ggplot

mpg

```
## # A tibble: 234 x 11
##    manufacturer model    displ  year   cyl trans    drv     cty   hwy fl    class
##    <chr>        <chr>    <dbl> <int> <int> <chr>    <chr> <int> <int> <chr> <chr>
##  1 audi         a4         1.8  1999     4 auto(l~ f        18    29 p     comp~
##  2 audi         a4         1.8  1999     4 manual~ f        21    29 p     comp~
##  3 audi         a4         2    2008     4 manual~ f        20    31 p     comp~
##  4 audi         a4         2    2008     4 auto(a~ f        21    30 p     comp~
##  5 audi         a4         2.8  1999     6 auto(l~ f        16    26 p     comp~
##  6 audi         a4         2.8  1999     6 manual~ f        18    26 p     comp~
##  7 audi         a4         3.1  2008     6 auto(a~ f        18    27 p     comp~
##  8 audi         a4 quat~   1.8  1999     4 manual~ 4        18    26 p     comp~
##  9 audi         a4 quat~   1.8  1999     4 auto(l~ 4        16    25 p     comp~
## 10 audi         a4 quat~   2    2008     4 manual~ 4        20    28 p     comp~
## # ... with 224 more rows
```
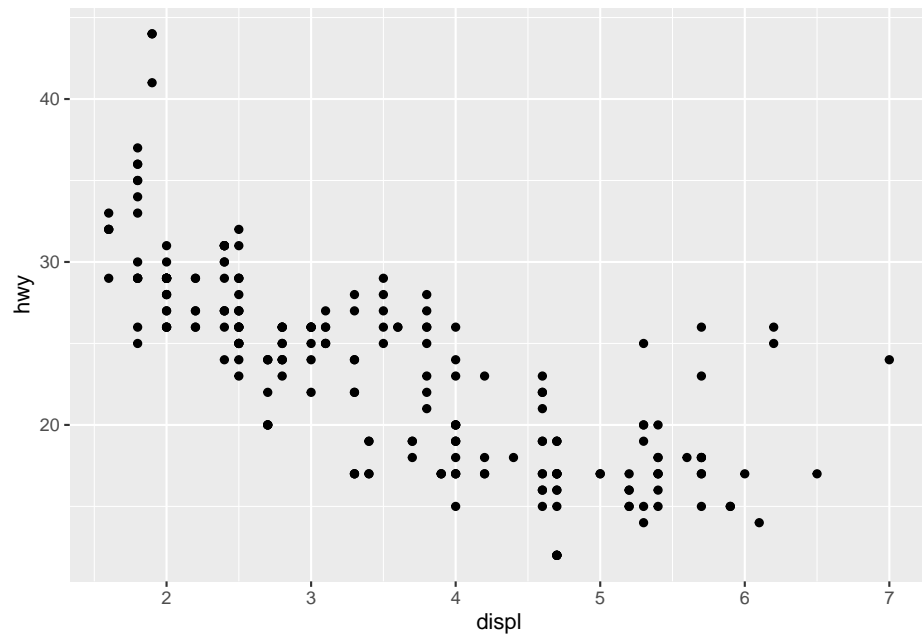
```r
ggplot(data=mpg)+geom_point(mapping=aes(x=displ,y=hwy))
```
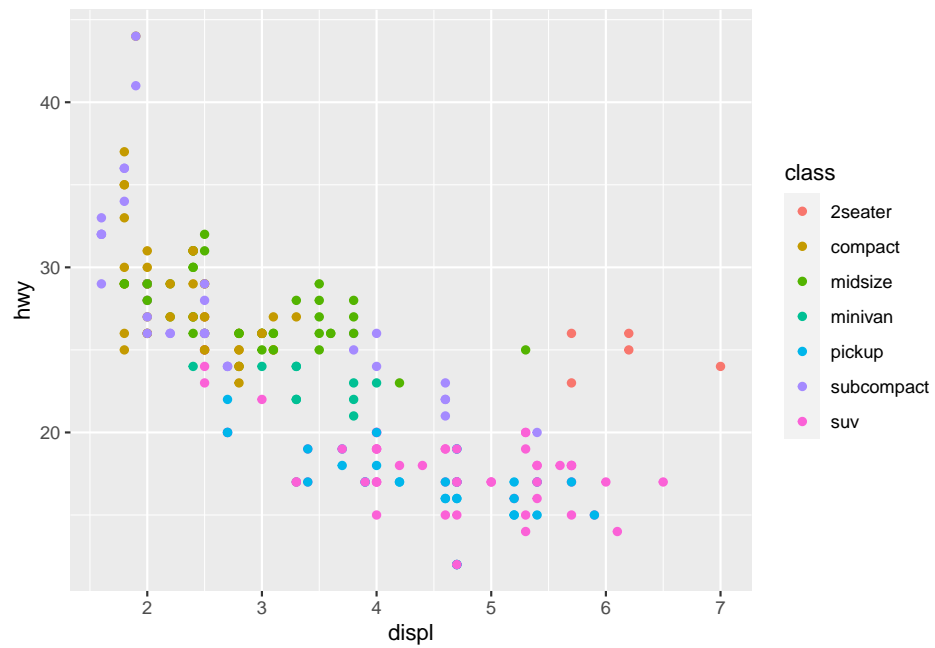
ggplot() 相当于一个不断叠加的图层

