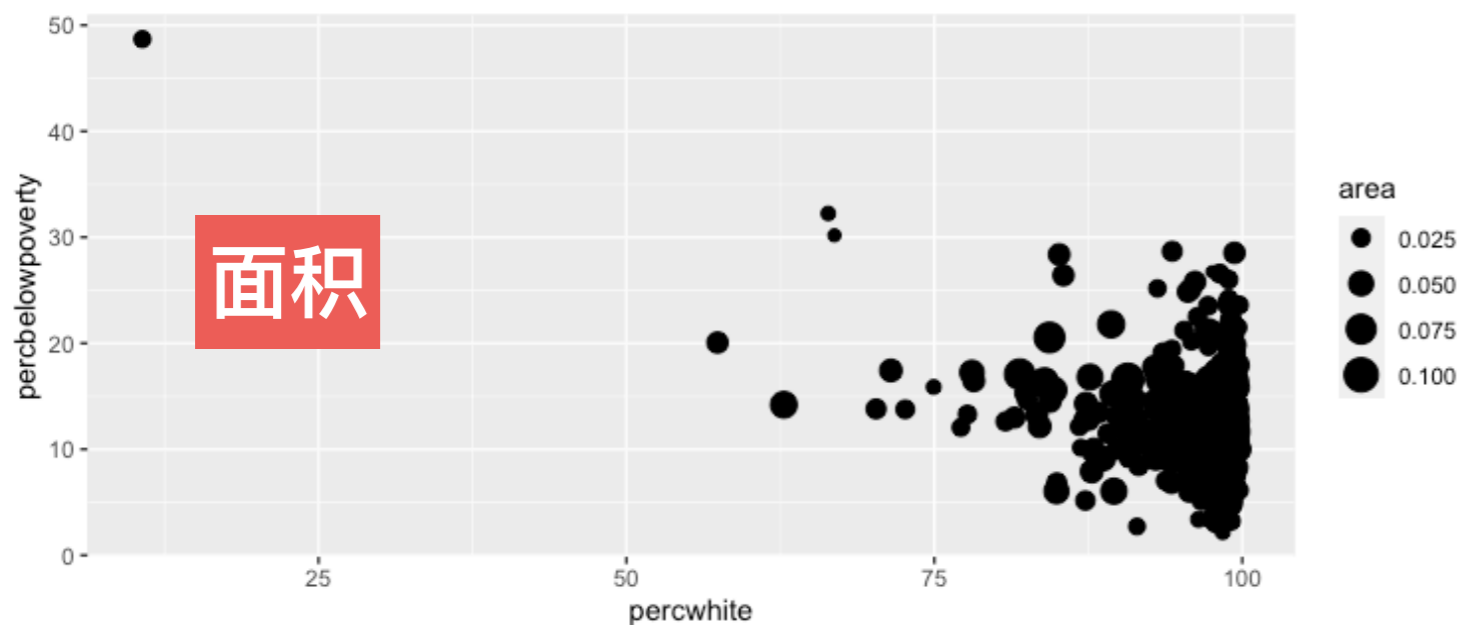
```
highest <-  
subset(economics,  
unemploy ==  
max(unemploy))  
unemp + geom_point(data  
= highest,  
size = 3, colour =  
alpha("red", 0.5))
```

