

ggplot2画图II



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中的数据集中的数据集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`:
- 复杂图形:
 - * `+; geom_xxx;`

用图层构建图形

- 数据
- 一组图形属性映射
- 几何对象
- 统计变换
- 位置调整

- data
- mapping
- geom
- stat
- position

数据必须是数据框

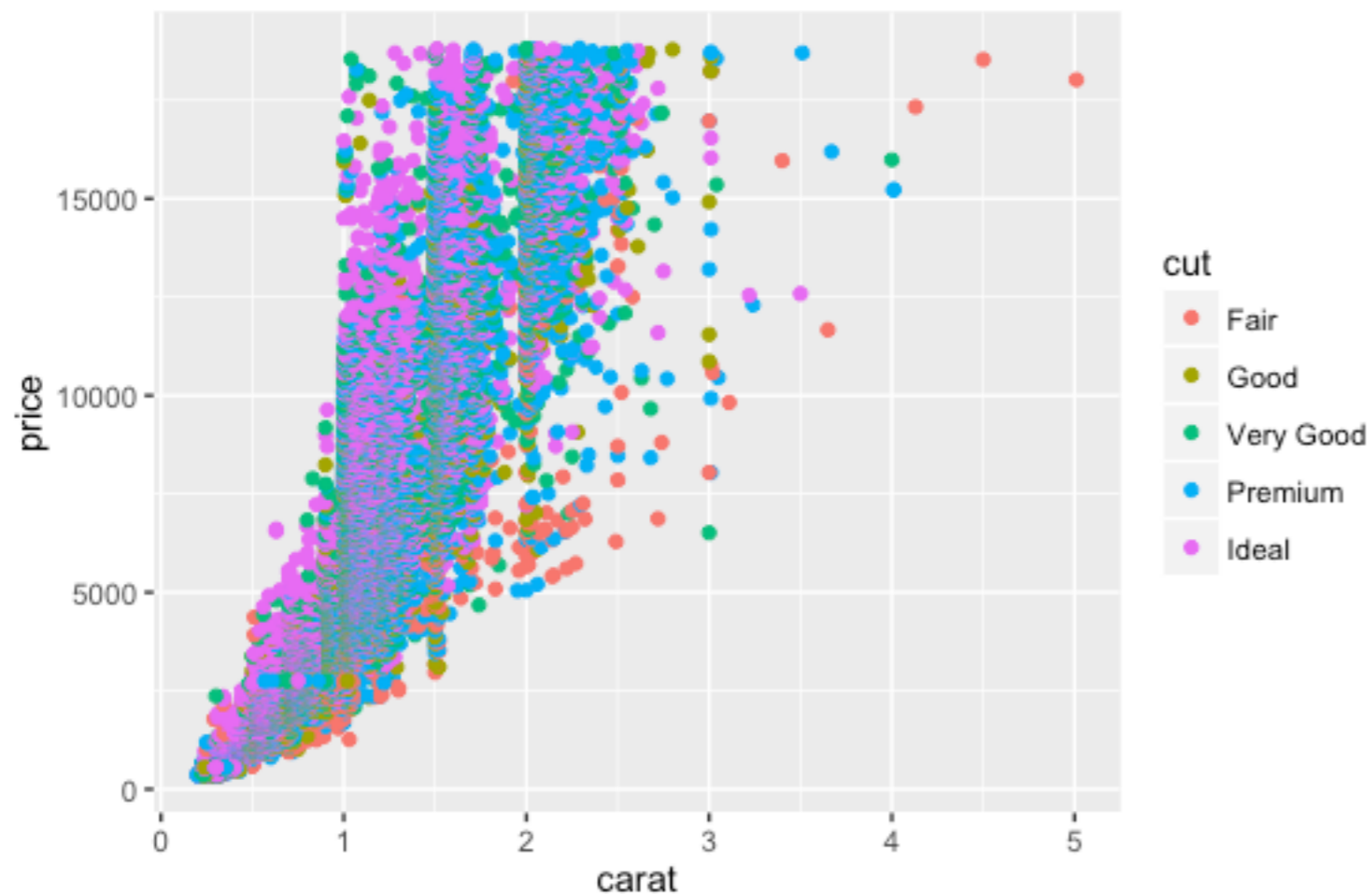
aes()

默认参数 参数设定

layer()

```
ggplot(data = NULL,  
       mapping = aes(),  
       ...,  
       environment = parent.frame())
```

layer()
自己查帮助



```
p <- ggplot(diamonds,  
           aes(carat,  
              price,  
              colour = cut),  
           )
```

p

```
p <- p + layer(geom = "point",  
              stat = "identity",  
              position = "identity")
```

p

```
geom(mapping = NULL,  
      data = NULL,  
      stat = "identity"  
      position = "identity"  
      ...,  
      na.rm = FALSE,  
      show.legend = NA,  
      inherit.aes = TRUE  
    )
```

见教材ggplot2的58页

```
geom_point()  
geom_line()  
geom_path()  
geom_bar()  
geom_histogram()  
geom_smooth()  
geom_density()  
geom_jitter()  
geom_text()  
geom_hline()  
geom_vline()  
geom_blank()  
geom_area()  
geom_abline()  
...
```



```
stat(mapping = NULL,  
      data = NULL,  
      geom/stat = ""  
      position = "identity"  
      ...,  
      na.rm = FALSE,  
      show.legend = NA,  
      inherit.aes = TRUE  
)
```

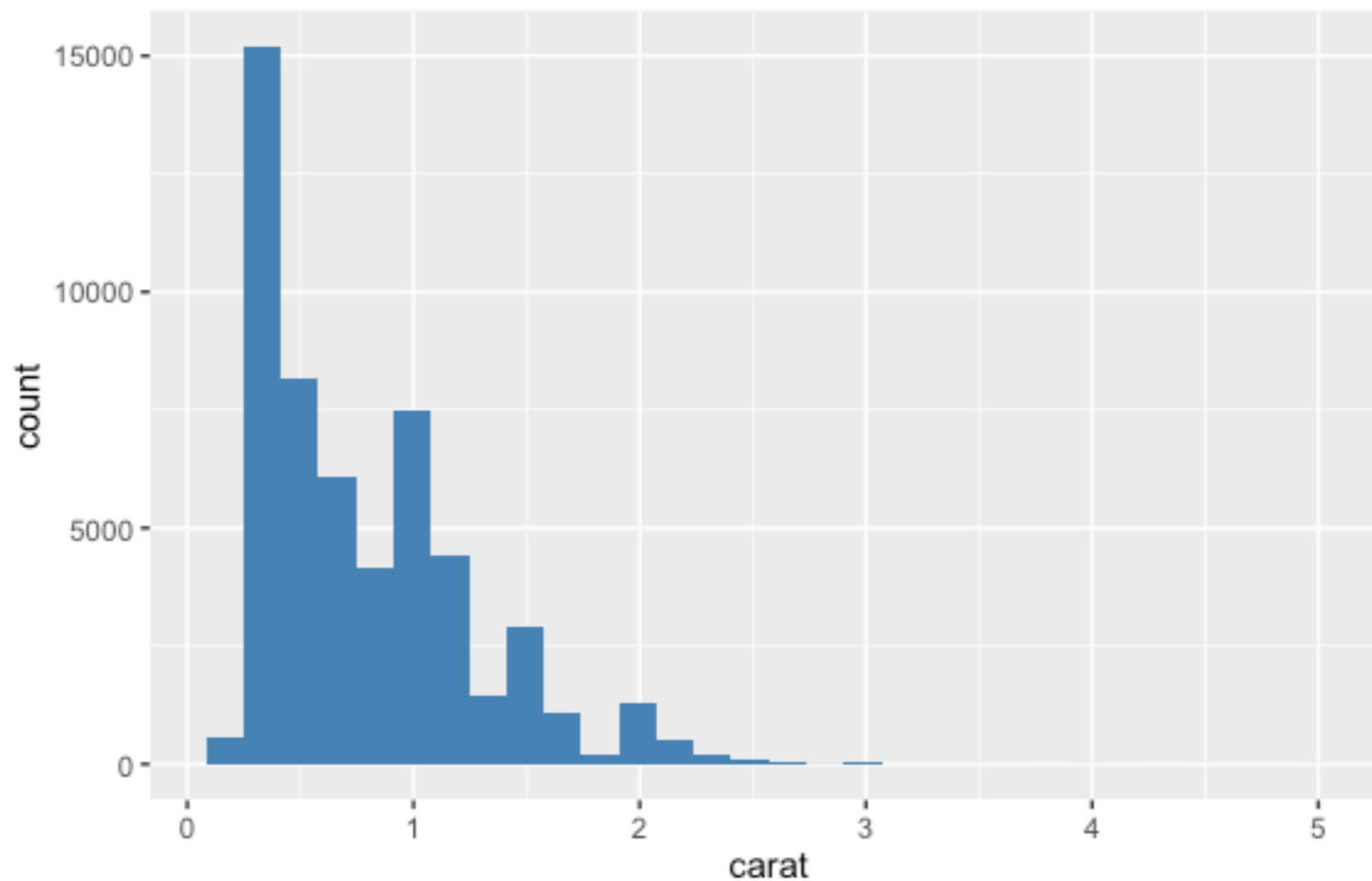
见教材ggplot2的60页

```
stat_identity()  
stat_smooth()  
stat_function()  
stat_boxplot()  
stat_density()  
stat_quantile()  
stat_sum()  
stat_summary()  
stat_unique()  
stat_bin()  
stat_bindot()  
...
```

```
p <- ggplot(diamonds, aes(x = carat))  
p <- p + layer(  
  geom = "bar",  
  stat = "bin",  
  position = "identity",  
  params = list(fill = "steelblue")  
)
```

p

```
p <- ggplot(diamonds,  
  aes(x = carat))  
p <- p + geom_histogram(bins = 30,  
  fill = "steelblue")  
p
```



```
> p <- ggplot(msleep, aes(sleep_rem / sleep_total, awake))  
> summary(p)
```

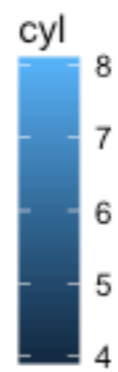
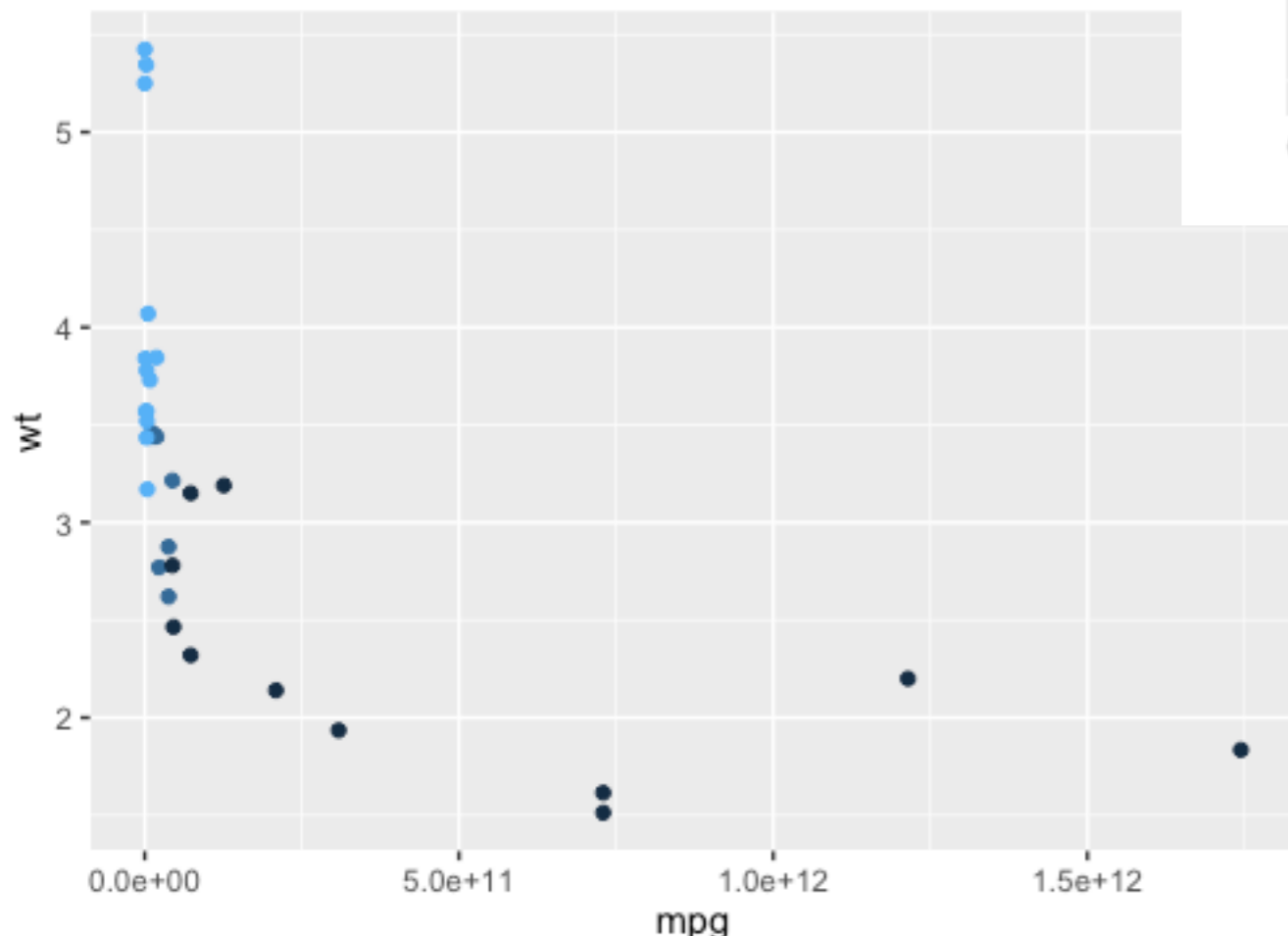
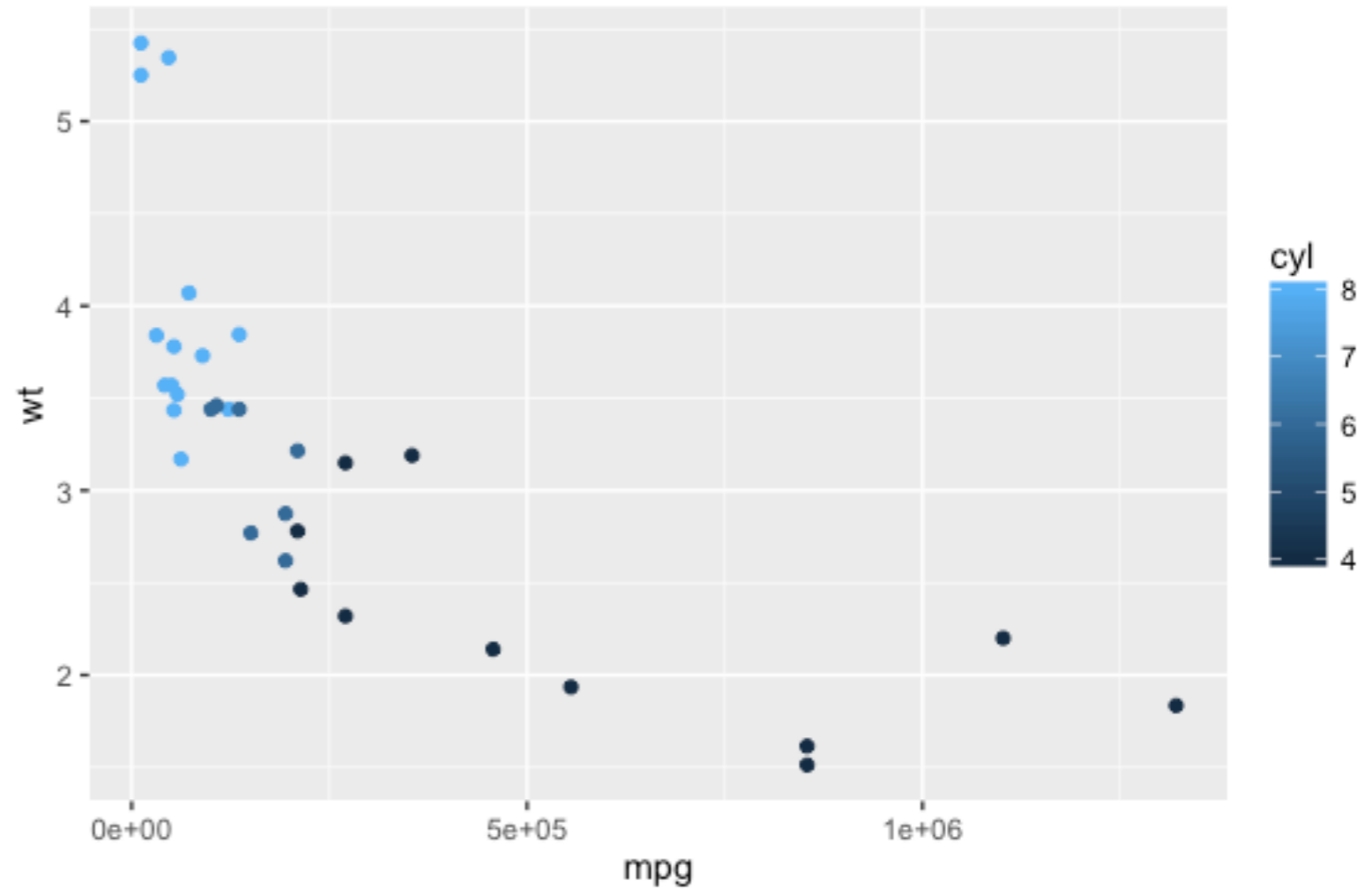
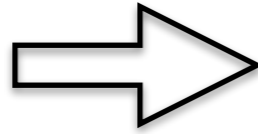
```
data: name, genus, vore, order, conservation, sleep_total, sleep_rem,  
      sleep_cycle, awake, brainwt, bodywt [83x11]  
mapping: x = sleep_rem/sleep_total, y = awake  
faceting: facet_null()
```

```
> p <- p + geom_point()  
> summary(p)
```

```
data: name, genus, vore, order, conservation, sleep_total, sleep_rem,  
      sleep_cycle, awake, brainwt, bodywt [83x11]  
mapping: x = sleep_rem/sleep_total, y = awake  
faceting: facet_null()
```

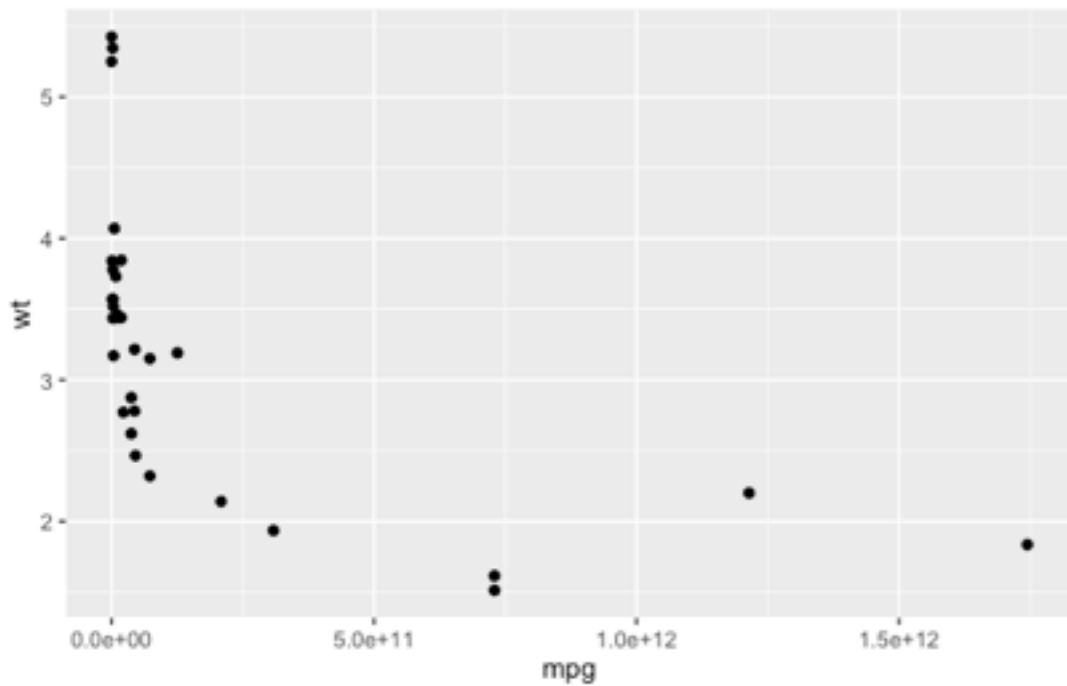
```
-----  
geom_point: na.rm = FALSE  
stat_identity: na.rm = FALSE  
position_identity
```

```
p <- ggplot(mtcars,  
  aes(mpg,  
    wt,  
    colour = cyl))  
+ geom_point()
```



```
mtcars <- transform(mtcars, mpg = mpg ^ 2)  
p %+% mtcars
```

aes()

