

ggplot2-01



课堂测试时间

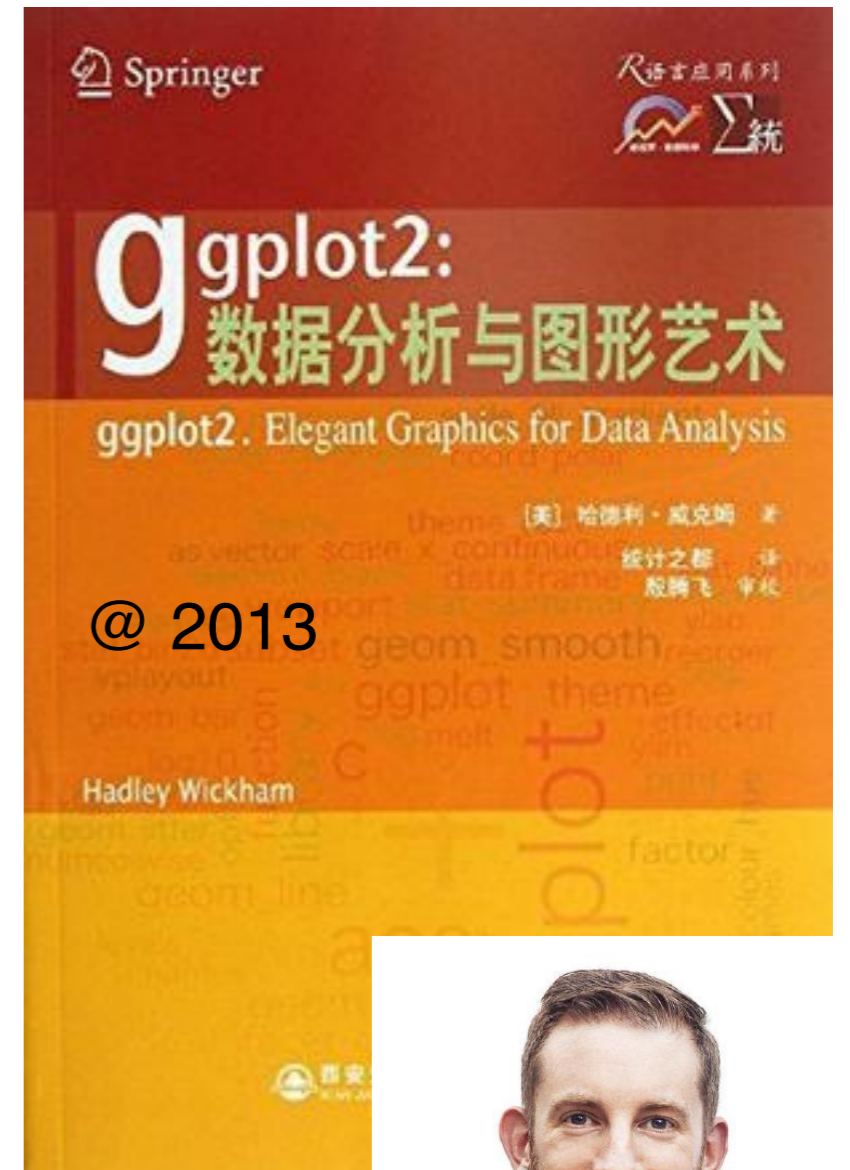
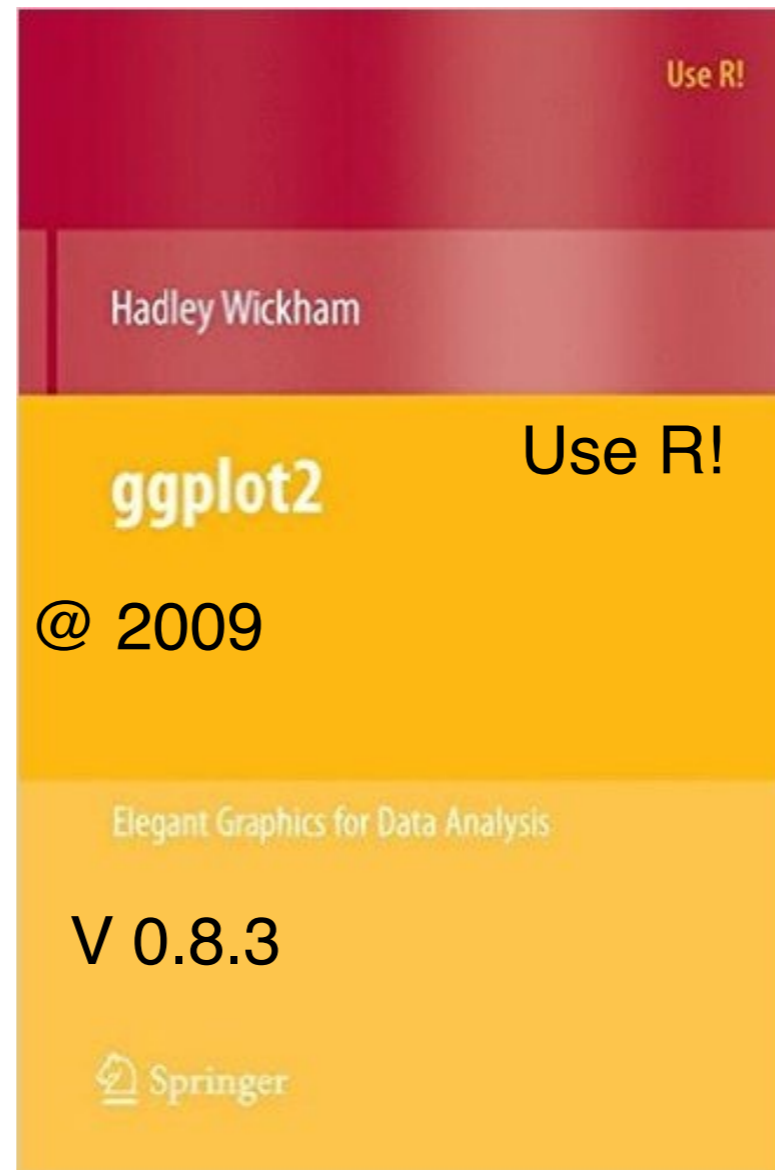
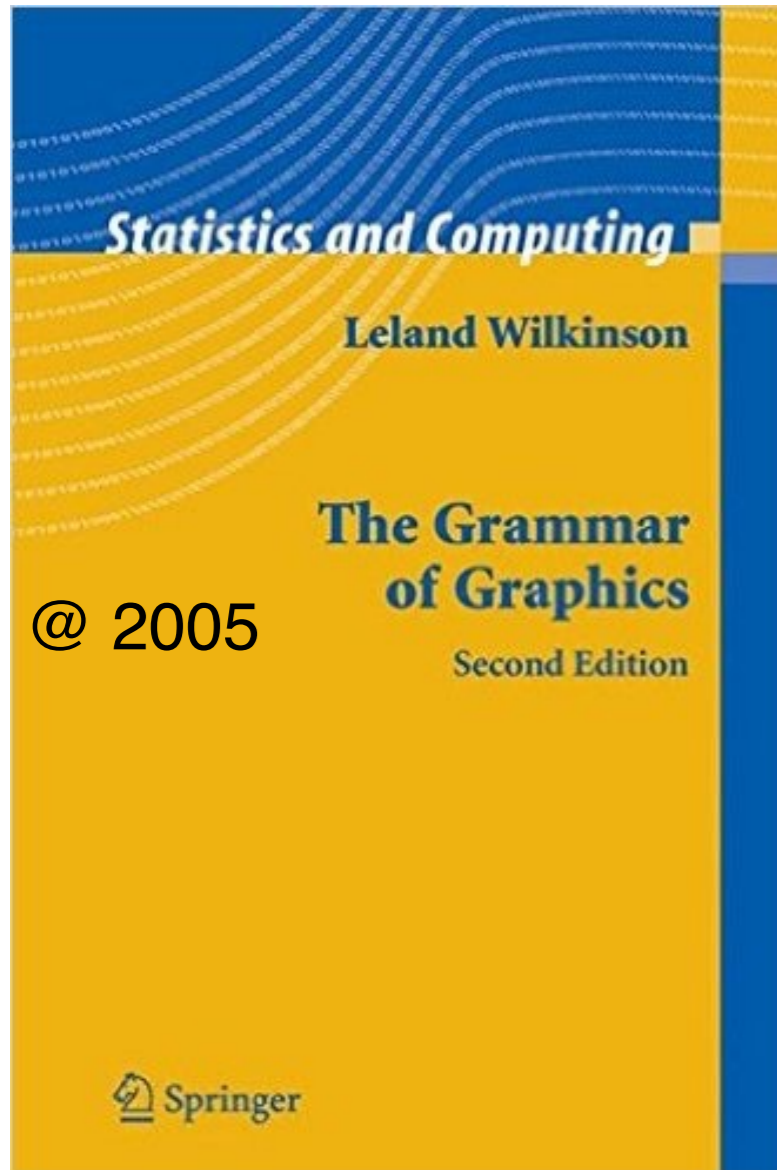
- [dapengde_DummyR_PM25.csv](#)是2003年8月在北京城区的三个高度（8米，100米，325米）测得的PM2.5的质量浓度日变化的统计数据，共4列25行。
 - ➔ 请画出一条折线表示h8和time的关系，要求是"time"和"pm2.5"分别是x轴的名称和y轴的名称，lty=1（表示line的type为1，表示直线）y轴的范围是0到200。
 - ➔ 在上图增加一条折线(使用lines()函数)表示h100和time的关系，要求颜色为红色，线型为虚线(lty=2)
 - ➔ 在上图中增加图例来表示上边画的两条折线，其中图例位置为（x=15，y=180）位置处，内容为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。

- 图例：
 - * 坐标; 边界标注; 标注(mar);horiz=TRUE;
- 线图：
 - * grid(); abline(); line(); lm(); arrows();
- 条形图：
 - * 堆积(beside); horiz=TRUE;
 - * 显示数字; 宽度、颜色和边界; 显示标注; 增加误差线
- 散点图：
 - * point();type="n"; corplot(); 增加抖动;
- 其余：
 - * par(); axis(); mtext(); jitter();

ggplot2 简介

<https://cran.r-project.org/web/packages/ggplot2/index.html>

V 3.3.0



- graphics、grid、lattice
- ggplot2

<http://hadley.nz/>



- 函数繁杂，语法复杂
- “笔纸”工作方式，不能增减
- 自动化低
- 主次不分

忘记一切

-
- 有理论基础，支持一套图形语法
 - 采用图层的设计方式，可增减
 - 媲美商业数据化软件的作图效果
 - 使用简单，定制容易（主题）

从头开始

```
install.packages("ggplot2")
```

- 数据 (data)
↕
映射 (mapping) ↔ 图形属性 (aesthetic attributes)
-

- 几何对象 (geometric object)
- 统计变换 (statistical transformation **s**)
- 标度 (scale)
- 坐标系 (coordinate system)
- 分面 (facet)

qplot

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

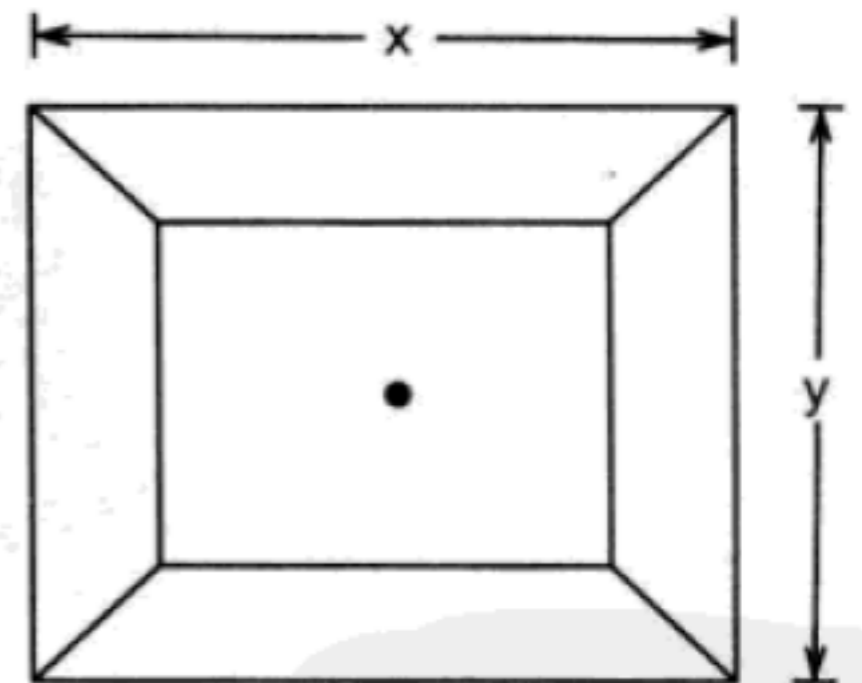
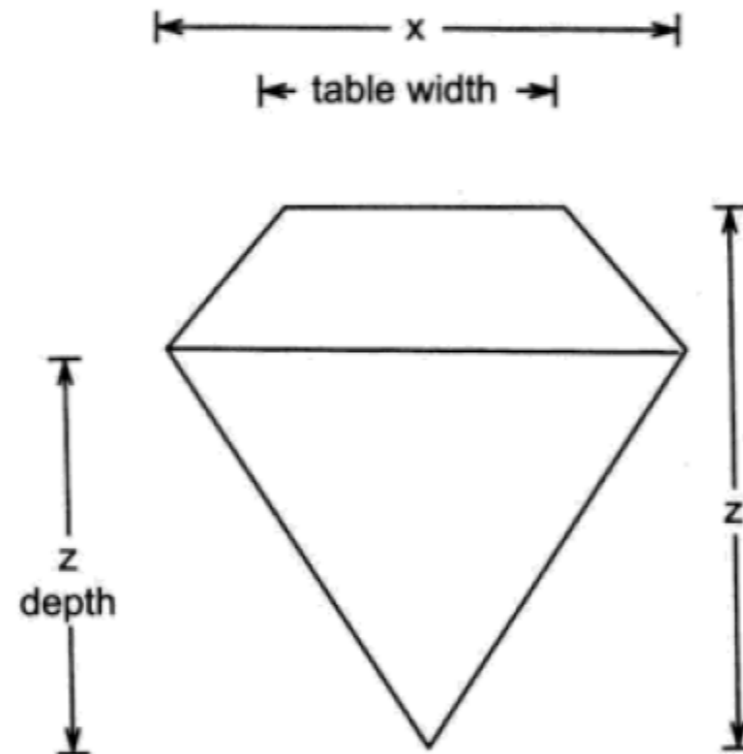
cut: 切工

color: 颜色

clarity: 净度

depty: 深度

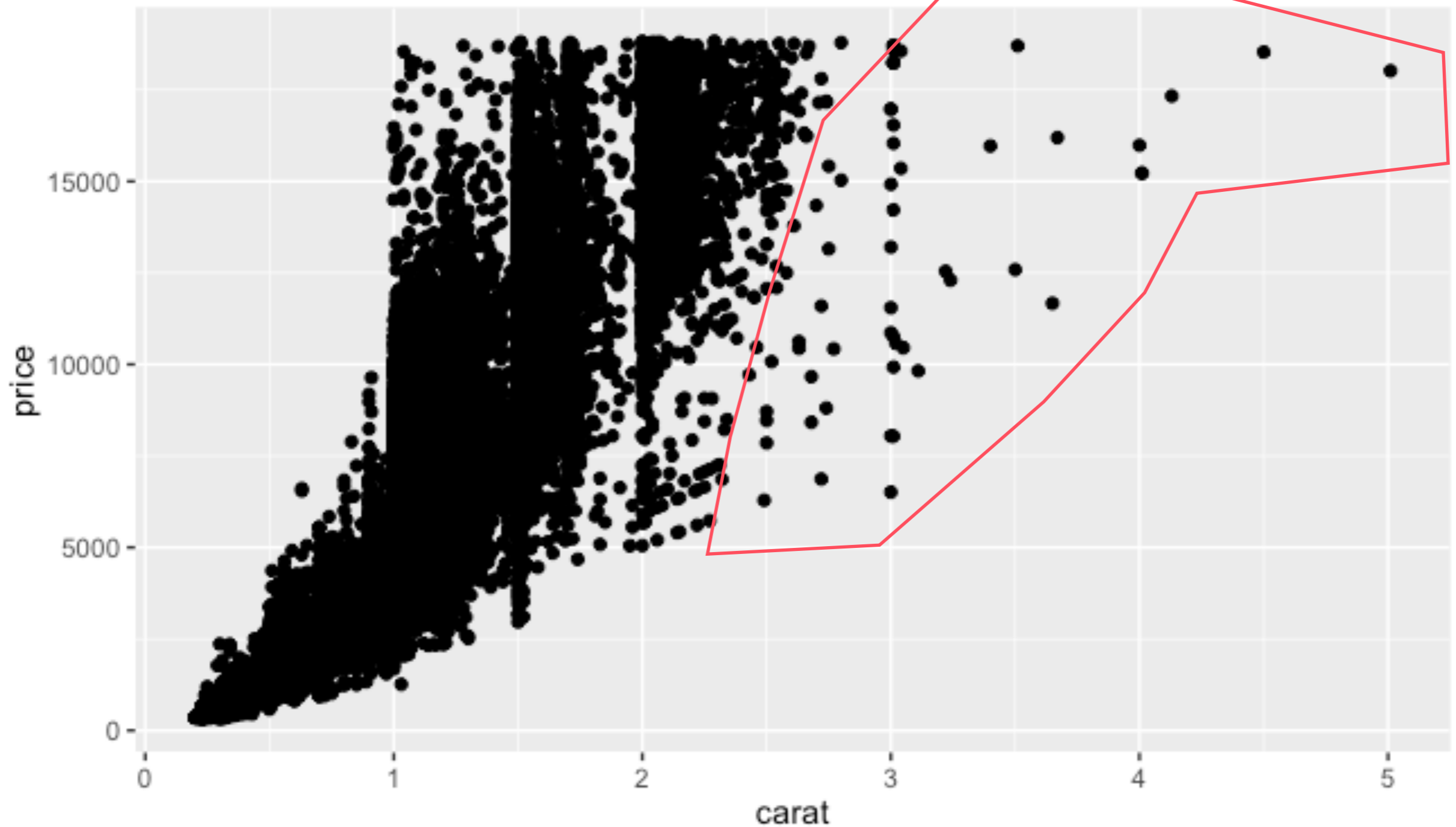
table: 钻面宽度



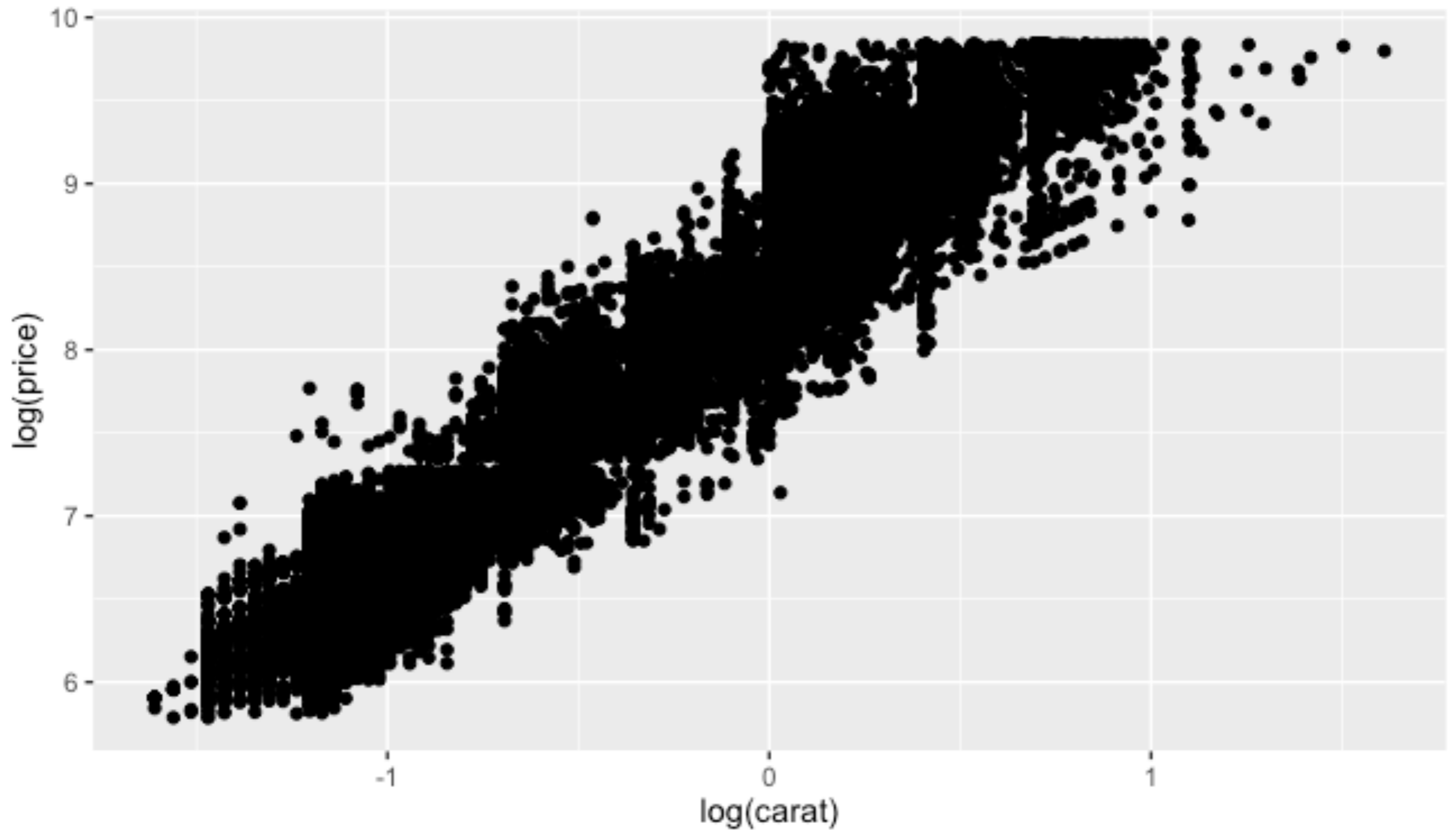
$$\text{depth} = z \text{ depth} / z * 100$$

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

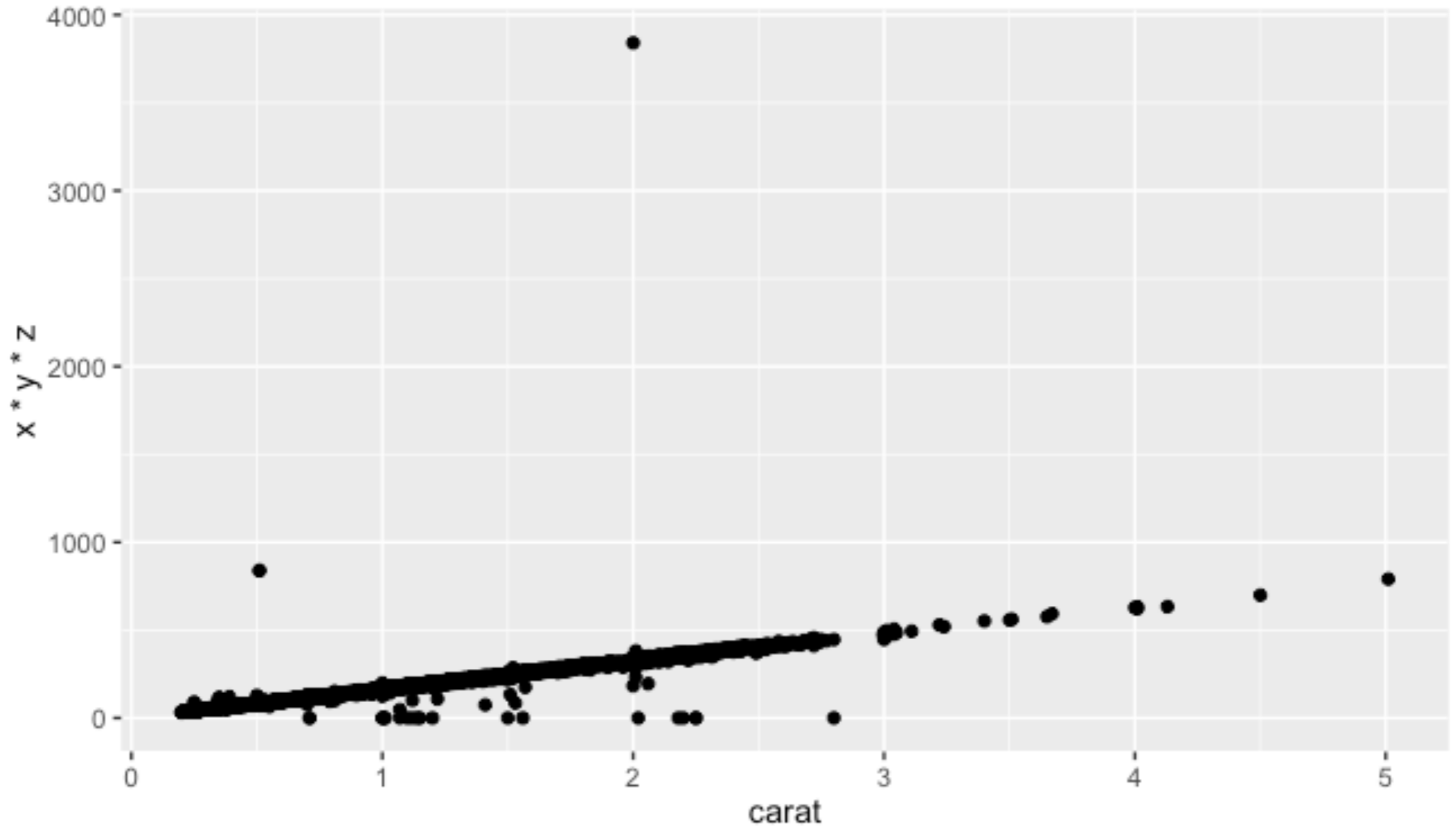
`qplot(carat, price, data = diamonds)`



```
qplot(log(carat), log(price), data = diamonds)
```