模板：ggplot(data=)+(mapping=aes())
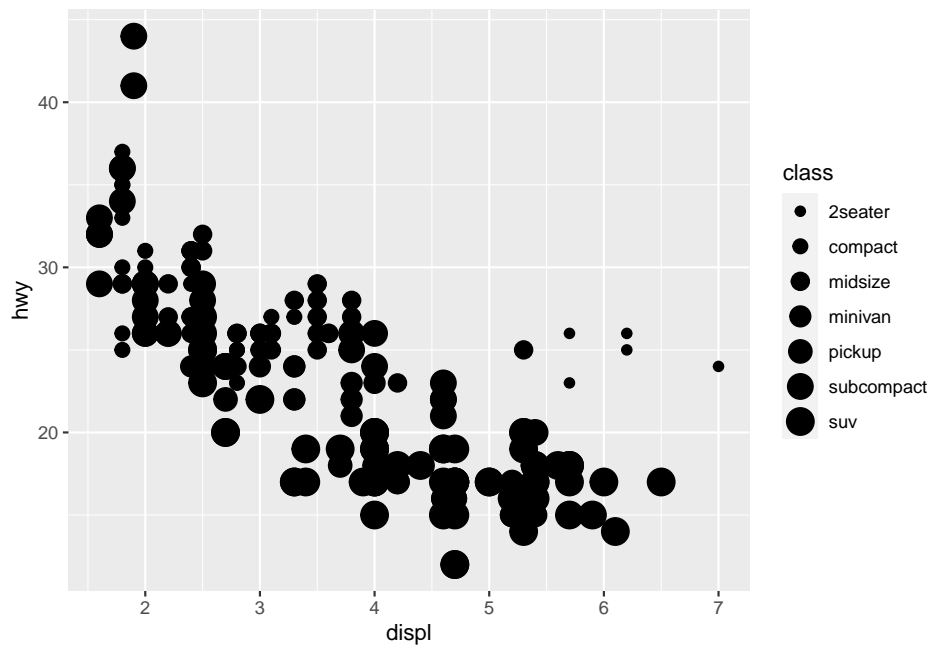
0.1 图形属性映射

```
ggplot(data=mpg)+
  geom_point(mapping=aes(x=displ,y=hwy,color=class))
```
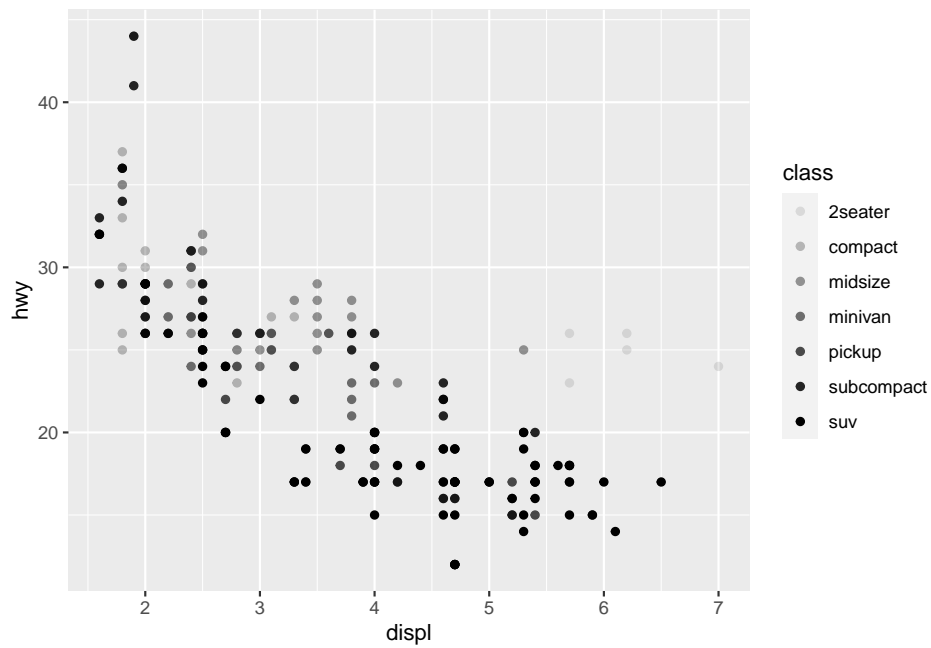
但是不建议将无序变量与 **size** 联系起来

```r
ggplot(data=mpg)+
  geom_point(mapping=aes(x=displ,y=hwy,size=class))
```

```
## Warning: Using size for a discrete variable is not advised.
```

```
ggplot(data=mpg)+
  geom_point(mapping=aes(x=displ,y=hwy,alpha=class))
```

```
## Warning: Using alpha for a discrete variable is not advised.
```

```
ggplot(data=mpg)+
  geom_point(mapping=aes(x=displ,y=hwy,shape=class))
```
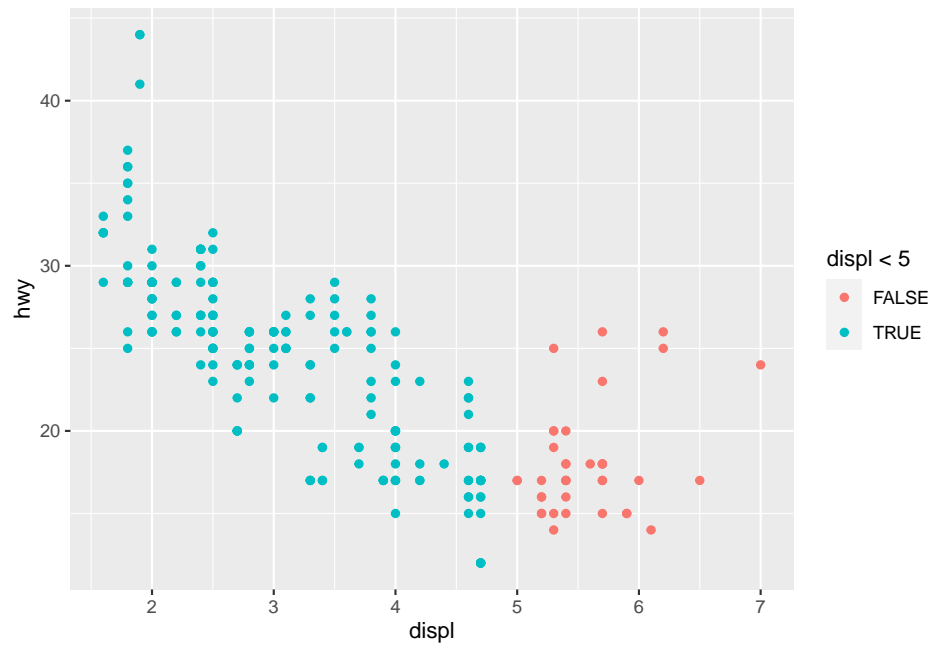
```
## Warning: The shape palette can deal with a maximum of 6 discrete values because
## more than 6 becomes difficult to discriminate; you have 7. Consider
## specifying shapes manually if you must have them.
```

```
## Warning: Removed 62 rows containing missing values (geom_point).
```
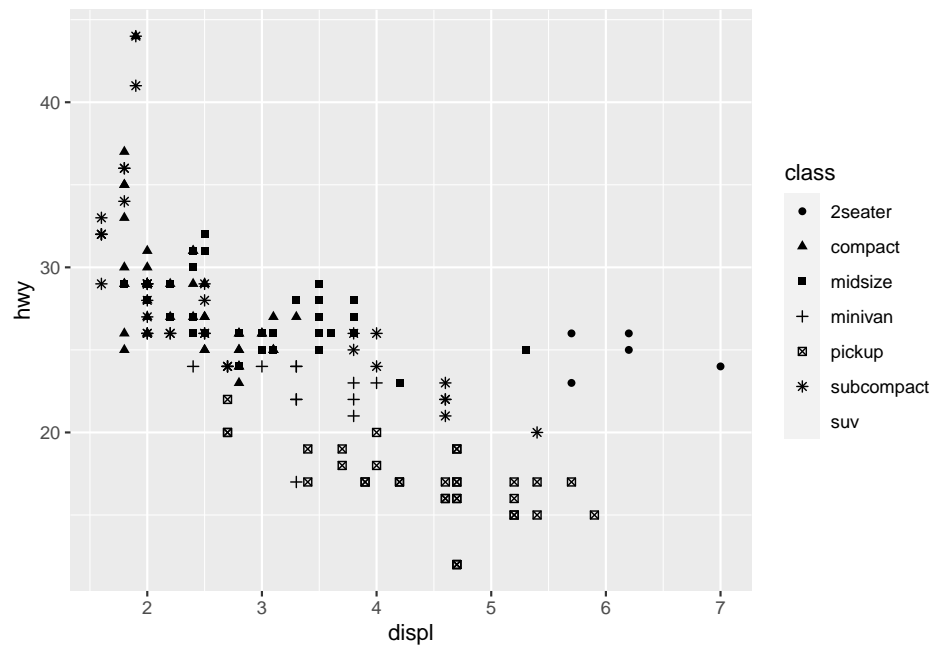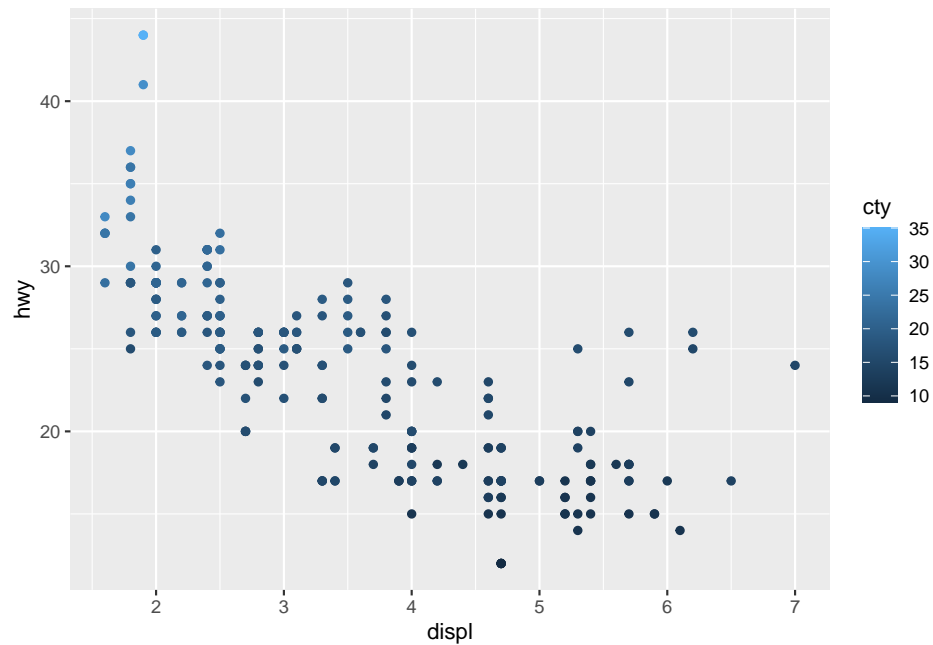
变量类型过多之后 **shape** 和 **size** 就显示不出来了

```r
ggplot(data=mpg)+
  geom_point(mapping=aes(x=displ,y=hwy,color=displ<5))
```
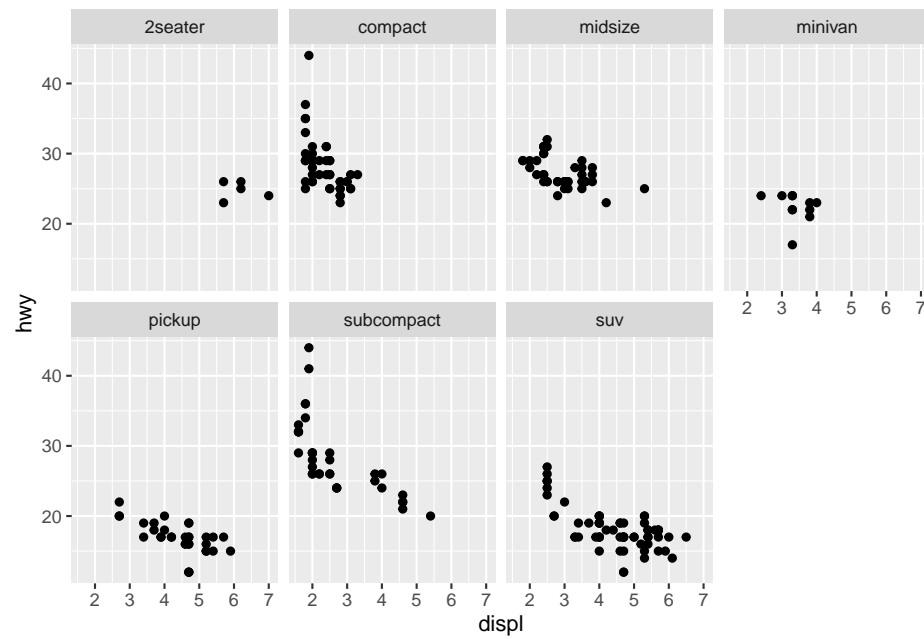
对于连续变量可以用 color 和 size 但是不能用 shape

```
ggplot(data=mpg)+
  geom_point(mapping=aes(x=displ,y=hwy,color=cty))
```
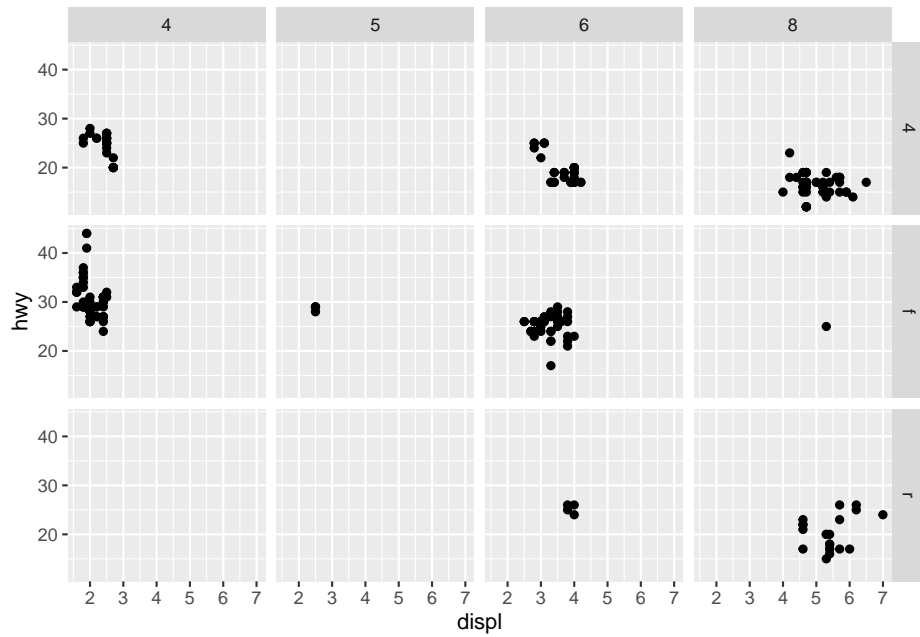