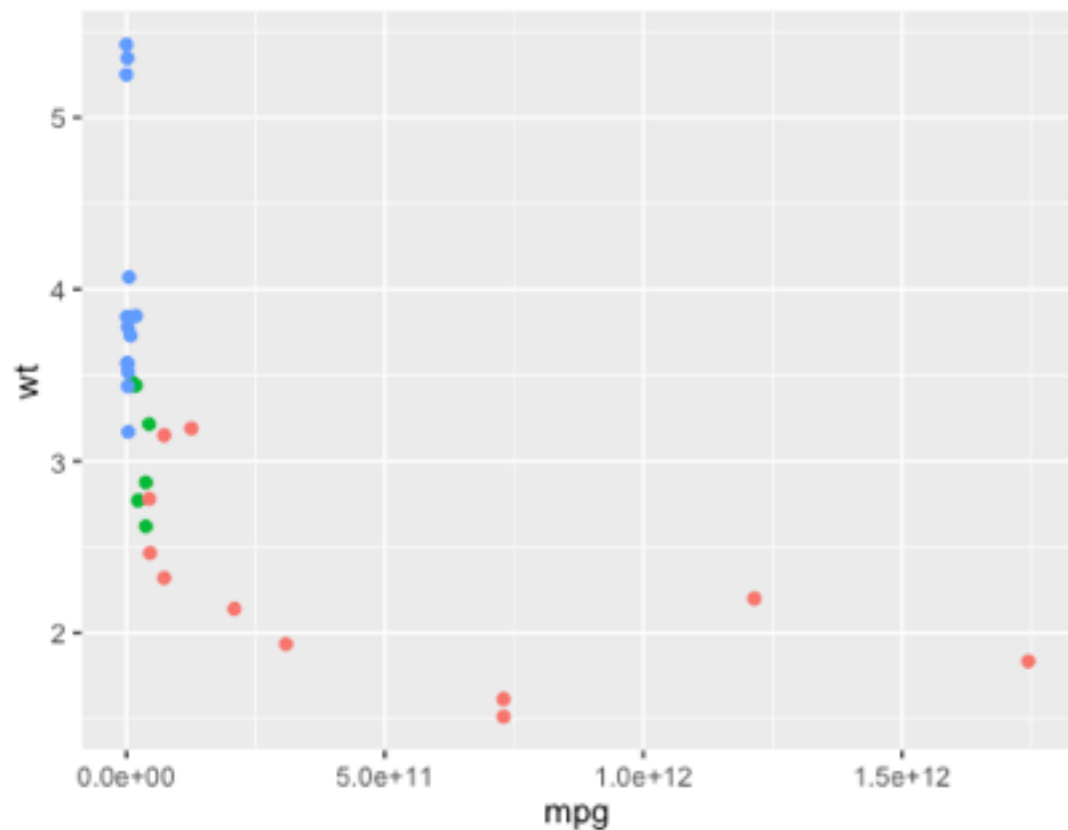


```
p <- ggplot(mtcars, aes(x = mpg, y = wt))
```

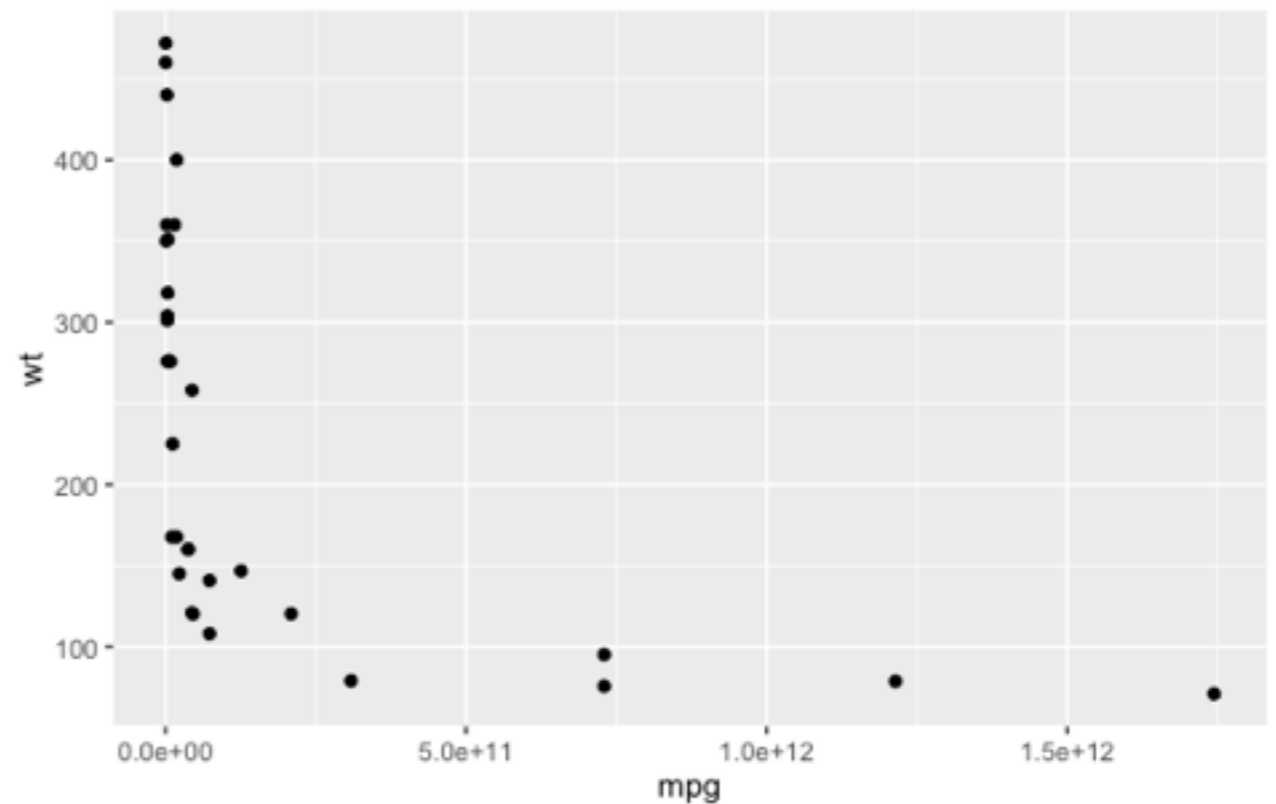
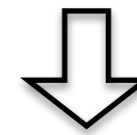


```
p + geom_point()
```

```
p + geom_point(aes(colour = factor(cyl)))
```



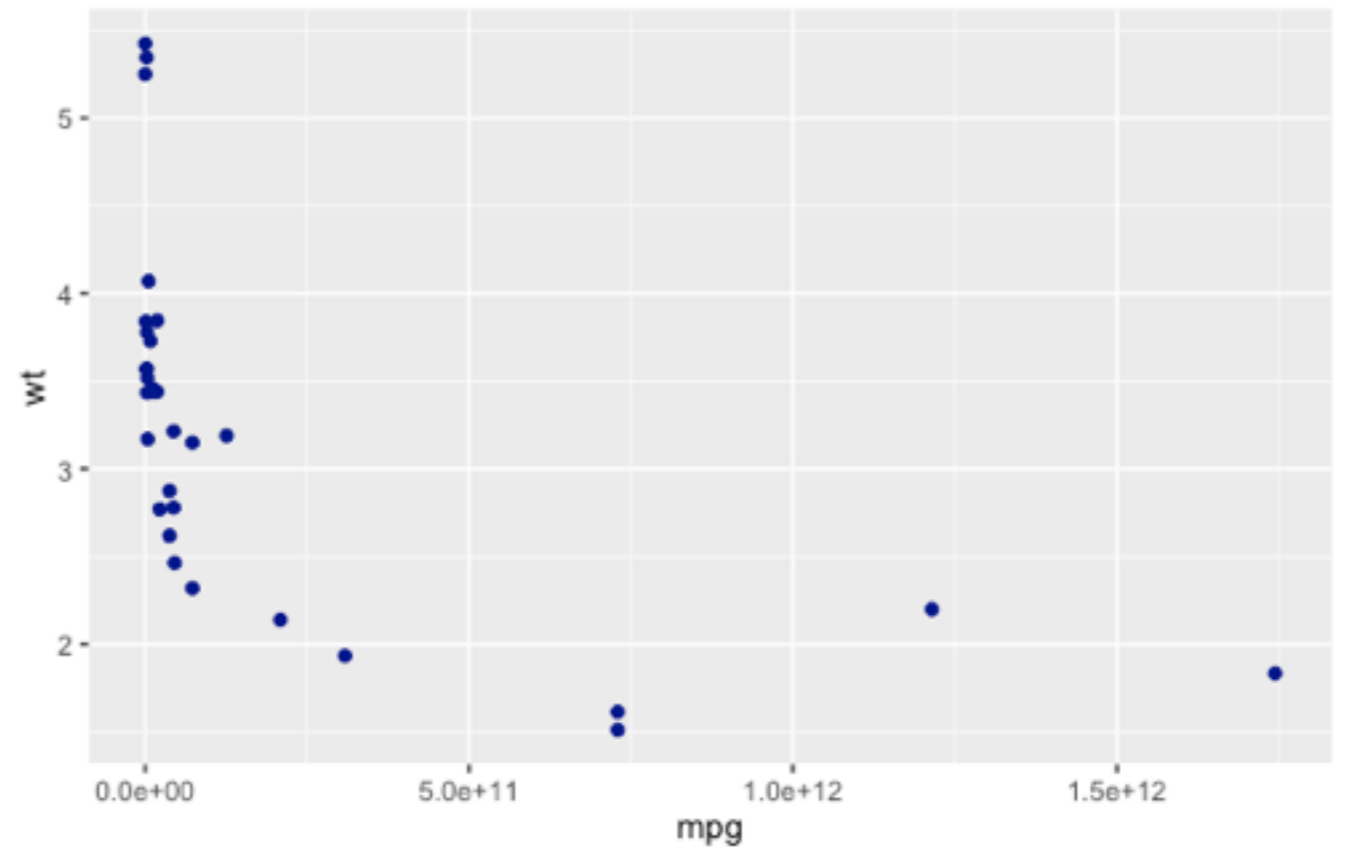
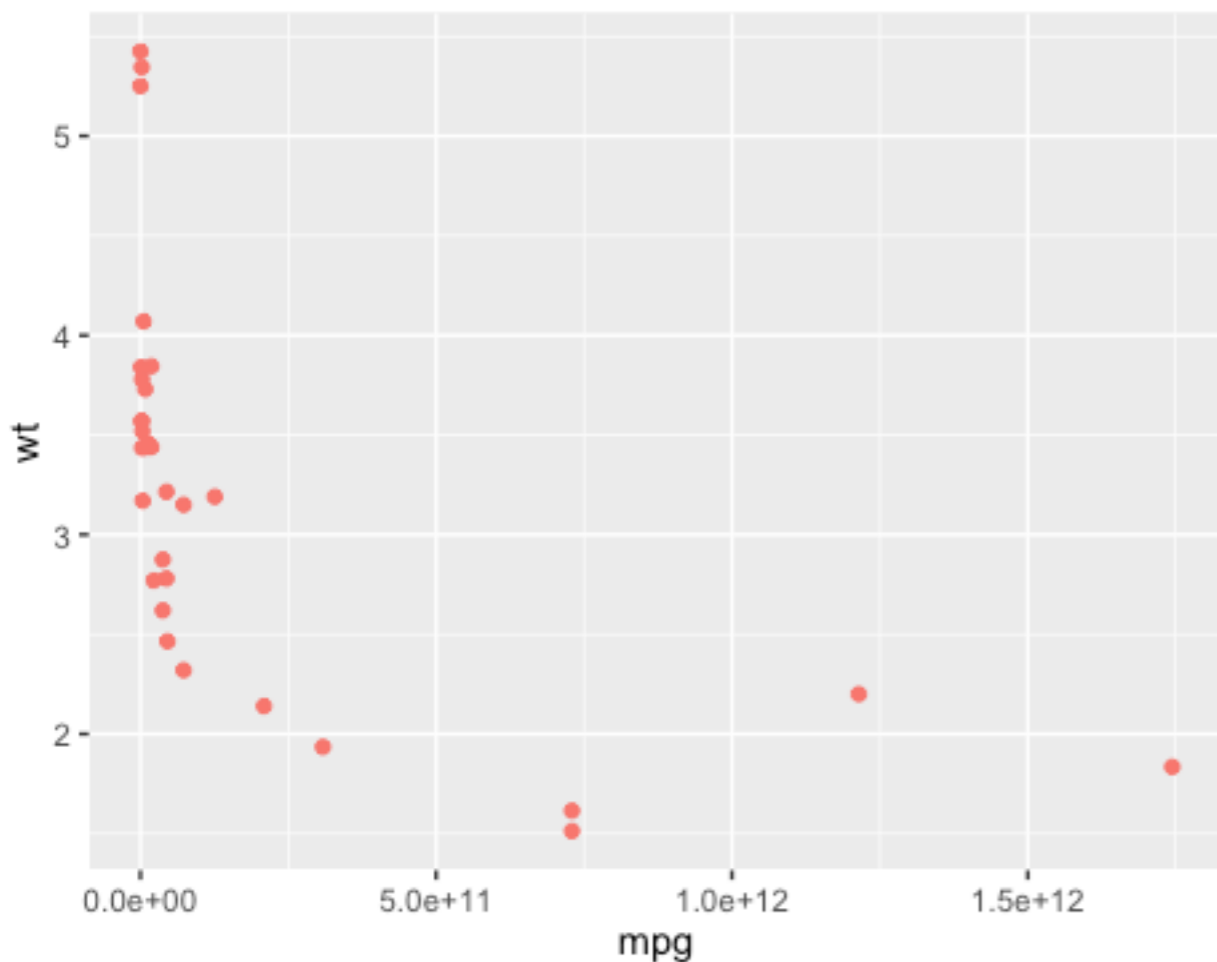
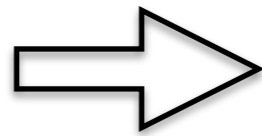
```
p + geom_point(aes(y = disp))
```



图形属性 vs. 图层属性

```
p <- ggplot(mtcars, aes(mpg, wt))
```

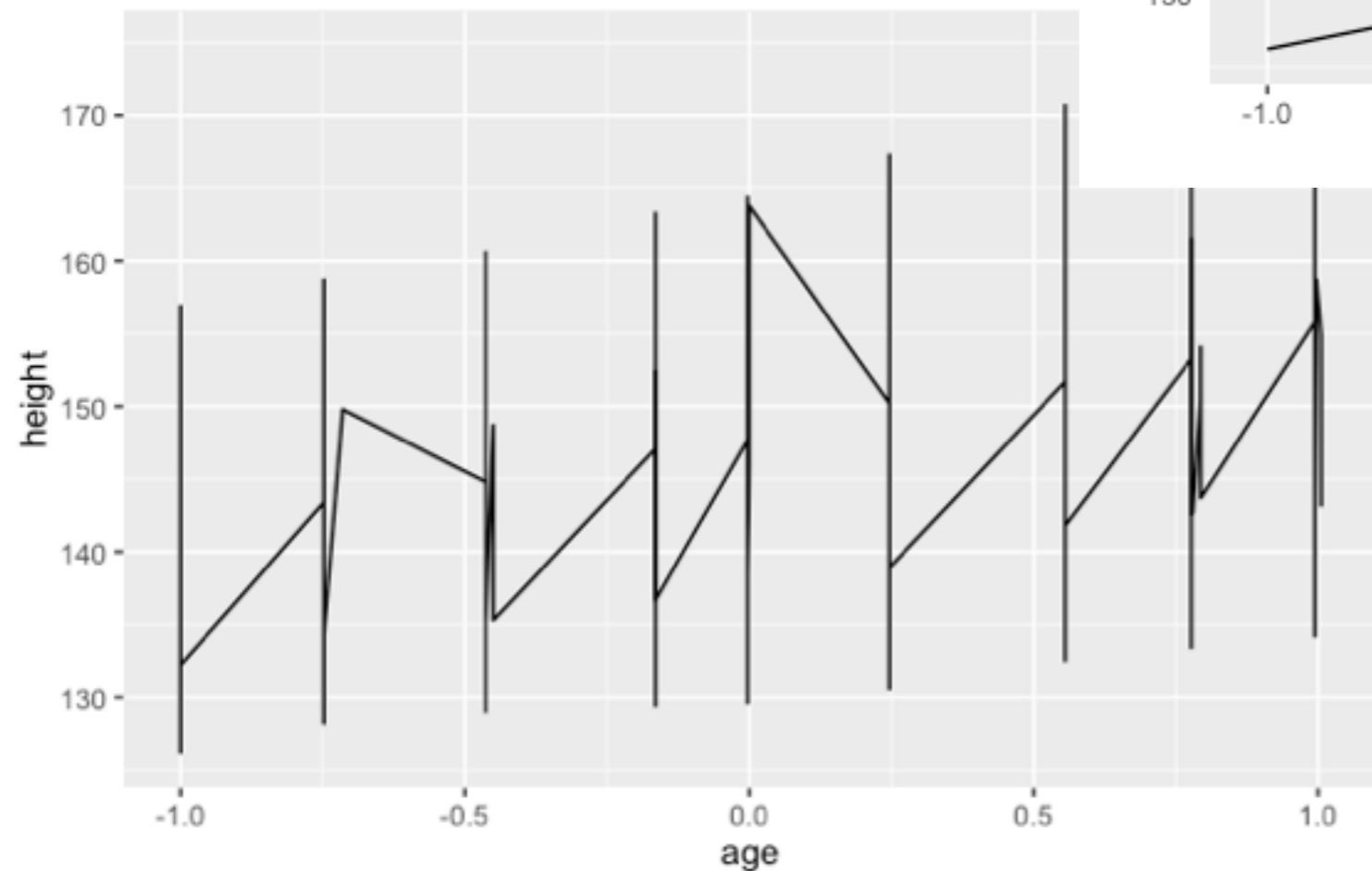
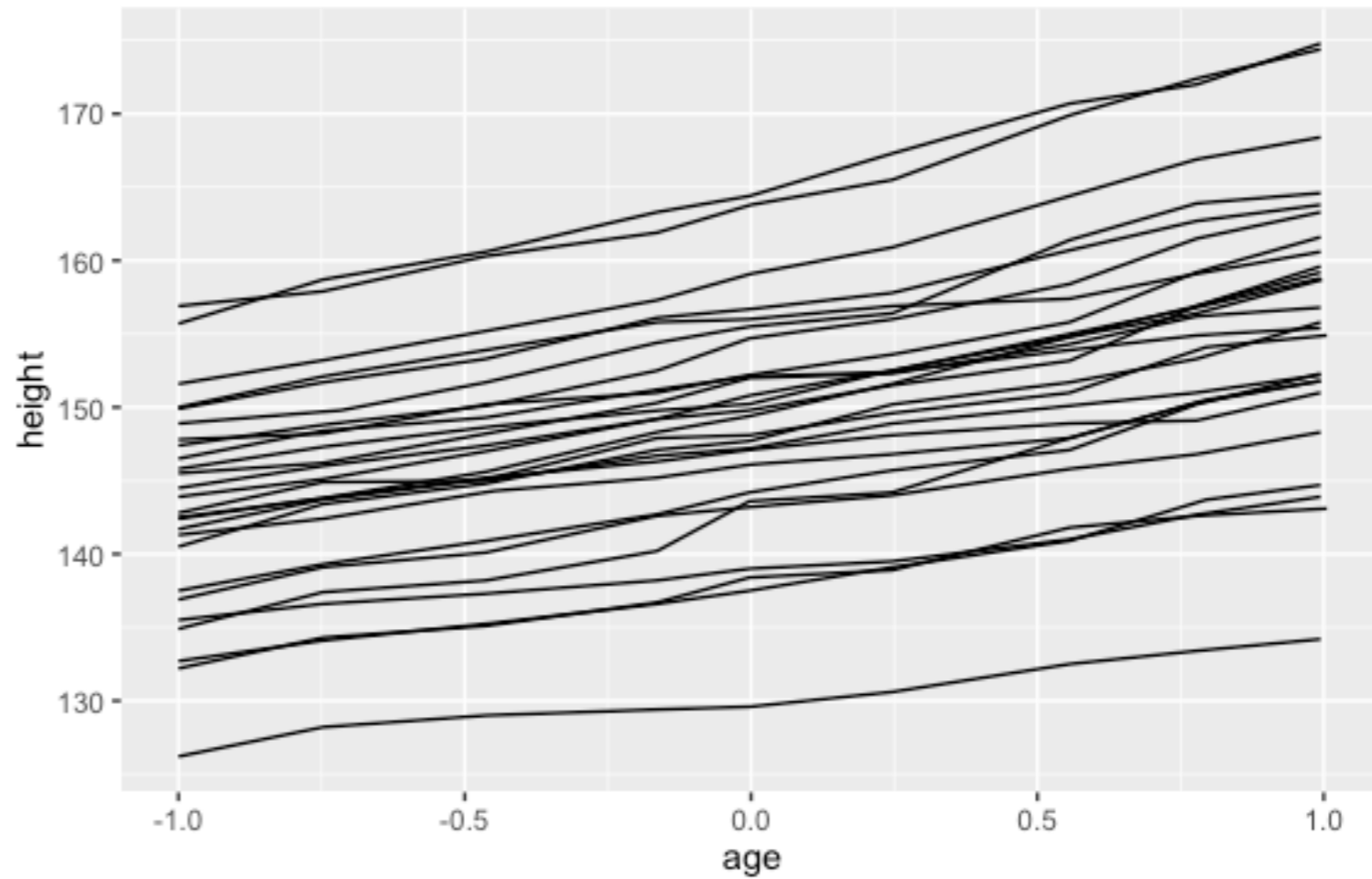
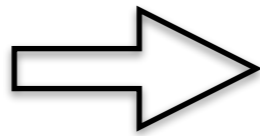
```
p + geom_point(colour = "darkblue")
```