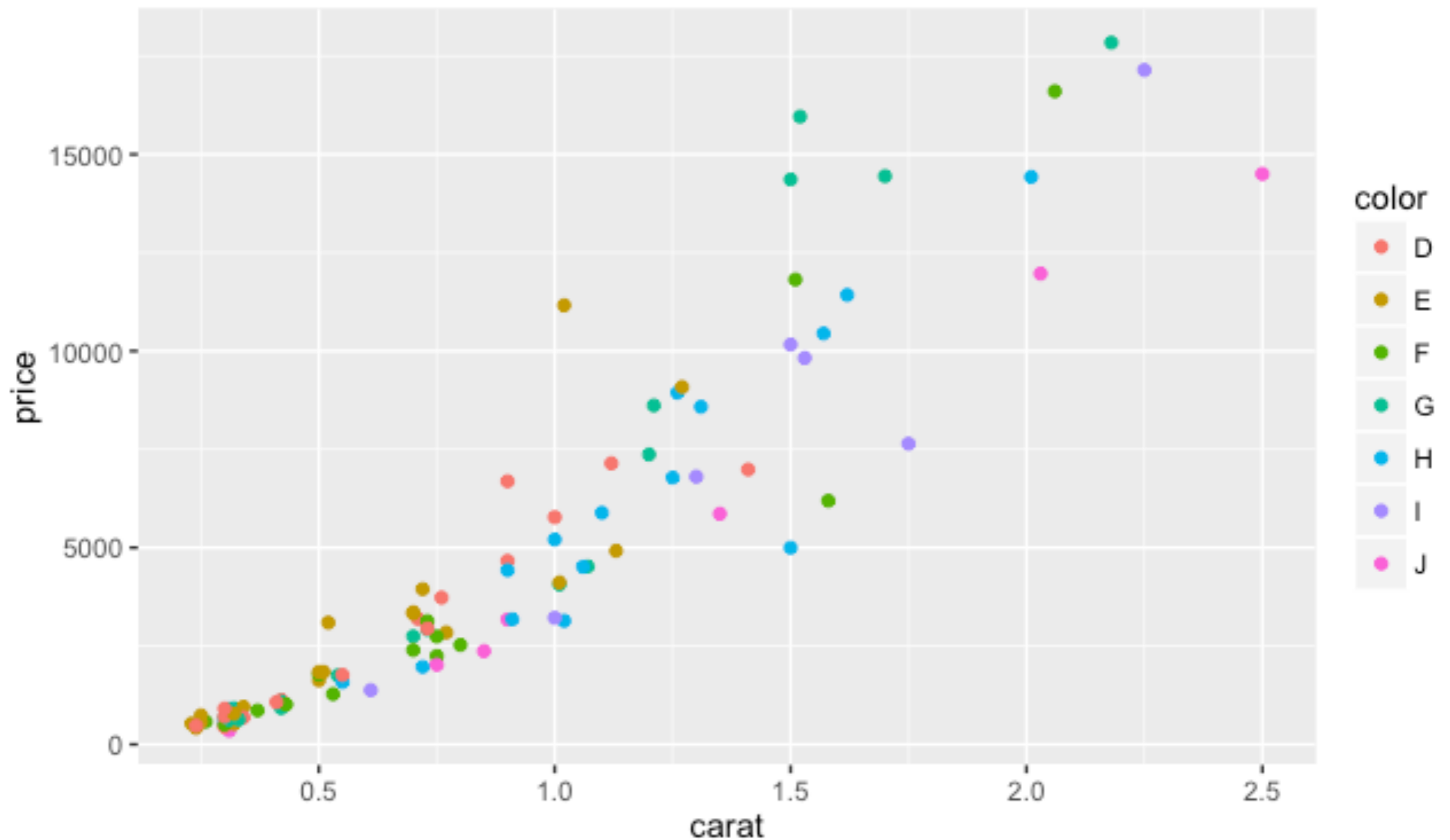


```
qplot(carat, x * y * z, data = diamonds)
```

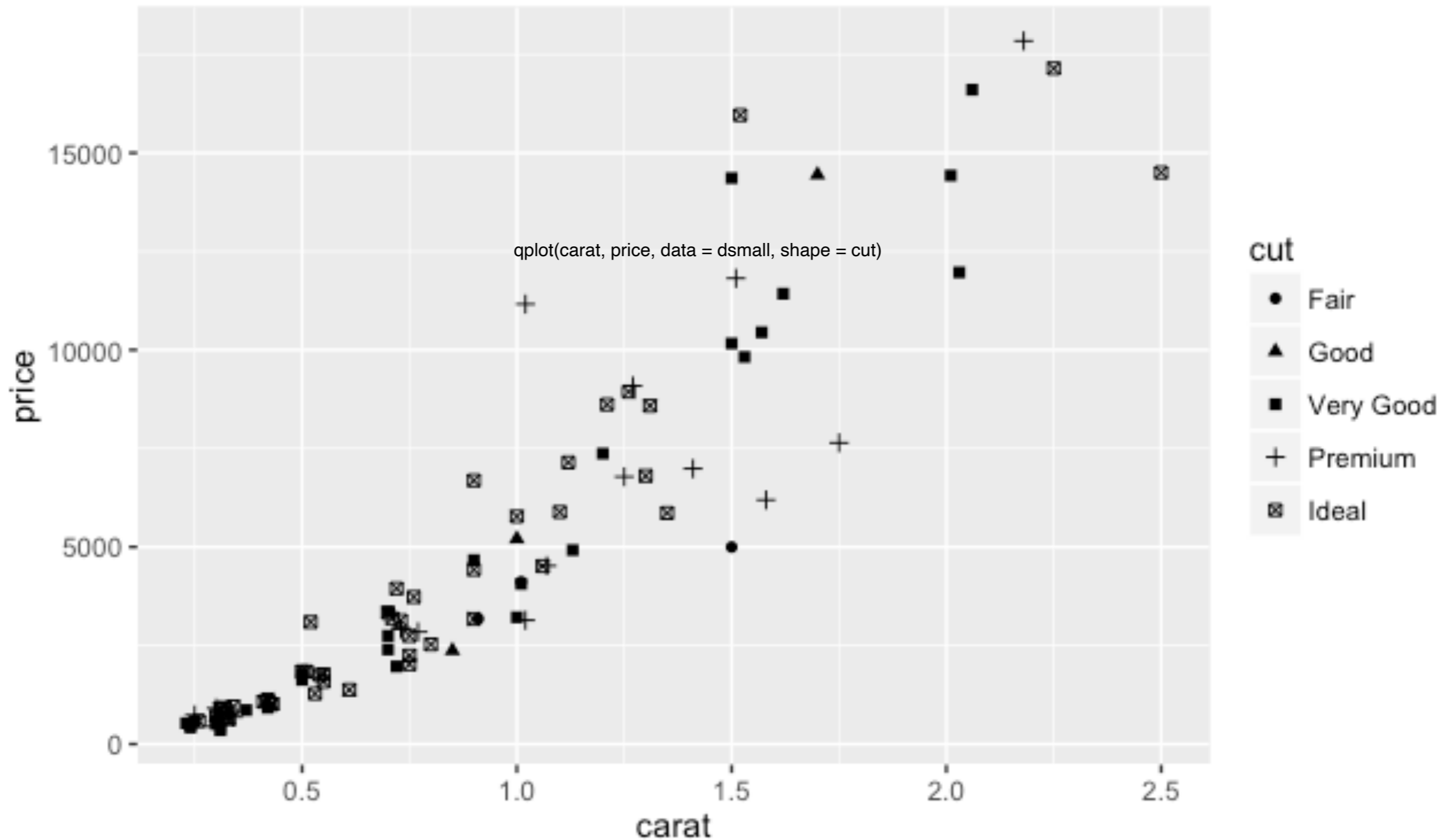


```
set.seed(1410)  dsmall <- diamonds[sample(nrow(diamonds), 100), ]
```

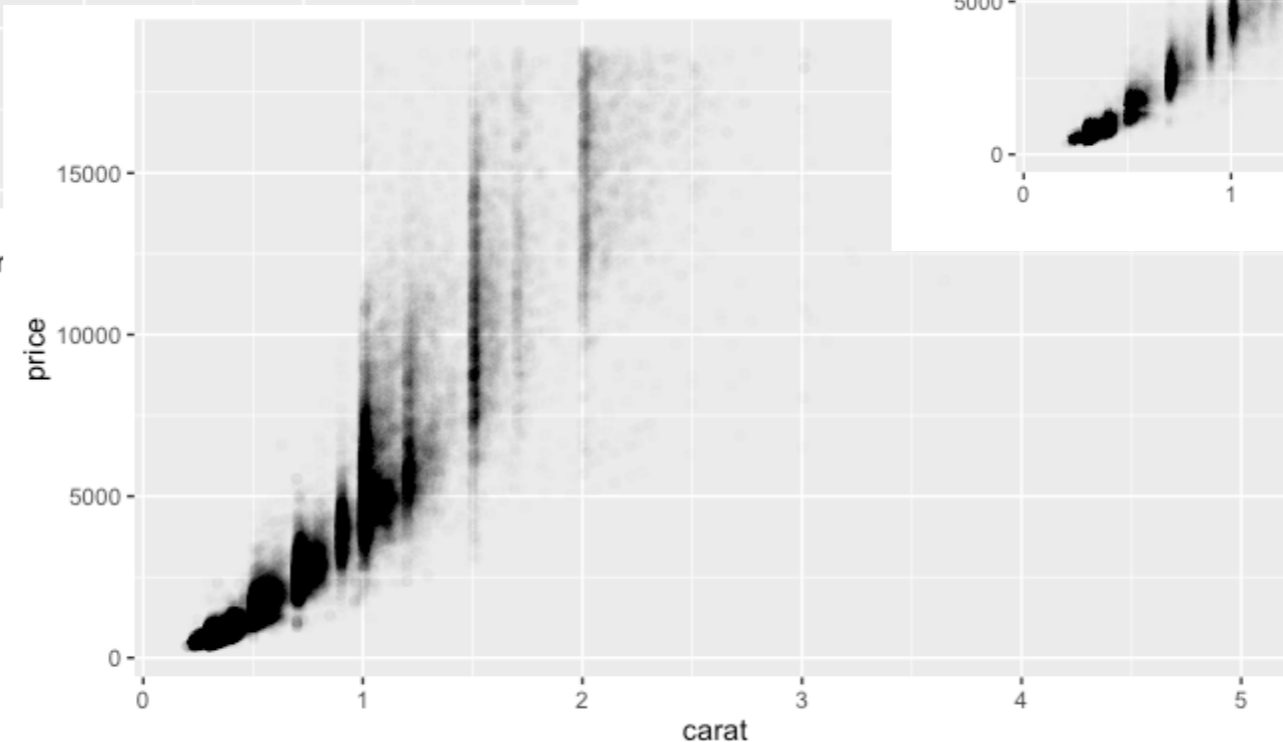
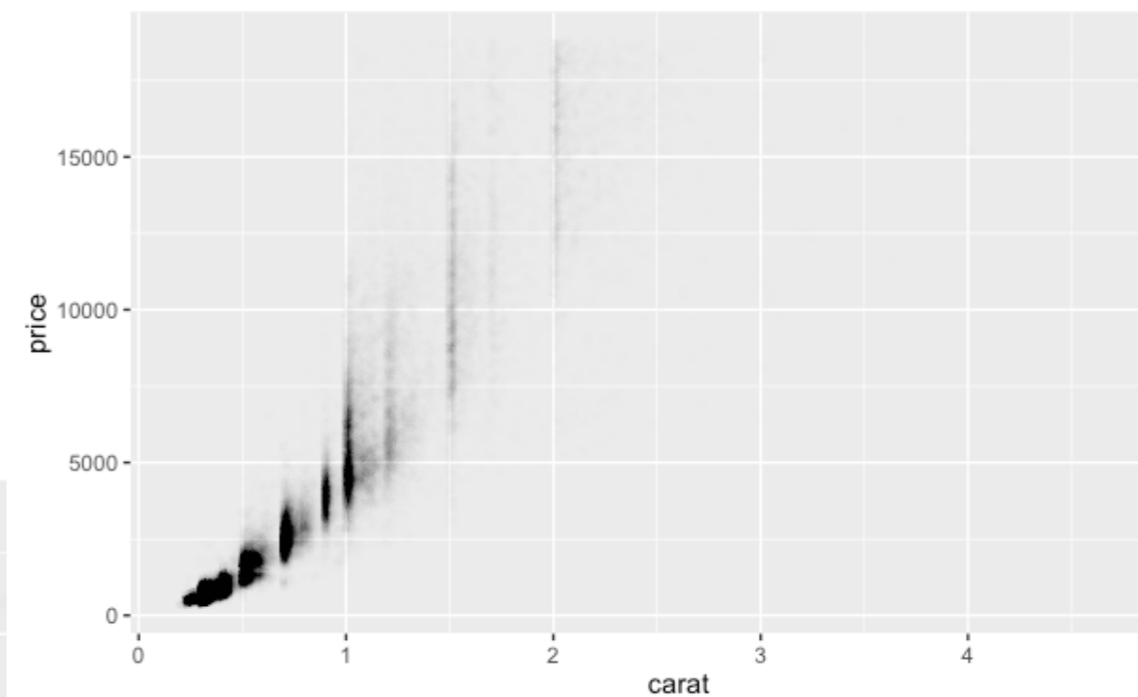
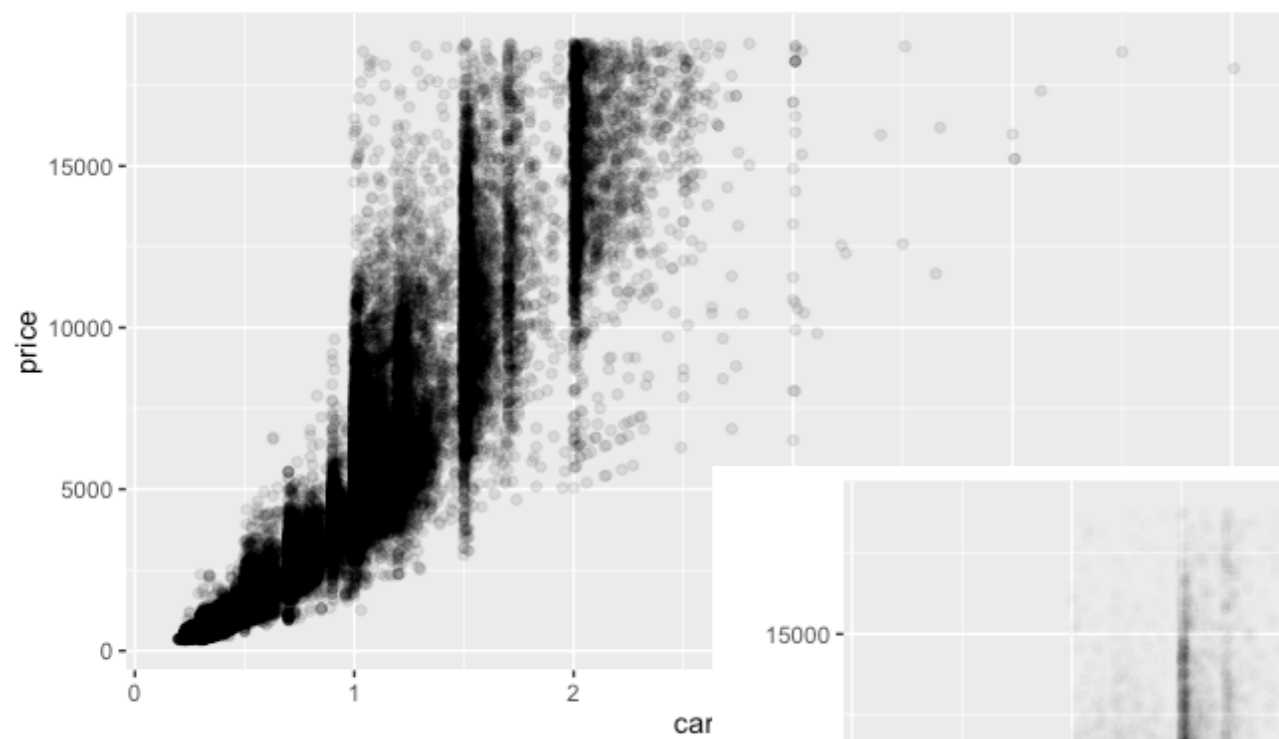
```
qplot(carat, price, data = dsmall, colour = color)
```



```
qplot(carat, price, data = dsmall, shape = cut)
```

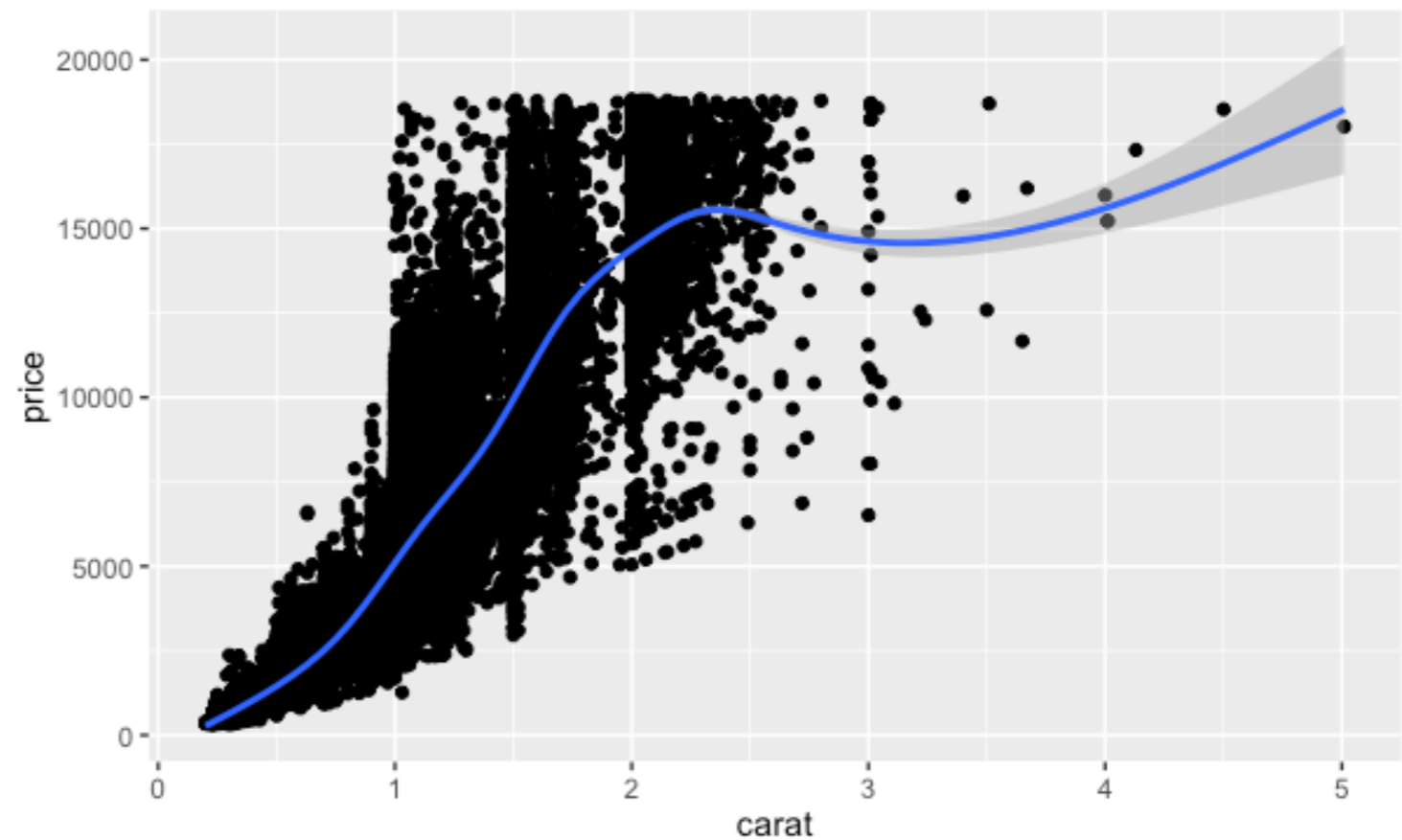
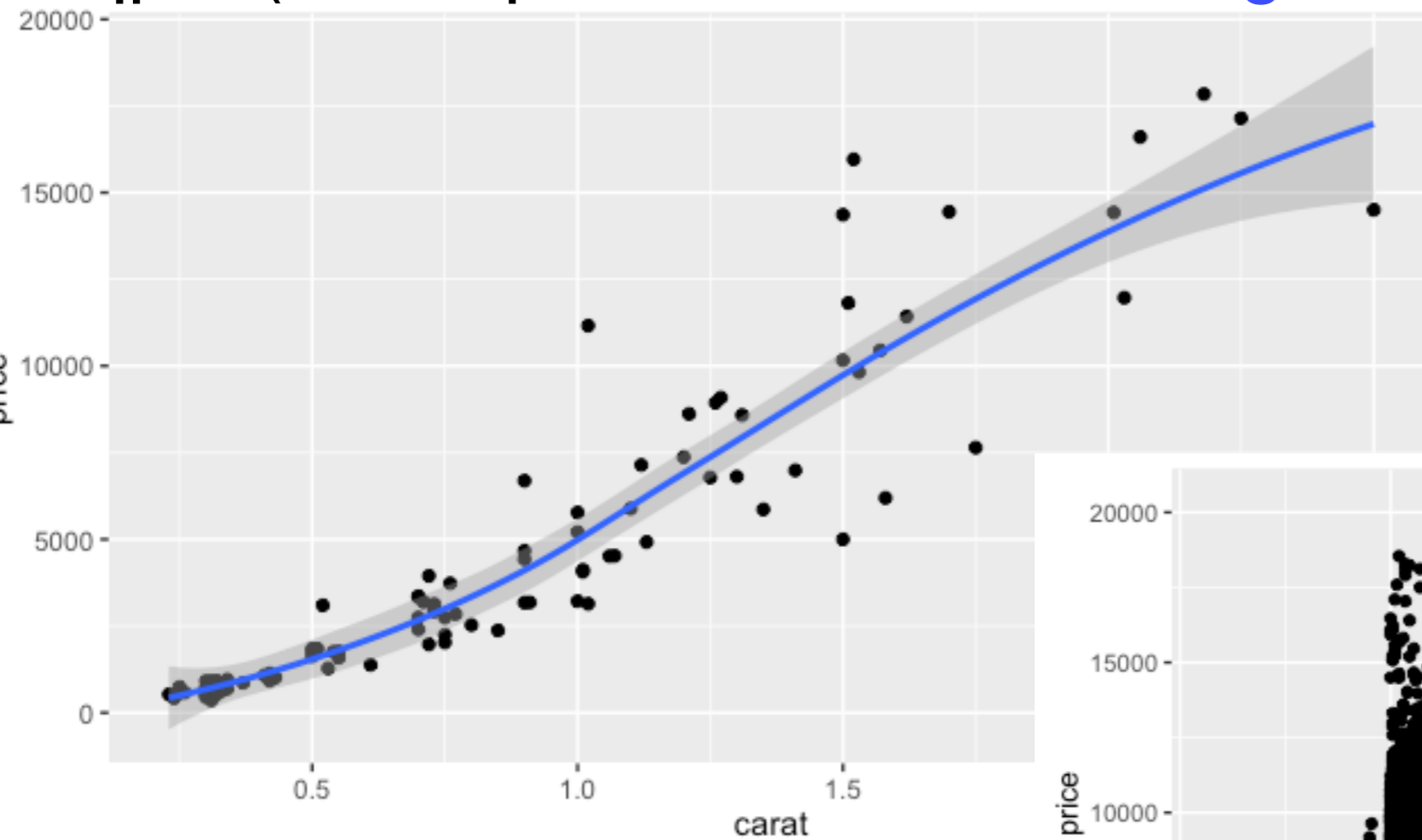


```
qplot(carat, price, data = diamonds, alpha = I(1/10))  
qplot(carat, price, data = diamonds, alpha = I(1/100))  
qplot(carat, price, data = diamonds, alpha = I(1/200))
```