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

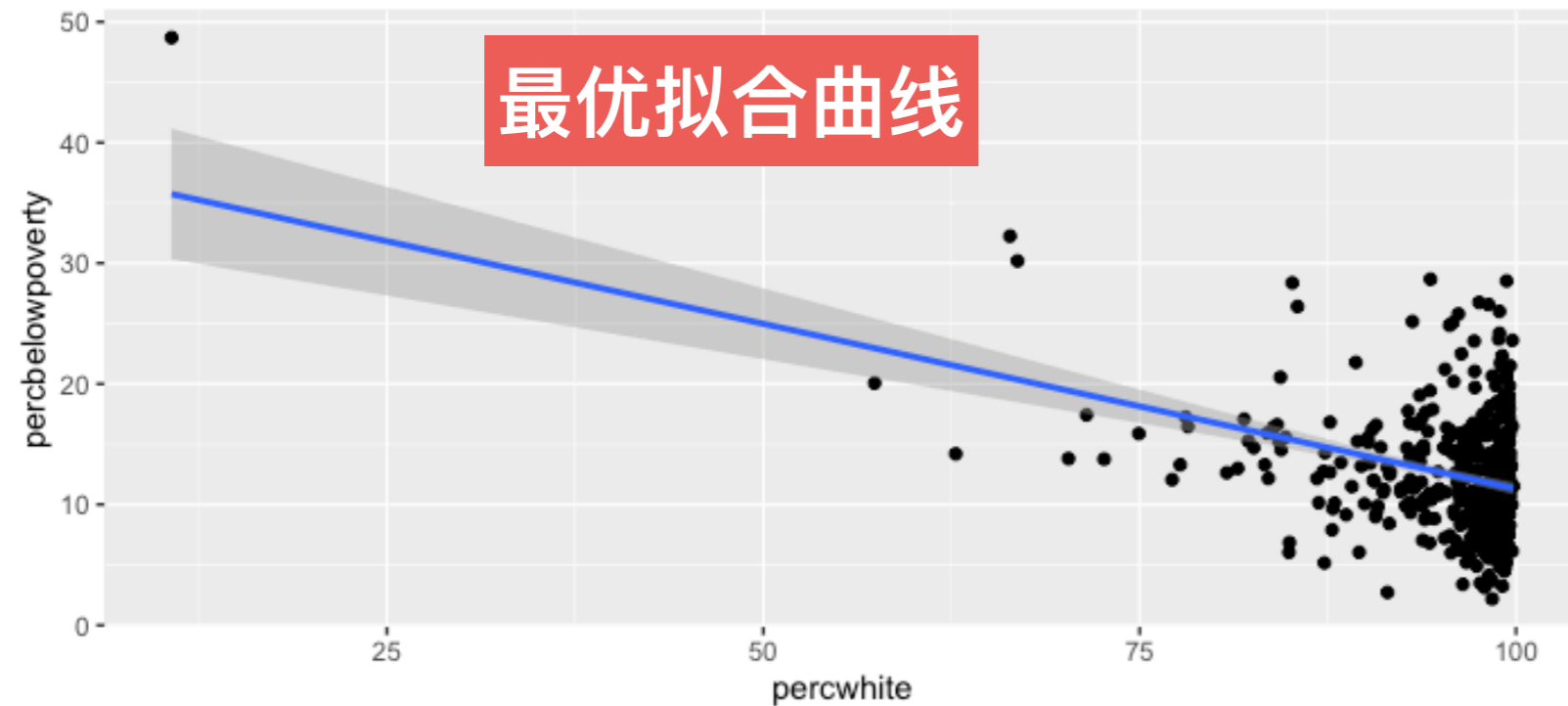


```
qplot(percwhite, percbelowpoverty, data =
midwest)
```