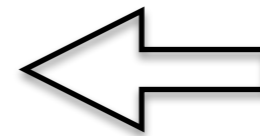
colour
● darkblue

```
p + geom_point(aes(colour = "darkblue"))
```

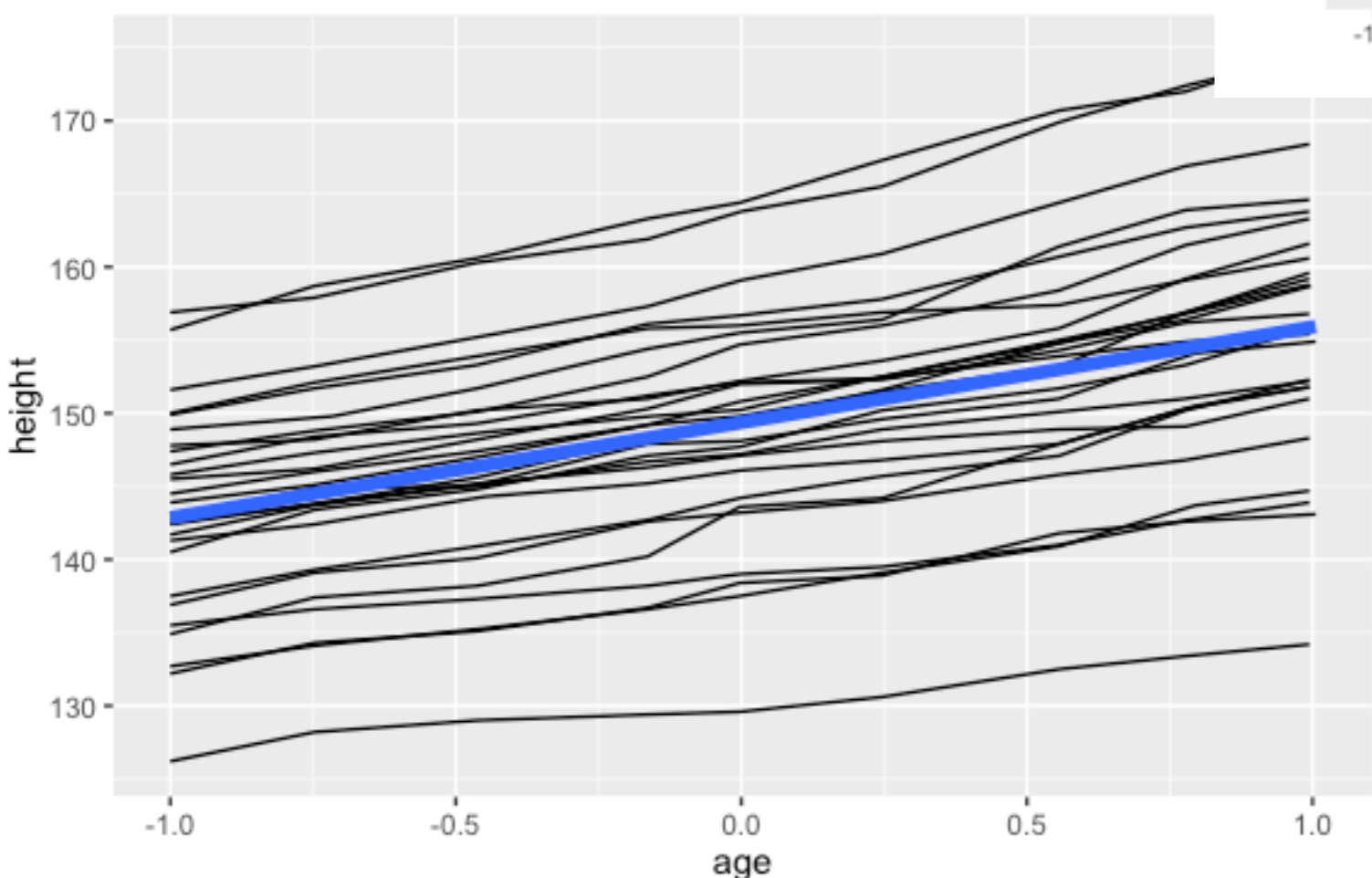
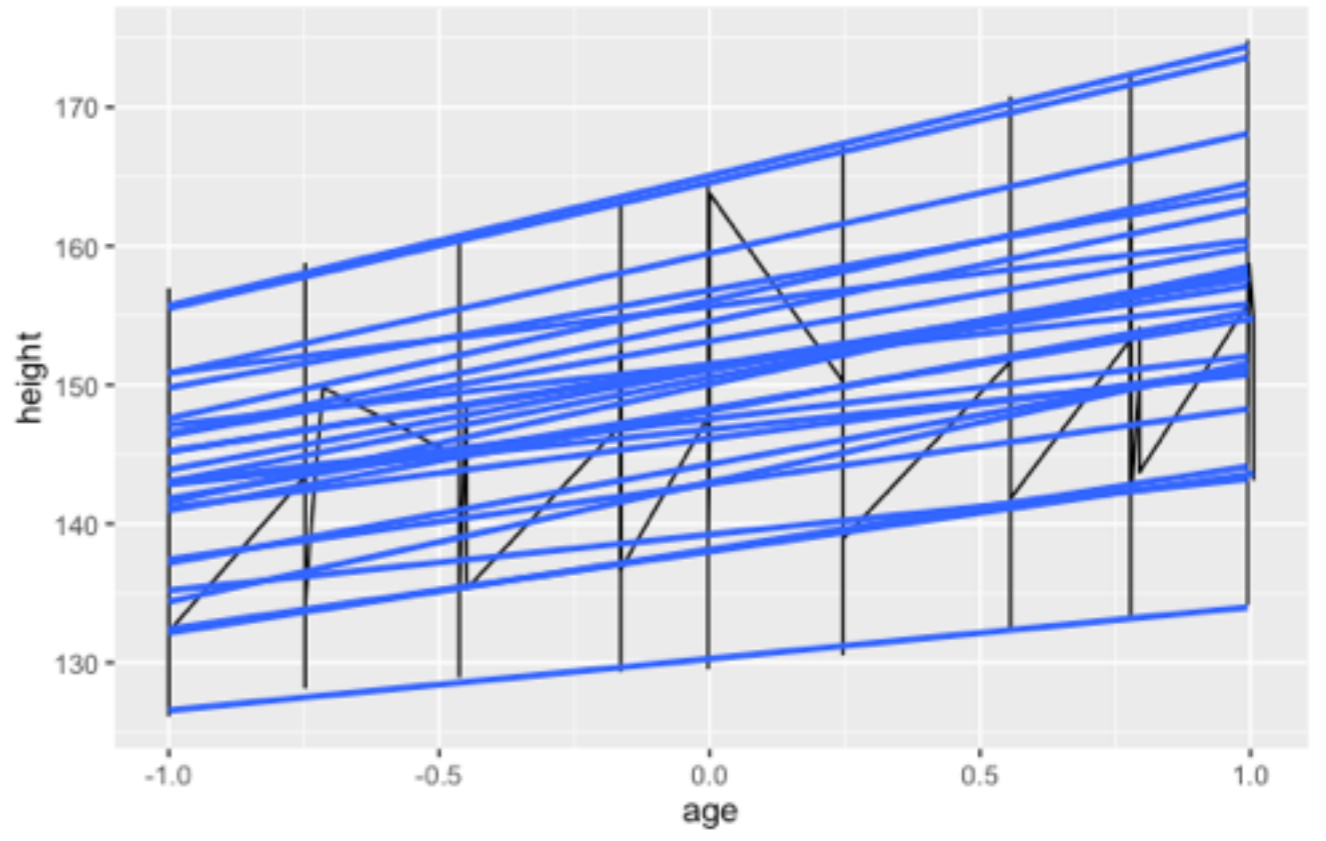
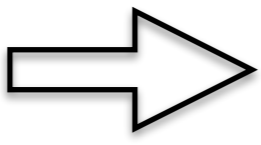
```
p <- ggplot(Oxboys,  
  aes(age,  
    height,  
    group = Subject)  
  )  
+ geom_line()
```



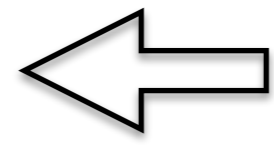
```
p <- ggplot(Oxboys,  
  aes(age,  
    height,  
    group = 1)  
  )  
+ geom_line()
```



```
p <- ggplot(Oxboys,  
  aes(age,  
    height,  
    group = Subject)  
  )  
p + geom_smooth(aes(group = Subject),  
  method="lm",  
  se = F)
```

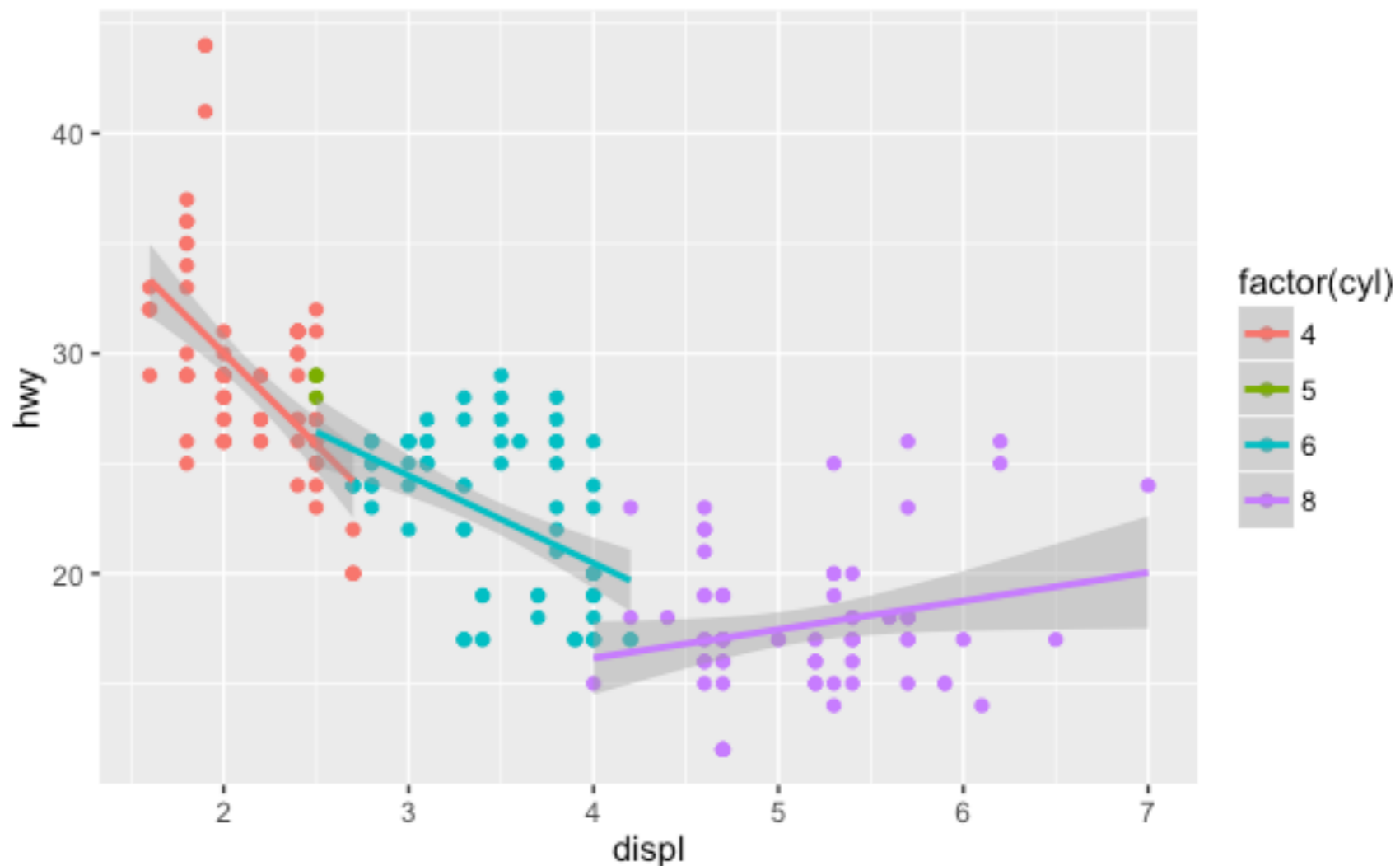


```
p <- ggplot(Oxboys,  
  aes(age,  
    height,  
    group = Subject)  
  )  
p + geom_smooth(aes(group = 1),  
  method="lm",  
  se = F)
```



工具箱

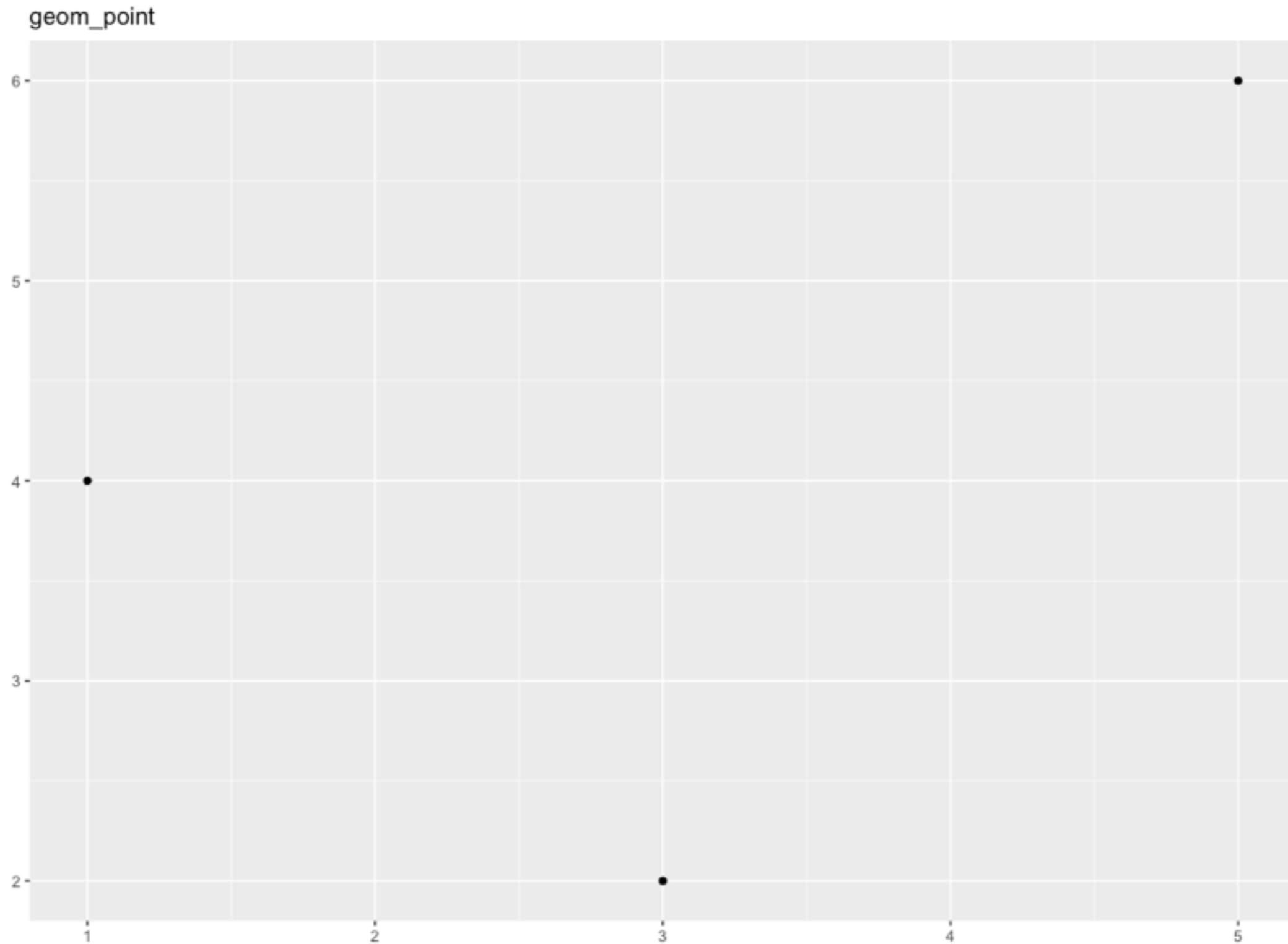
- 展示数据本身
- 展示数据的统计摘要
- 添加额外的元数据、上下文信息和注解



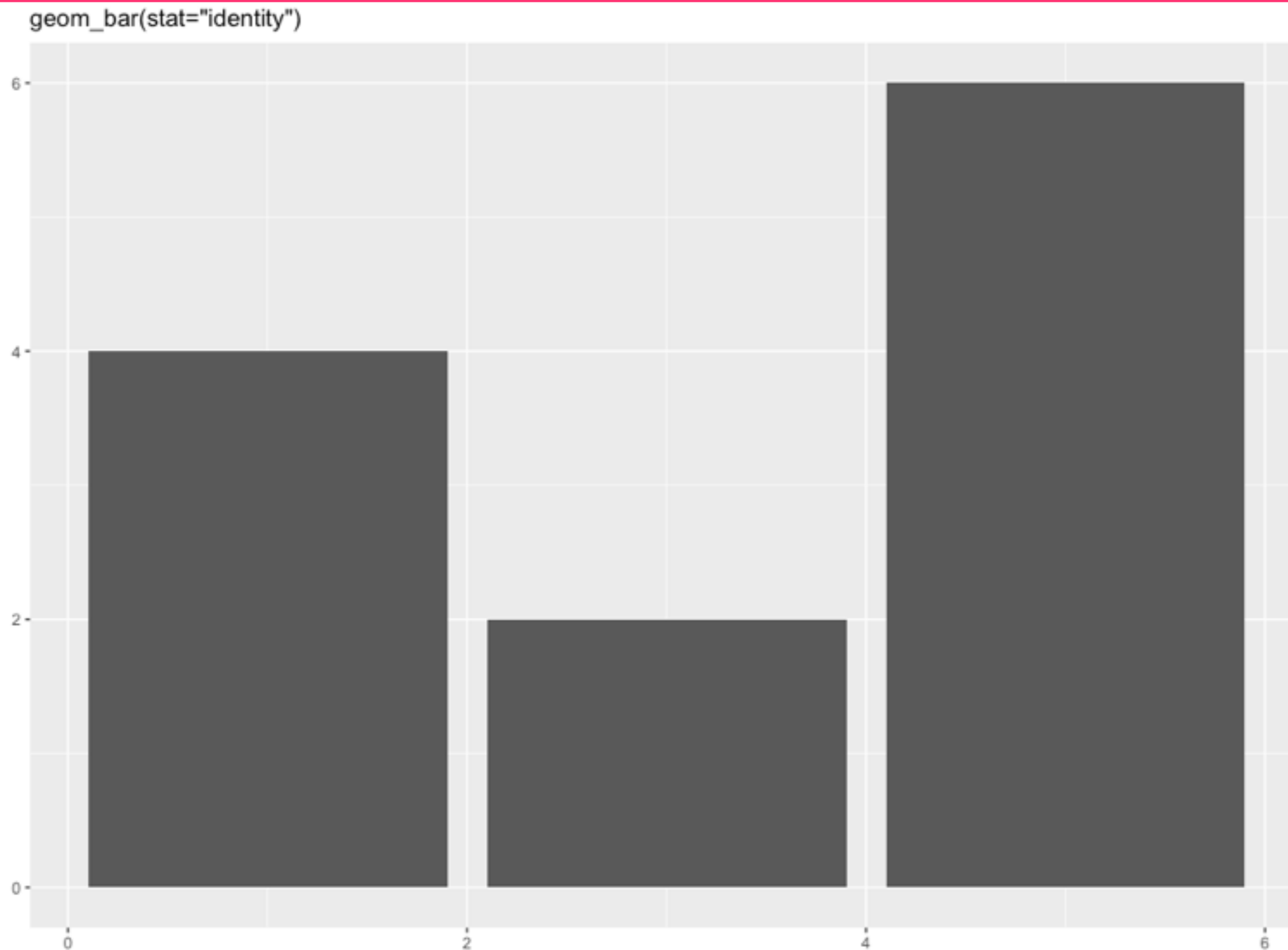
- `geom_area()`: 面积图
 - `geom_bar(stat="identity")`: 条形图
 - `geom_line()`: 线条图
 - `geom_point()`: 散点图
 - `geom_text()`: 添加标签
 - `geom_tile()`: 色深图、水平图
-

```
> 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)
```