0.2 分面

facet_wrap() 第一个参数为~+ 变量名, 变量必须是离散的。

```
ggplot(data=mpg)+
  geom_point(mapping=aes(x=displ,y=hwy))+
  facet_wrap(~class,nrow=2)
```
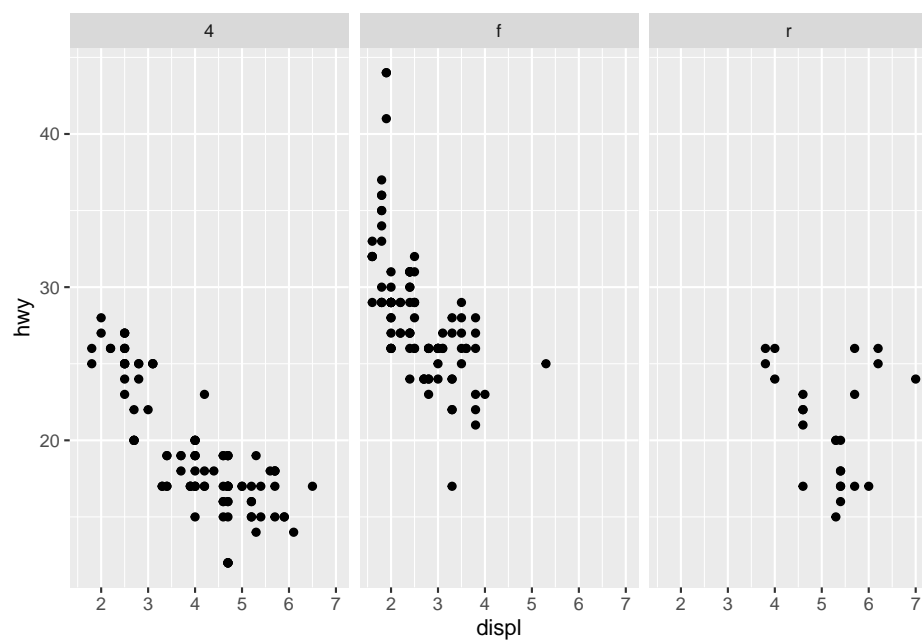
facet_grid() 可以对两个变量进行分面

```
ggplot(data=mpg)+
  geom_point(mapping=aes(x=displ,y=hwy))+
  facet_grid(drv~cyl)
```
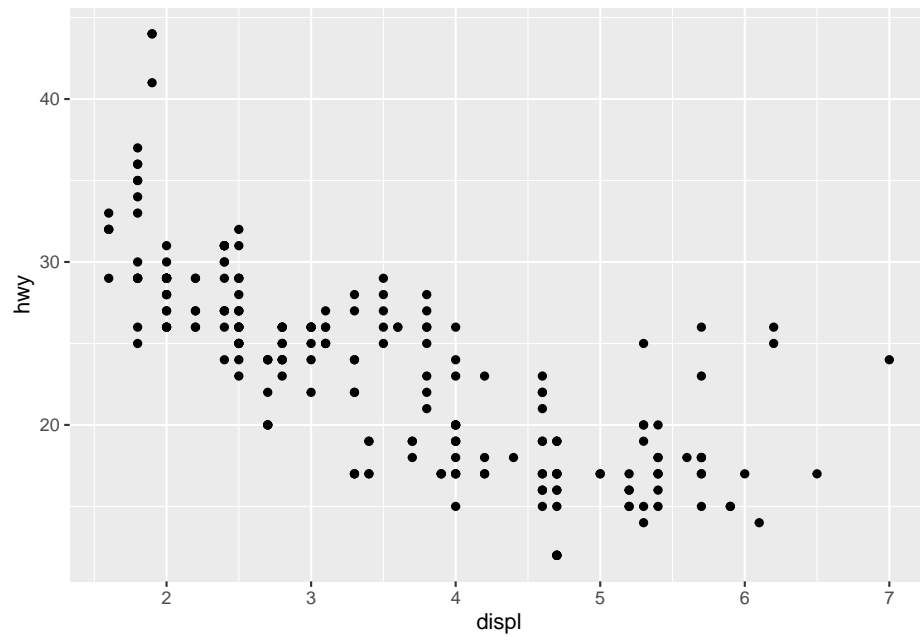
空白图表示同时满足二者的车并不存在