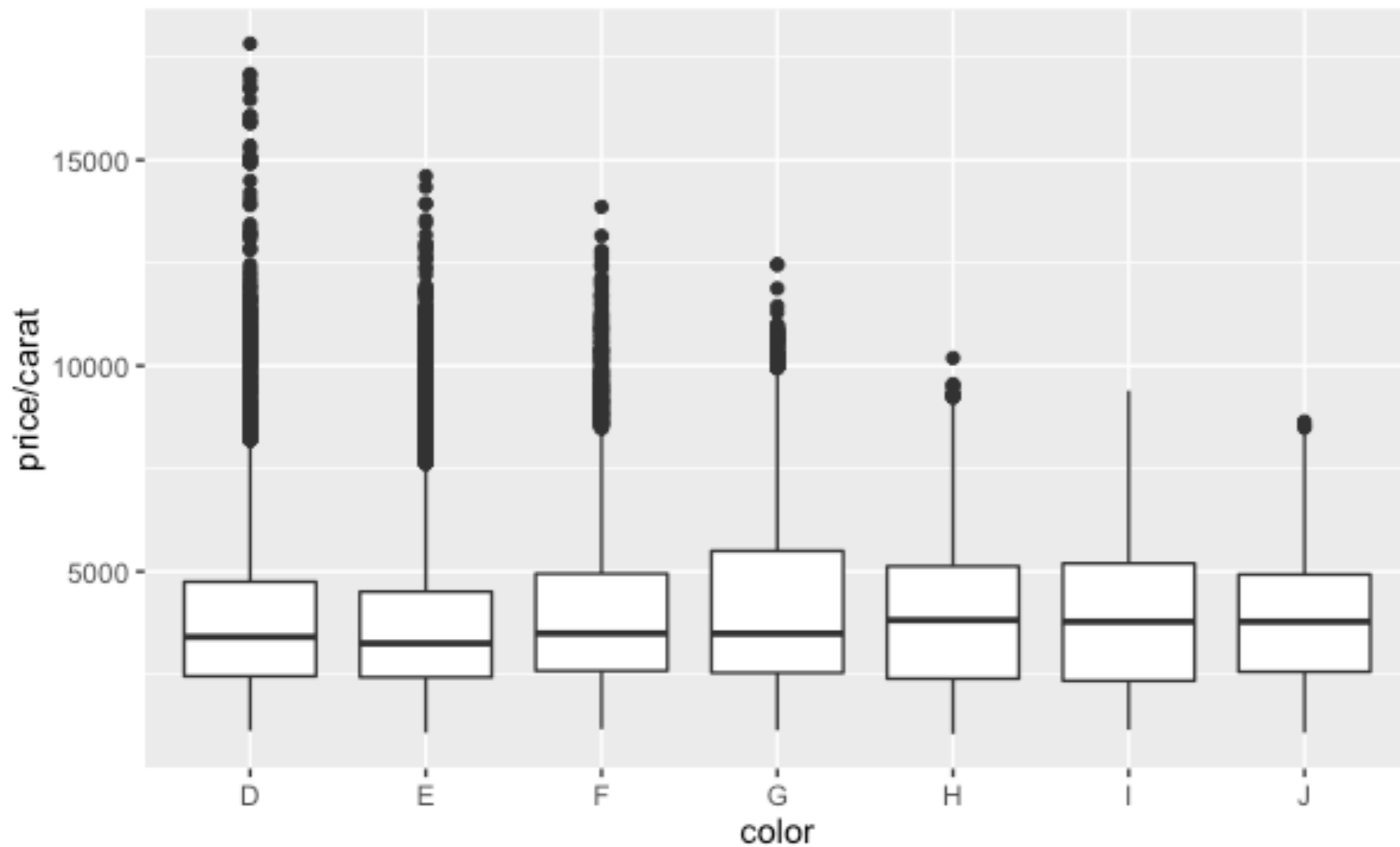
- `point`: 散点图 `geom = "point"`
- `smooth`: 平滑曲线和标准误
- `boxplot`: 箱线图
- `path`、`line`: 连线（曲线图、路径图）
- `histogram`: 直方图
- `freqpoly`: 频率多边形
- `density`: 密度曲线
- `bar`: 柱状图（条形图）

`qplot(carat, price, data = dsmall, geom = c("point", "smooth"))`

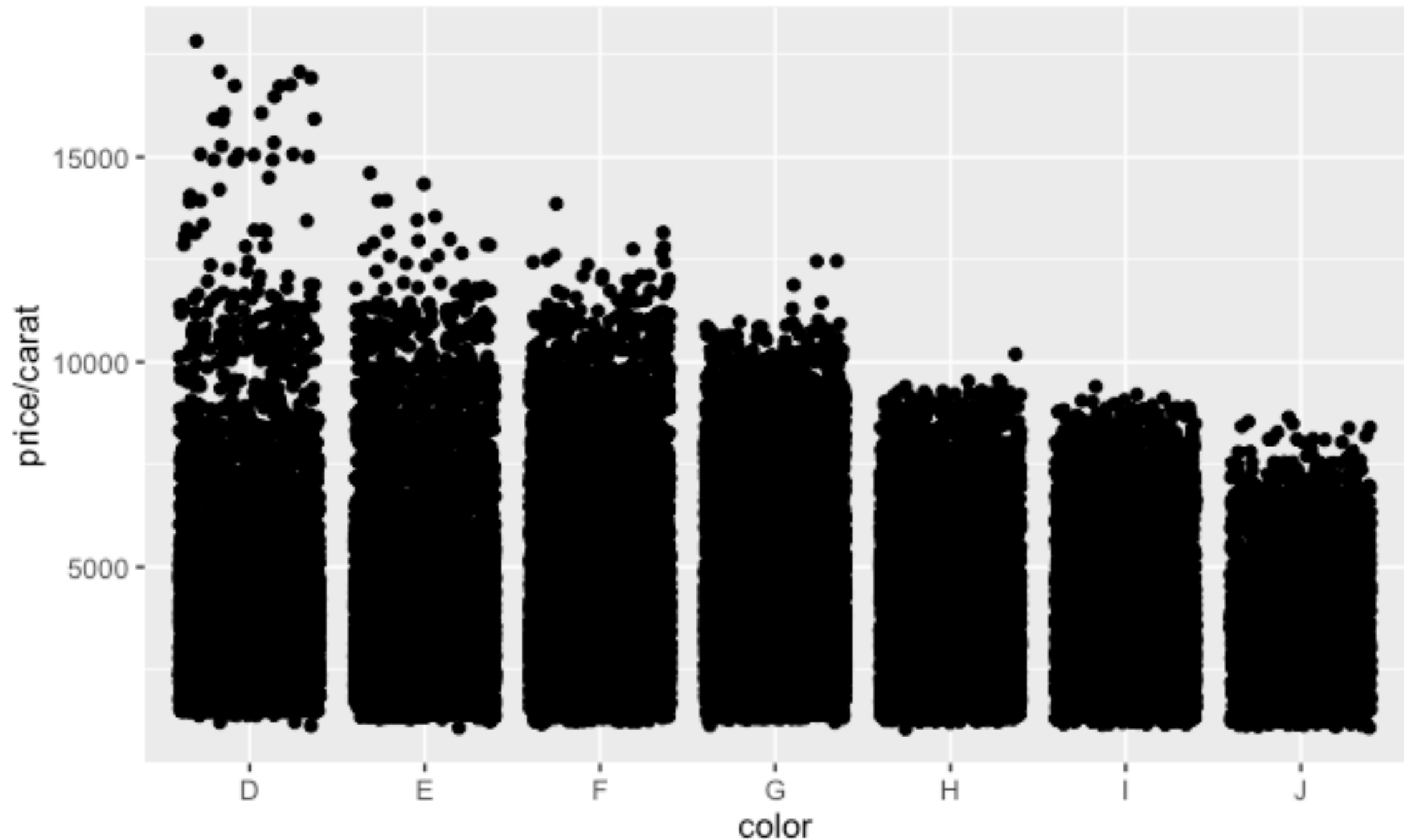


`qplot(carat, price, data = diamonds, geom = c("point", "smooth"))`

```
qplot(color, price / carat, data = diamonds, geom = "boxplot")
```



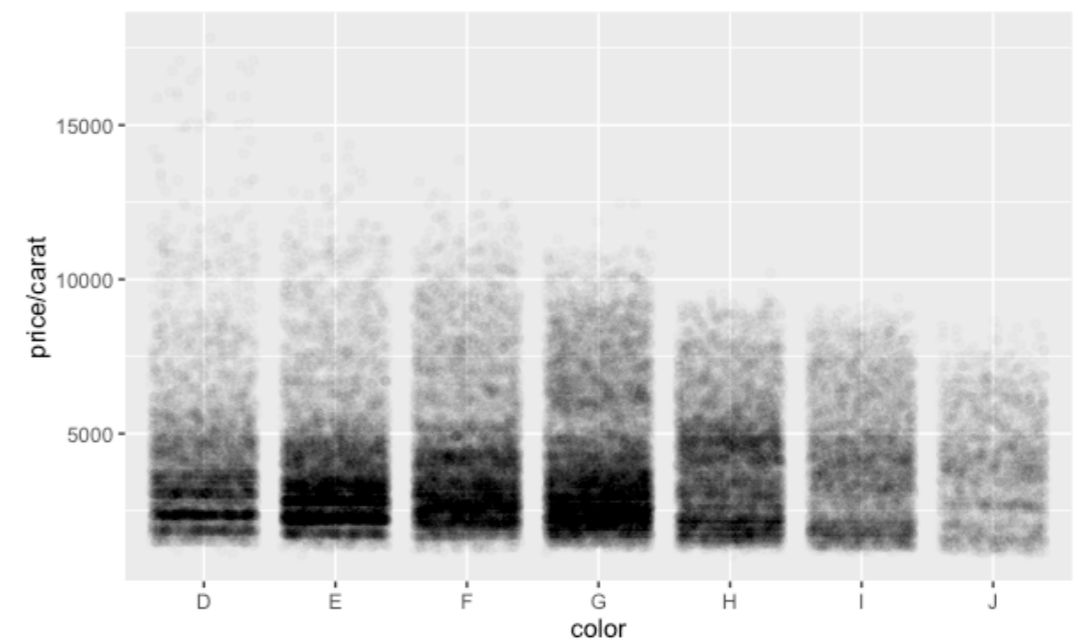
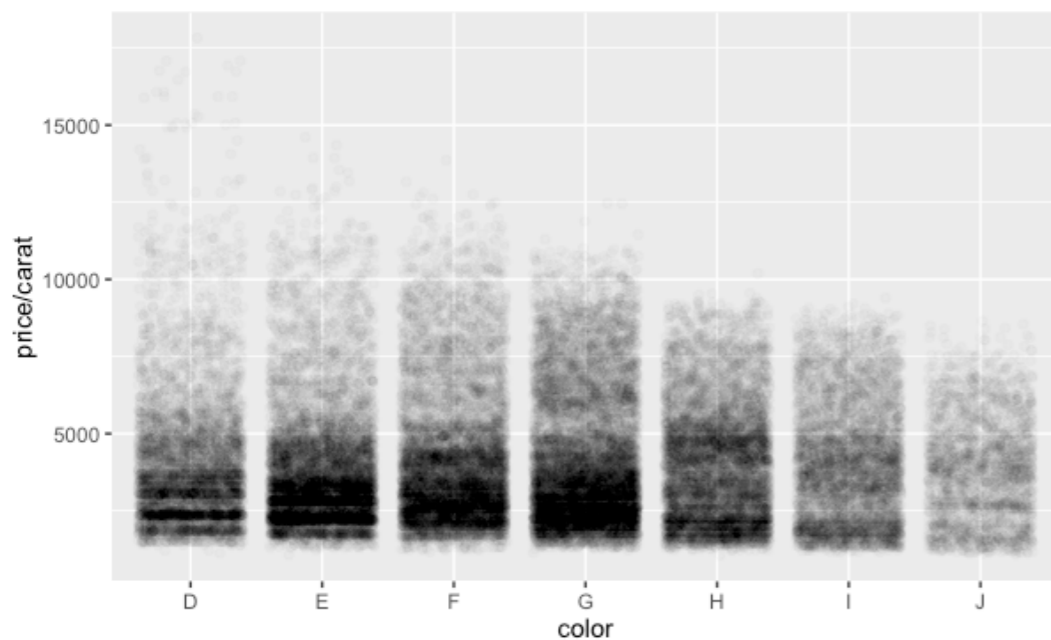
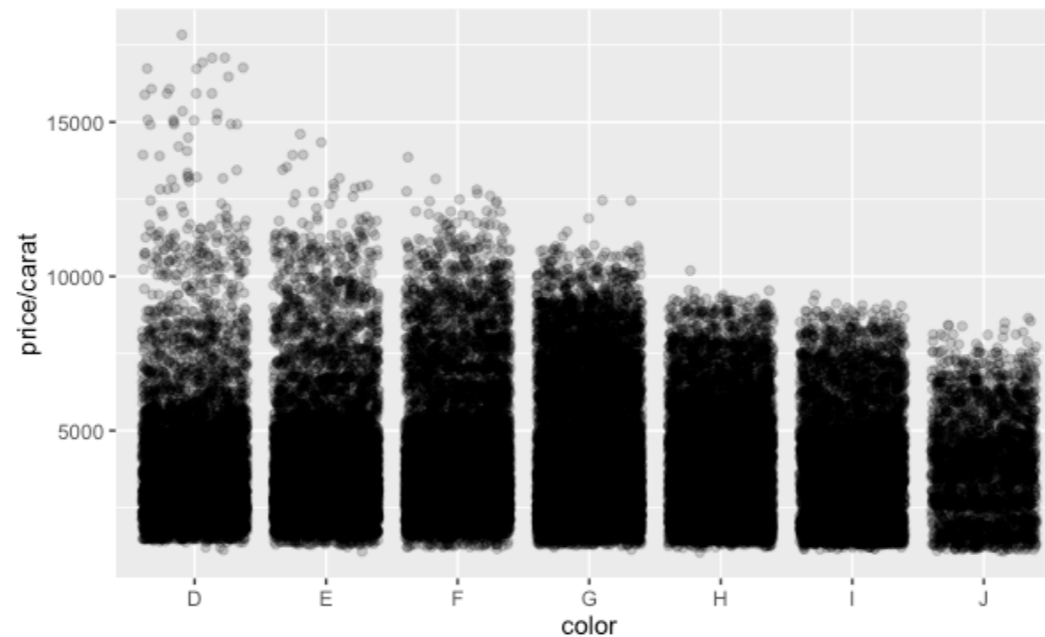
`qplot(color, price / carat, data = diamonds, geom = "jitter")`



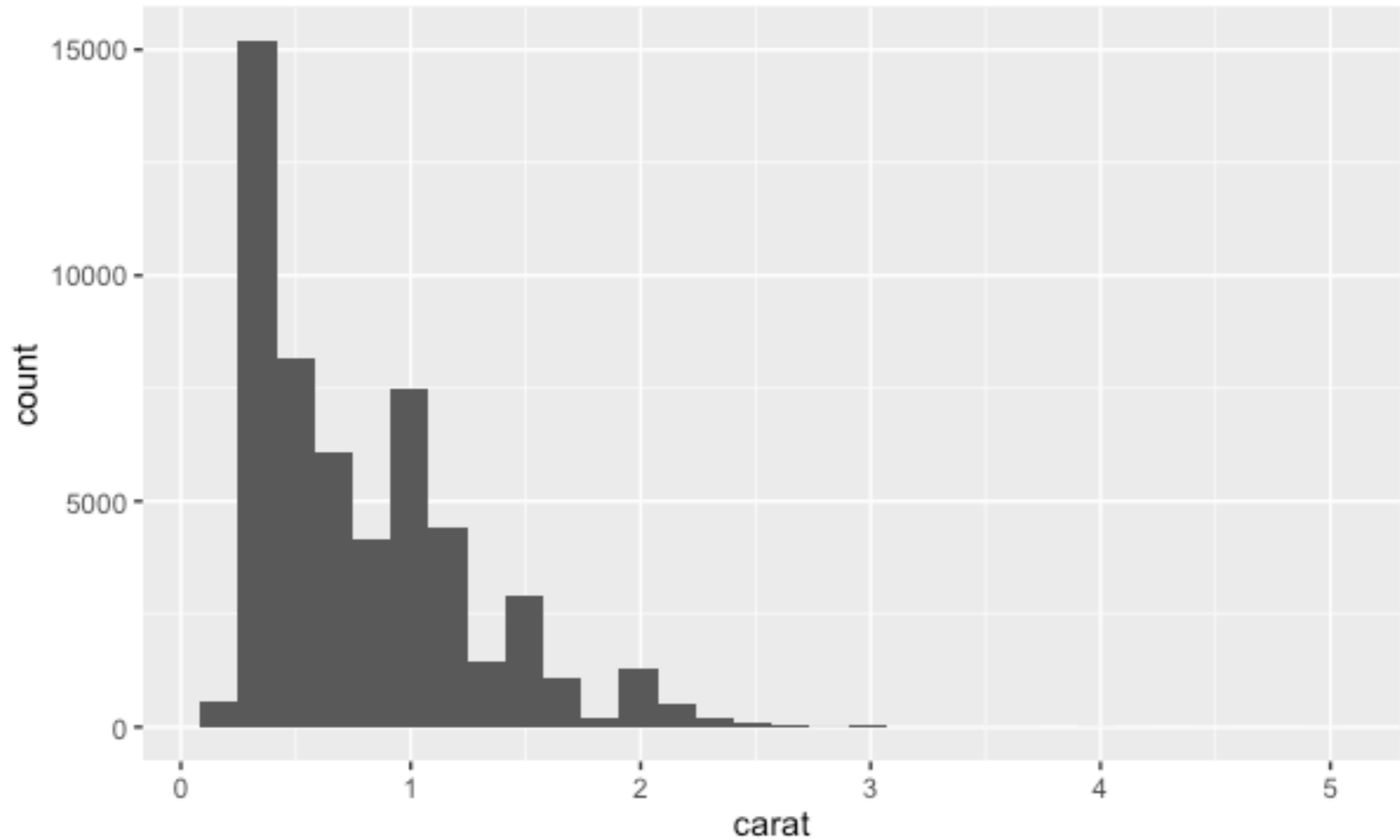
`qplot(color, price / carat, data = diamonds, geom = "jitter", alpha = I(1 / 5))`

`qplot(color, price / carat, data = diamonds, geom = "jitter", alpha = I(1 / 50))`

`qplot(color, price / carat, data = diamonds, geom = "jitter", alpha = I(1 / 200))`



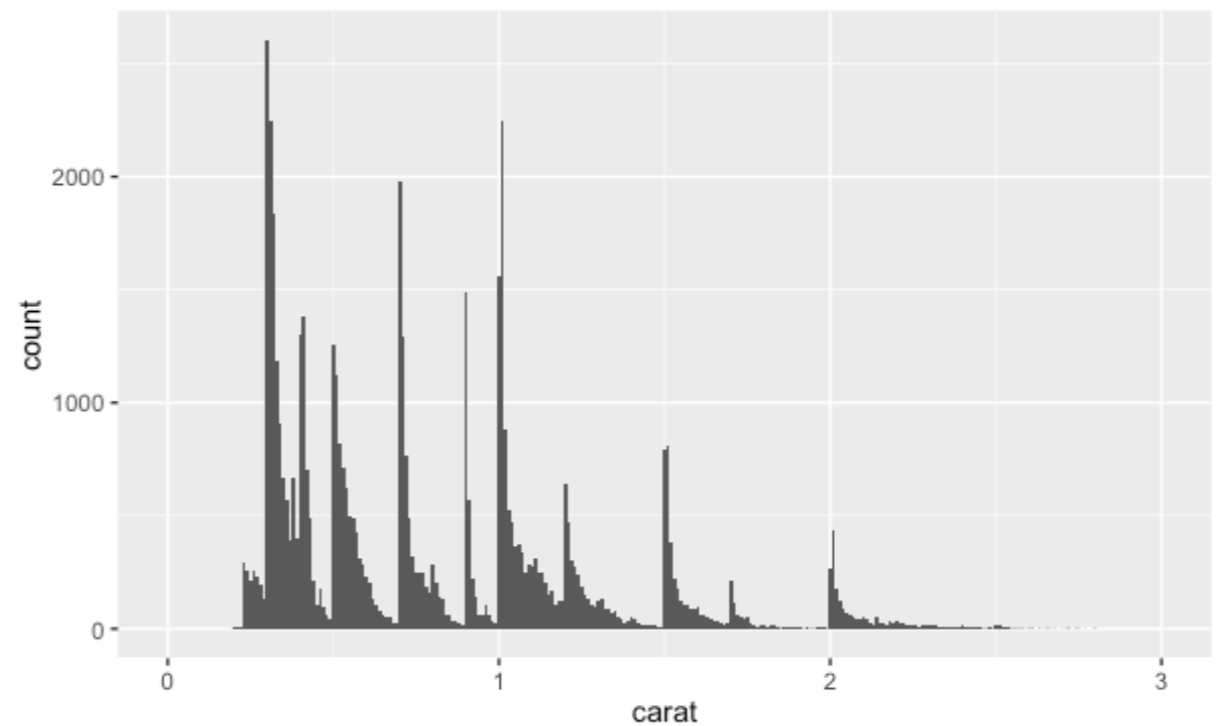
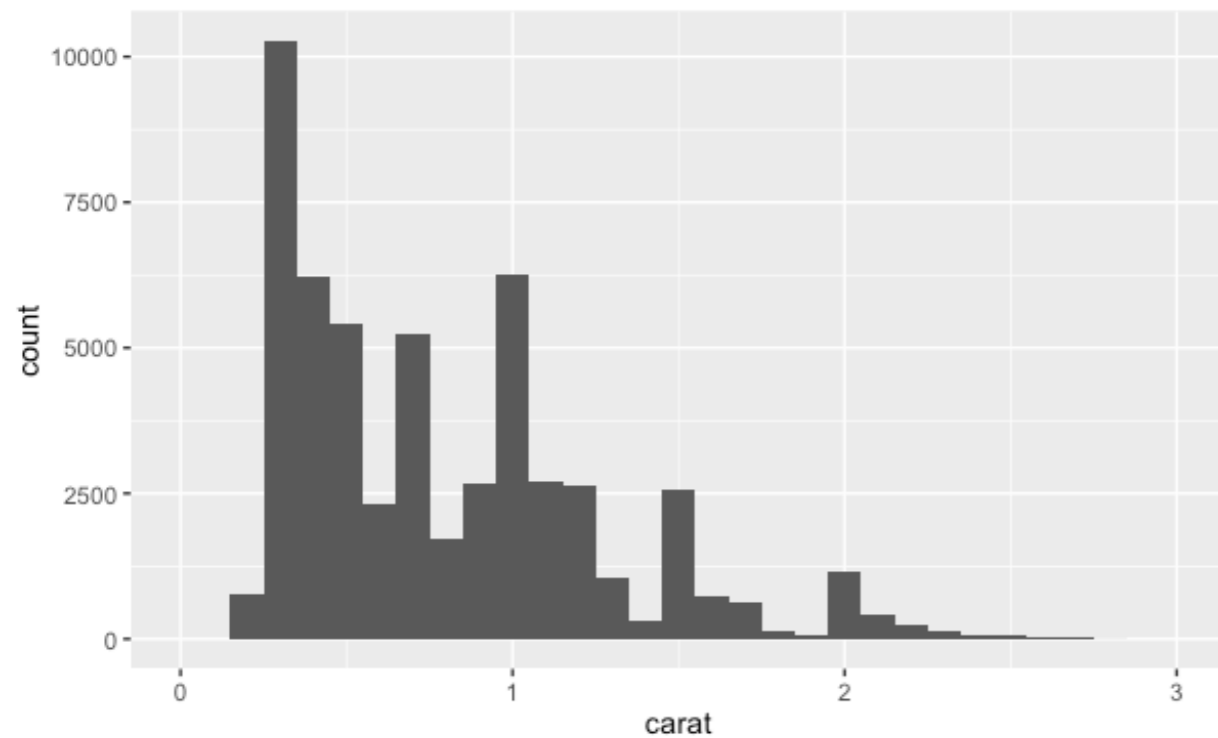
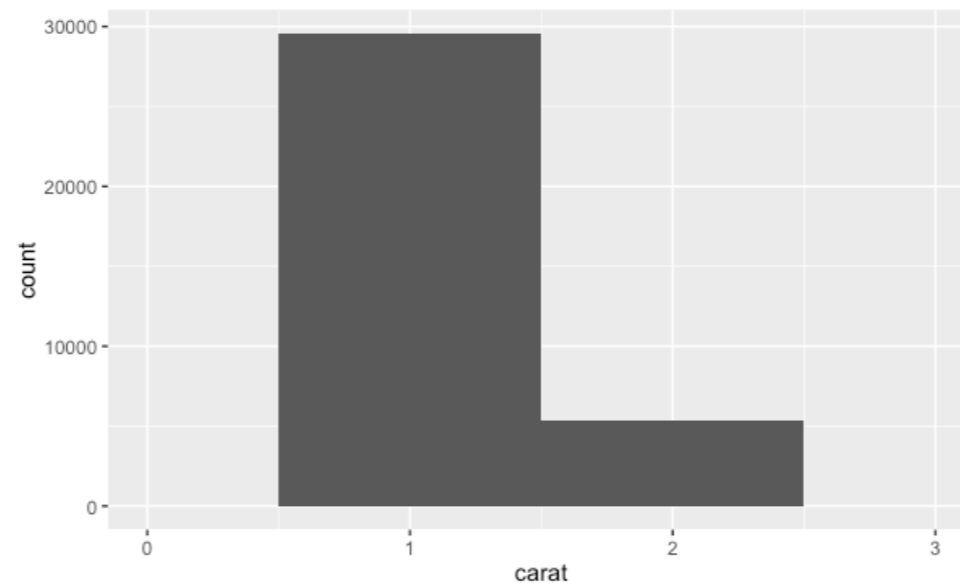
```
qplot(carat, data = diamonds, geom = "histogram")
```