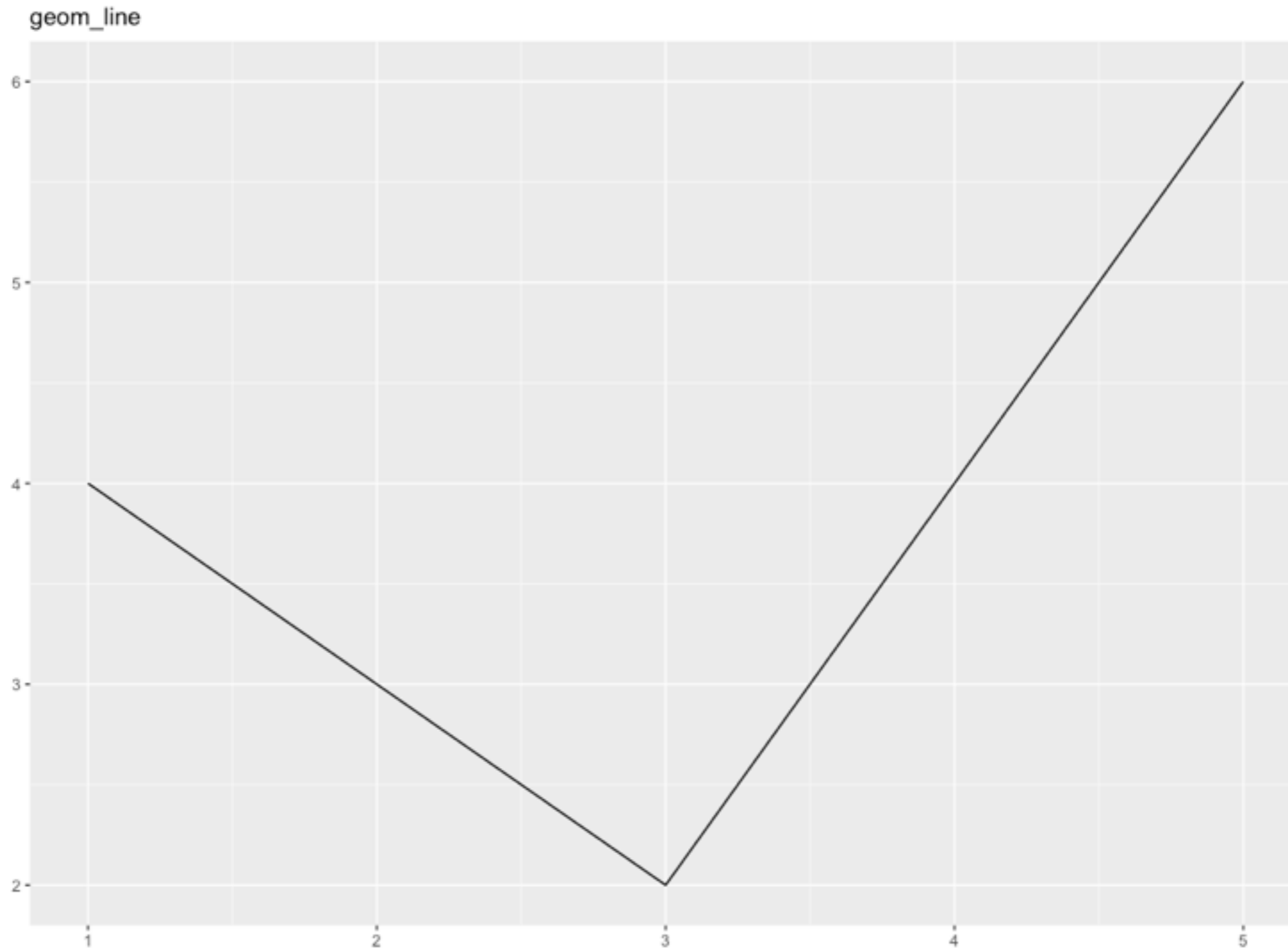
```
> p + geom_point() + labs(title = "geom_point")
```



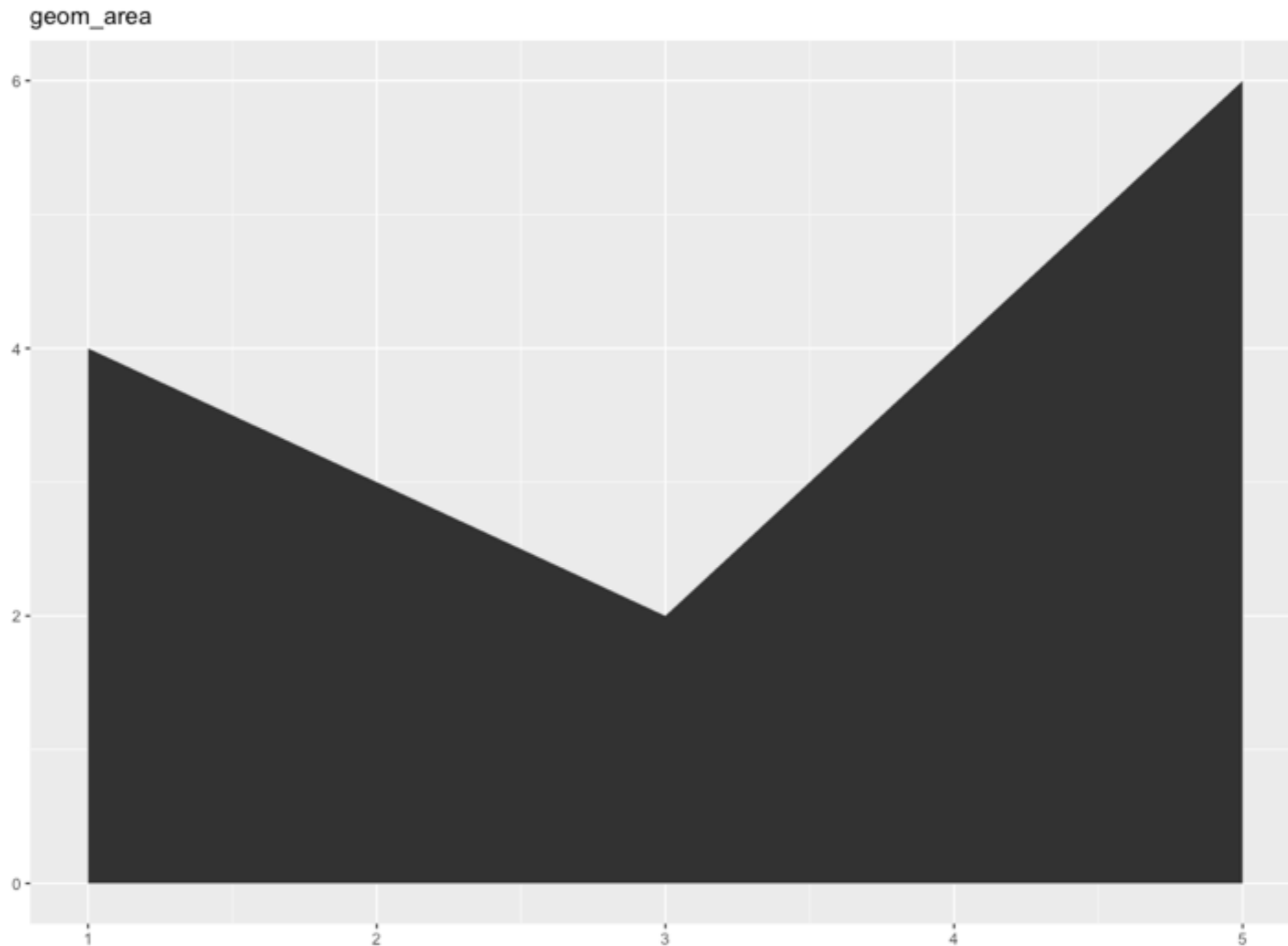
```
> p + geom_bar(stat="identity") +  
+ labs(title = "geom_bar(stat=\"identity\")")
```



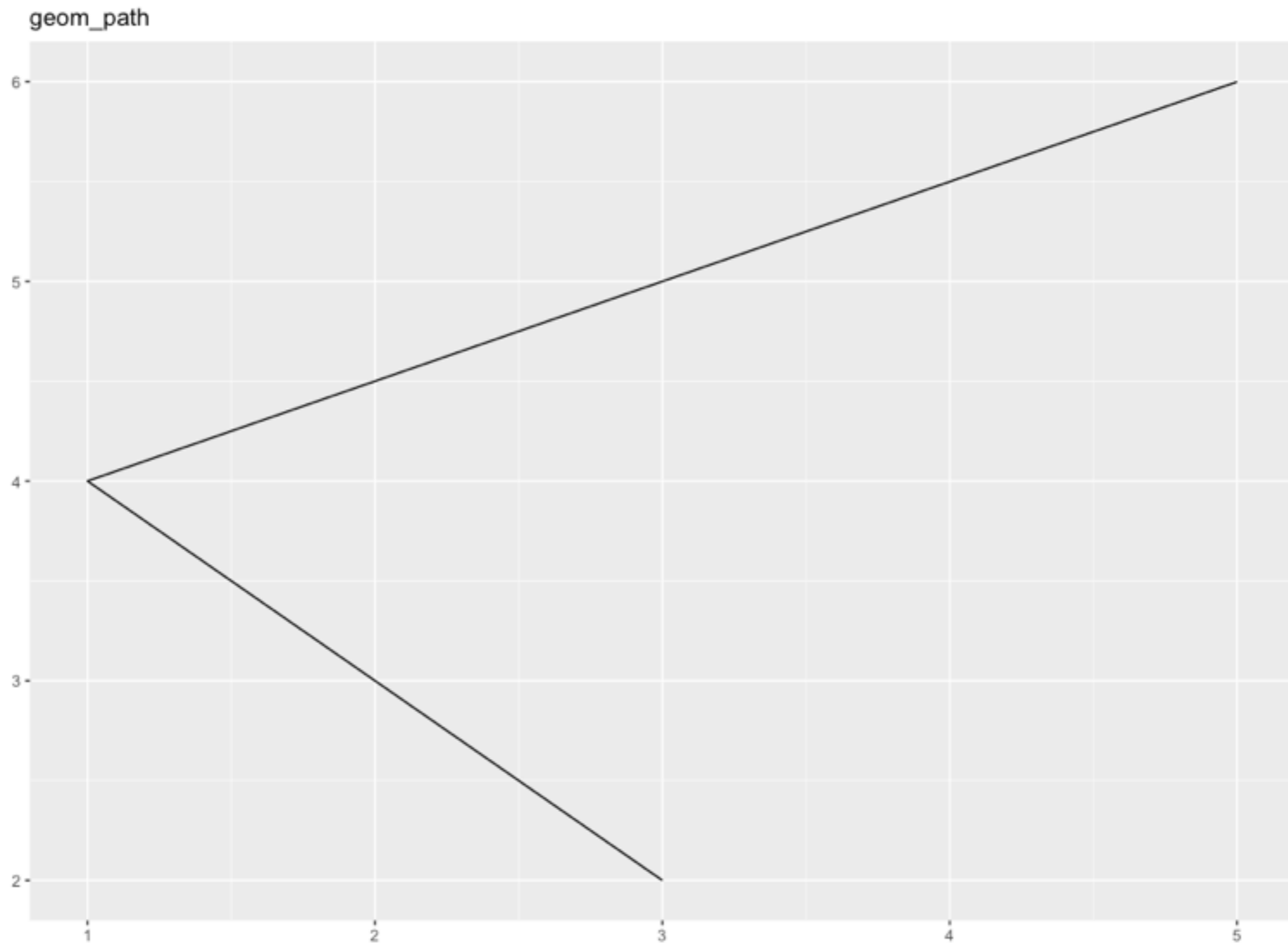
```
> p + geom_line() + labs(title = "geom_line")
```



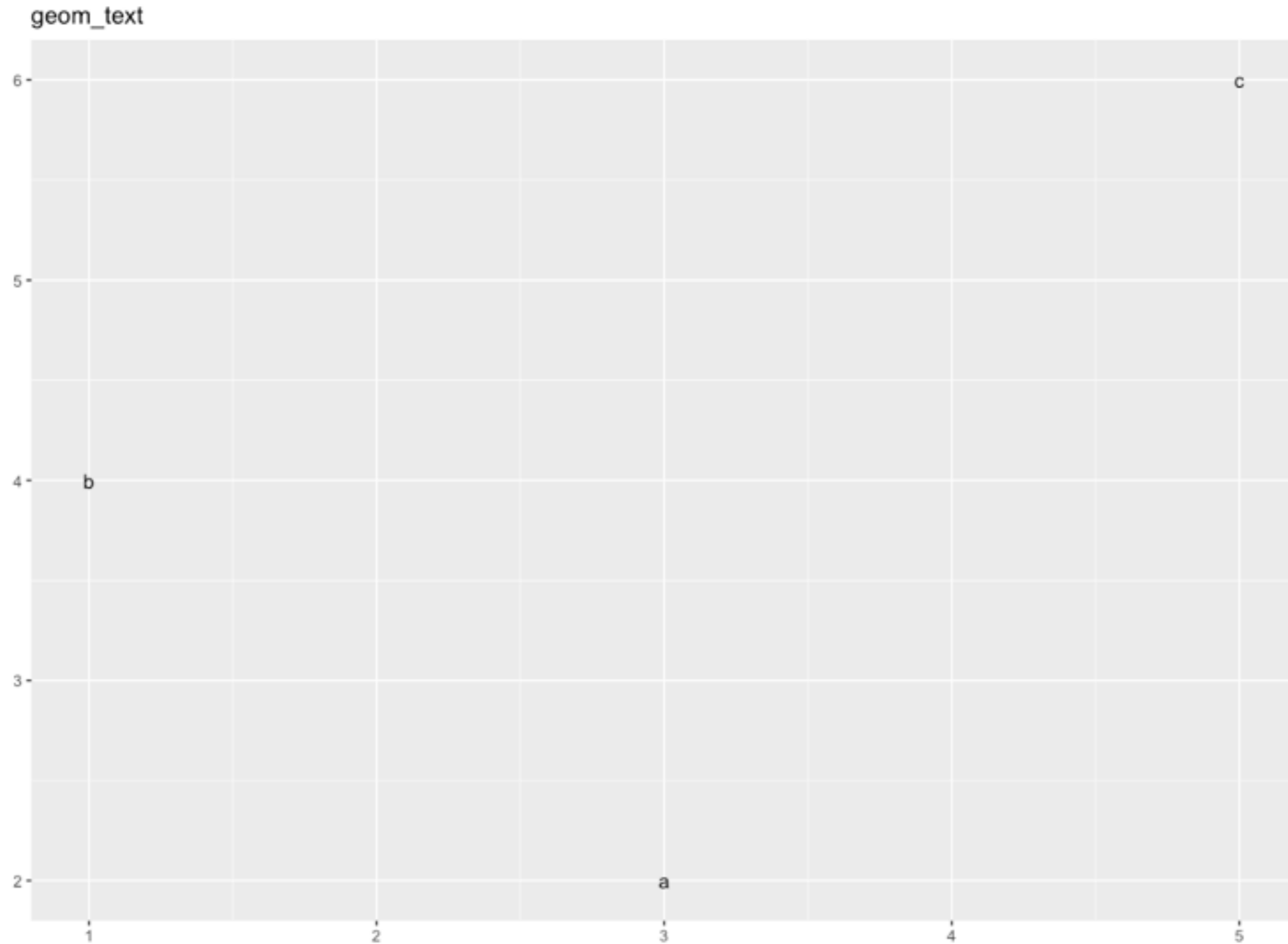
```
> p + geom_area() + labs(title = "geom_area")
```



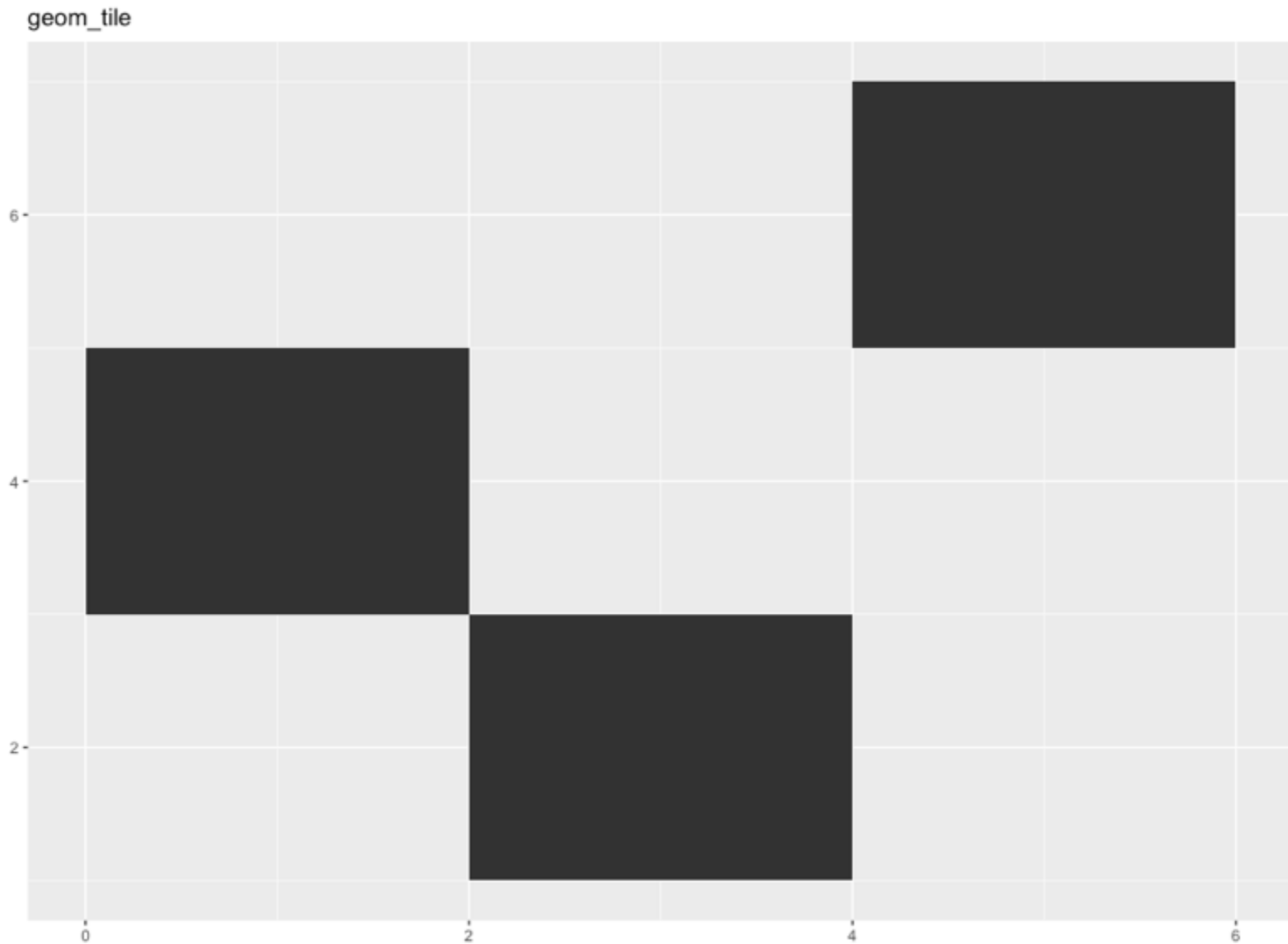
```
> p + geom_path() + labs(title = "geom_path")
```