真对连续变量分面的话就会有许多

```r
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
    facet_grid(.~drv) # 与facet_wrap(~drv)相同
```
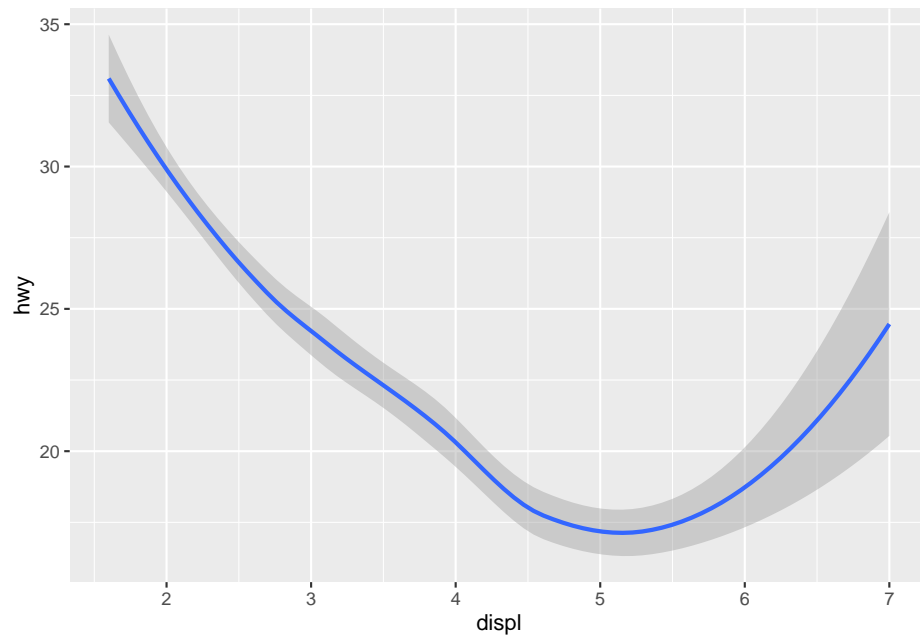
0.3 几何图像

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```
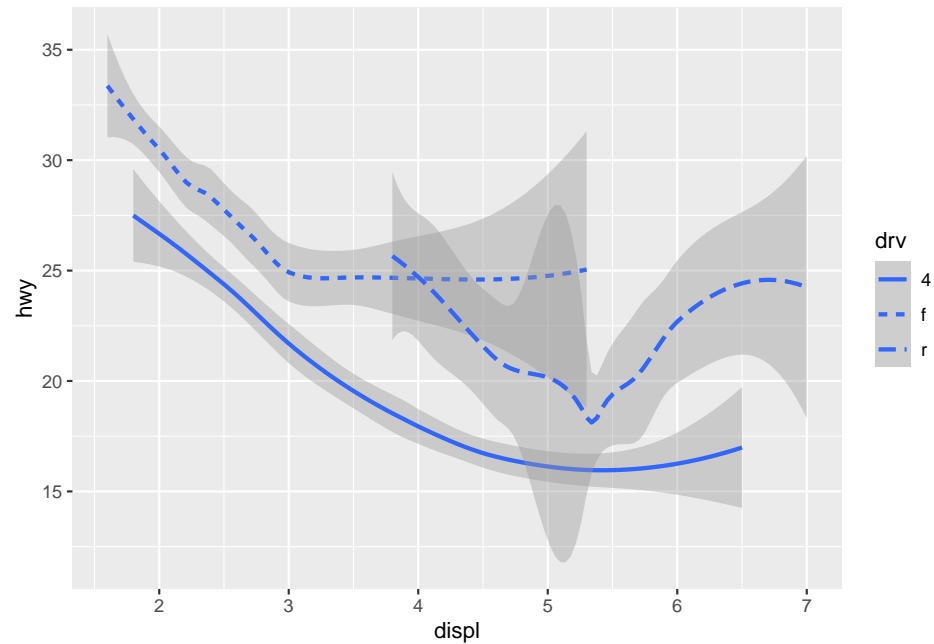
```
ggplot(data = mpg) +
  geom_smooth(mapping = aes(x = displ, y = hwy))
```

`geom_smooth()` using method = 'loess' and formula 'y ~ x'

```r
ggplot(data = mpg) +
  geom_smooth(mapping = aes(x = displ, y = hwy,linetype=drv))
```
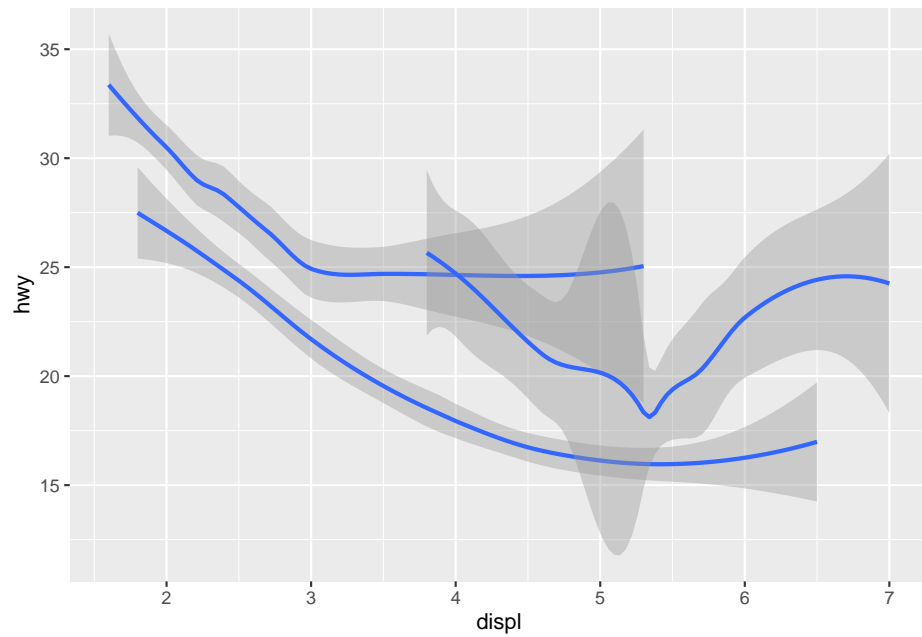
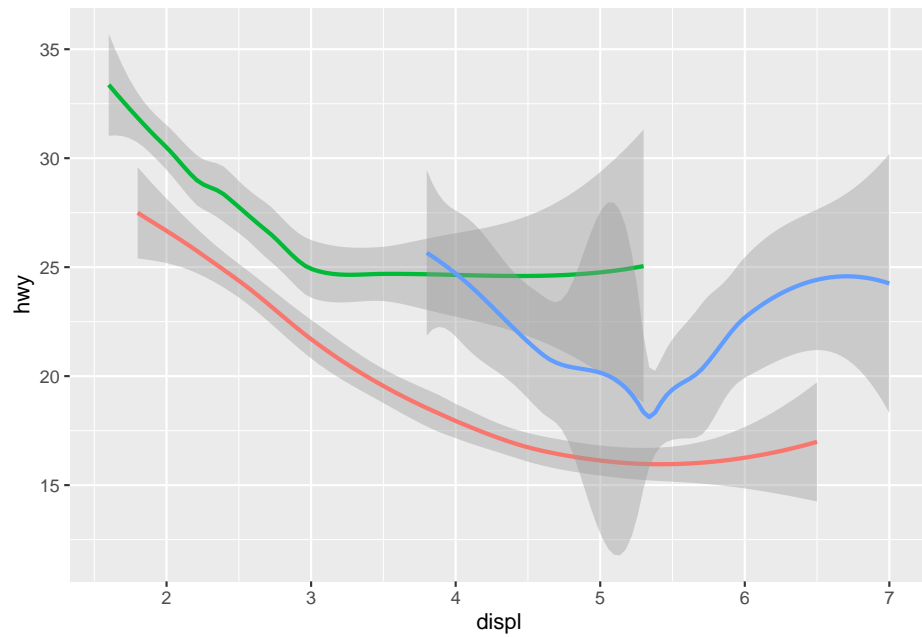## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