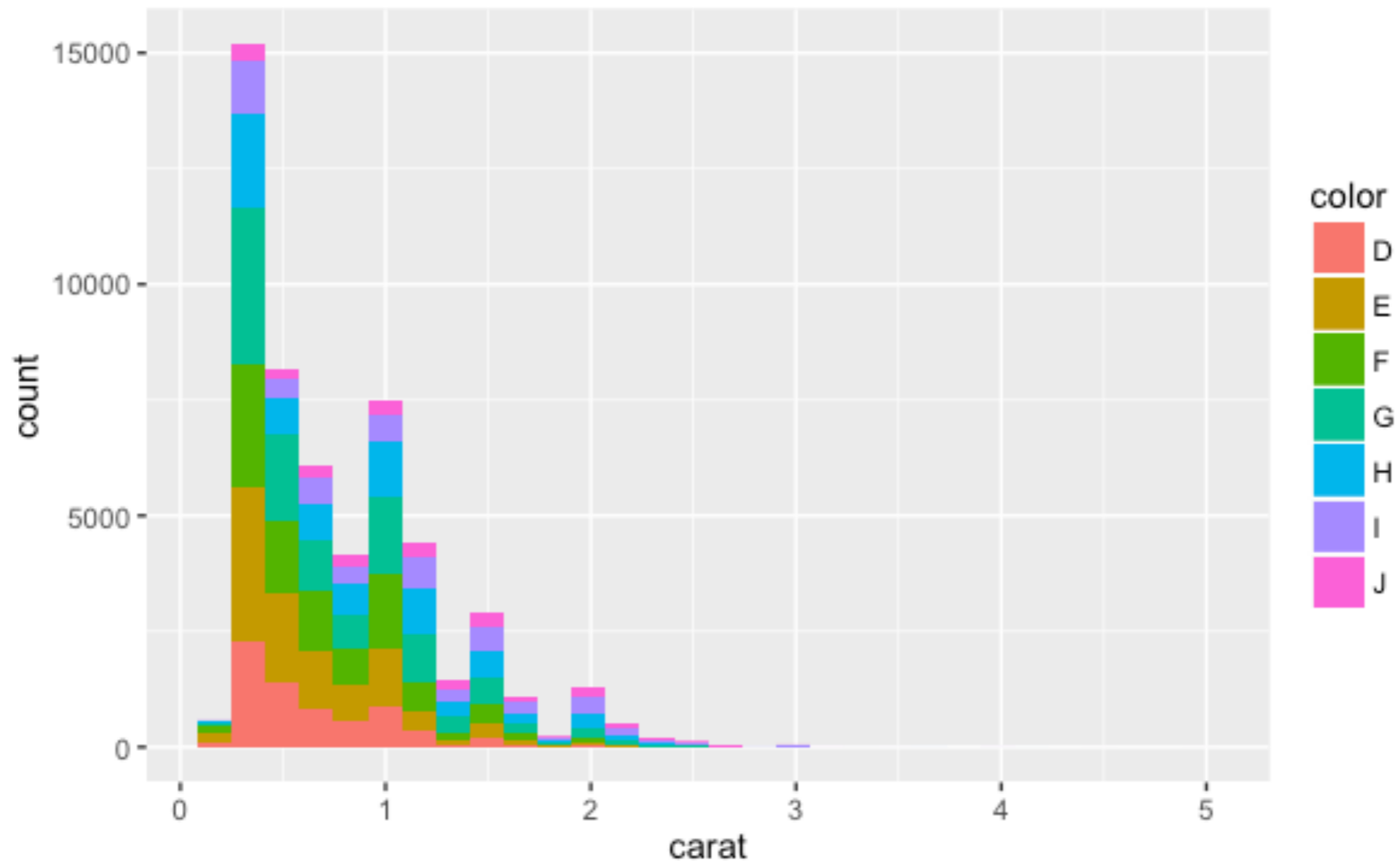
```
qplot(carat, data = diamonds, geom = "histogram", binwidth = 1, xlim = c(0,3))
```

```
qplot(carat, data = diamonds, geom = "histogram", binwidth = 0.1, xlim = c(0,3))
```

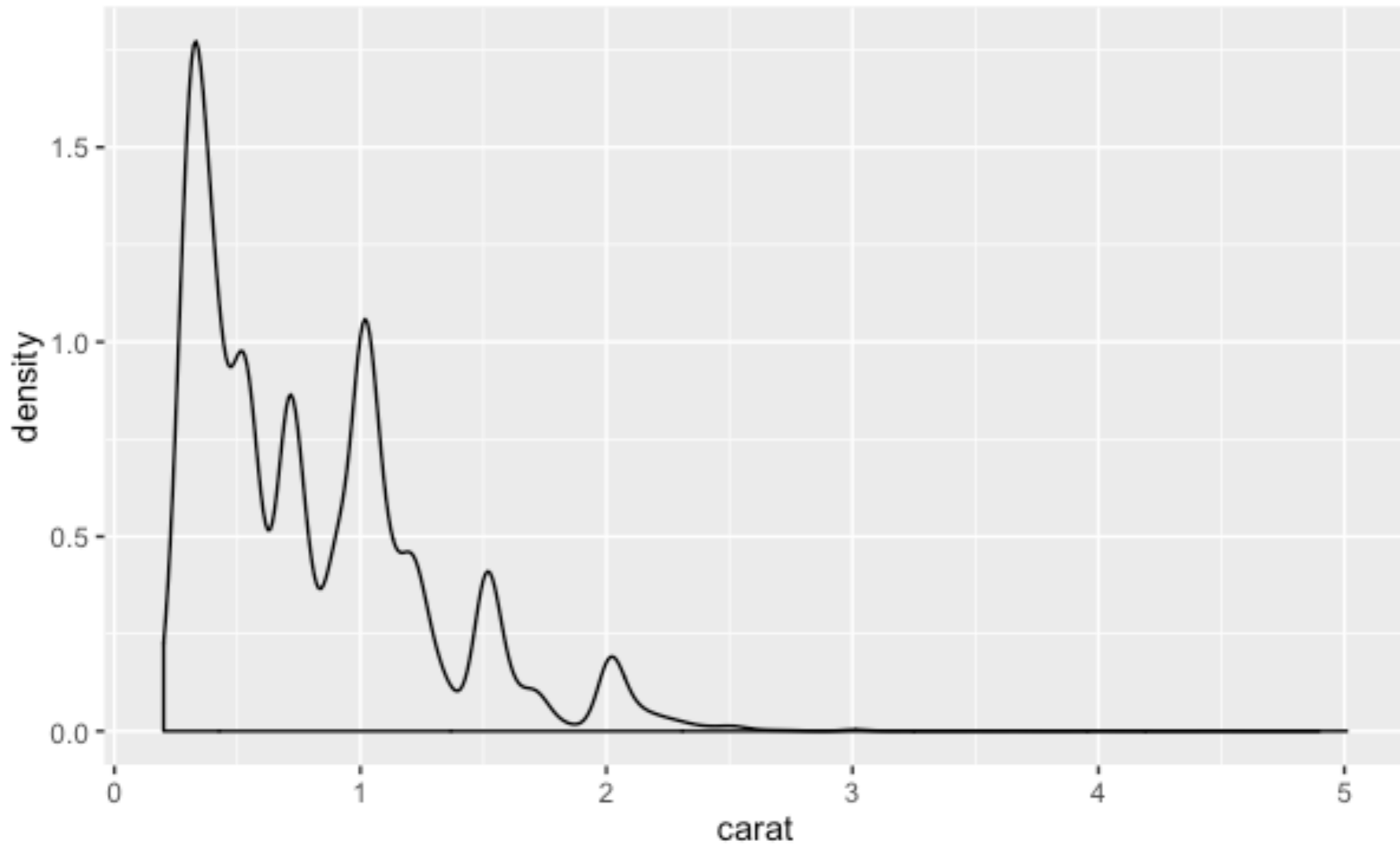
```
qplot(carat, data = diamonds, geom = "histogram", binwidth = 0.01, xlim = c(0,3))
```



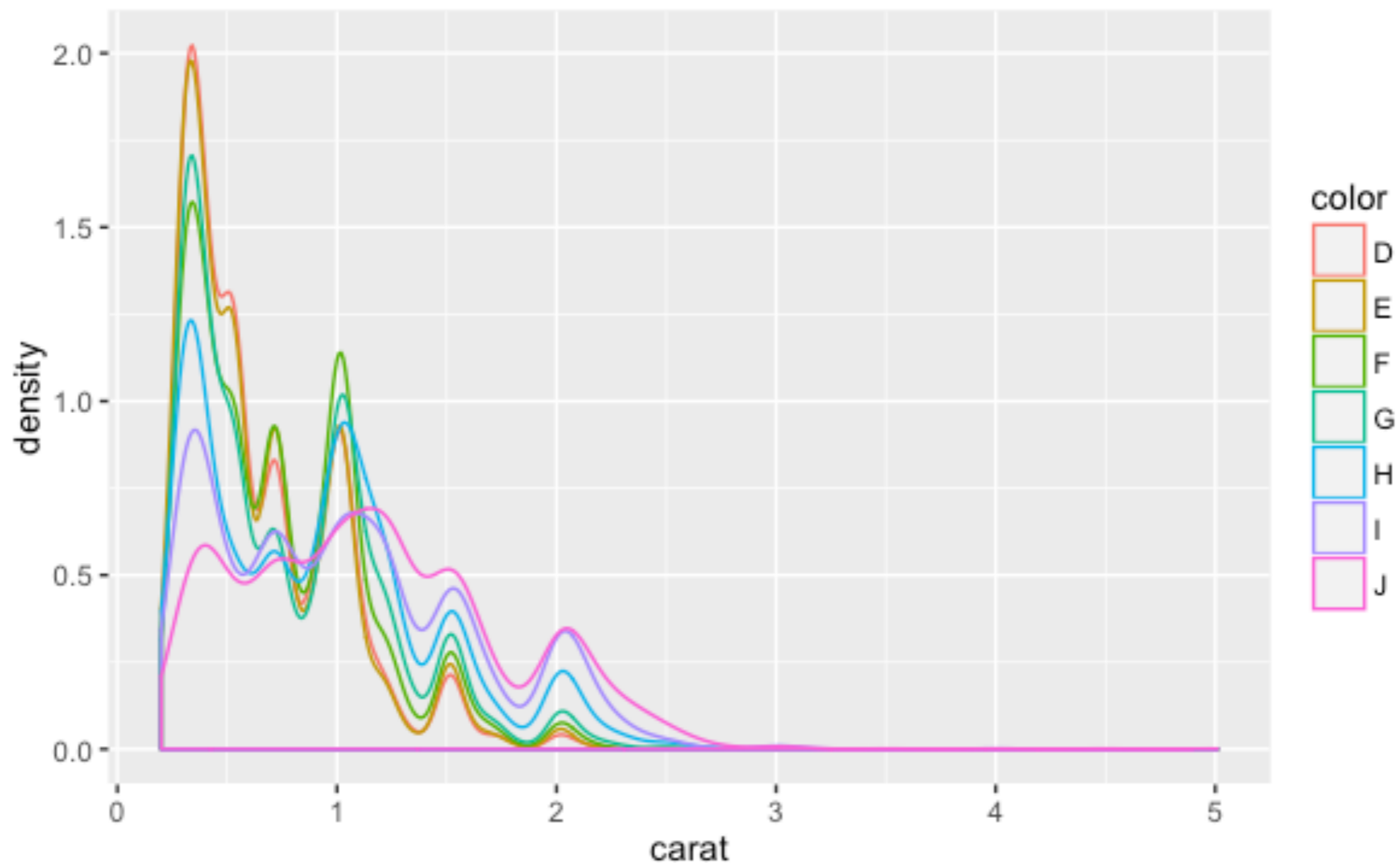
```
qplot(carat, data = diamonds, geom = "histogram", fill = color)
```



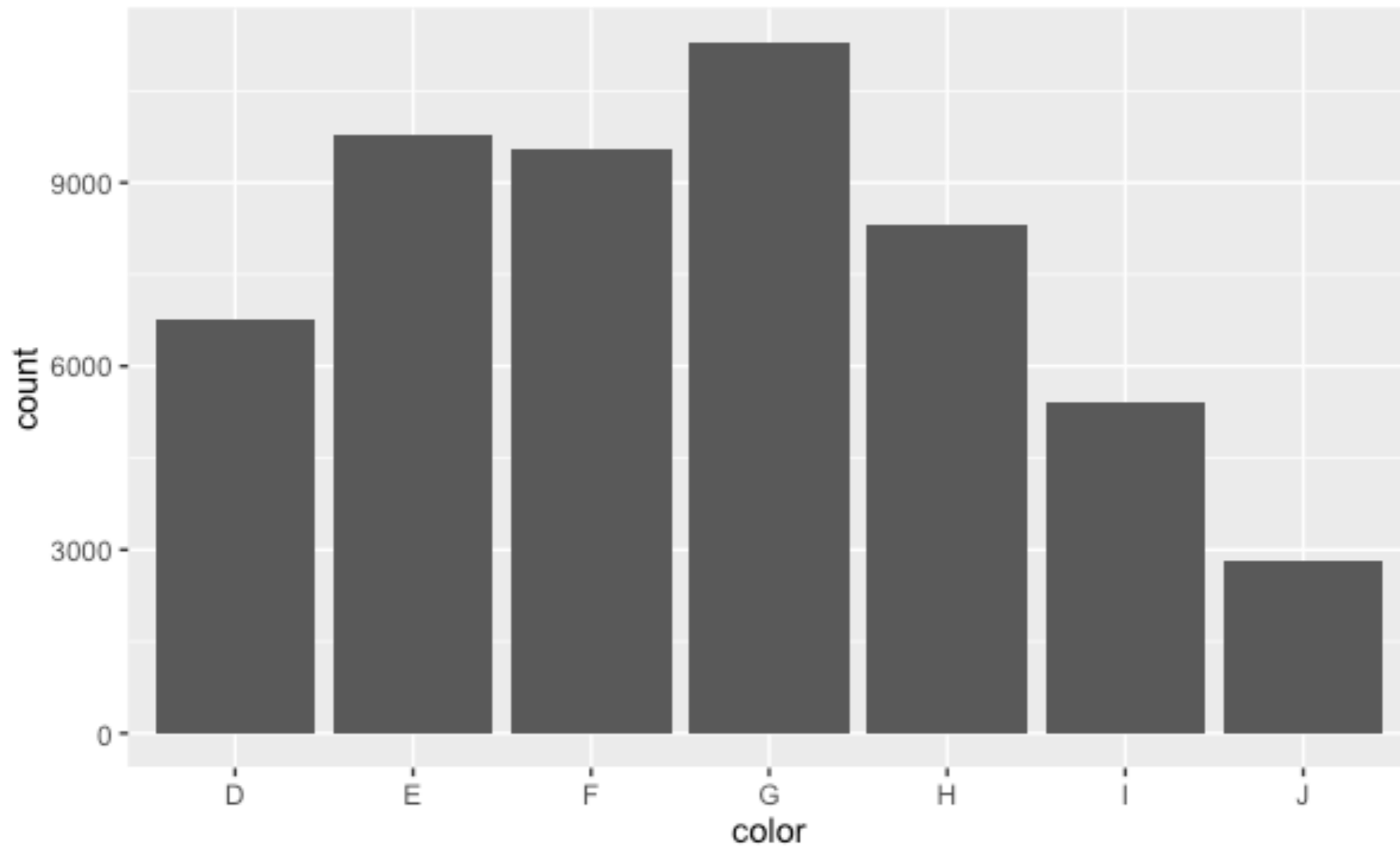

```
qplot(carat, data = diamonds, geom = "density")
```



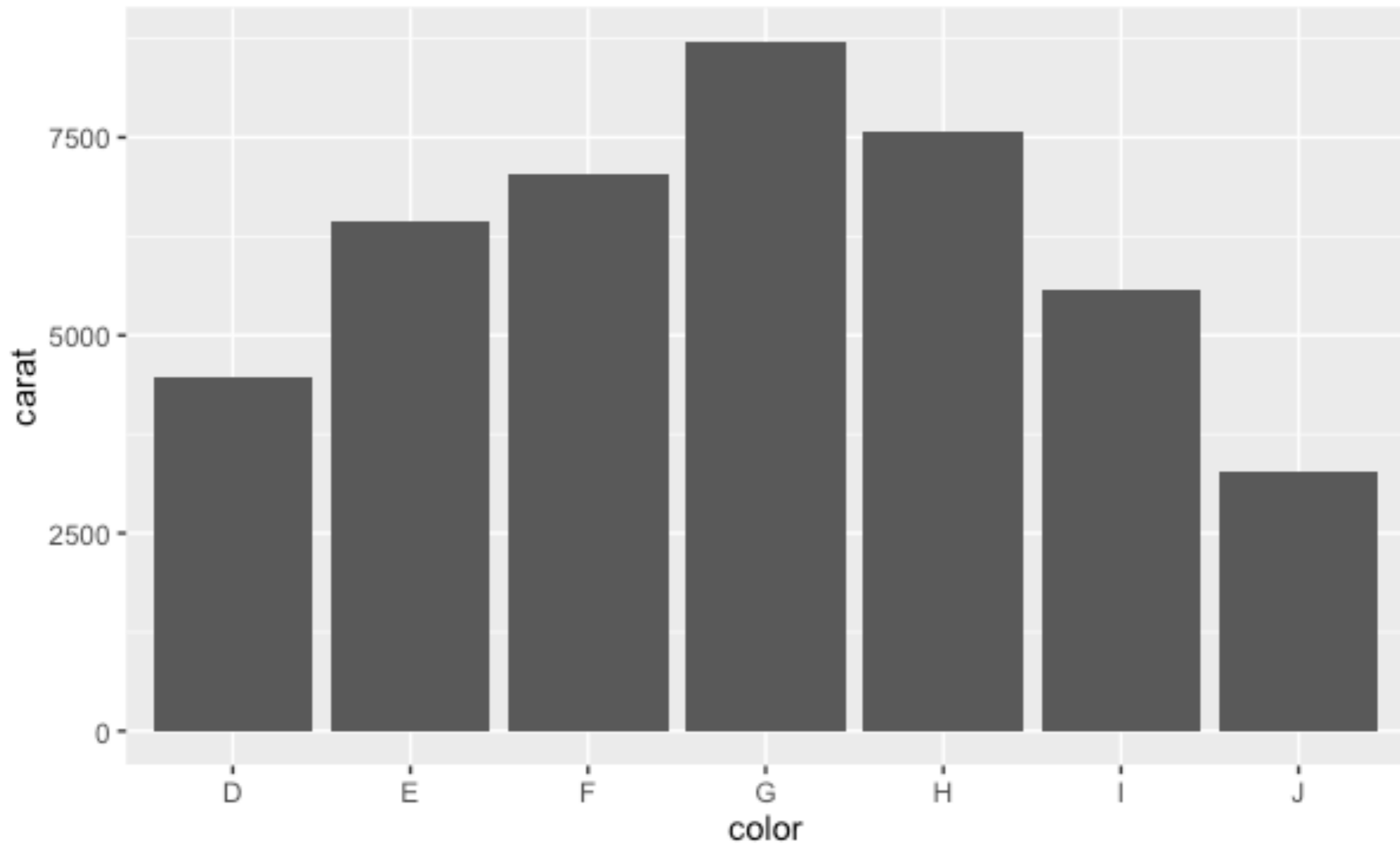
```
qplot(carat, data = diamonds, geom = "density", colour = color)
```

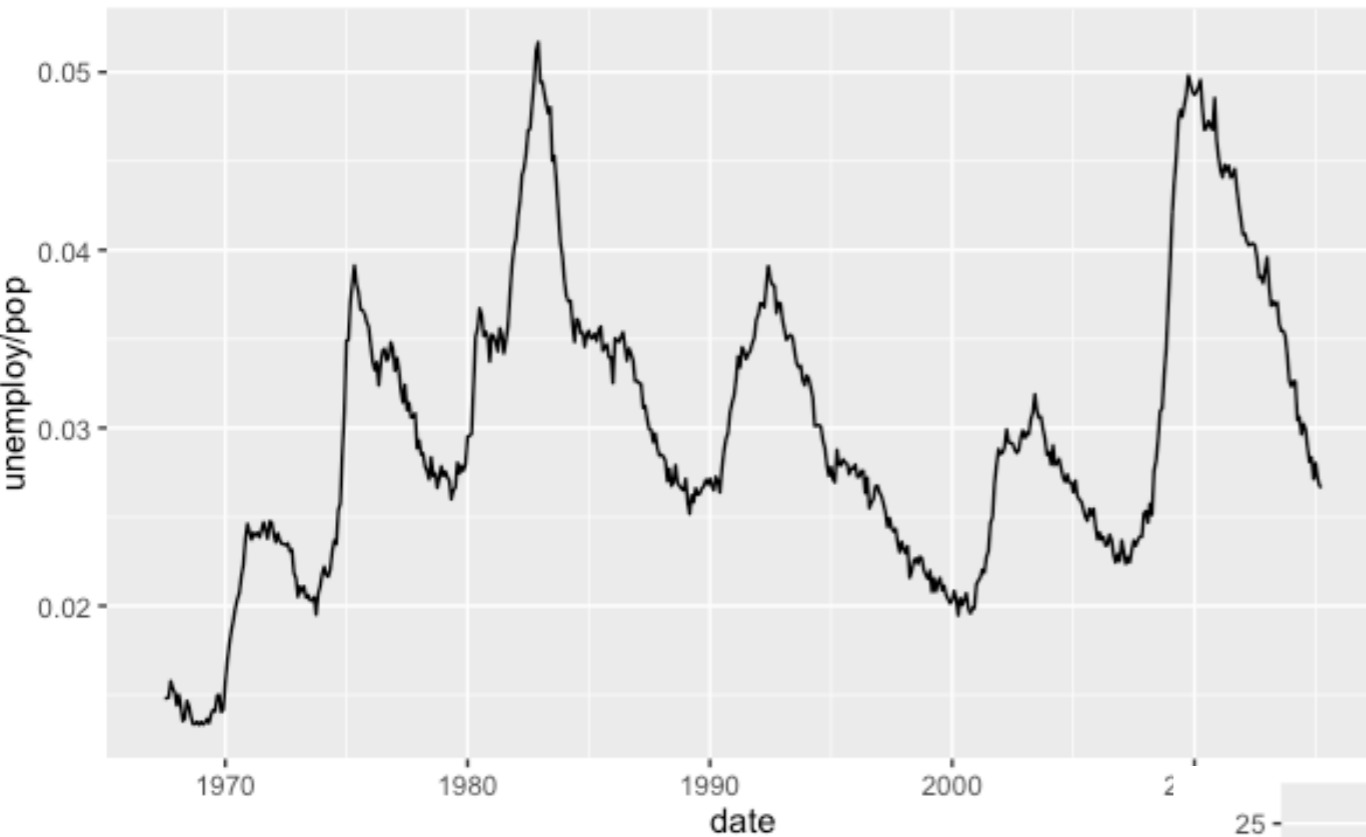


```
qplot(color, data = diamonds, geom = "bar")
```