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



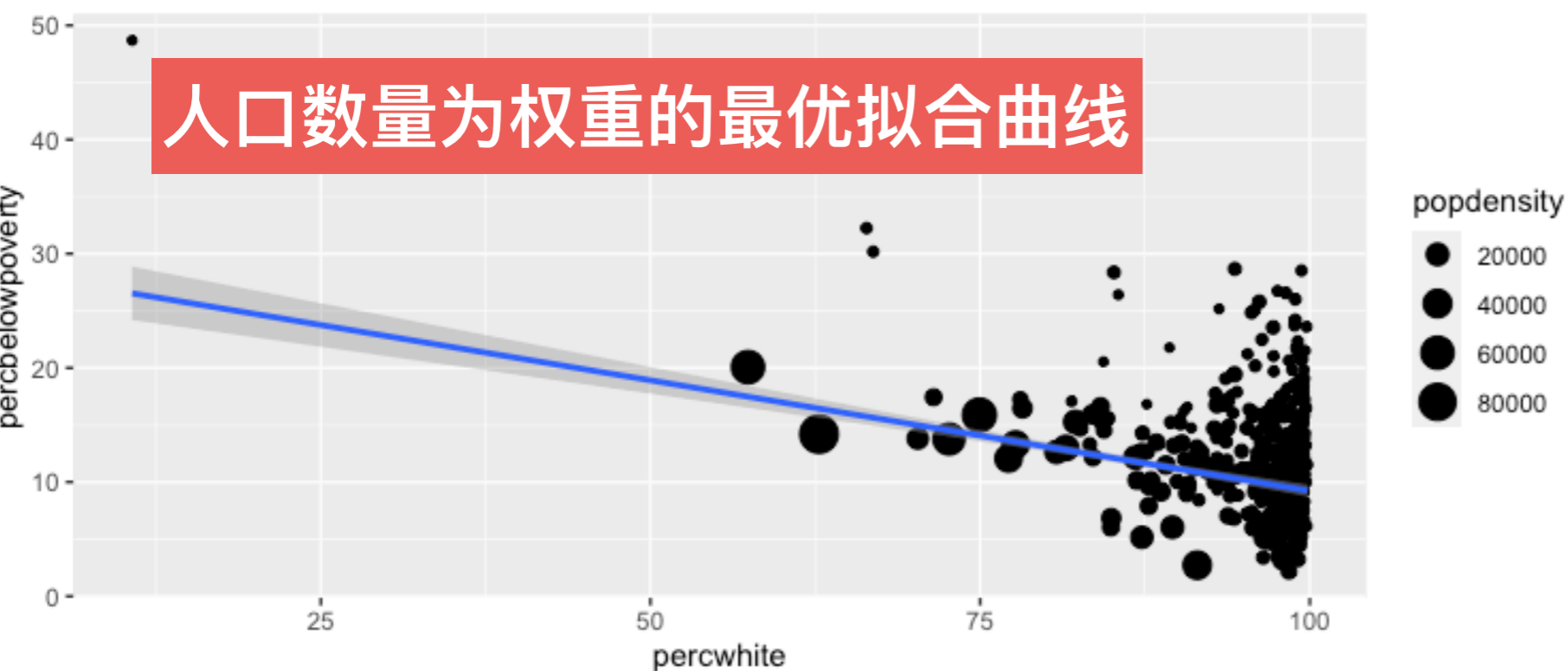
```
qplot(percwhite, percbelowpoverty, data =
midwest, size = area) +
scale_size_area()
```