几何图形有 30 多种

```r
ggplot(data = mpg) +
  geom_smooth(mapping = aes(x = displ, y = hwy,group=drv))
```

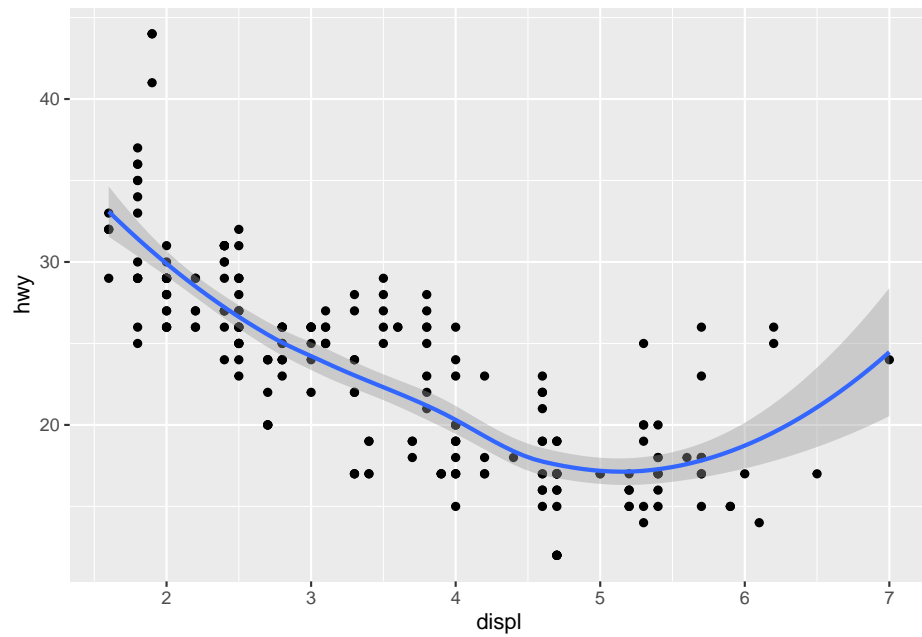## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

```r
ggplot(data = mpg) +
  geom_smooth(mapping = aes(x = displ, y = hwy,color=drv),
              show.legend=FALSE) #边注释是否存在
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```
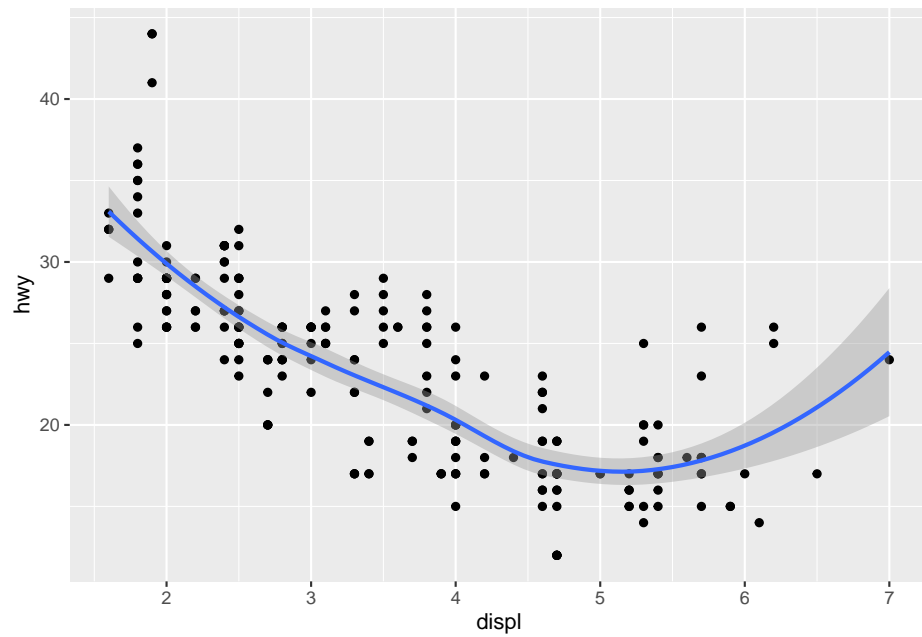
还可以多个图层叠加

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
    geom_smooth(mapping = aes(x = displ, y = hwy))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

但是这样有代码重复，下面是避免的方法

```r
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point() +
    geom_smooth()
```
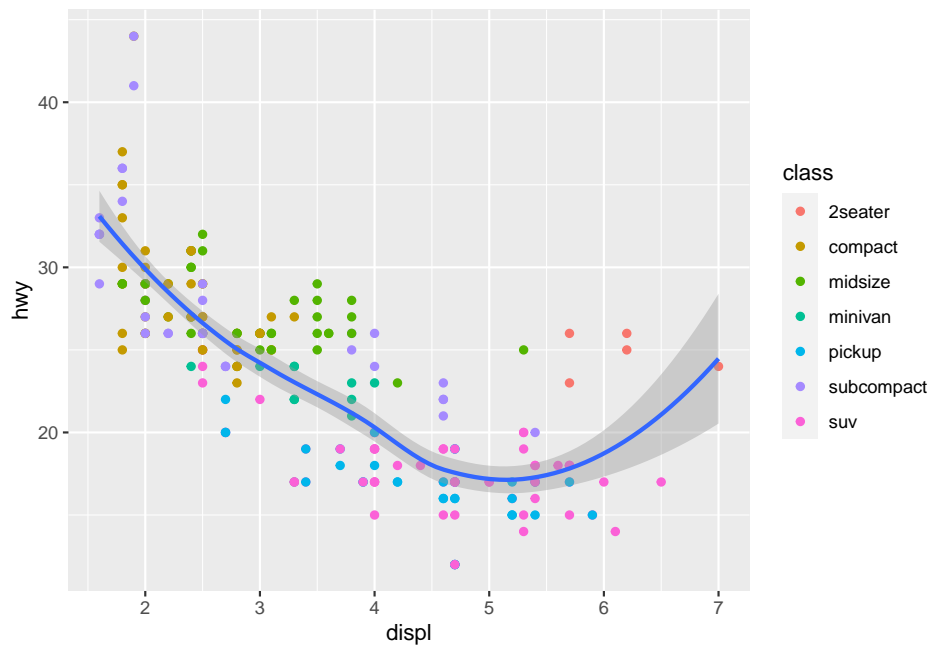
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

几何函数内的 mapping 只对局部有效

```r
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(color = class)) +
    geom_smooth()
```
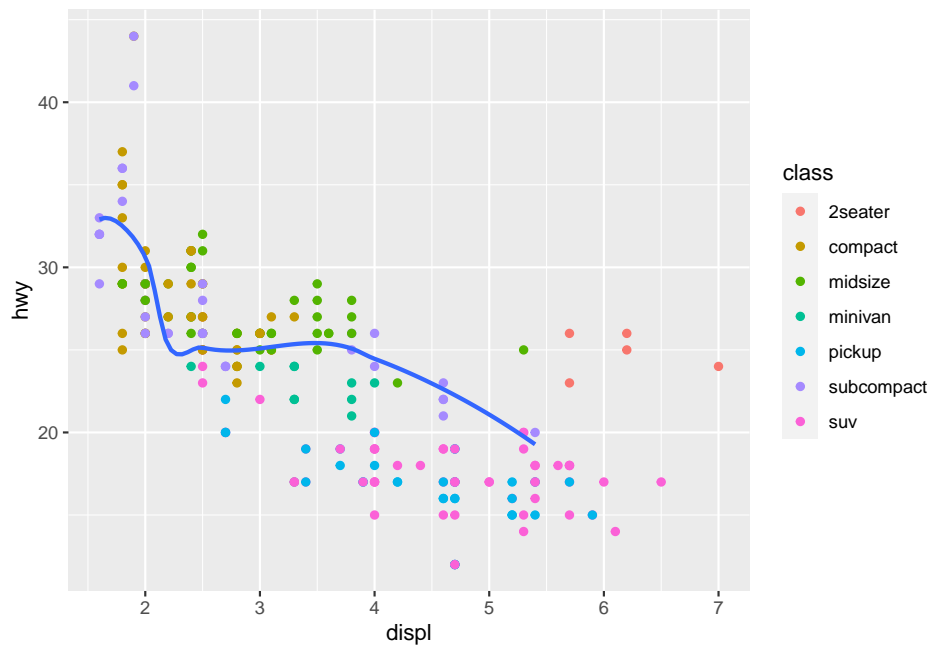
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

不同图层也可以用不同的 **data**