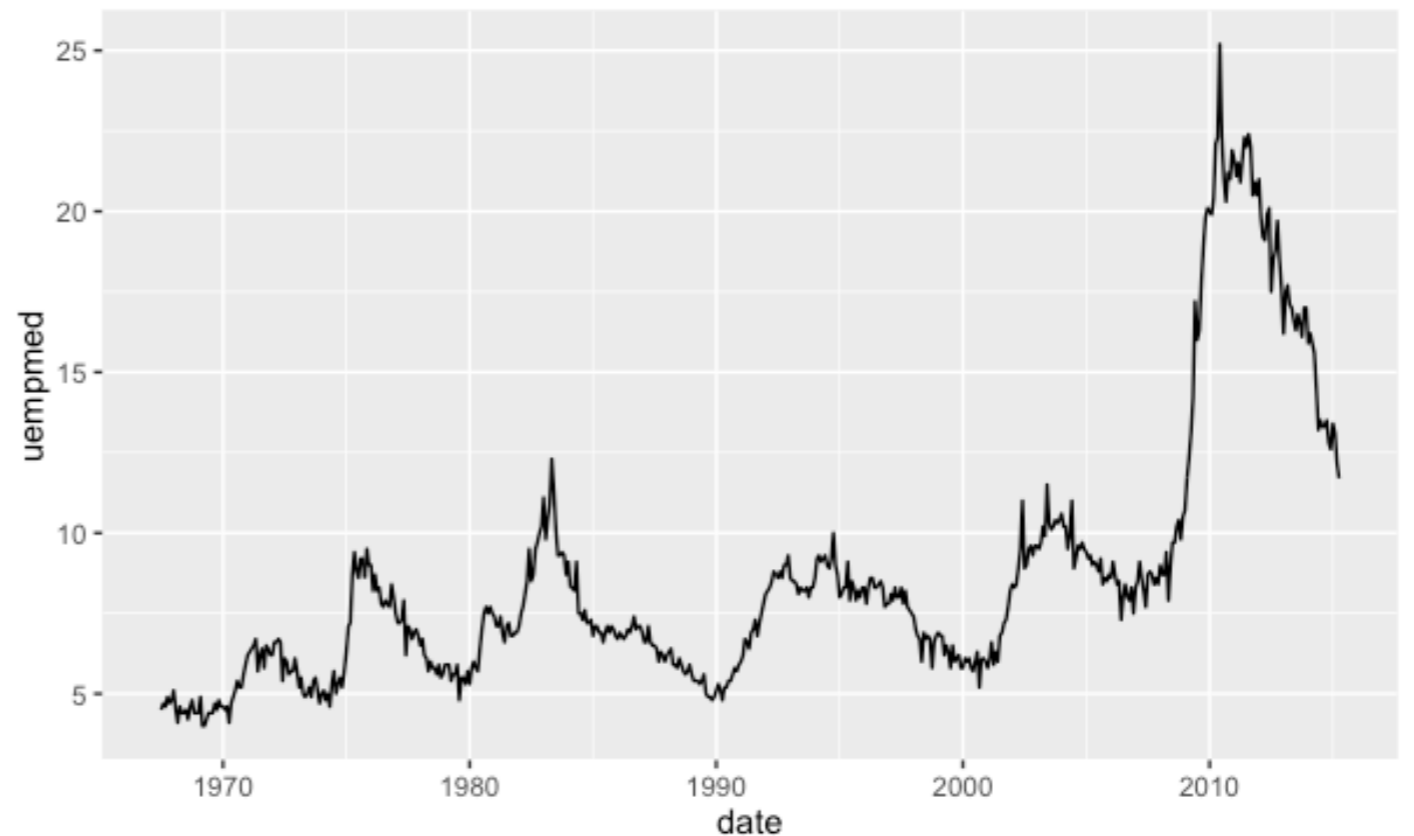
```
qplot(color, data = diamonds, geom = "bar", weight = carat) +  
scale_y_continuous("carat")
```

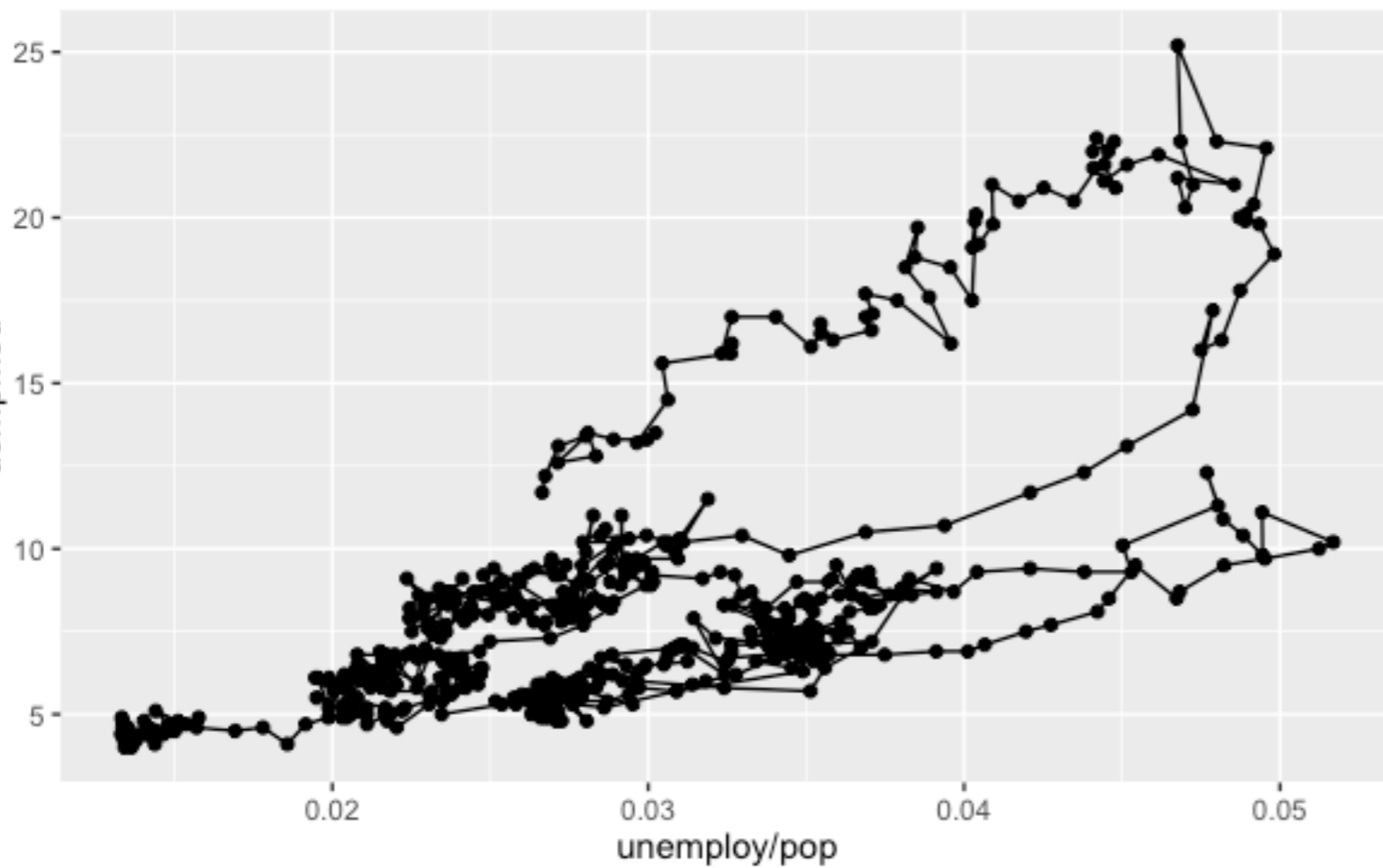




```
qplot(date, unemploy / pop,  
data = economics, geom =  
"line")
```

```
qplot(date, unempmed,  
data = economics,  
geom = "line")
```

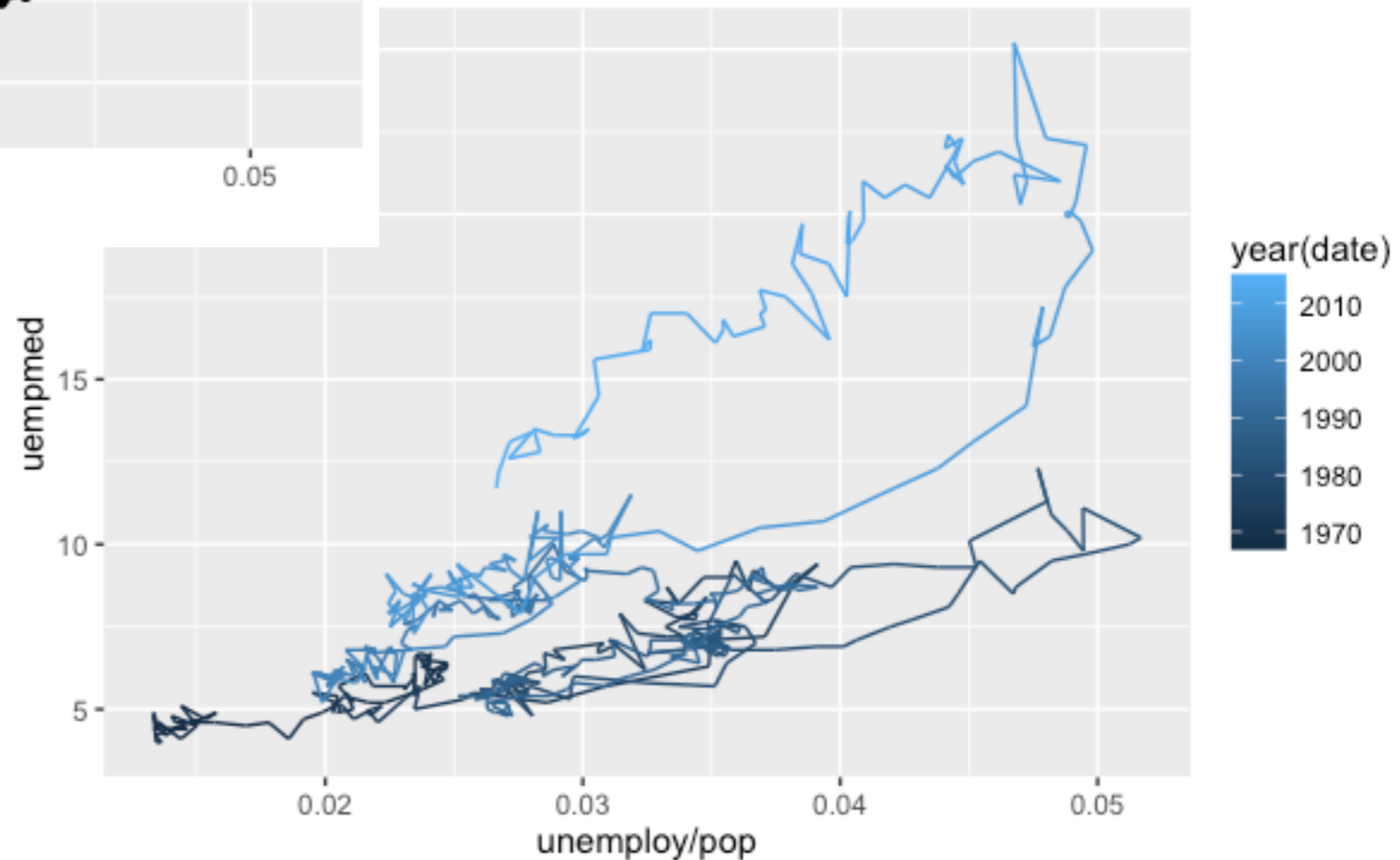




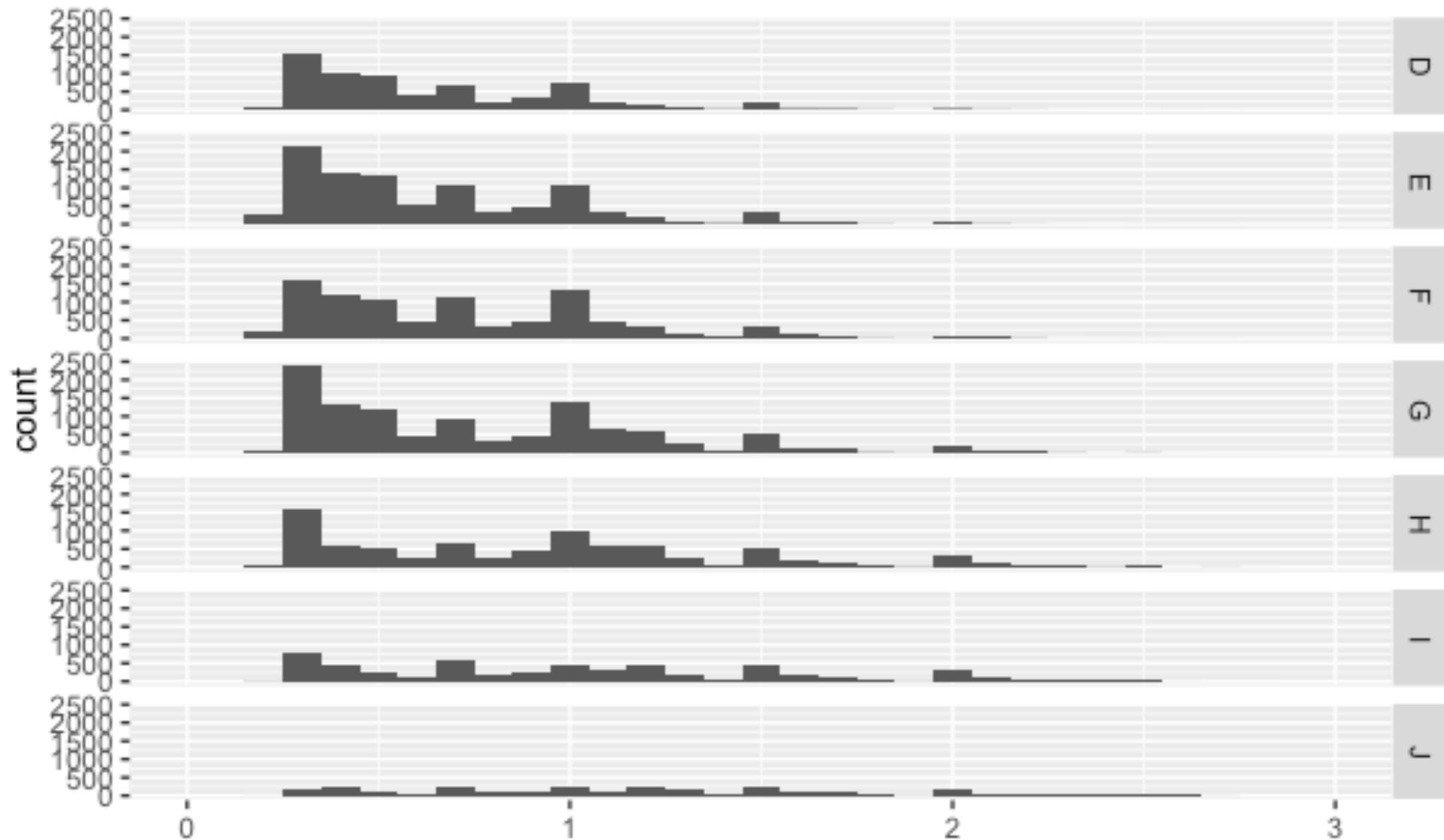
```
year <- function(x) as.POSIXlt(x)  
$year + 1900
```

```
qplot(unemploy / pop, uempmed,  
data = economics,  
geom = c("point", "path"))
```

```
qplot(unemploy / pop,  
uempmed, data = economics,  
geom = "path",  
colour = year(date))
```



```
qplot(carat, data = diamonds, facets = color ~ .,  
      geom = "histogram", binwidth = 0.1, xlim = c(0, 3))
```

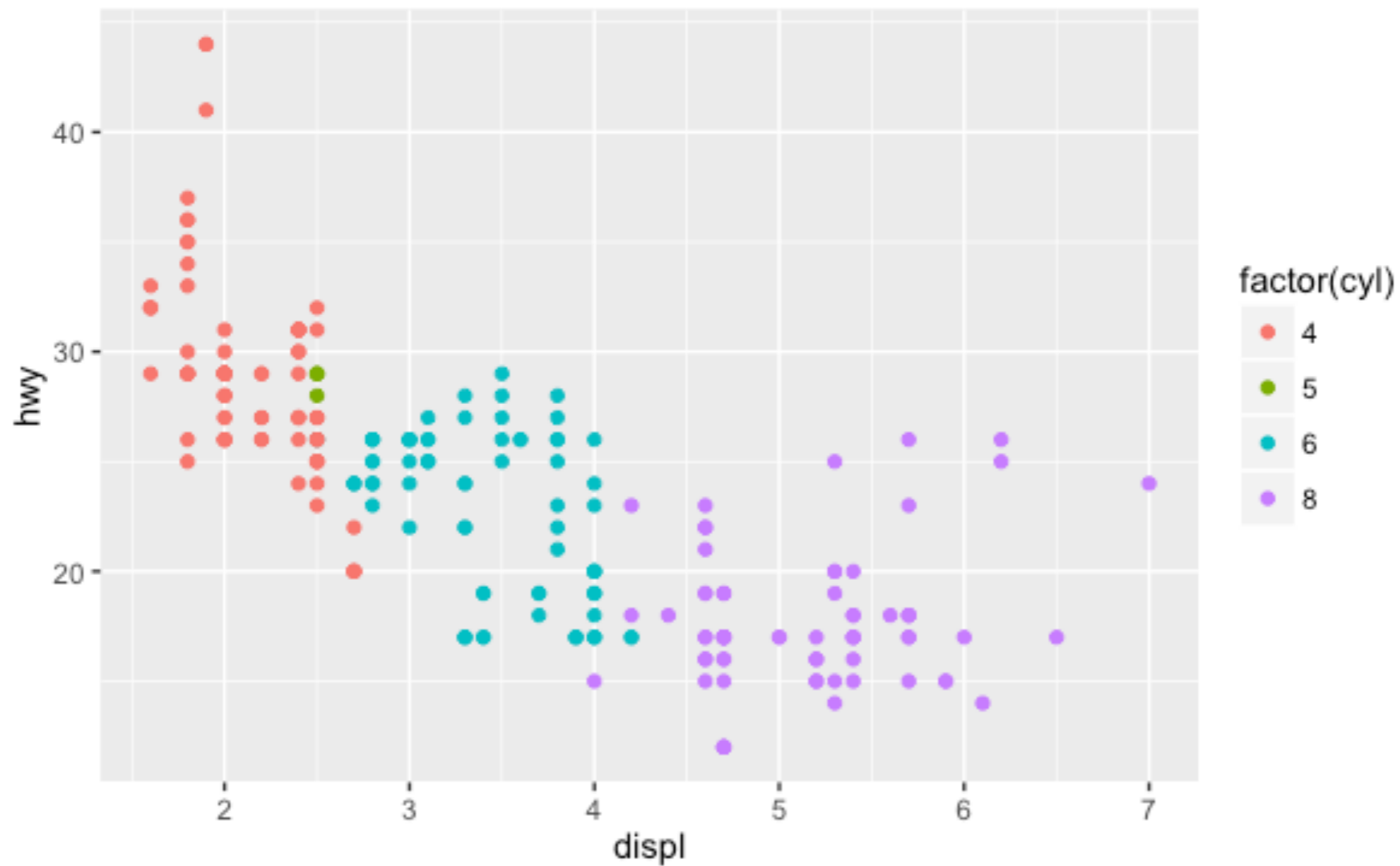


- xlim
- ylim
- log
- main
- xlab
- ylab

语法突破

manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
audi	a4	1.80	1999	4	auto(l5)	f	18	29	p	compact
audi	a4	1.80	1999	4	manual(m5)	f	21	29	p	compact
audi	a4	2.00	2008	4	manual(m6)	f	20	31	p	compact
audi	a4	2.00	2008	4	auto(av)	f	21	30	p	compact
audi	a4	2.80	1999	6	auto(l5)	f	16	26	p	compact
audi	a4	2.80	1999	6	manual(m5)	f	18	26	p	compact
audi	a4	3.10	2008	6	auto(av)	f	18	27	p	compact
audi	a4 quattro	1.80	1999	4	manual(m5)	4	18	26	p	compact
audi	a4 quattro	1.80	1999	4	auto(l5)	4	16	25	p	compact
audi	a4 quattro	2.00	2008	4	manual(m6)	4	20	28	p	compact

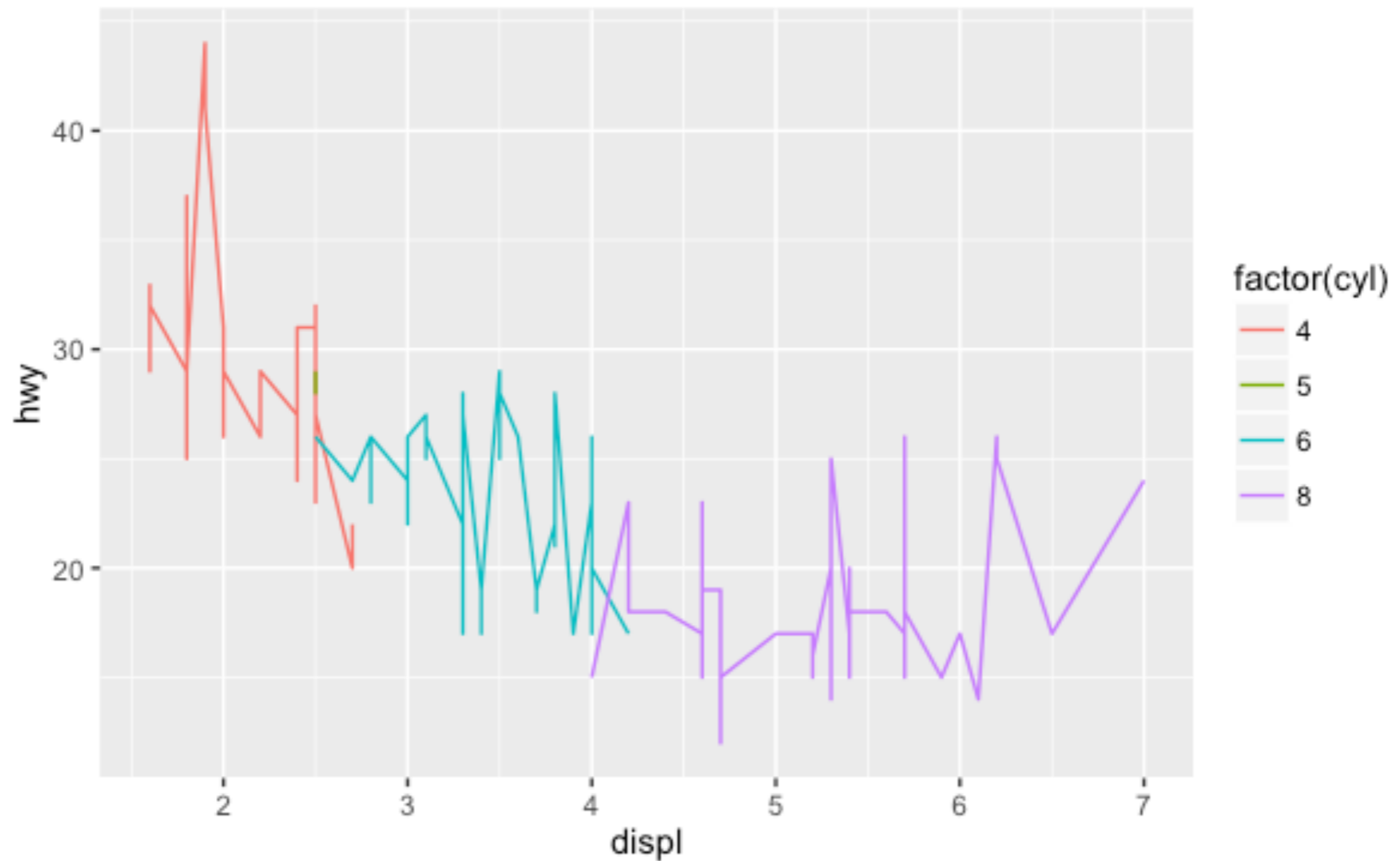
```
qplot(displ, hwy, data = mpg, colour = factor(cyl))
```



Disp映射到x坐标, hwy映射到y坐标, cyl映射到颜色

manufacturer	model	disp	year	cyl	cty	hwy	class	x	y	colour
audi	a4	1.8	1999	4	18	29	compact	1.8	29	4
audi	a4	1.8	1999	4	21	29	compact	1.8	29	4
audi	a4	2.0	2008	4	20	31	compact	2.0	31	4
audi	a4	2.0	2008	4	21	30	compact	2.0	30	4
audi	a4	2.8	1999	6	16	26	compact	2.8	26	6
audi	a4	2.8	1999	6	18	26	compact	2.8	26	6
audi	a4	3.1	2008	6	18	27	compact	3.1	27	6
audi	a4 quattro	1.8	1999	4	18	26	compact	1.8	26	4
audi	a4 quattro	1.8	1999	4	16	25	compact	1.8	25	4
audi	a4 quattro	2.0	2008	4	20	28	compact	2.0	28	4

```
qplot(displ, hwy, data=mpg, colour=factor(cyl), geom="line")
```