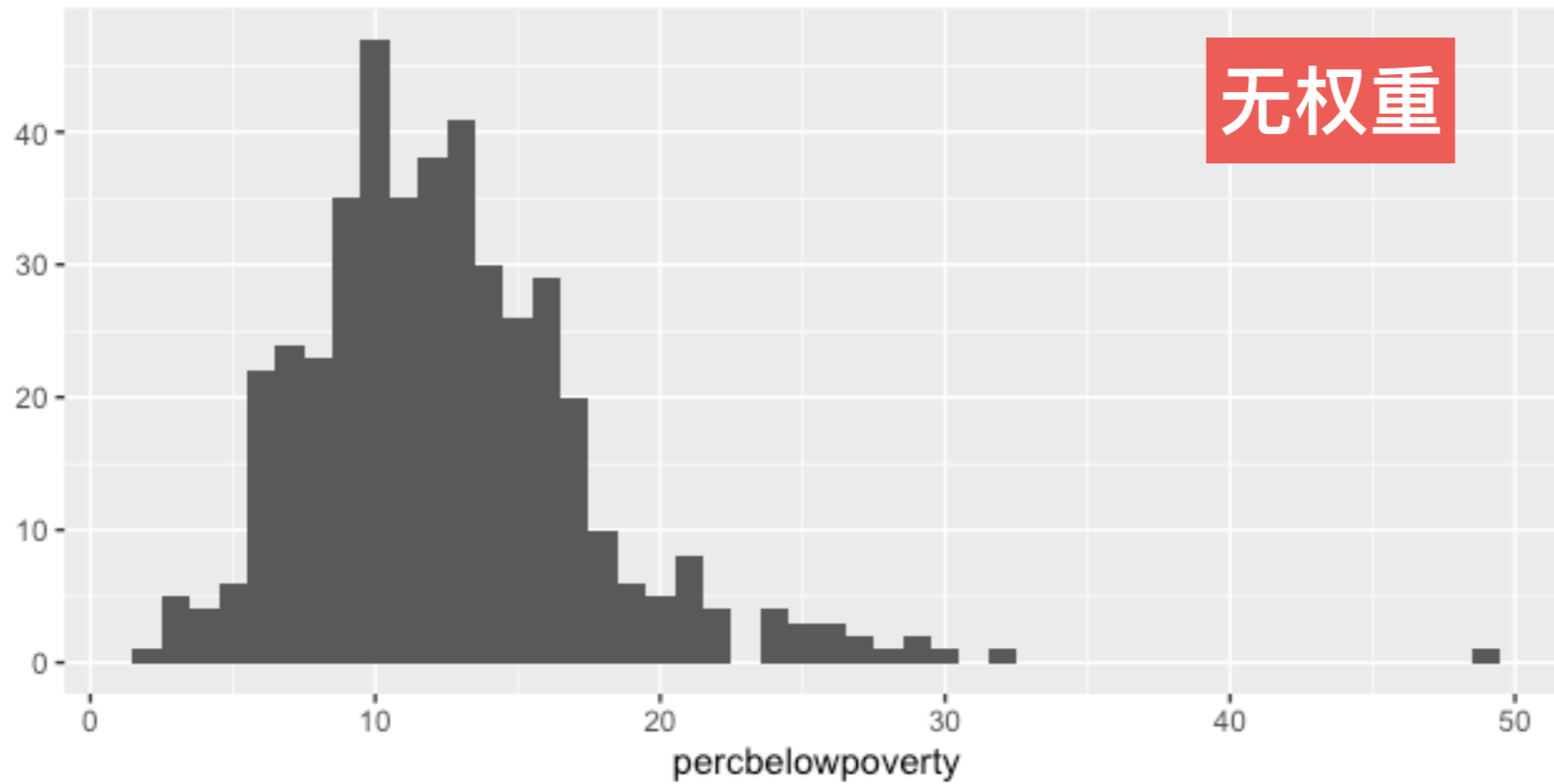


```
lm_smooth <-  
geom_smooth(method = lm, size =  
1)
```