```r
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(color = class)) +
    geom_smooth(
      data=filter(mpg,class=="subcompact"),
      se=FALSE #se指line的阴影部分
    )
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

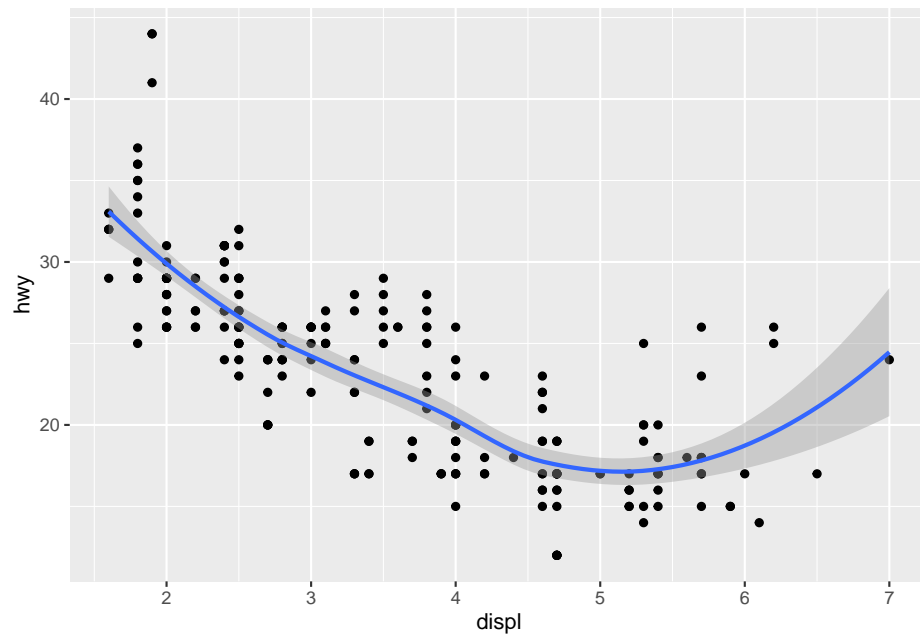geom_boxplot()：绘制箱线图

geom_histogram()：绘制直方图

geom_bar()：绘制条形图

geom_path()：在各个数据点之间连线（路径图）
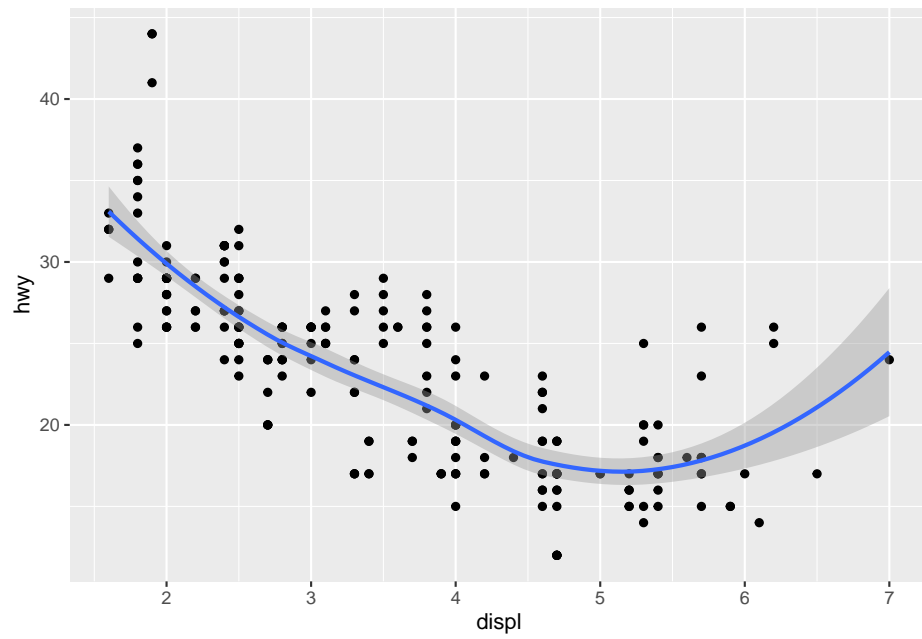
还有很多。

下面两个代码没有区别：

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point() +
    geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```
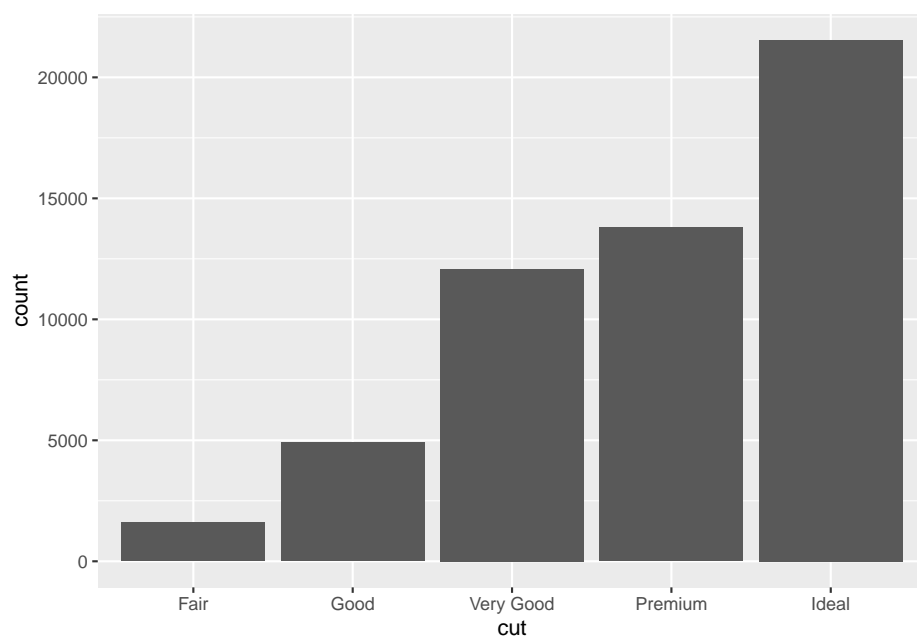
```
ggplot() +
  geom_point(
    data = mpg,
    mapping = aes(x = displ, y = hwy)
  )+
  geom_smooth(
    data = mpg,
    mapping = aes(x = displ, y = hwy)
  )
```