- 把数据从其计量单位（例如油耗的升数，里程等）转化为计算机能识别的显示要素（例如像素，颜色等）的过程，称为 **Scaling**

- 在右图中有几项scaling

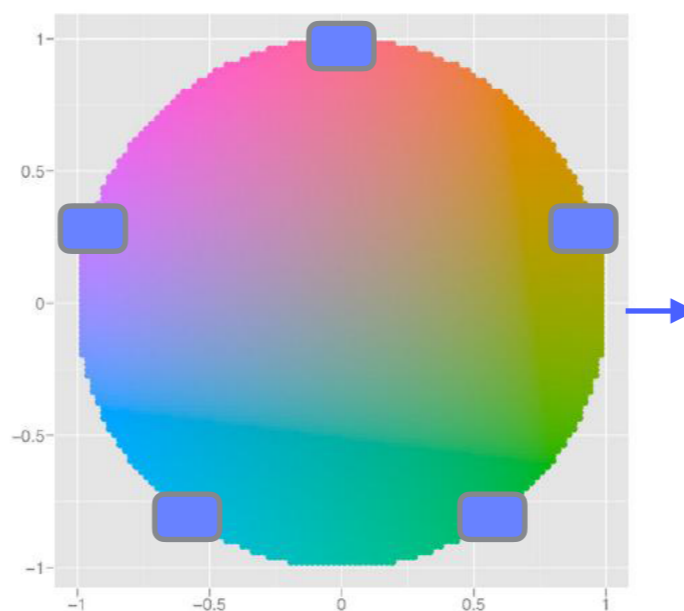
*将水平坐标x映射到[0,1]区间。这里不使用具体像素值的原因是grid包替我们完成最终的转换

*将垂直坐标y映射到[0,1]区间
*由坐标系统(coord)根据x,y

的组合最终定位，常见的坐标系统包括直角坐标系，极坐标系，球面映射等

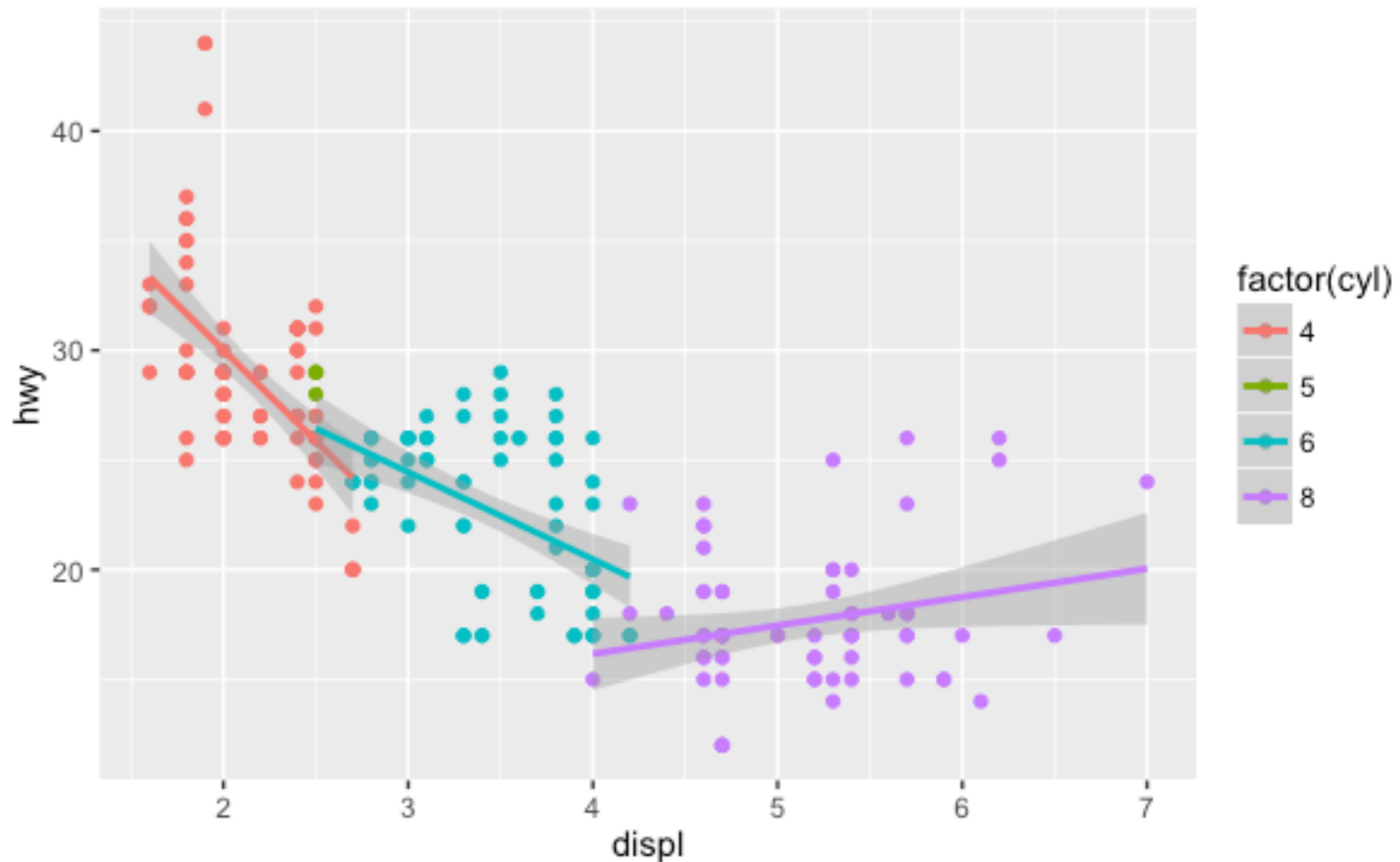
*颜色的scaling

x	y	colour
1.8	29	4
1.8	29	4
2.0	31	4
2.0	30	4
2.8	26	6
2.8	26	6
3.1	27	6
1.8	26	4
1.8	25	4
2.0	28	4

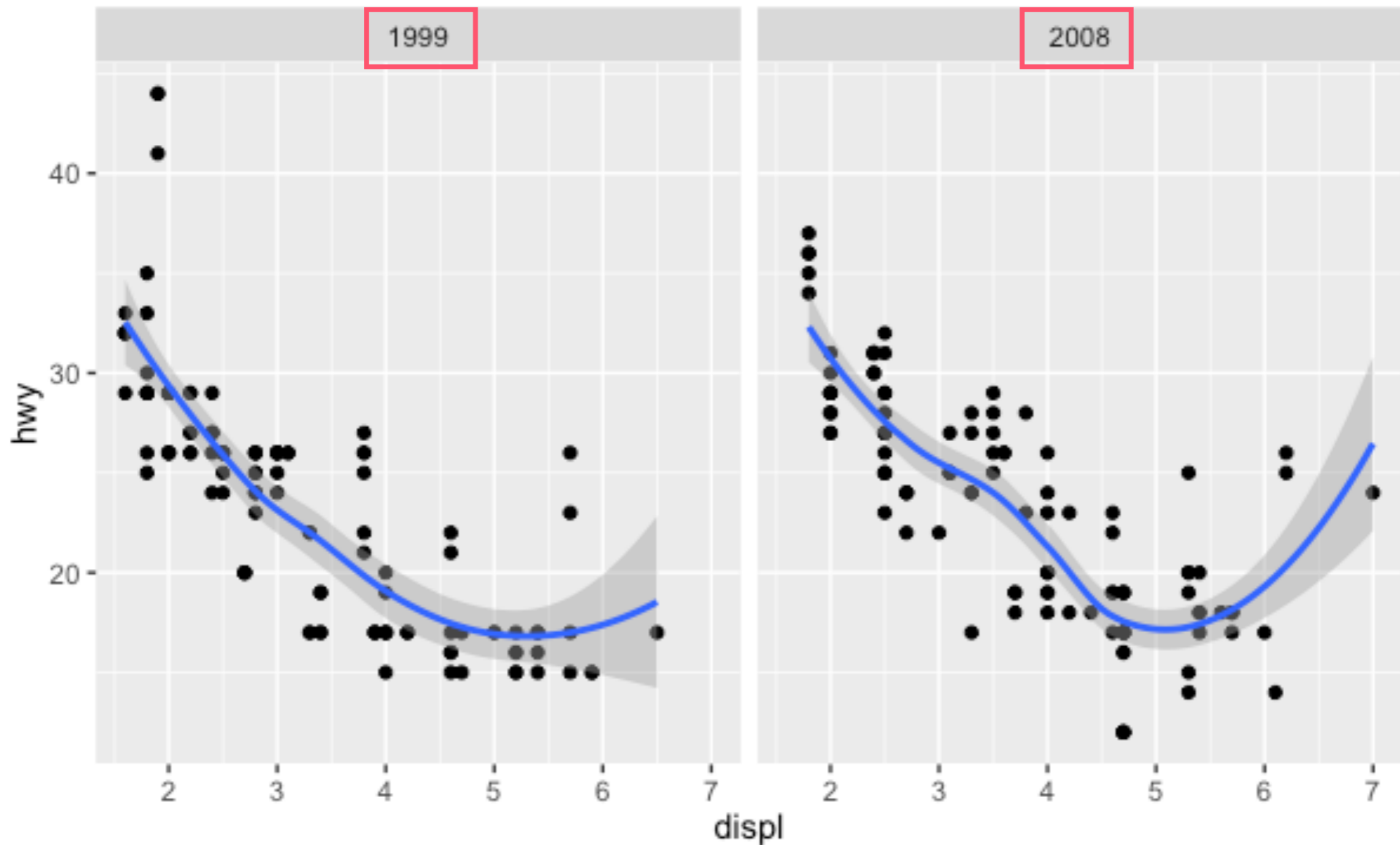


x	y	colour	size	shape
0.037	0.531	#FF6C91	1	19
0.037	0.531	#FF6C91	1	19
0.074	0.594	#FF6C91	1	19
0.074	0.562	#FF6C91	1	19
0.222	0.438	#00C1A9	1	19
0.222	0.438	#00C1A9	1	19
0.278	0.469	#00C1A9	1	19
0.037	0.438	#FF6C91	1	19
0.037	0.406	#FF6C91	1	19
0.074	0.500	#FF6C91	1	19

```
qplot(displ, hwy, data=mpg, colour=factor(cyl)) +  
geom_smooth(data= subset(mpg, cyl != 5), method="lm")
```

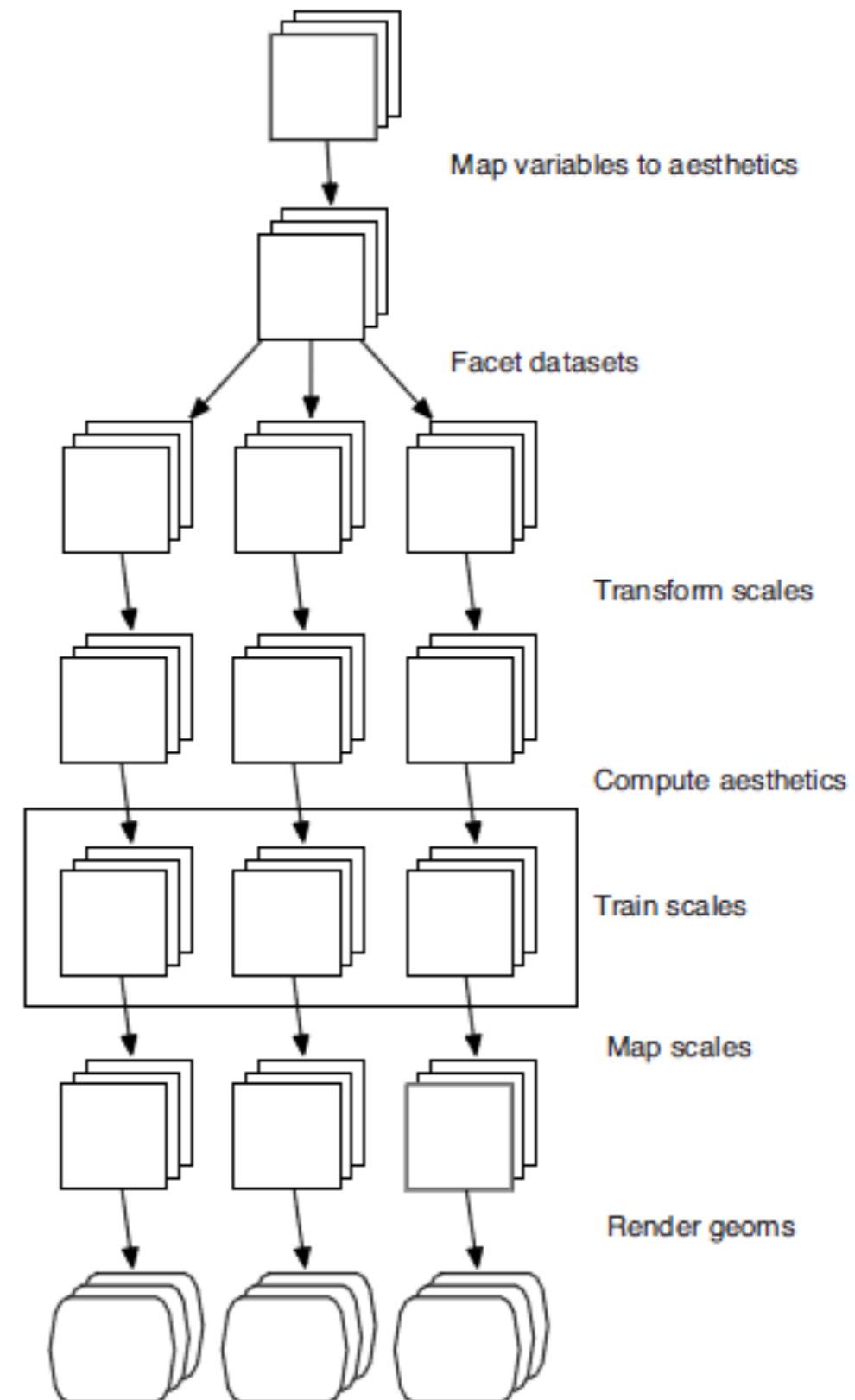


```
qplot(displ, hwy, data=mpg, facets = . ~ year) + geom_smooth()
```

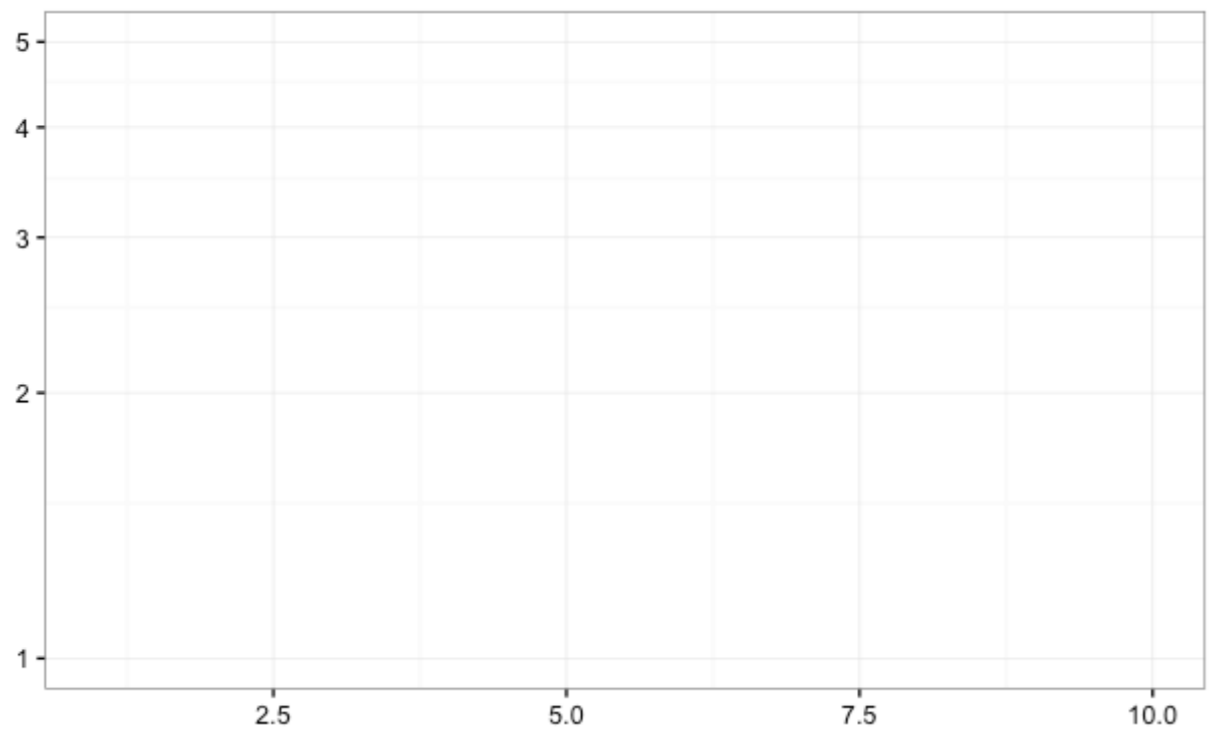
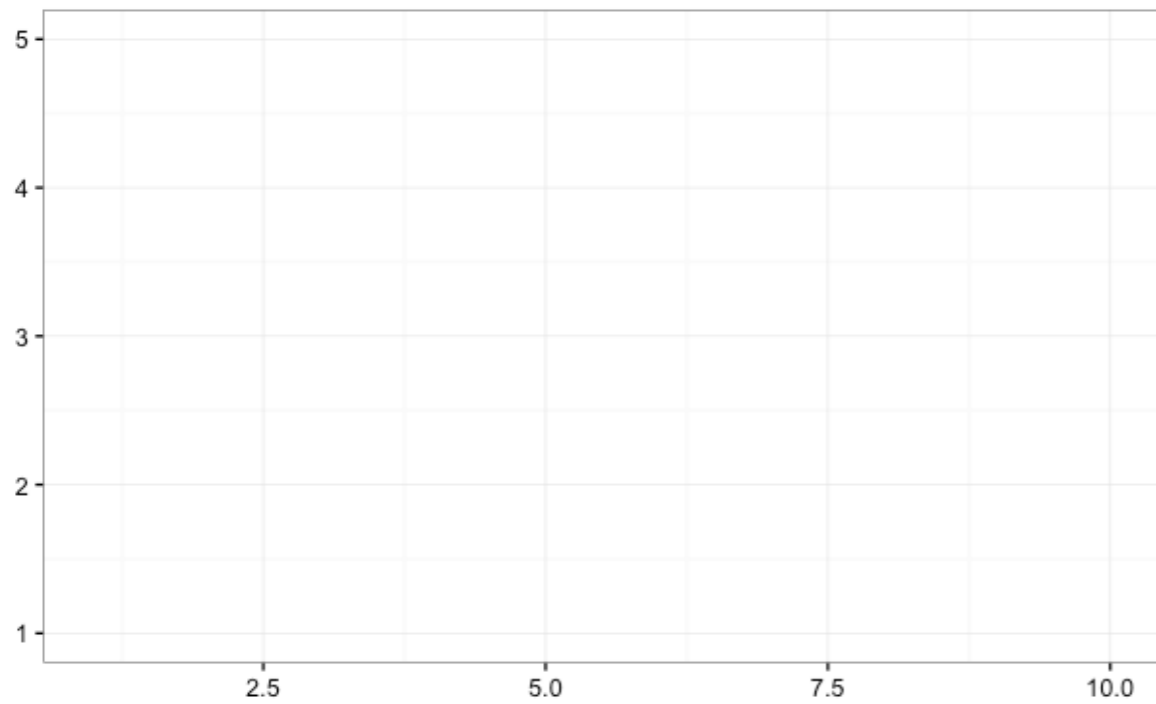