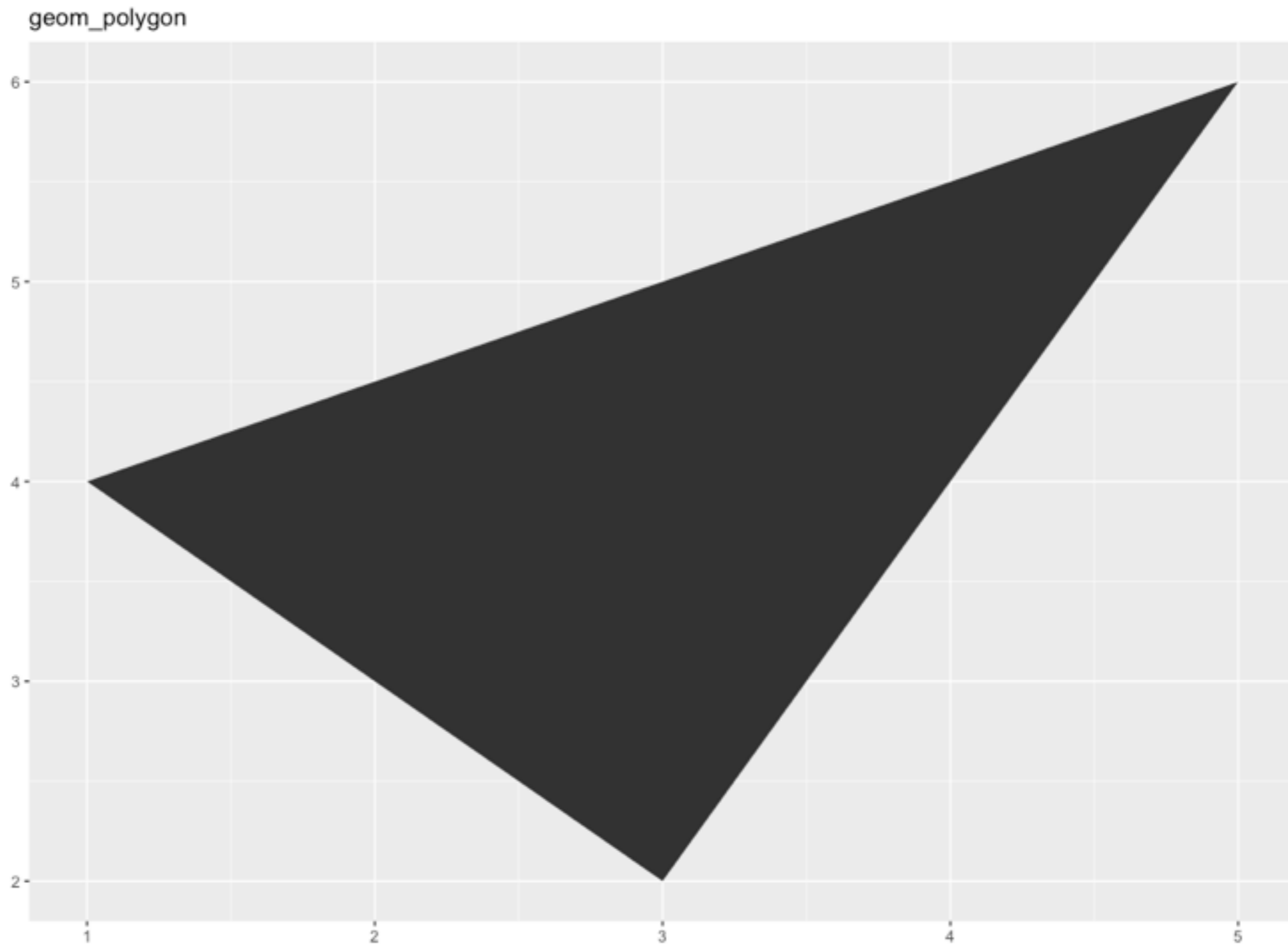

```
> p + geom_text() + labs(title = "geom_text")
```



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



```
> 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

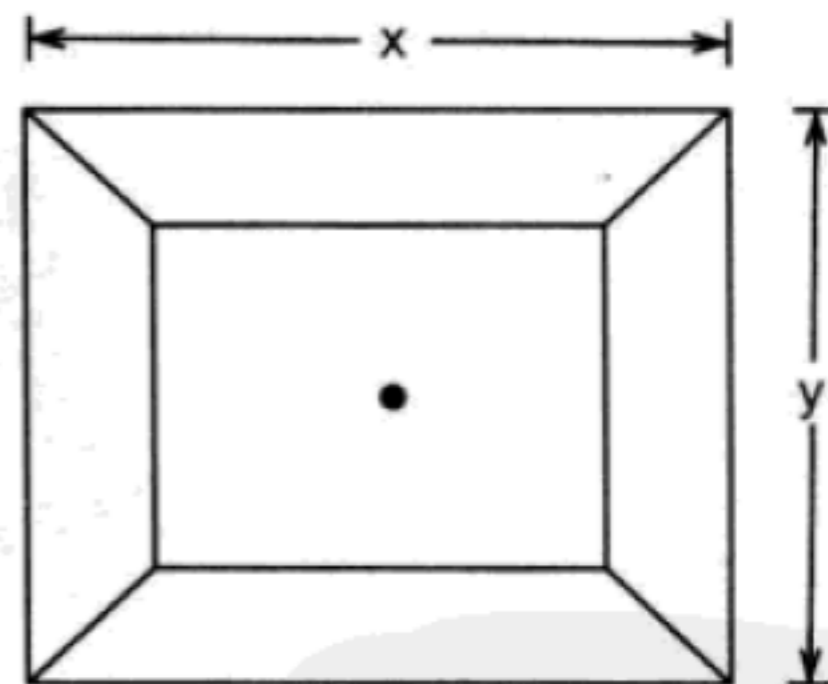
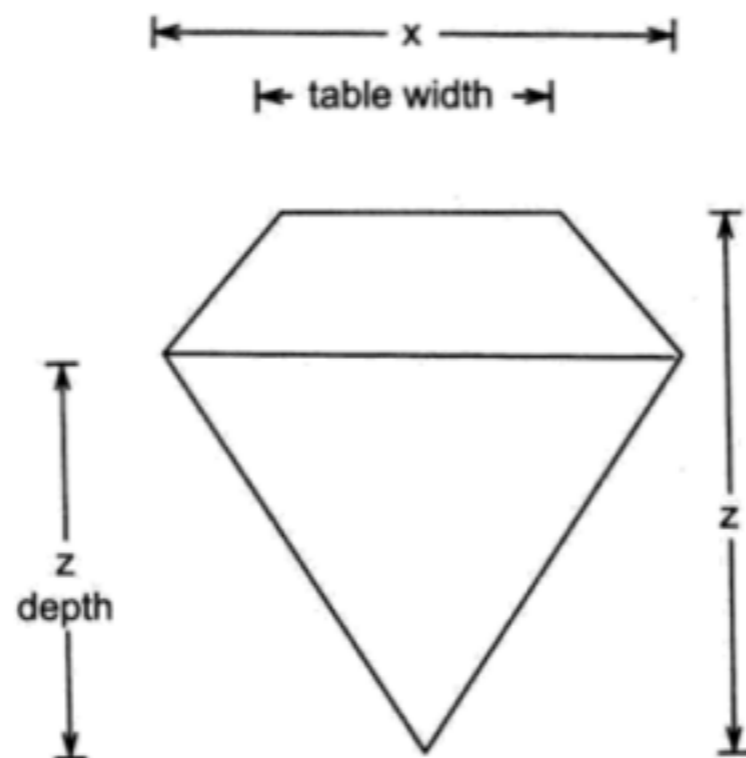
cut: 切工

color: 颜色

clarity: 净度

depty: 深度

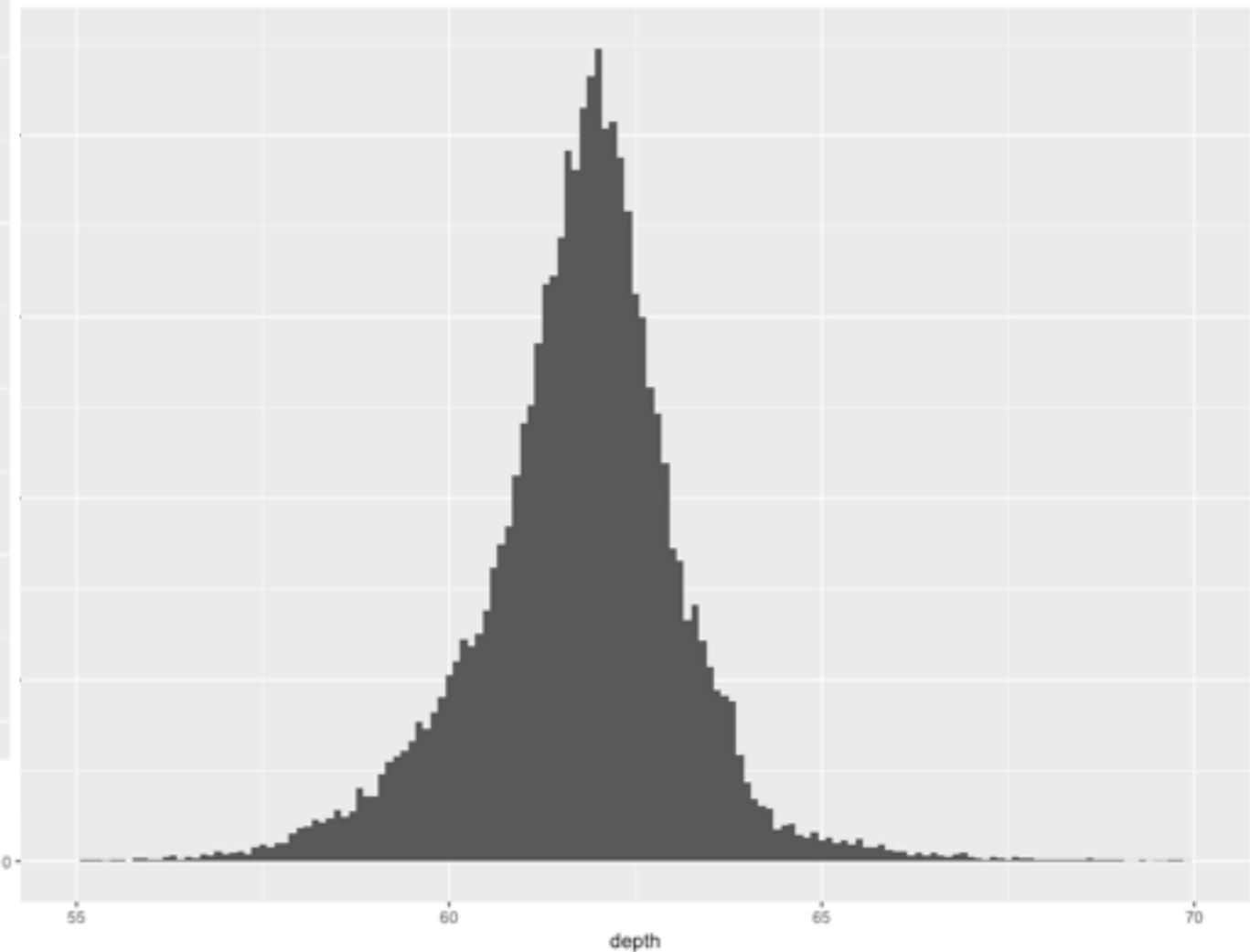
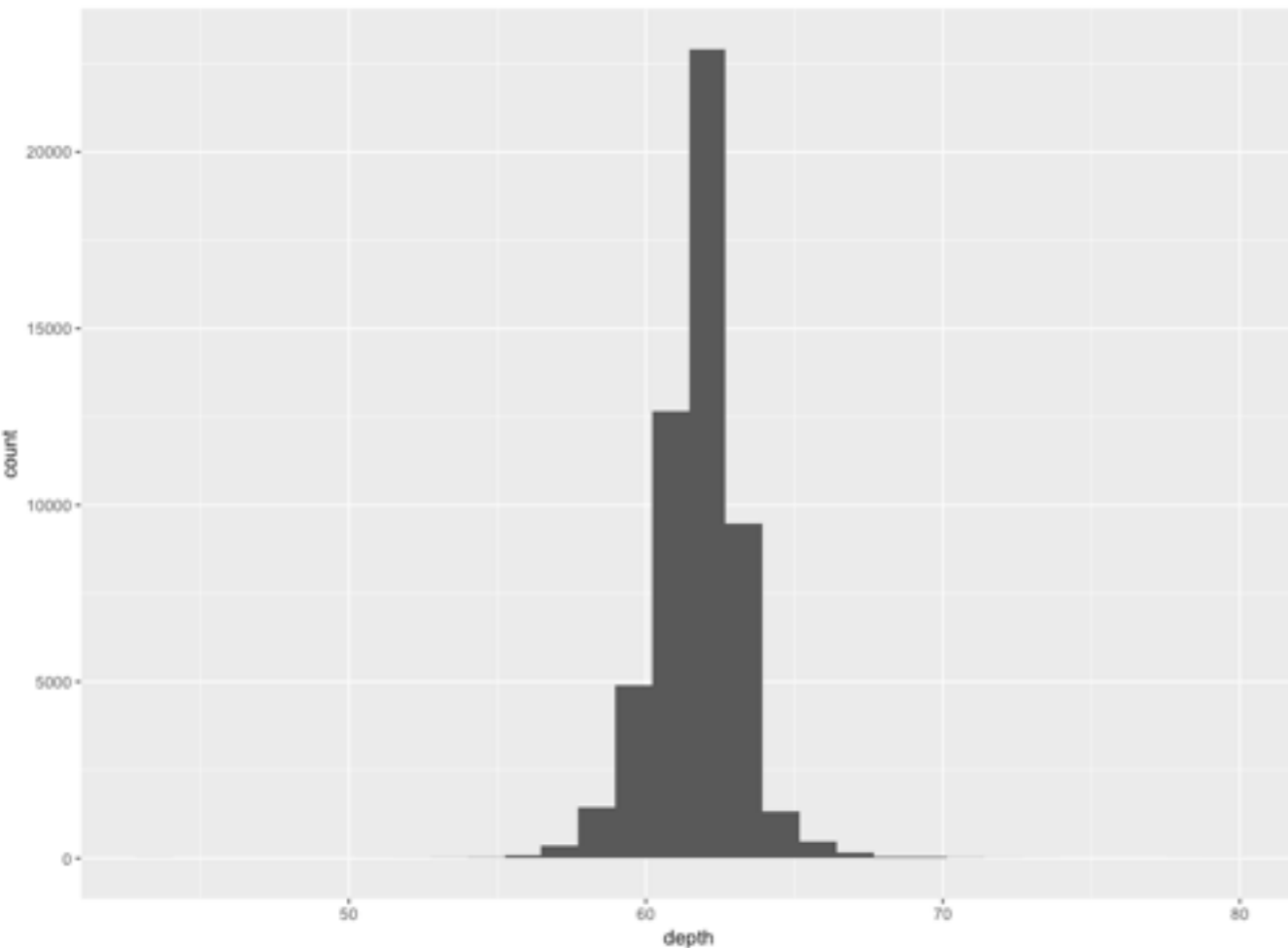
table: 钻面宽度



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

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

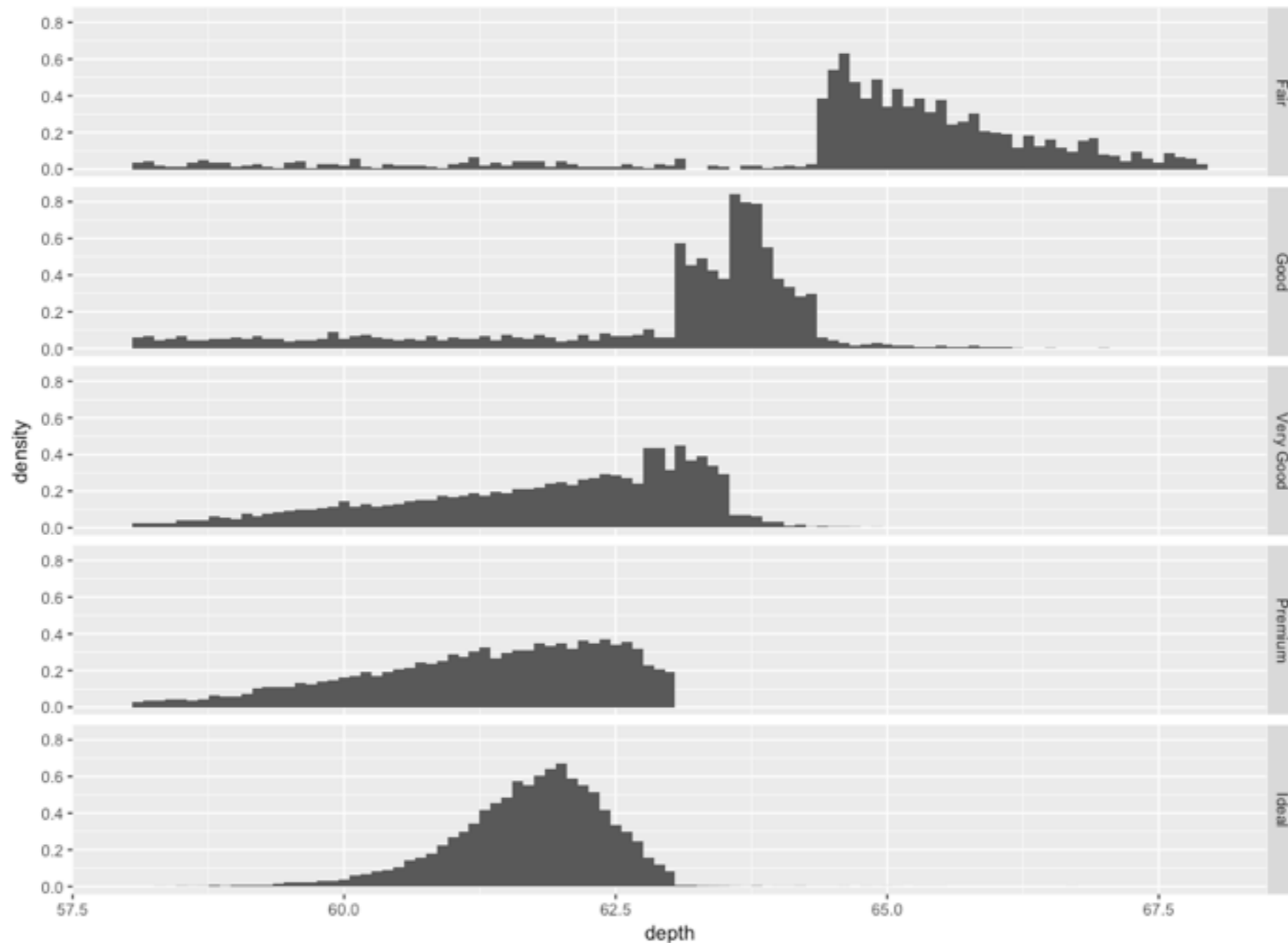
```
> qplot(depth, data=diamonds, geom="histogram")
```



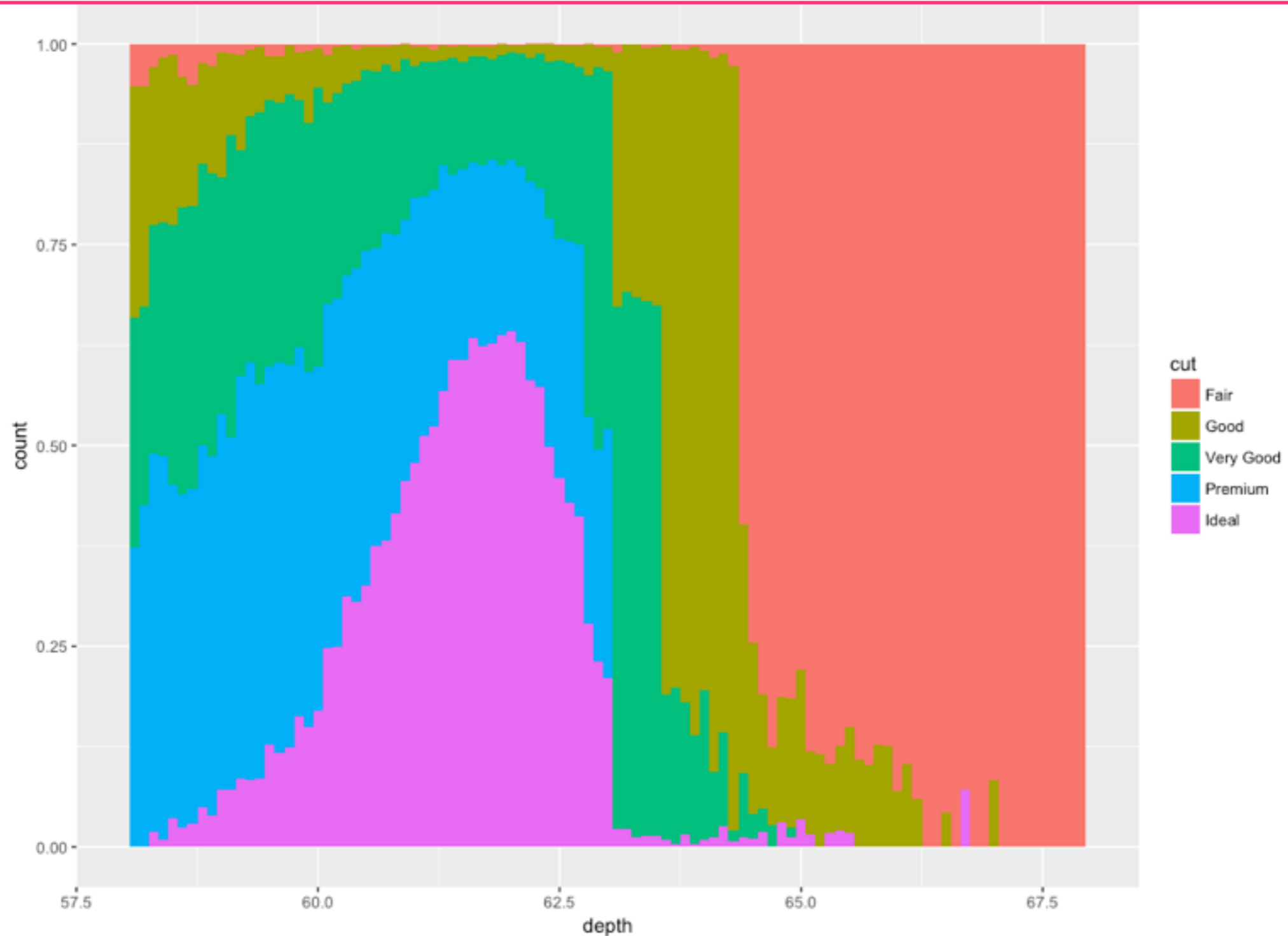
```
> qplot(depth, data=diamonds, geom="histogram", xlim=c(55, 70), binwidth=0.1)
```

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 ~ .)
```