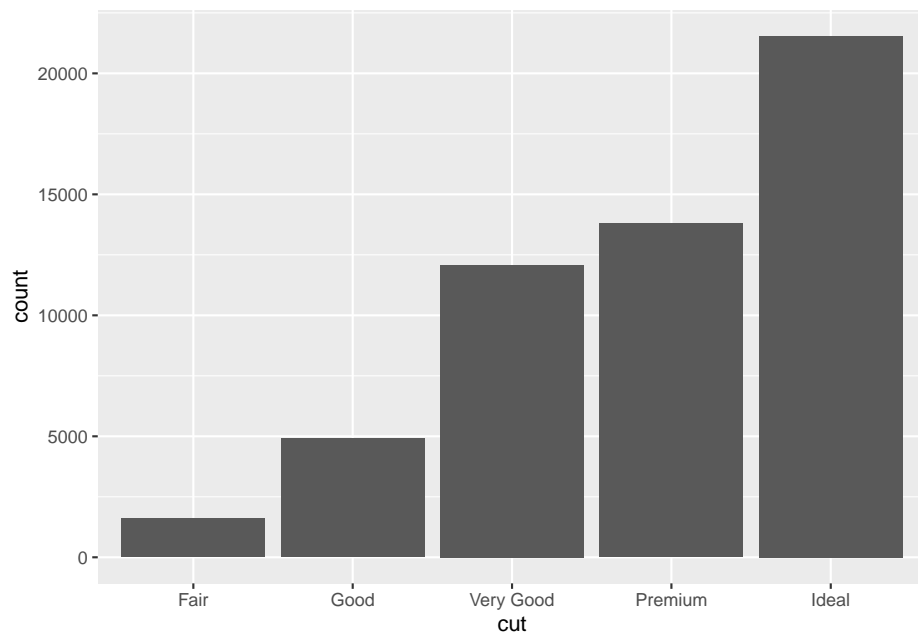## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