- 将变量映射到图形属性
- 对数据进行分面处理
- 标度转换
- 计算图形属性
- 标度训练
- 标度影射
- 渲染几何对象

图层

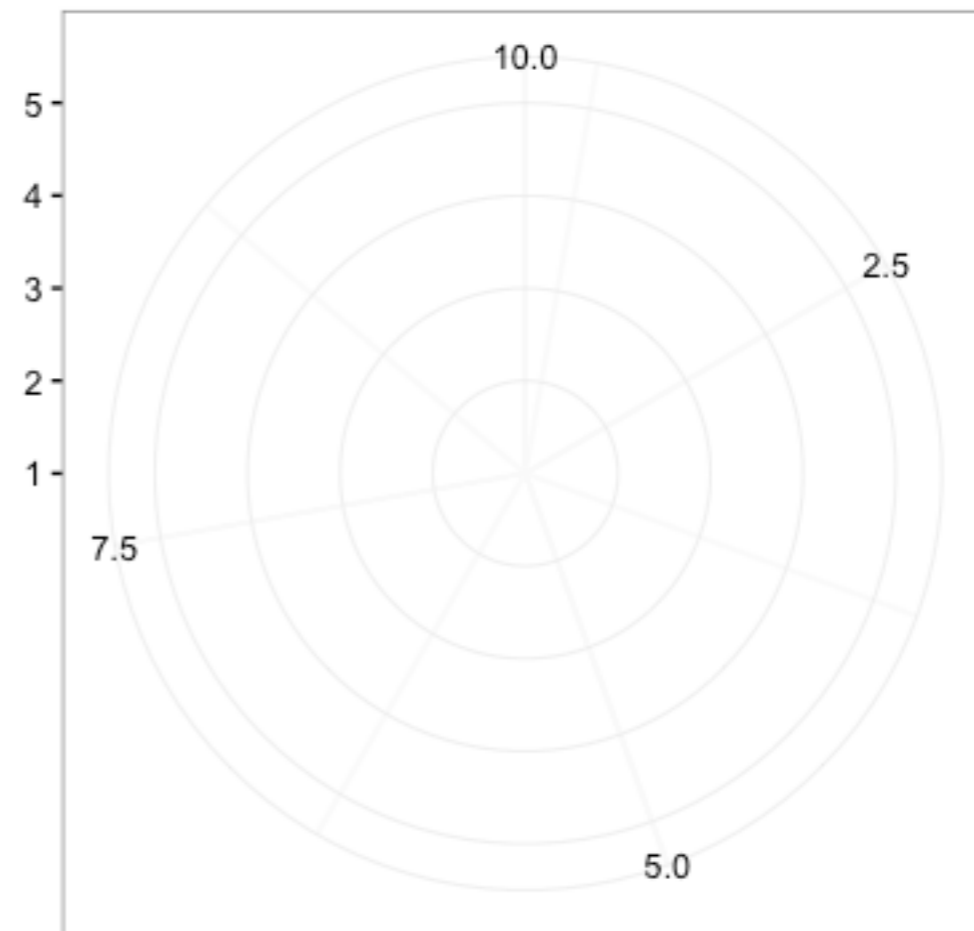


笛卡尔



半对数

极坐标

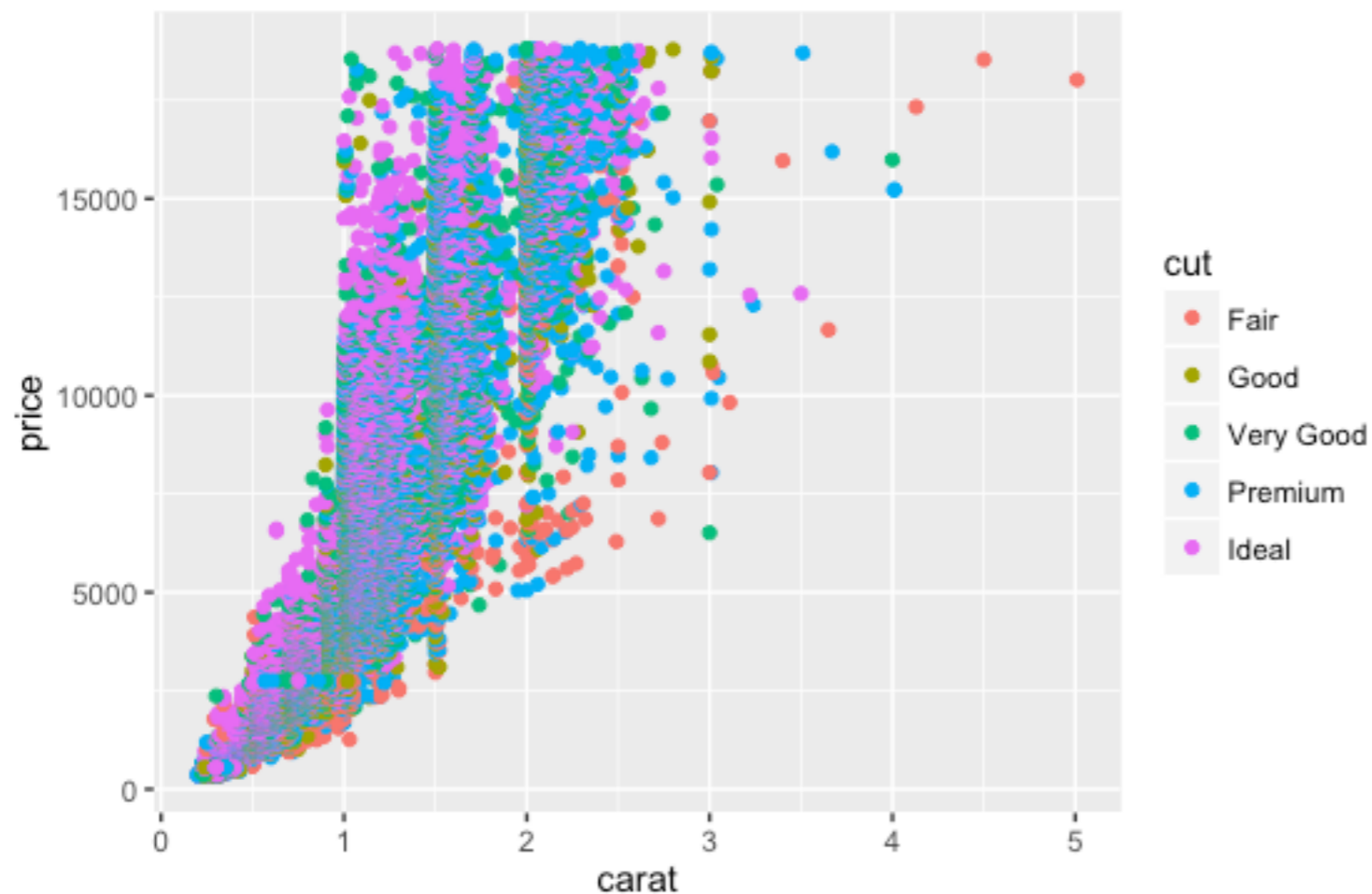


极坐标

用图层构建图形

```
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  
)
```

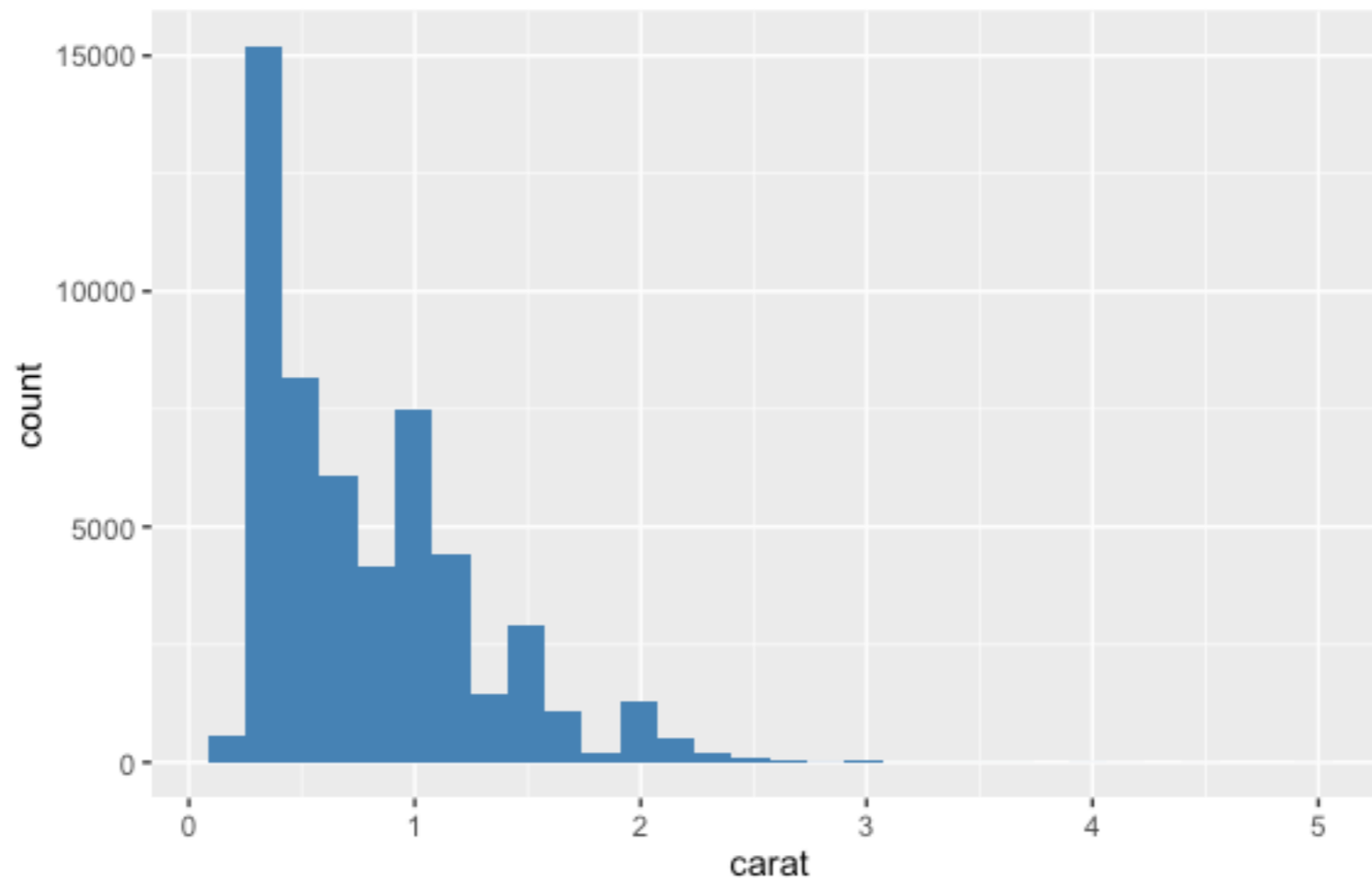
见教材ggplot2的60页

```
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()
```

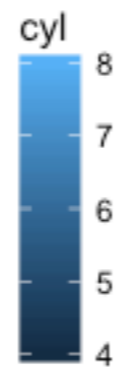
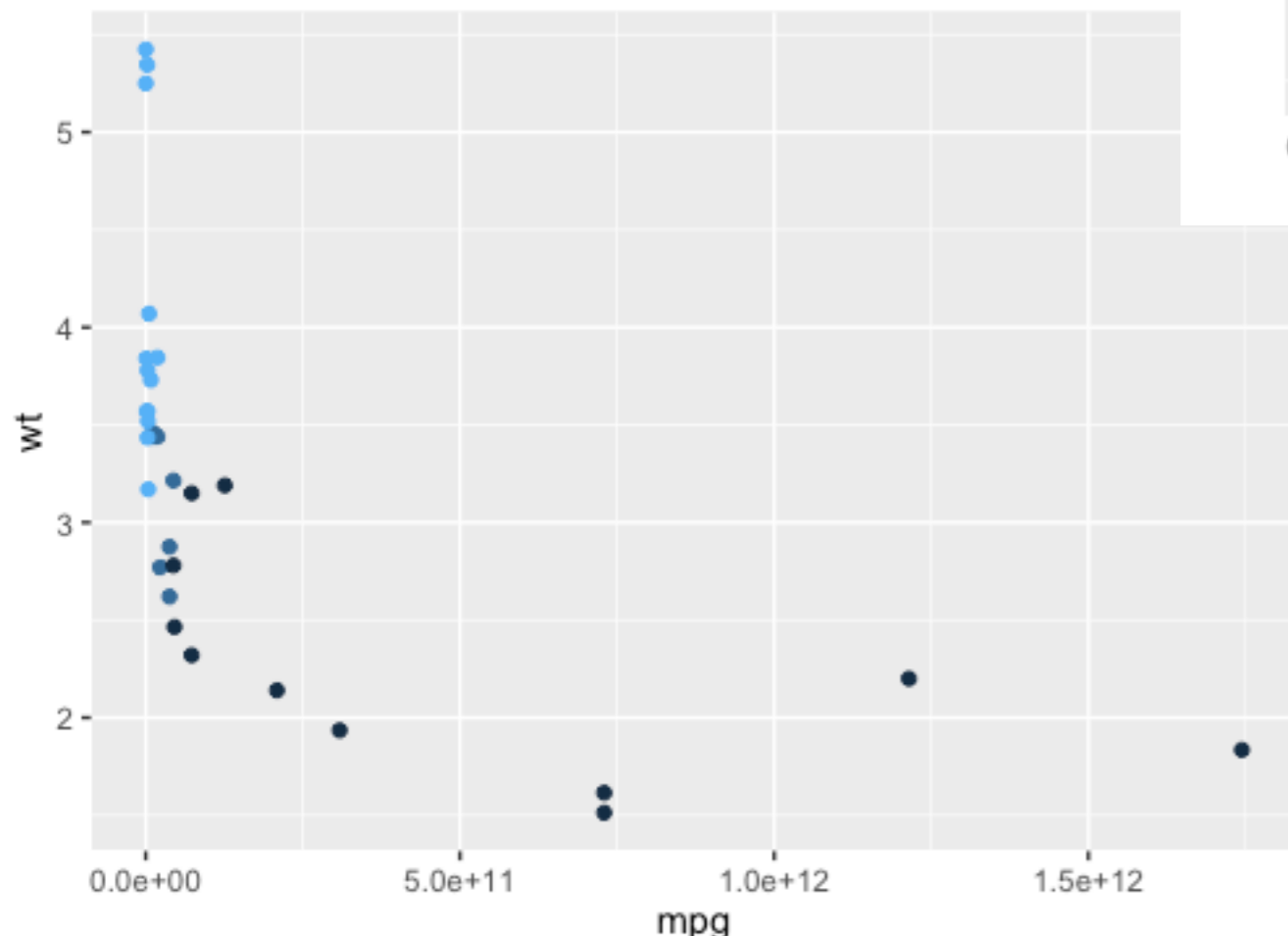
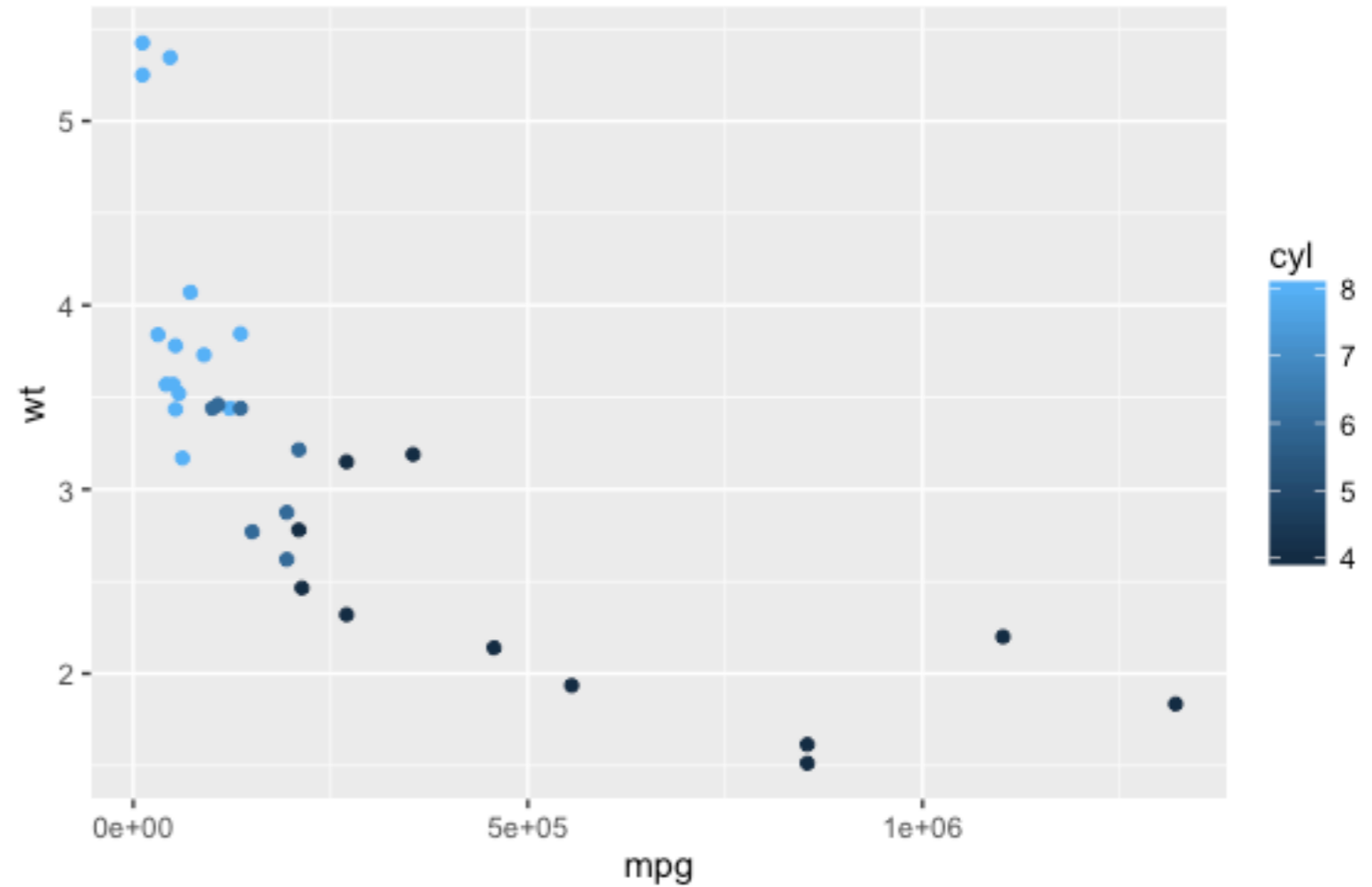
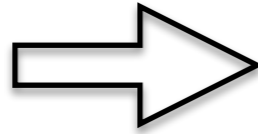
```
> p <- p + geom_point()  
> summary(p)
```

```
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()
```

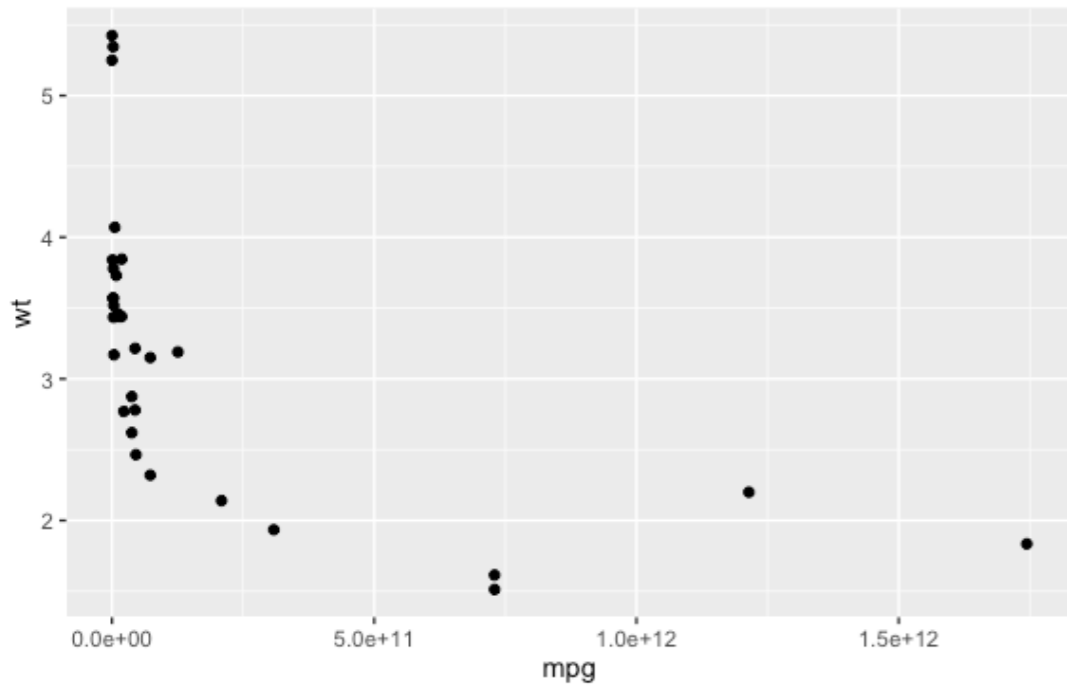
```
-----  
geom_point: na.rm = FALSE  
stat_identity: na.rm = FALSE  
position_identity
```



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



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

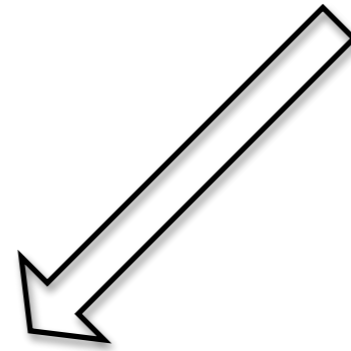


```
p <- ggplot(mtcars, aes(x = mpg, y = wt))
```

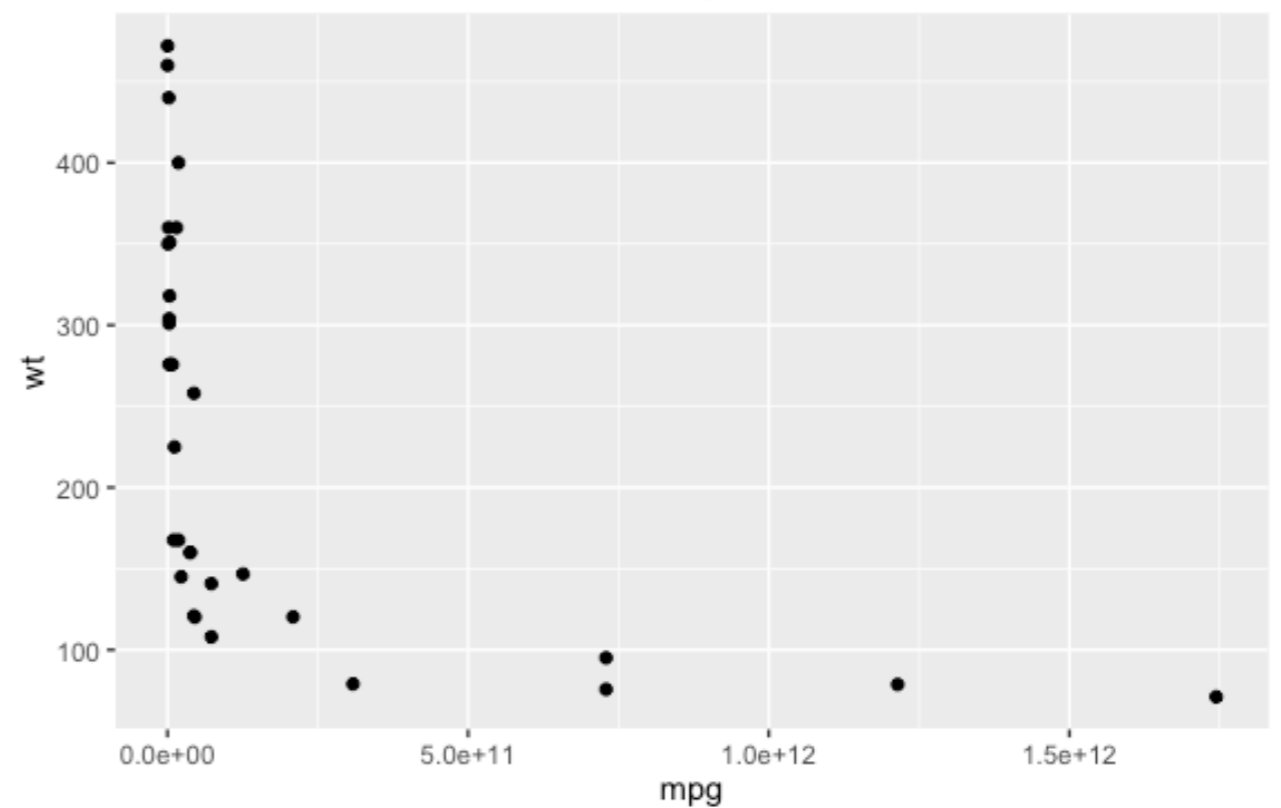
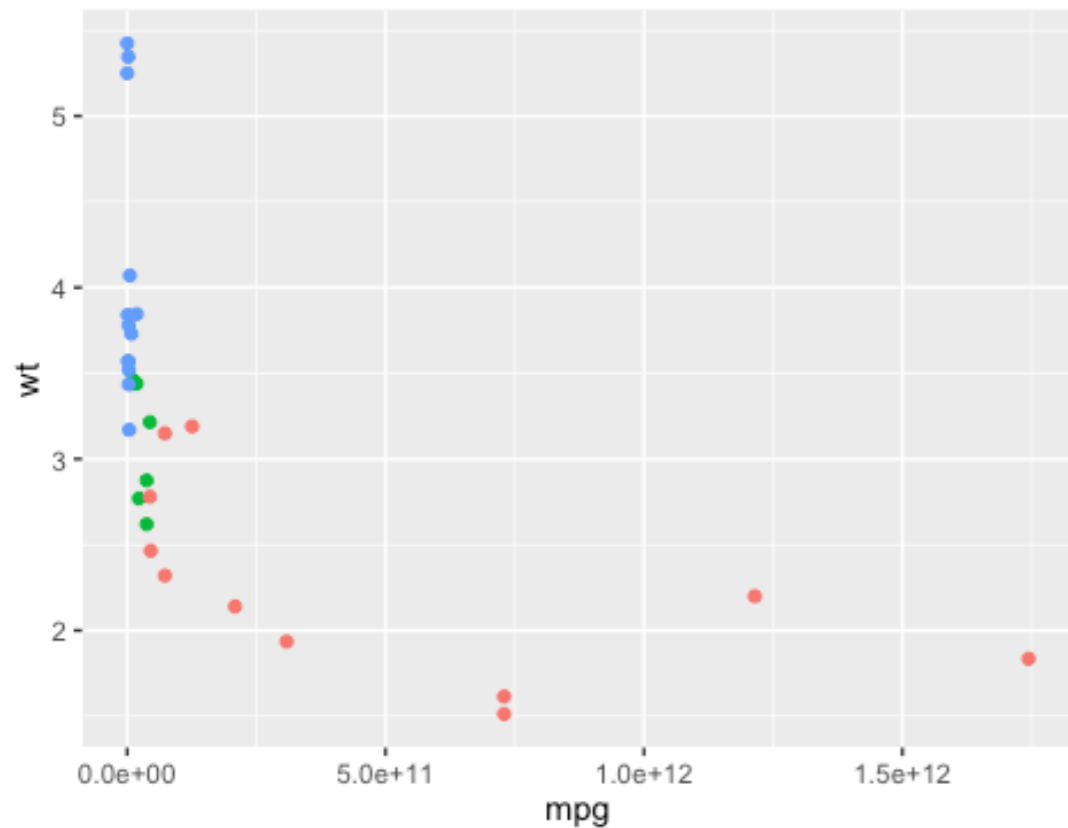
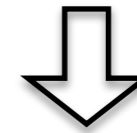


```
p + geom_point()
```

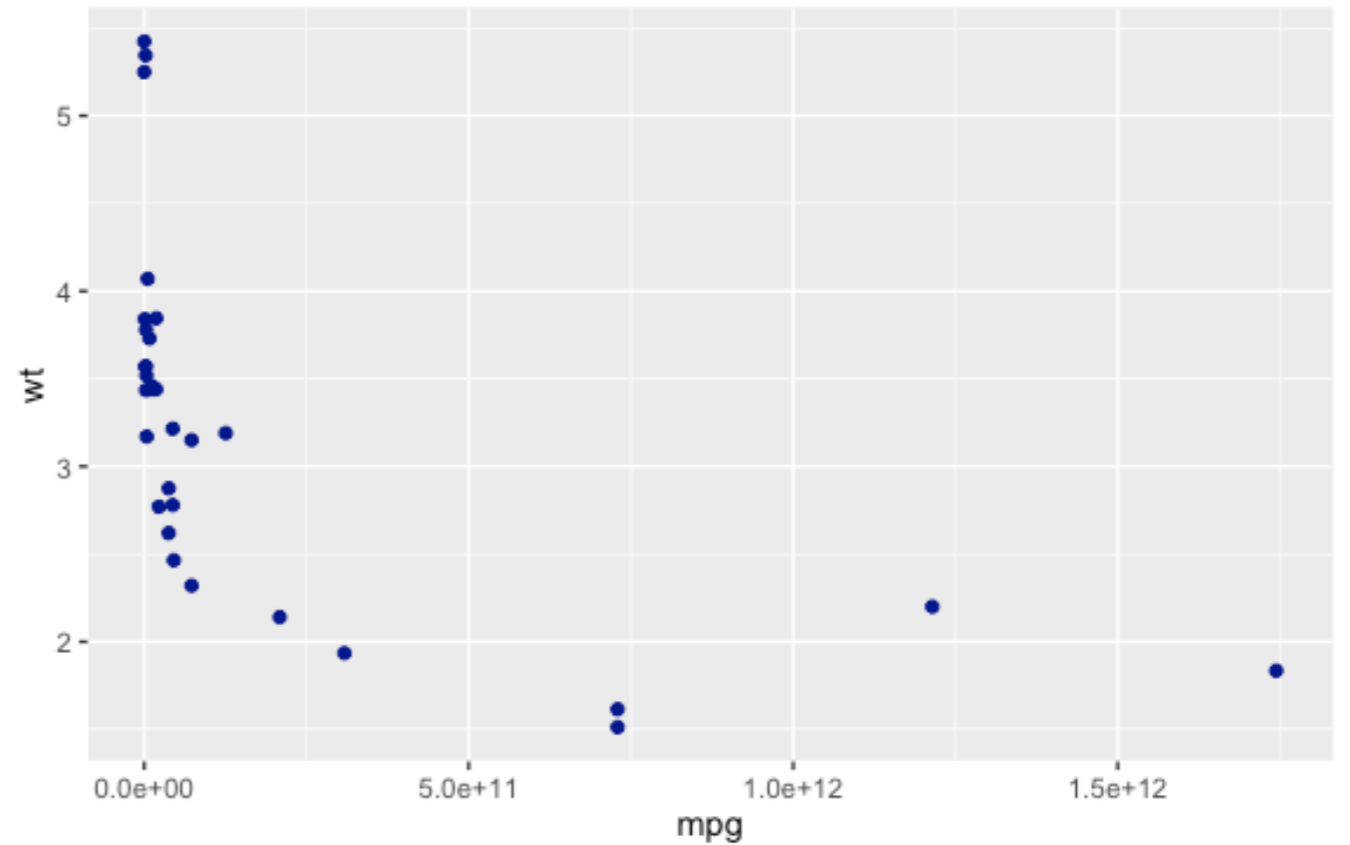
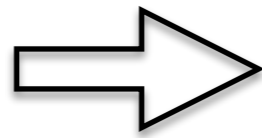
```
p + geom_point(aes(colour = factor(cyl)))
```