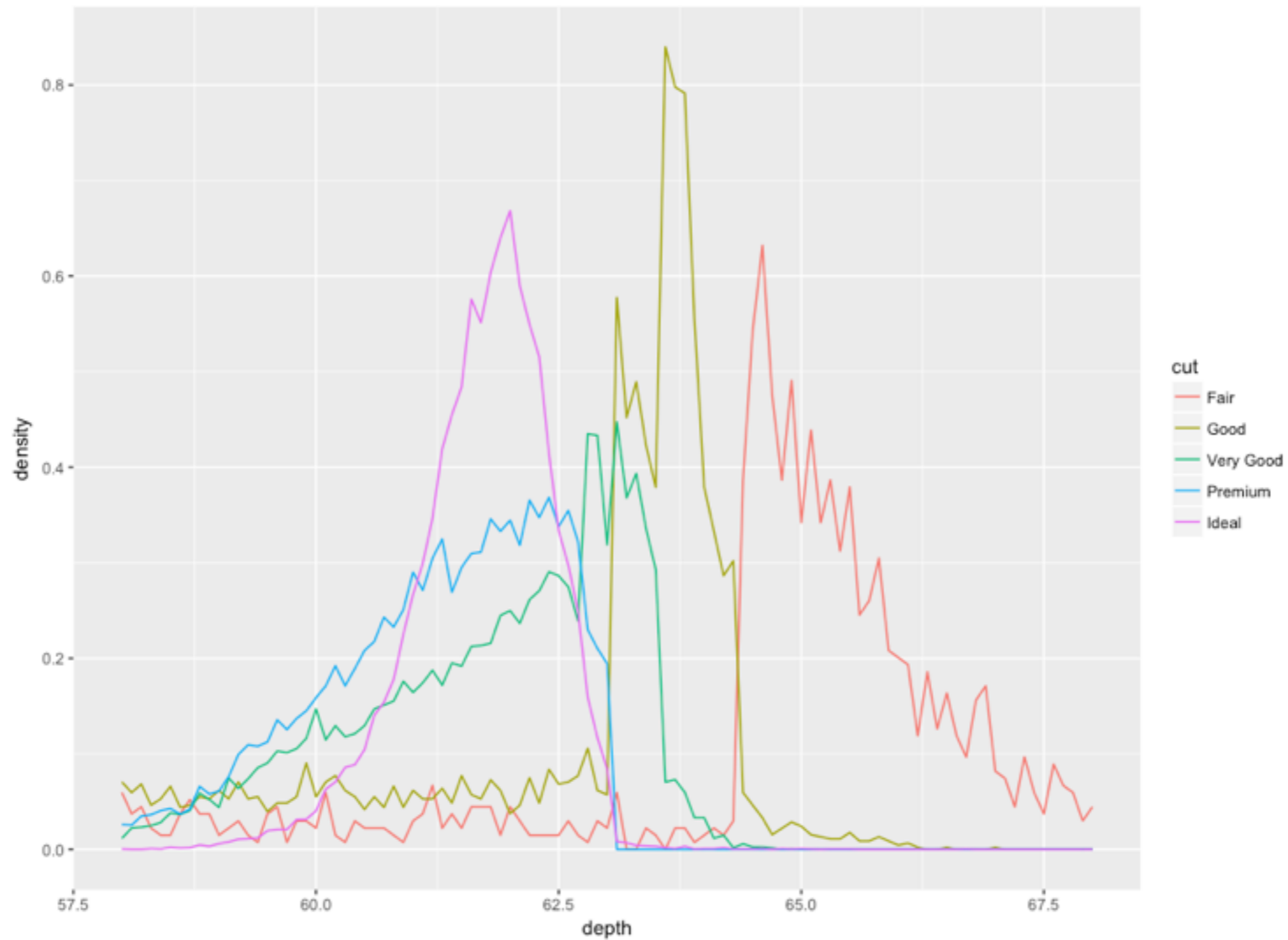


```
> depth_dist + geom_histogram(aes(fill = cut), binwidth = 0.1,  
+   position = "fill")
```



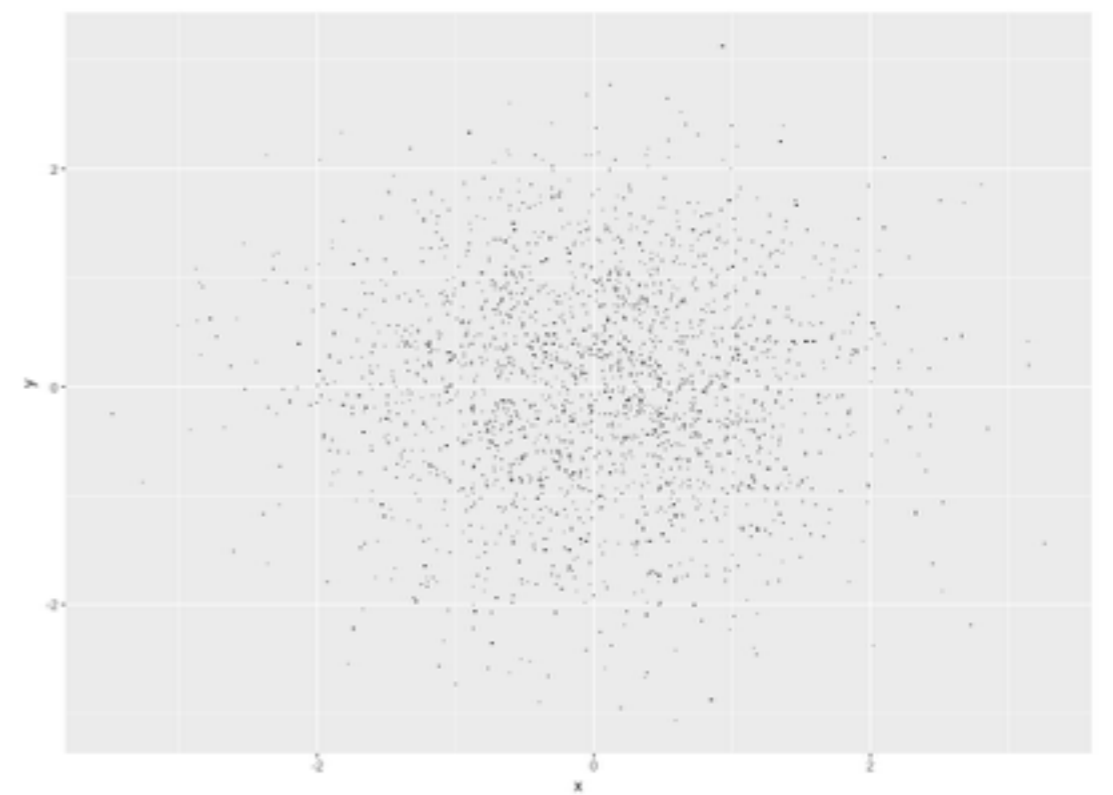
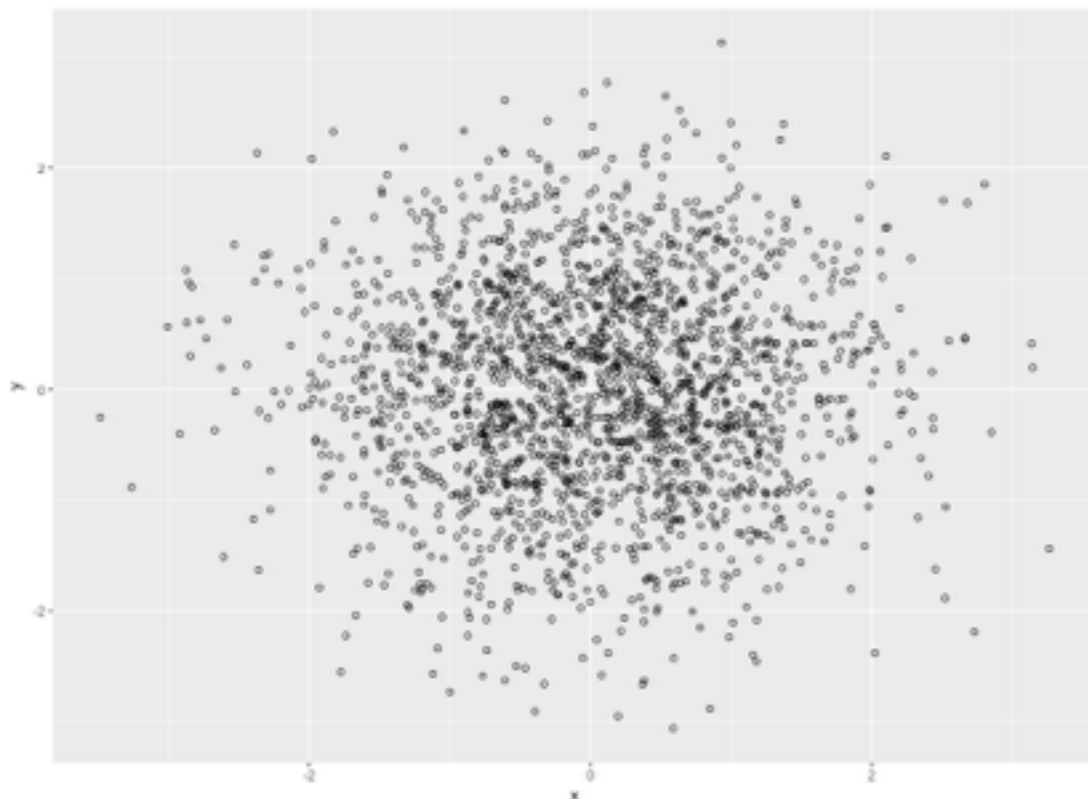
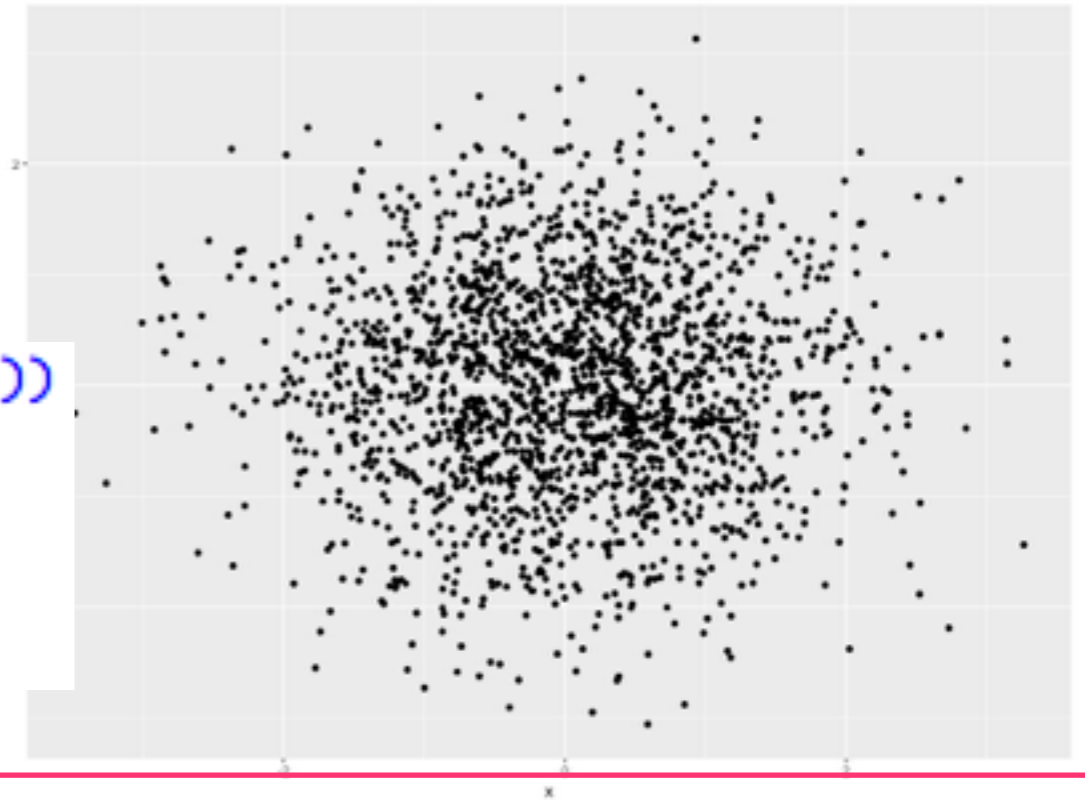
条件密度图

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



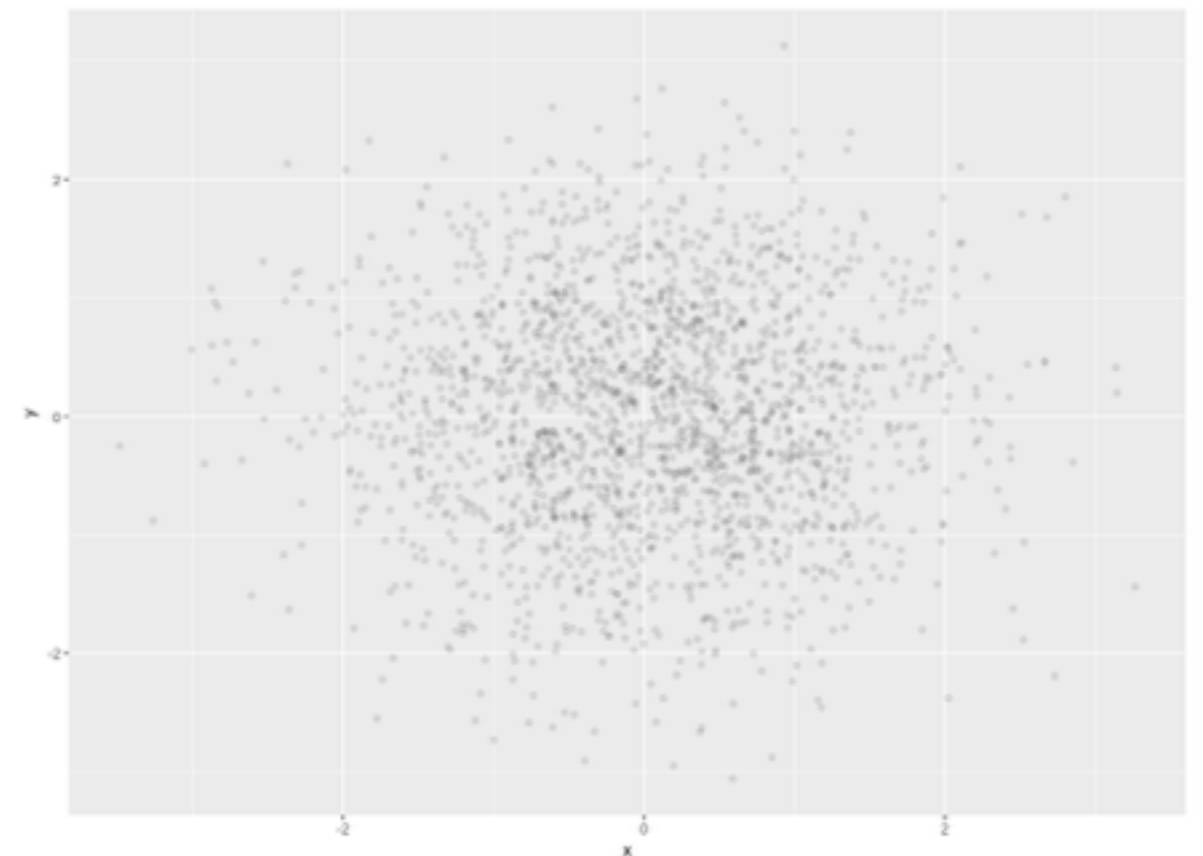
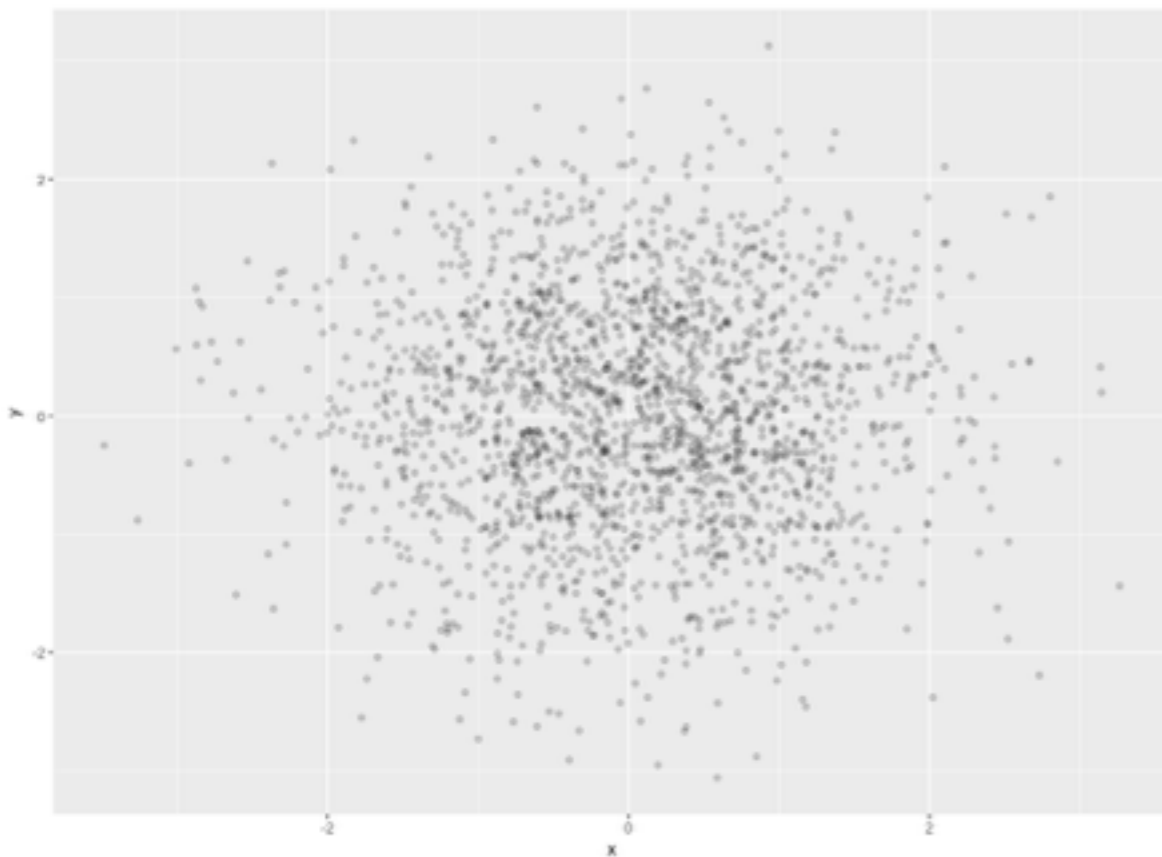
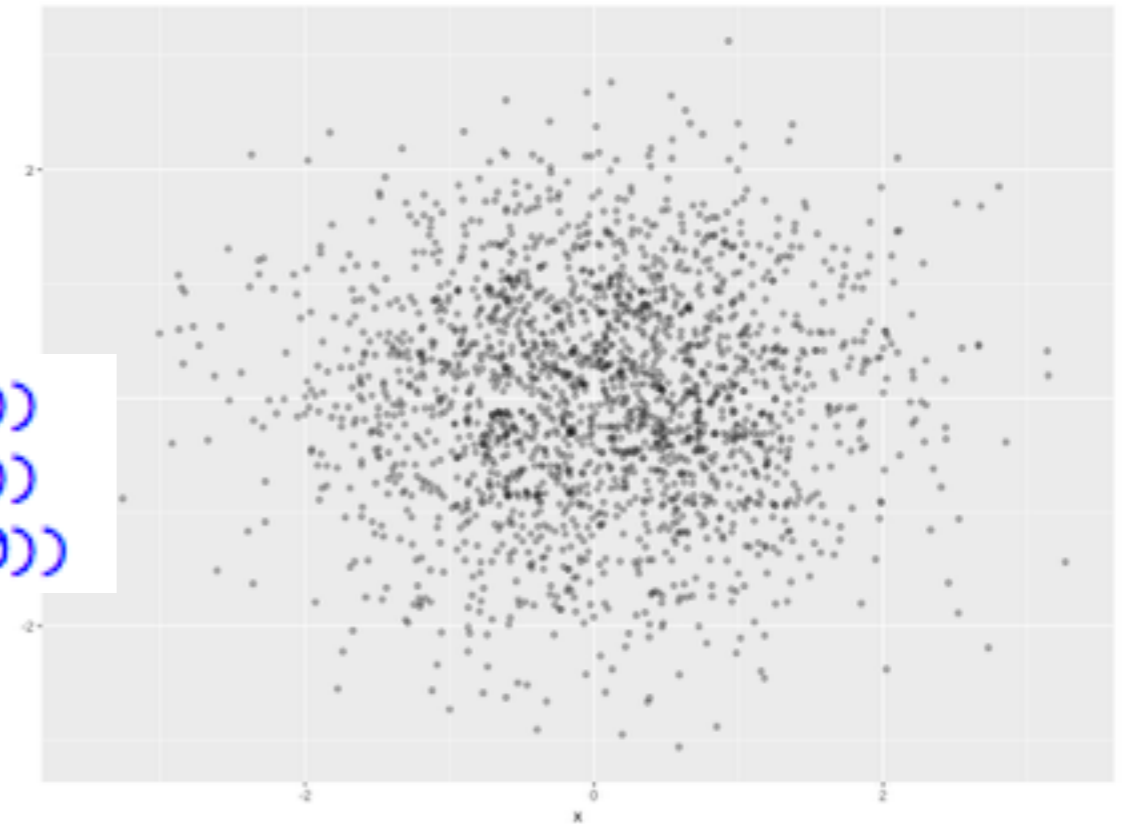
使用点的大小