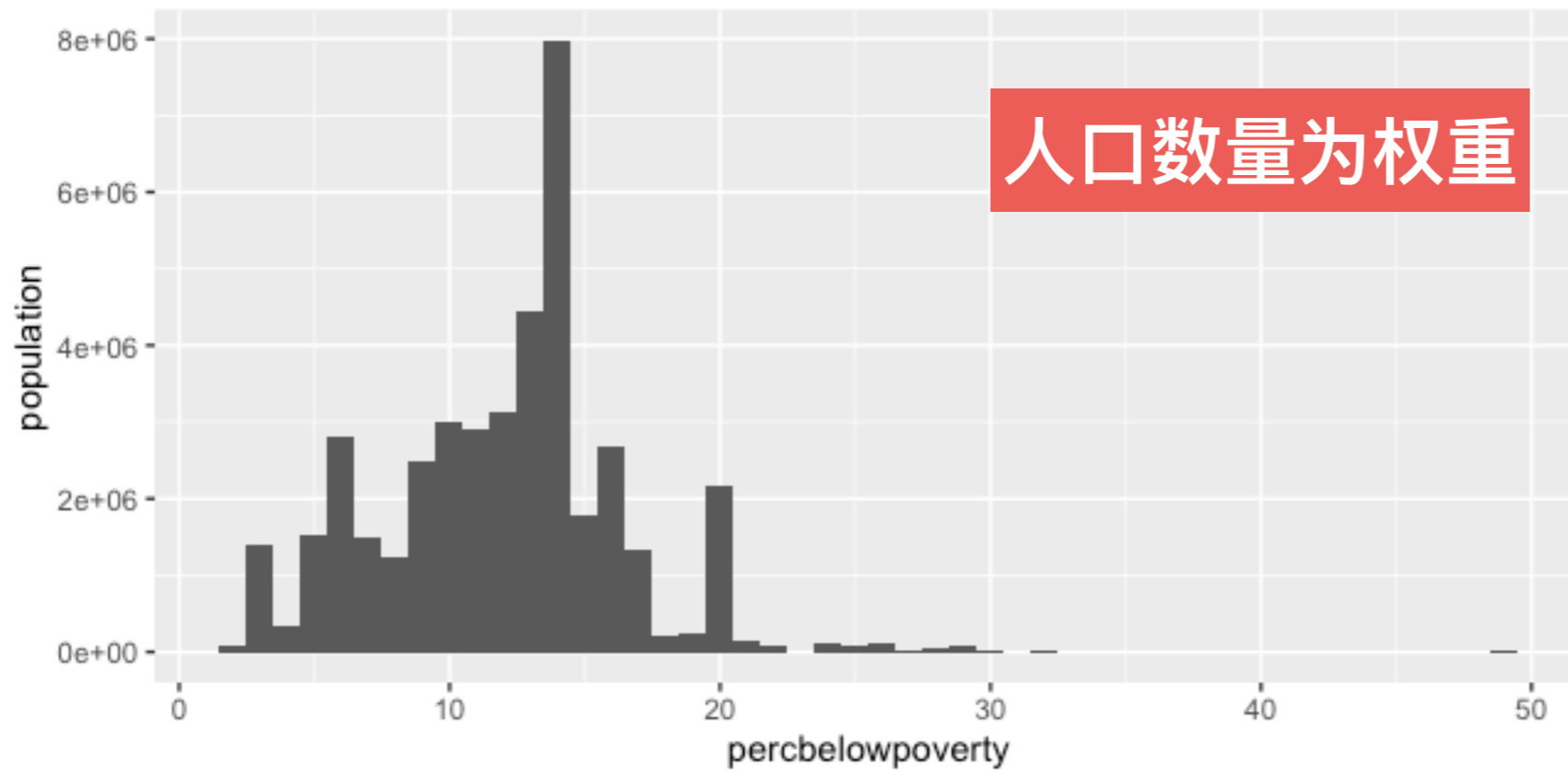
```
qplot(percwhite, percbelowpoverty,  
data = midwest) + lm_smooth
```