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

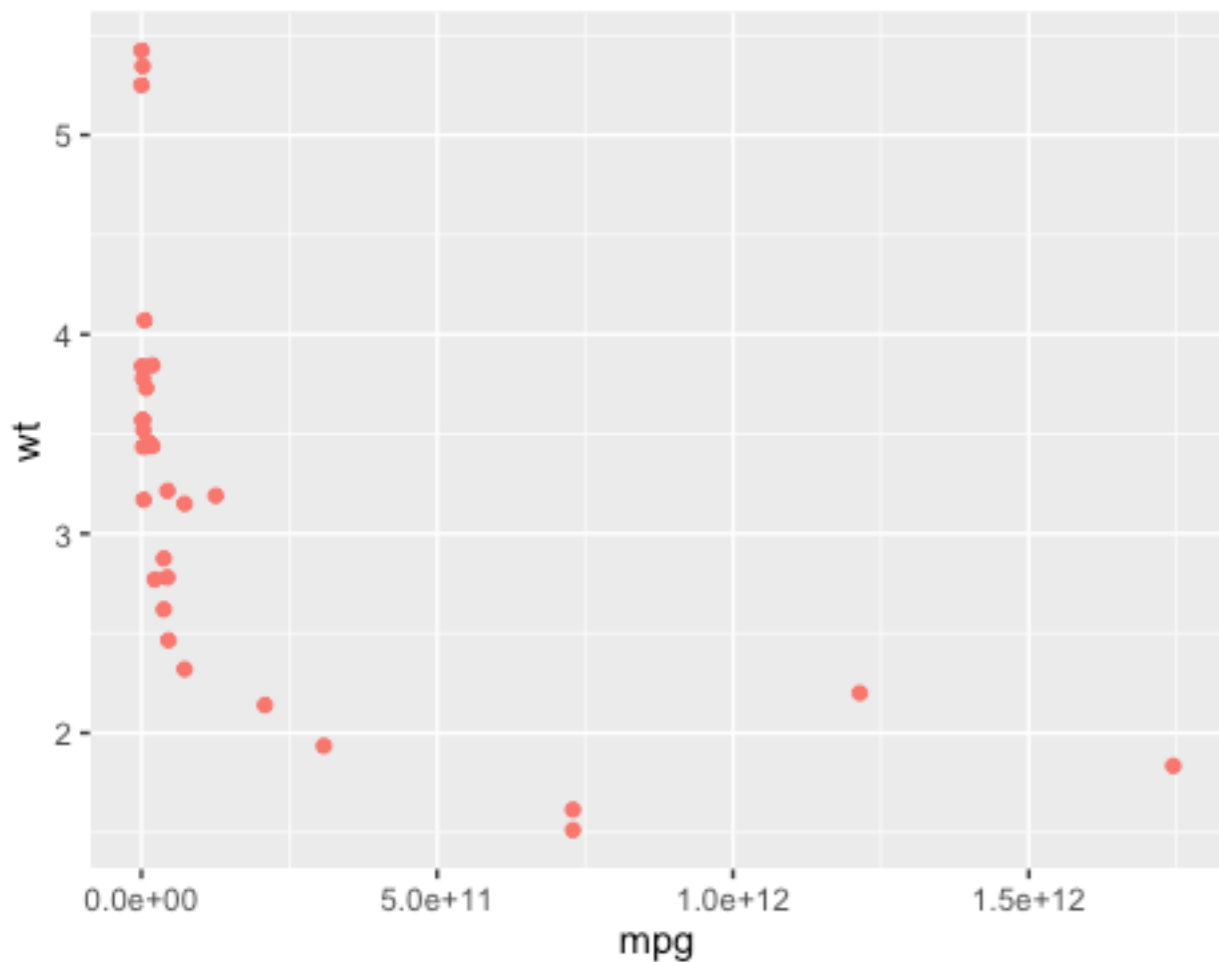


```
p <- ggplot(mtcars, aes(mpg, wt))  
p + geom_point(colour = "darkblue")
```

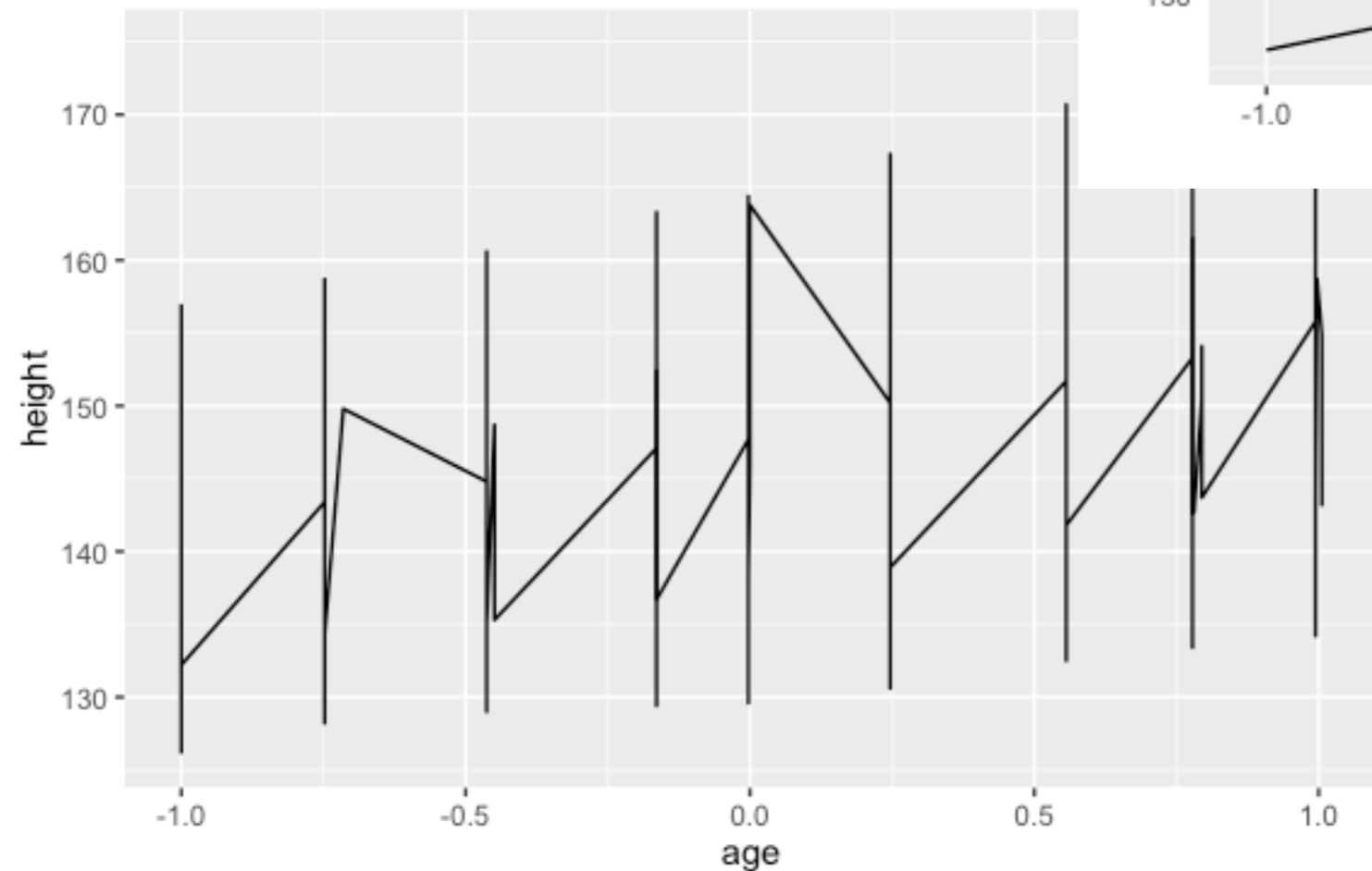
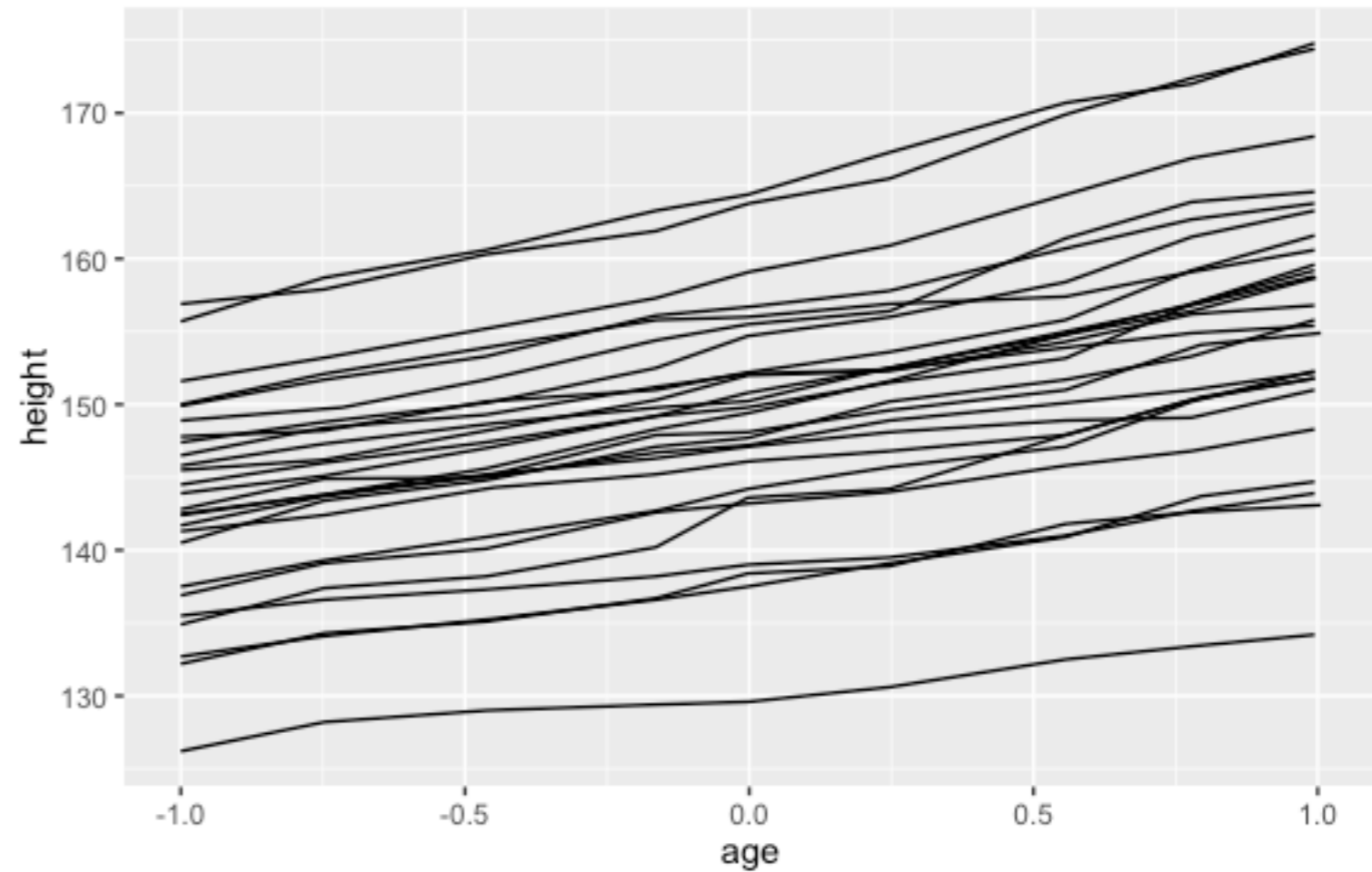
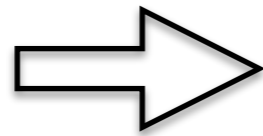


colour
• darkblue

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

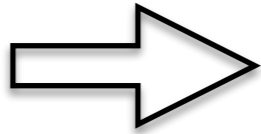


```
p <- ggplot(Oxboys,  
  aes(age,  
    height,  
    group = Subject)  
  )  
+ geom_line()
```

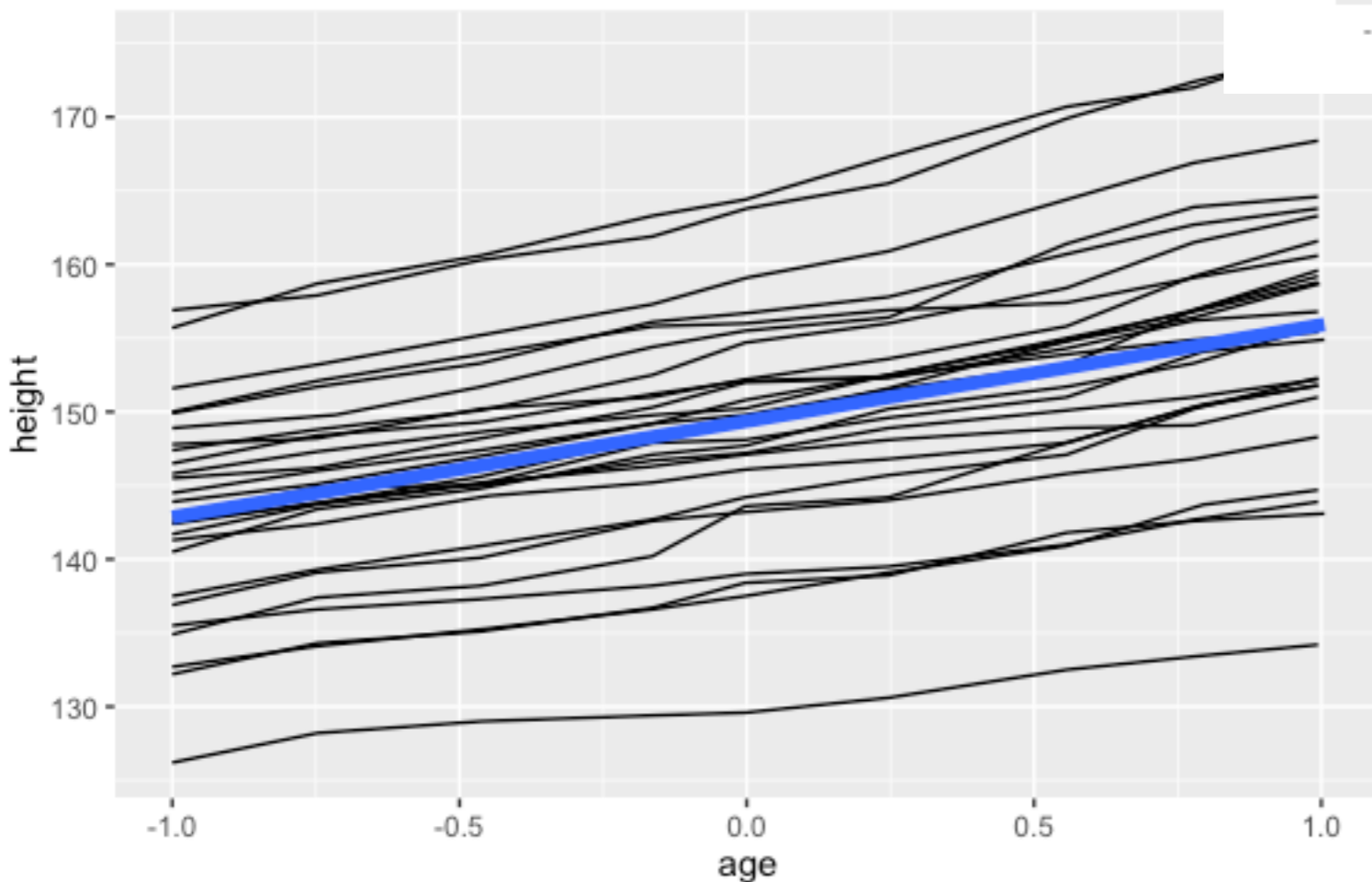
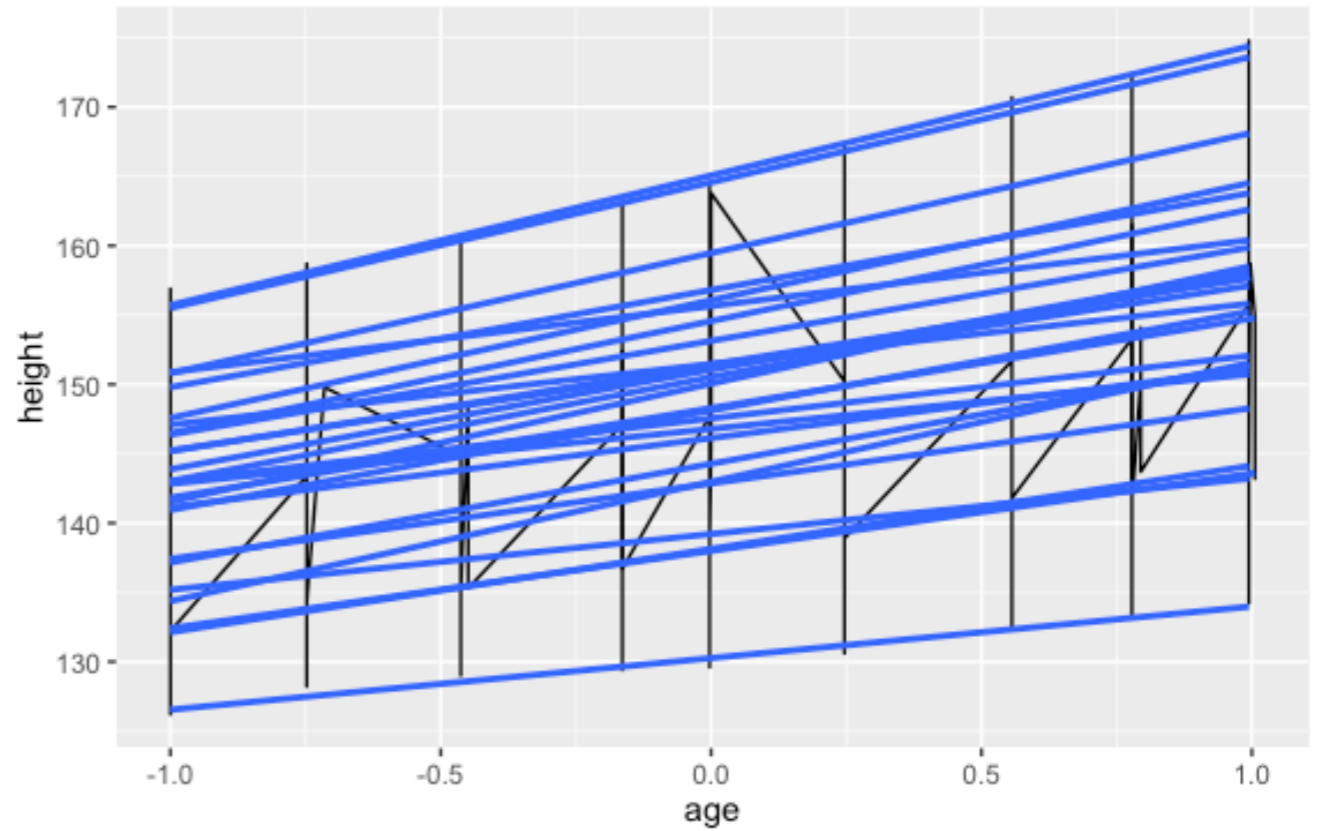


```
p <- ggplot(Oxboys,  
  aes(age,  
    height,  
    group = 1)  
  )  
+ geom_line()
```

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

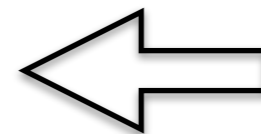


```
p + geom_smooth(aes(group = Subject),  
  method="lm",  
  se = F)
```



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

```
p + geom_smooth(aes(group = 1),  
  method="lm",  
  se = F)
```



提问时间!

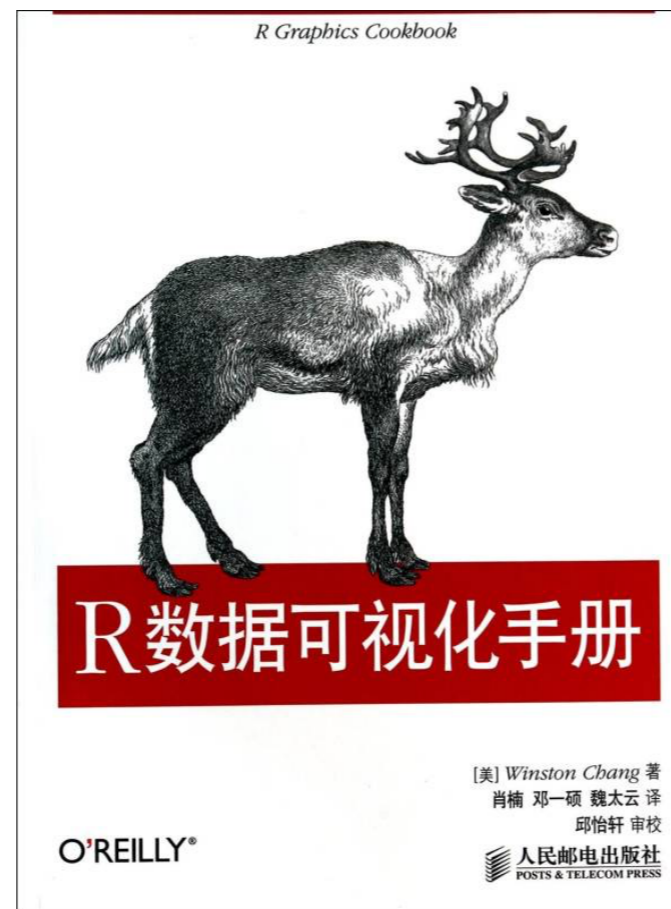
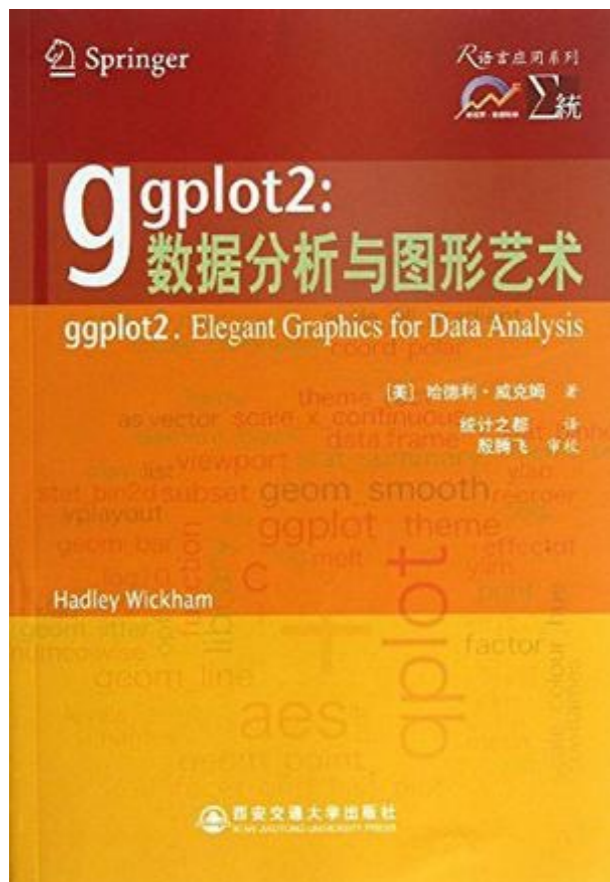
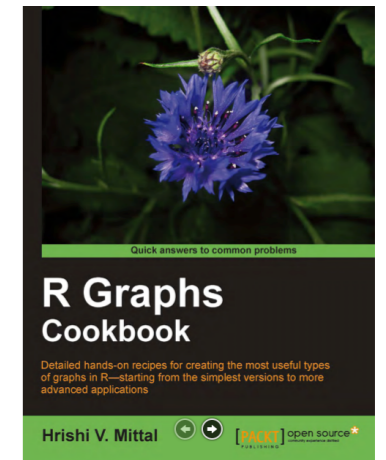
孙惠平

sunhp@ss.pku.edu.cn

R语言卡片

练习

- ggplot2的1-4章，熟悉所有例子。
- R数据可视化手册的2-5章，熟悉所有例子。
- 教材RIA（第二版）的第19章，熟悉所有例子。



INTERACTIVE COURSE

Introduction to Data Visualization with ggplot2

Start Course For Free

Bookmark

⌚ 4 hours ▶ 14 Videos ↔ 52 Exercises 👤 41,319 Participants 📊 4,300 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