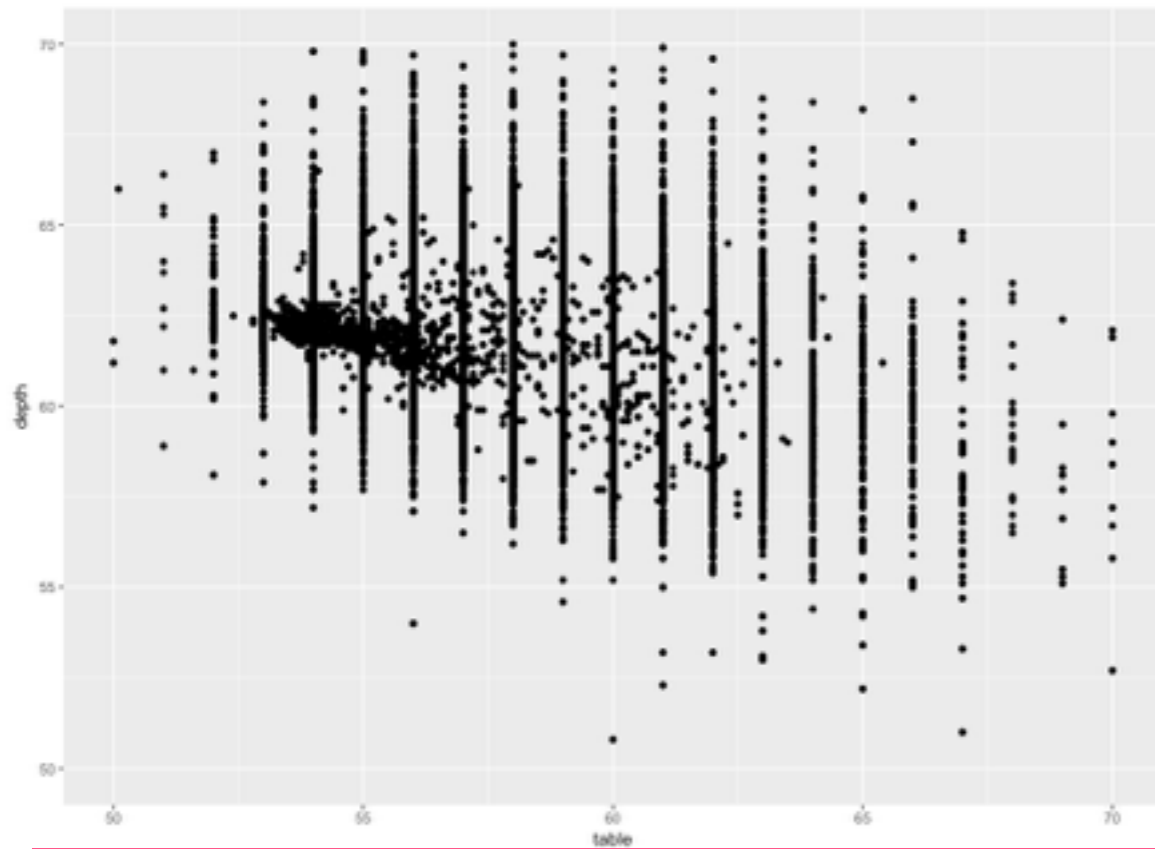
```
> df <- data.frame(x = rnorm(2000), y = rnorm(2000))  
> norm <- ggplot(df, aes(x, y))  
> norm + geom_point()  
> norm + geom_point(shape = 1)  
> norm + geom_point(shape = ".") # Pixel sized
```



使用点的透明度

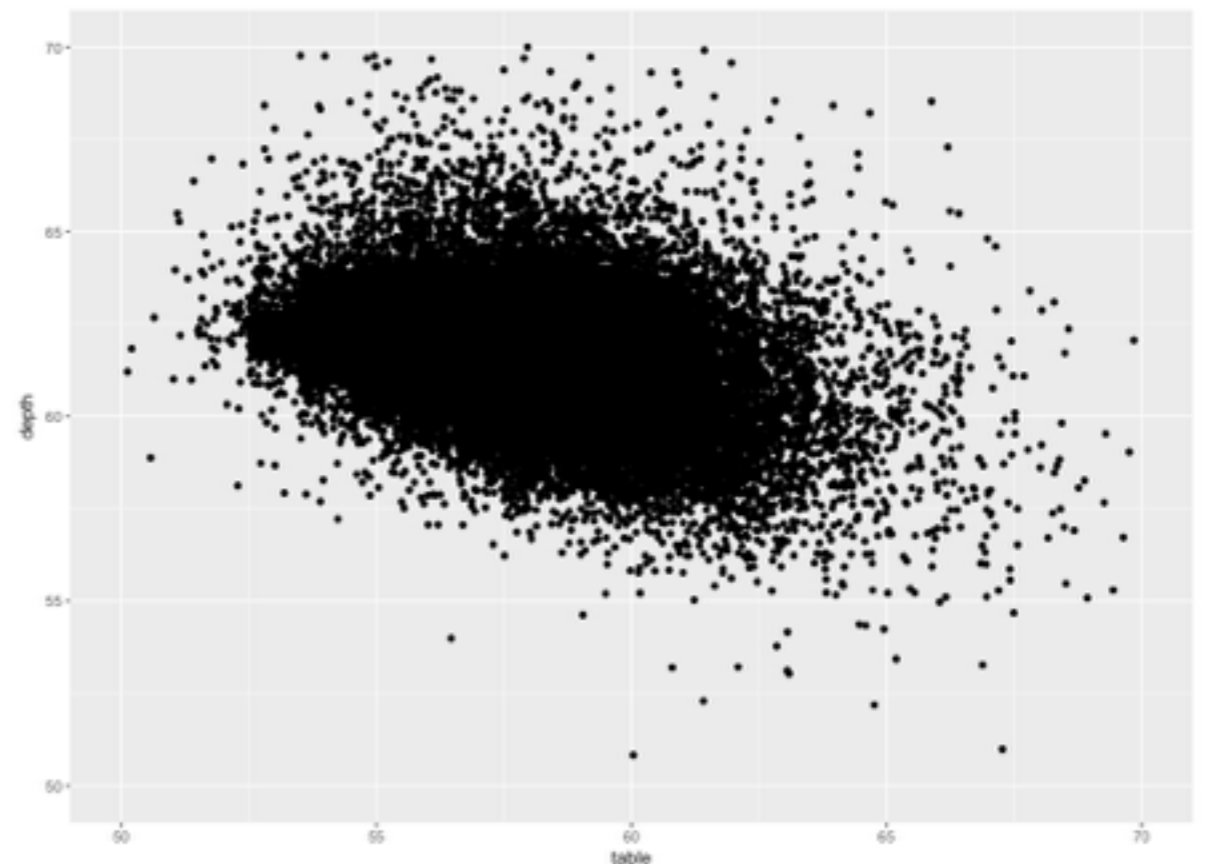
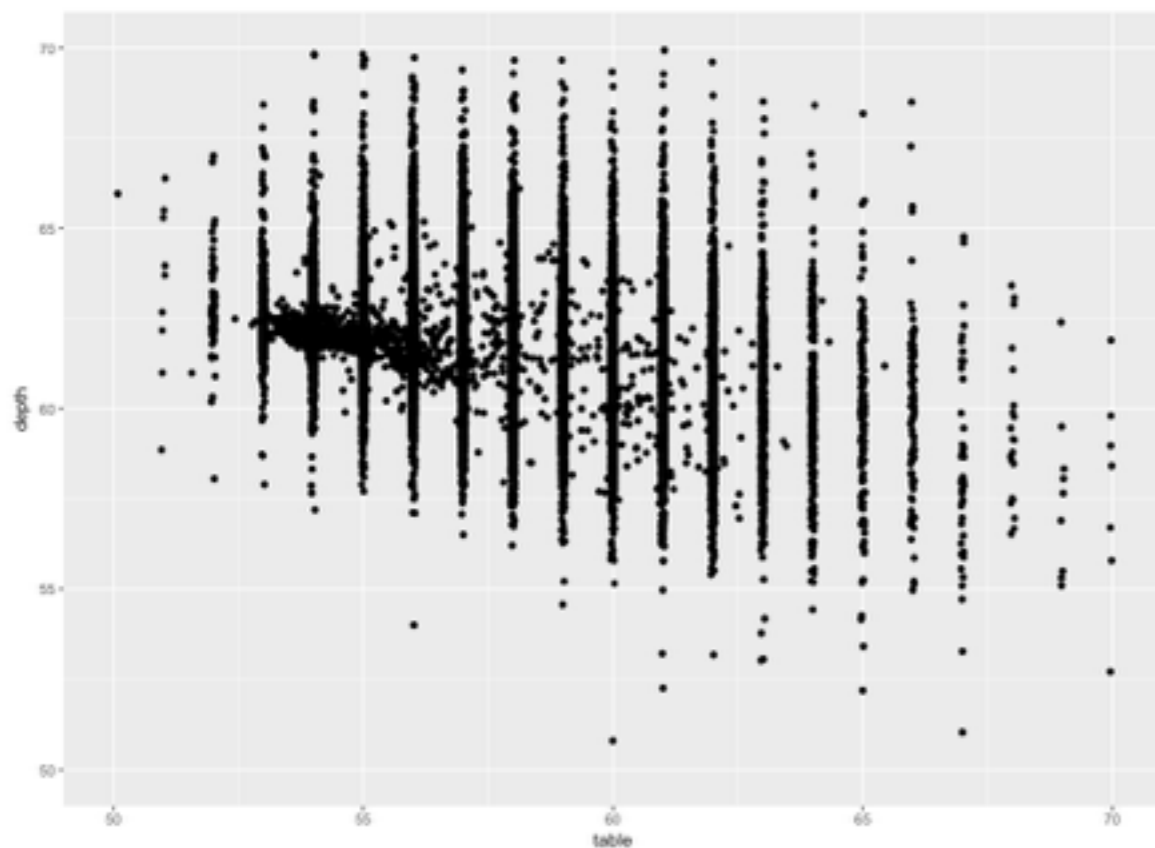
```
> norm + geom_point(colour = alpha("black", 1/3))  
> norm + geom_point(colour = alpha("black", 1/5))  
> norm + geom_point(colour = alpha("black", 1/10))
```

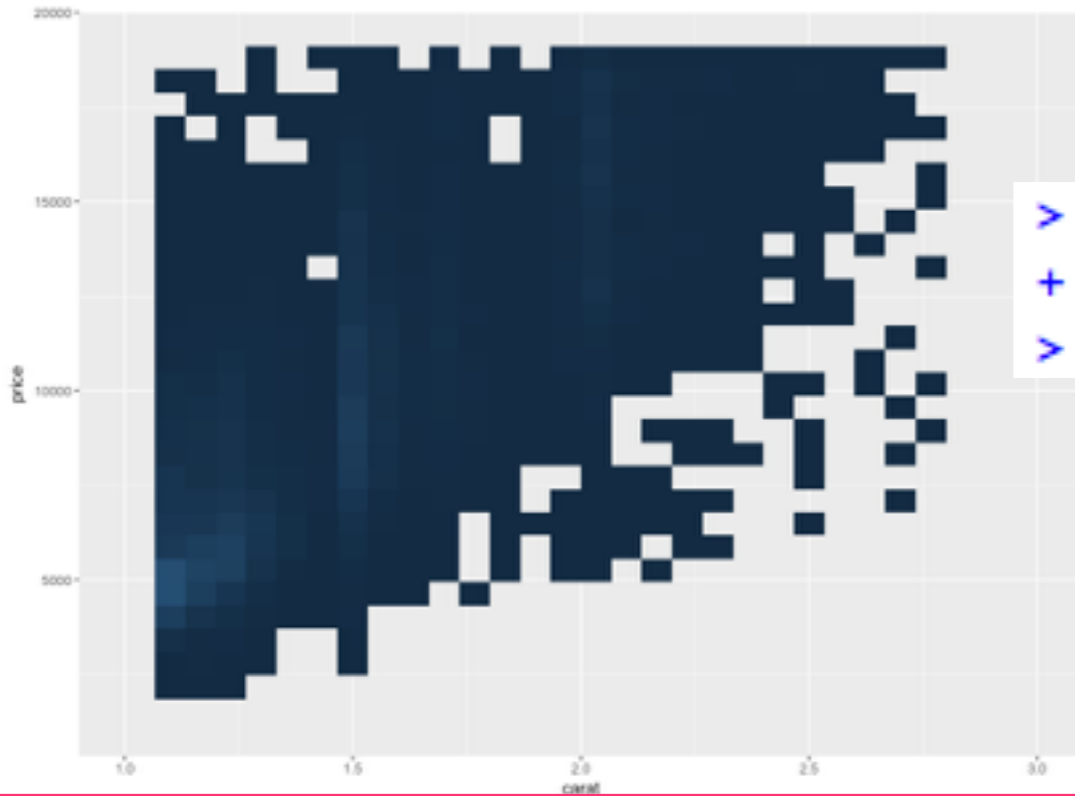




使用随机扰动

```
> 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)
```



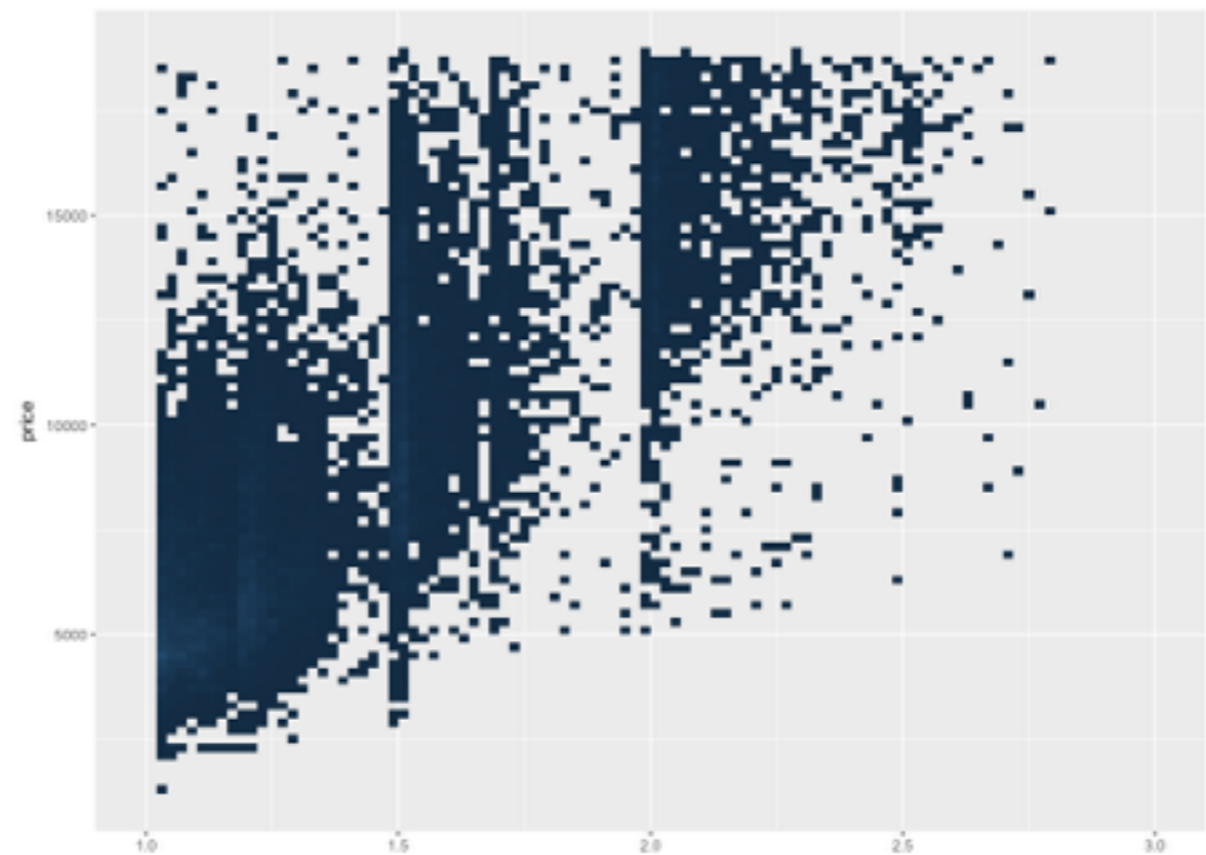
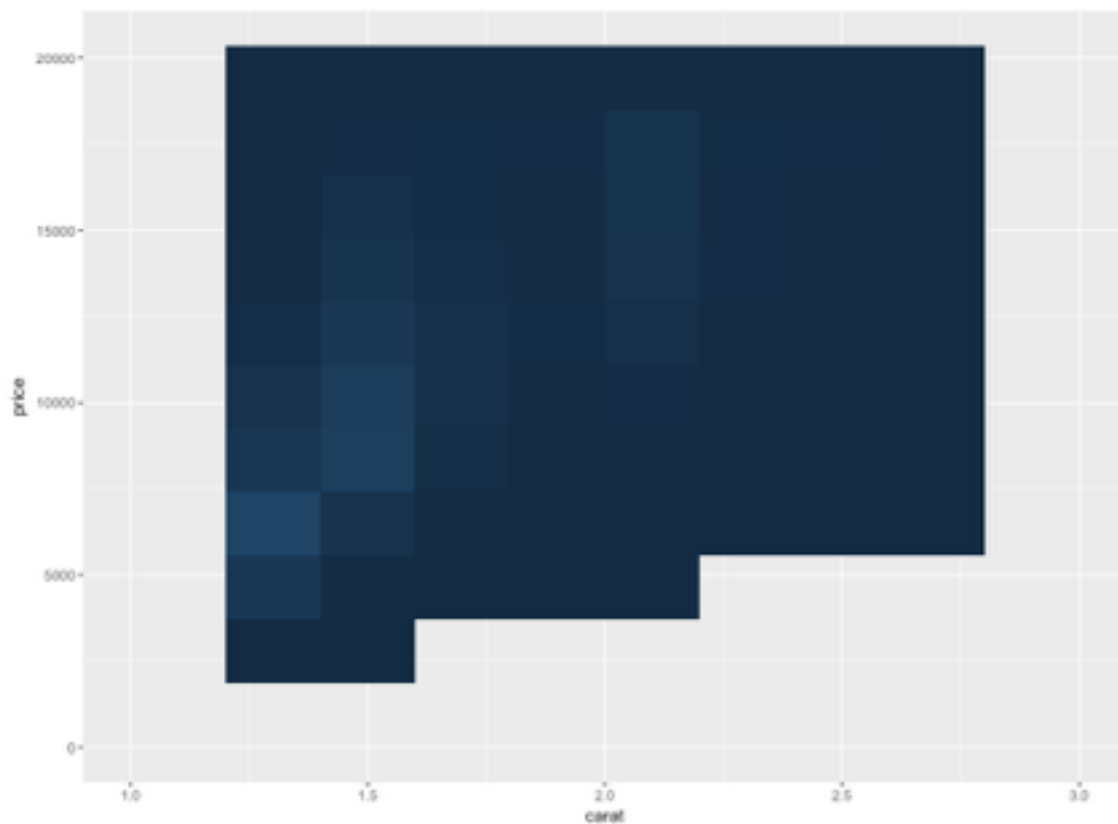