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





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



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

标度、坐标系和图例

定位

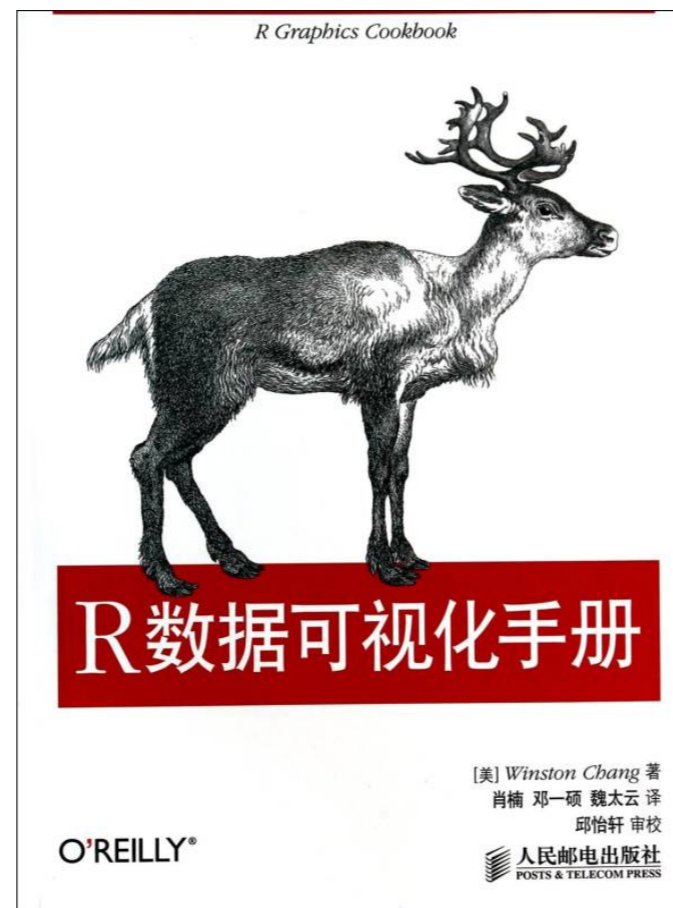
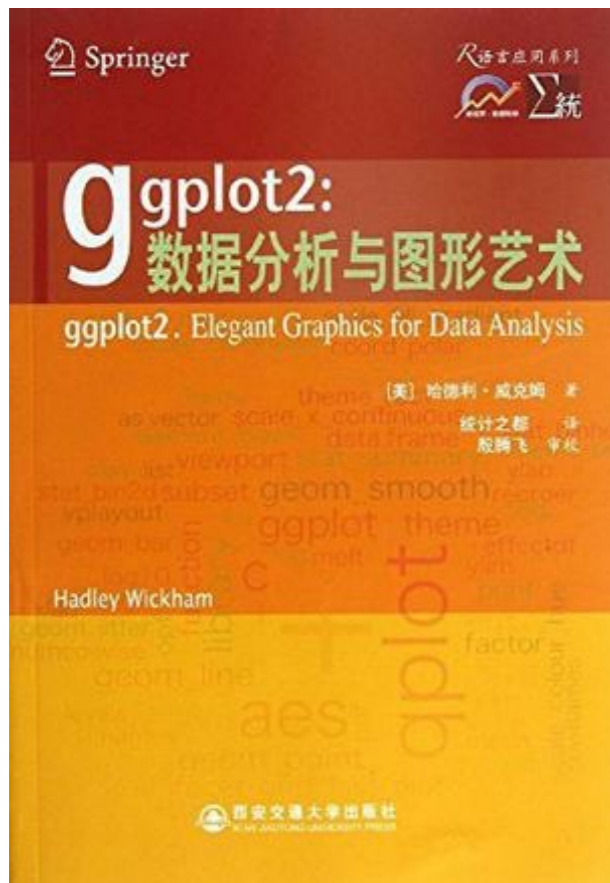
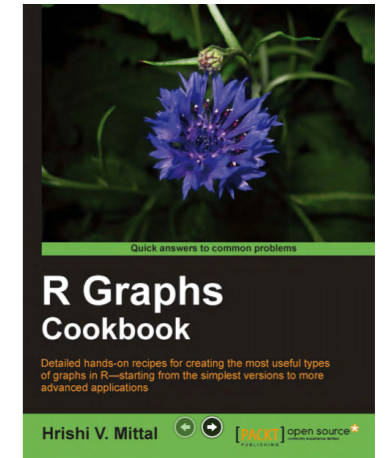
提问时间!

孙惠平

sunhp@ss.pku.edu.cn

练习

- 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



INTERACTIVE COURSE

Intermediate Data Visualization with ggplot2

[Start Course For Free](#) [Bookmark](#)

🕒 4 hours ▶ 14 Videos ↔ 52 Exercises 👤 15,336 Participants 📊 4,350 XP

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