使用分箱计数

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

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

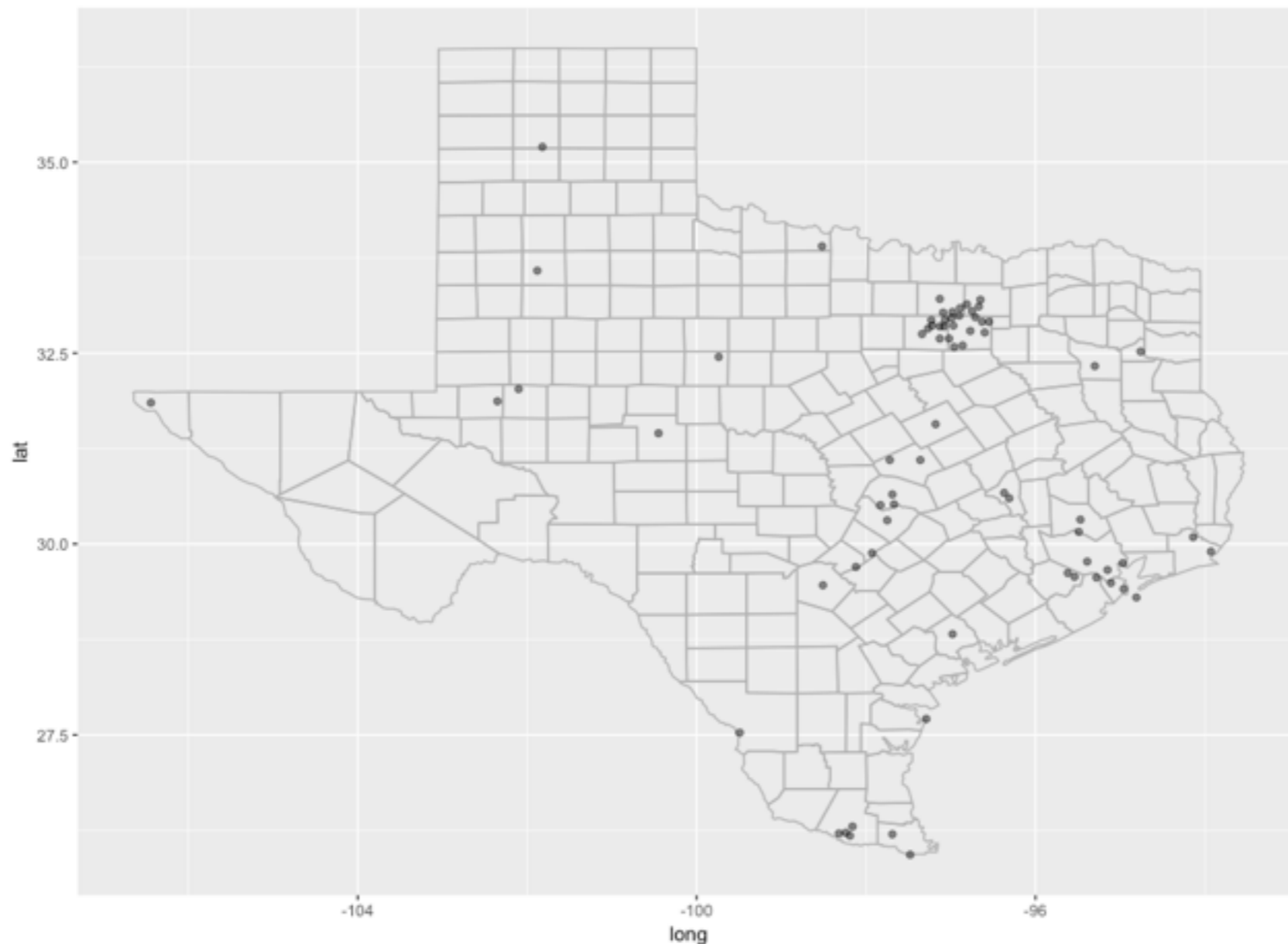
```
> d + stat_bin2d(binwidth=c(0.02, 200))
```



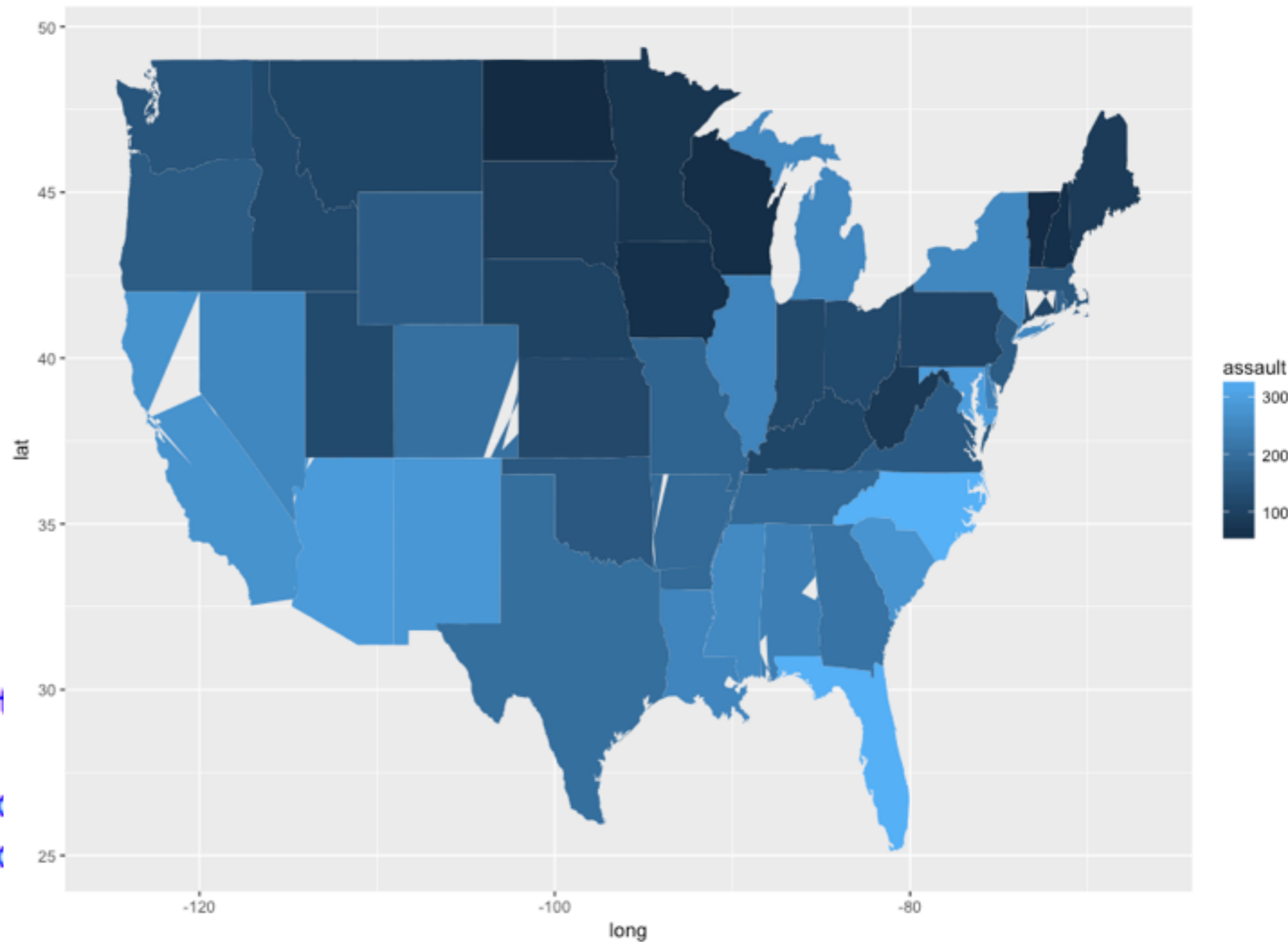

```
> library(maps)
> data(us.cities)
> big_cities <- subset(us.cities, pop > 500000)
> qplot(long, lat, data = big_cities) + borders("state", size = 0.5)
```



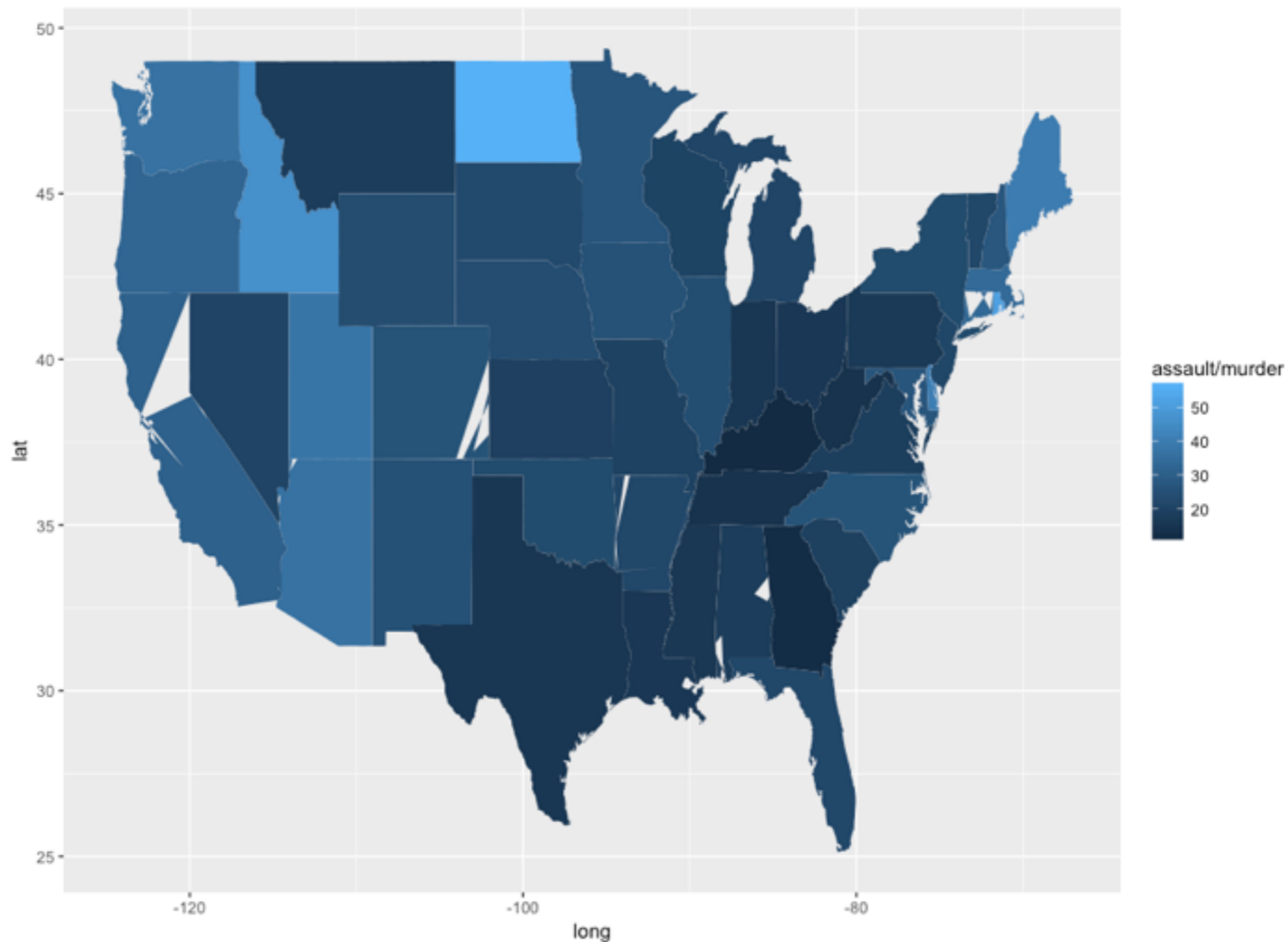
```
> tx_cities <- subset(us.cities, country.etc == "TX")  
> ggplot(tx_cities, aes(long, lat)) +  
+   borders("county", "texas", colour = "grey70") +  
+   geom_point(colour = alpha("black", 0.5))
```



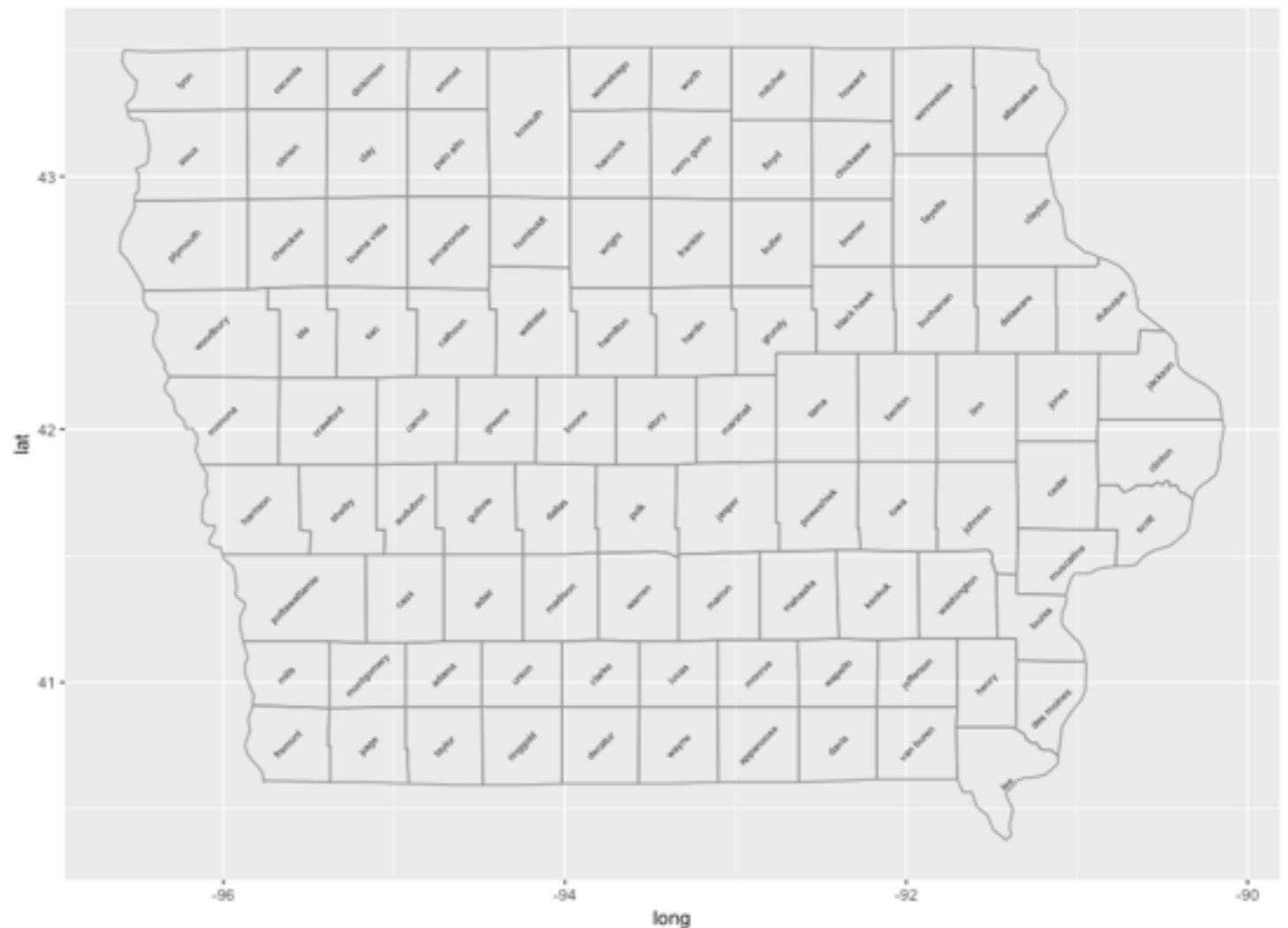
```
> library(maps)
> states <- map_data("state")
> arrests <- USArrests
> names(arrests) <- tolower(names(arrests))
> arrests$region <- tolower(USArrests$region)
>
> choro <- merge(states, arrests, by = "region")
> choro <- choro[order(choro$order), ]
> qplot(long, lat, data = choro, group = group,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



标度、坐标系和图例

定位

提问时间!

孙惠平

sunhp@ss.pku.edu.cn

练习

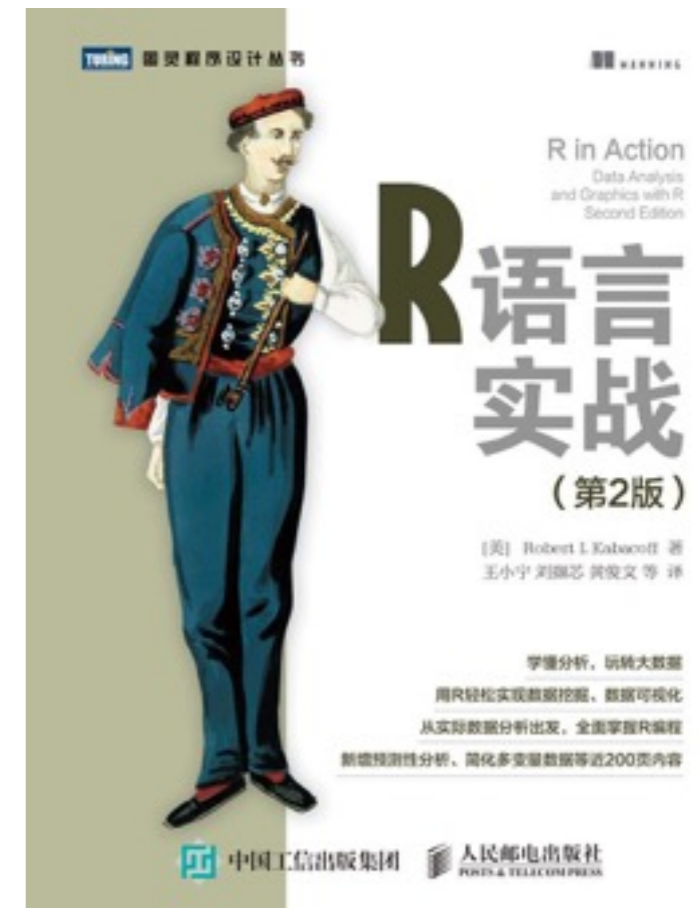
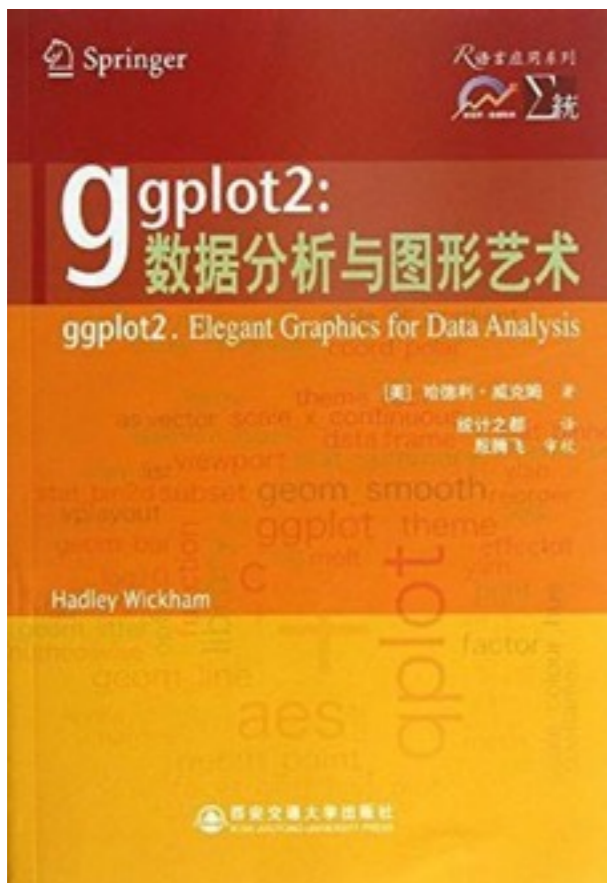


Learn R, in R.

swirl teaches you R programming and data science
interactively, at your own pace, and right in the R console!

`install_course_github("pkusdatanalysis", "C9_GGplot_02")`

- 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



谢谢!

孙惠平

sunhp@ss.pku.edu.cn