0.4 统计变换

```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut))
```
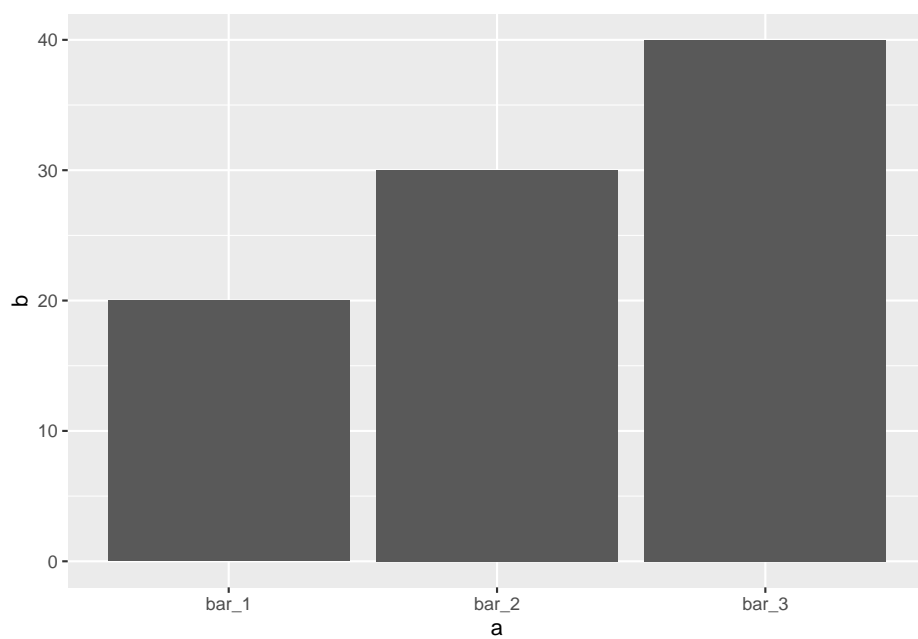
几何对象函数和统计变换函数可以互换使用，因为每个几何对象函数都有一个默认统计变换，每个统计变换都有一个默认几何对象函数。

```
ggplot(data = diamonds) +
  stat_count(mapping = aes(x = cut))
```
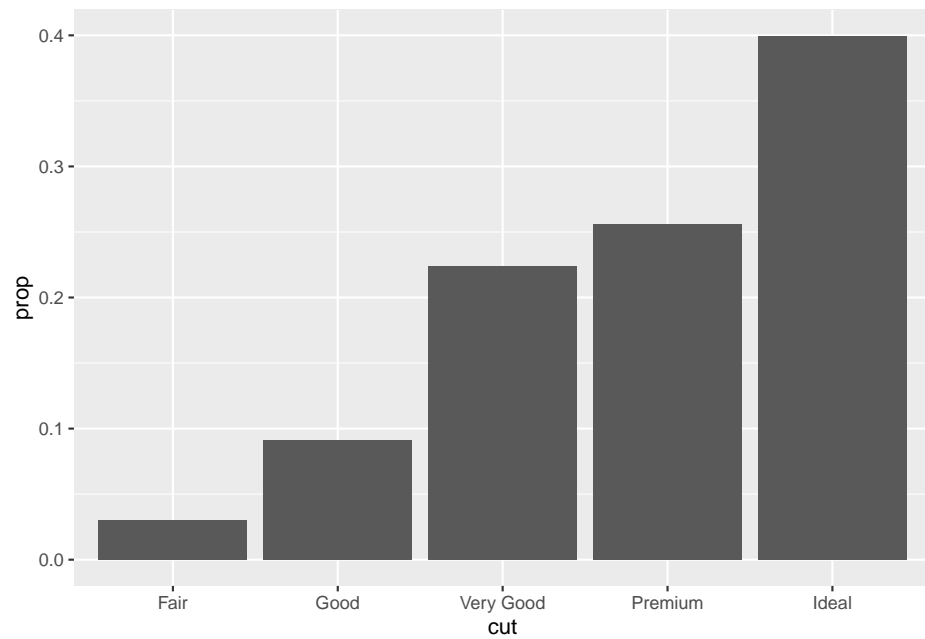
有时我们会想覆盖默认的统计变换：

```r
demo<-tribble(
  ~a,~b,
  "bar_1",20,
  "bar_2",30,
  "bar_3",40
)

ggplot(data=demo)+
  geom_bar(aes(x=a,y=b),stat="identity")
```
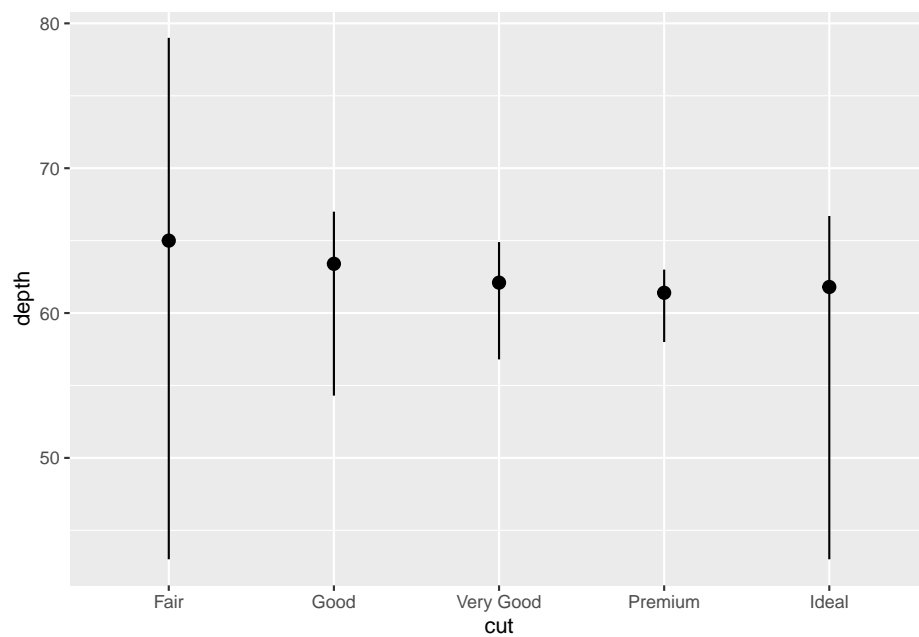
有时可能会想覆盖从统计变换生成的变量到图形的默认映射：

```
ggplot(data=diamonds)+
  geom_bar(
    mapping = aes(x=cut,y=..prop..,group=1)
  )
```
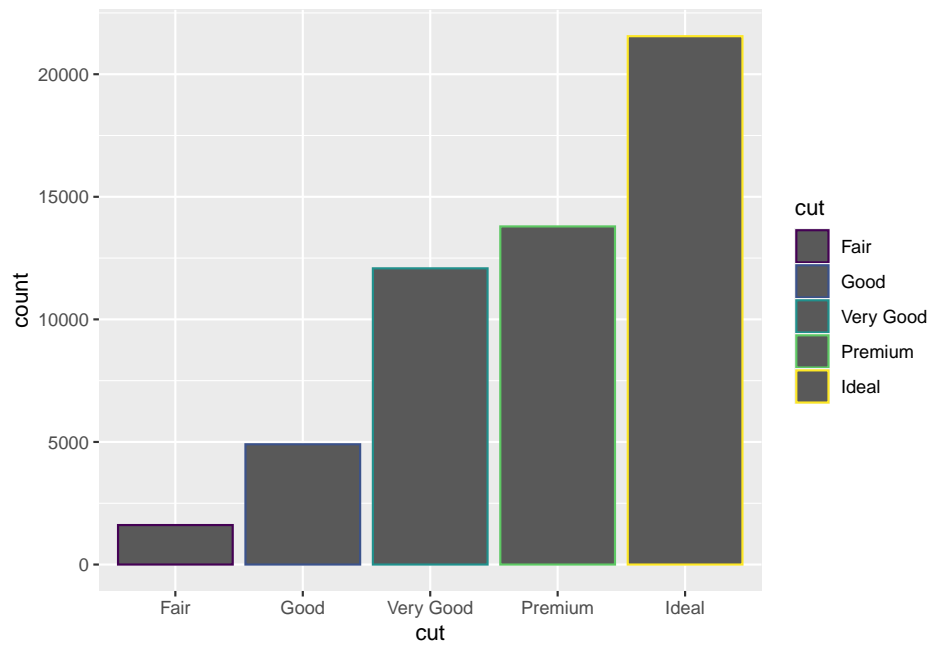
有时想在代码中强调统计变换:

```
ggplot(data = diamonds) +
  stat_summary(
    mapping = aes(x = cut, y = depth),
    fun.min = min,
    fun.max = max,
    fun = median
  )
```
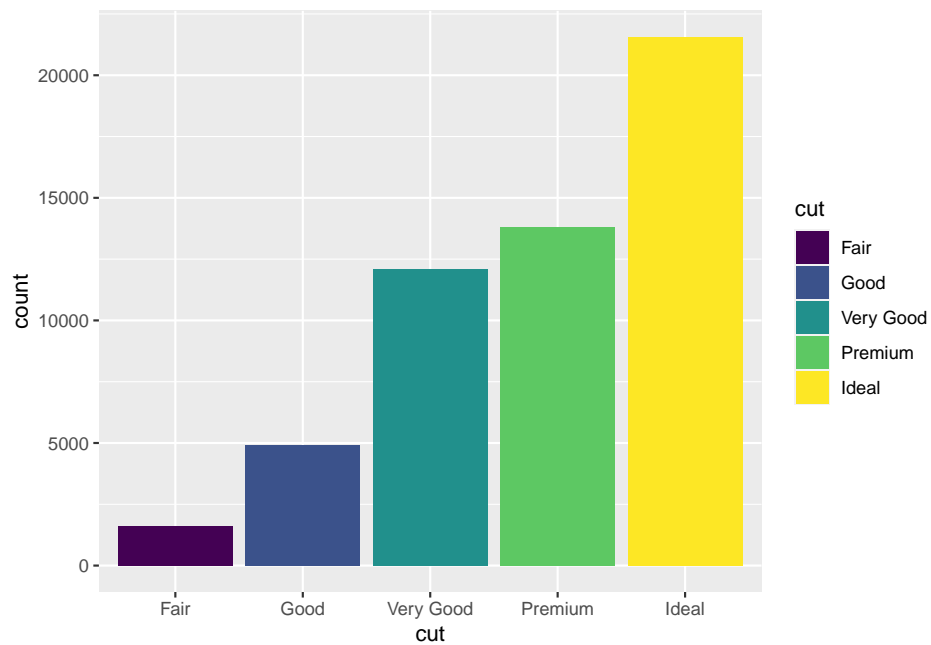
0.5 位置调整
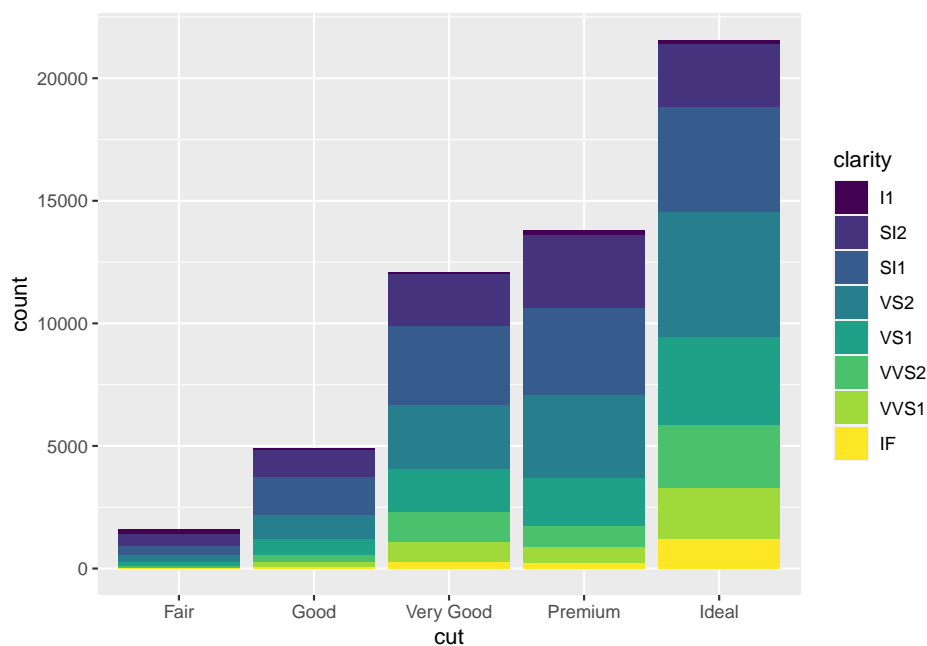
可以利用 color 或者 fill 为条形图上色

```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, color = cut))
```

```r
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, fill = cut))
```
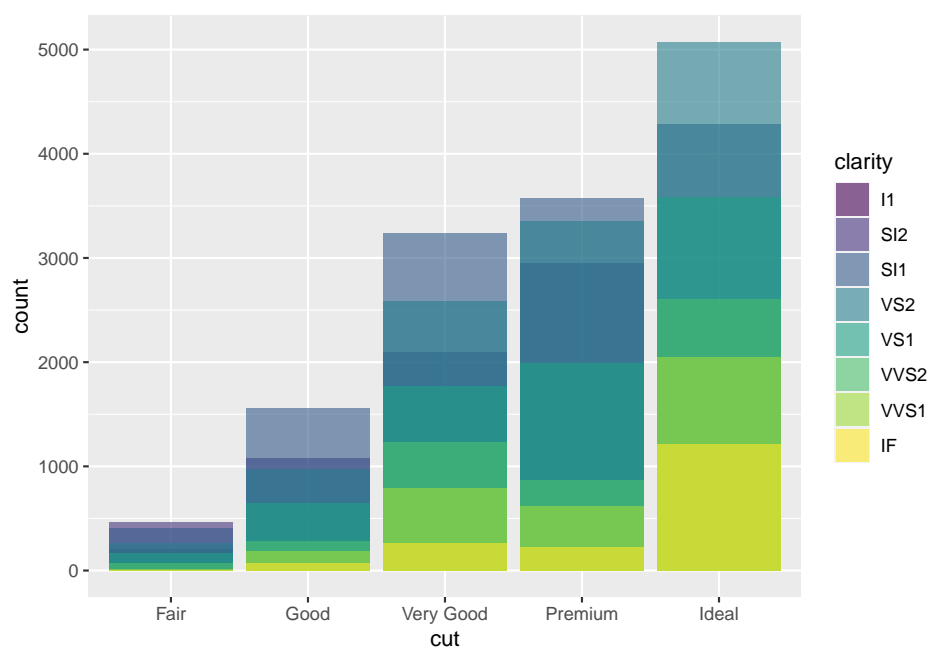
```
ggplot(data = diamonds) +
  geom_bar(mapping = aes(x = cut, fill = clarity))
```


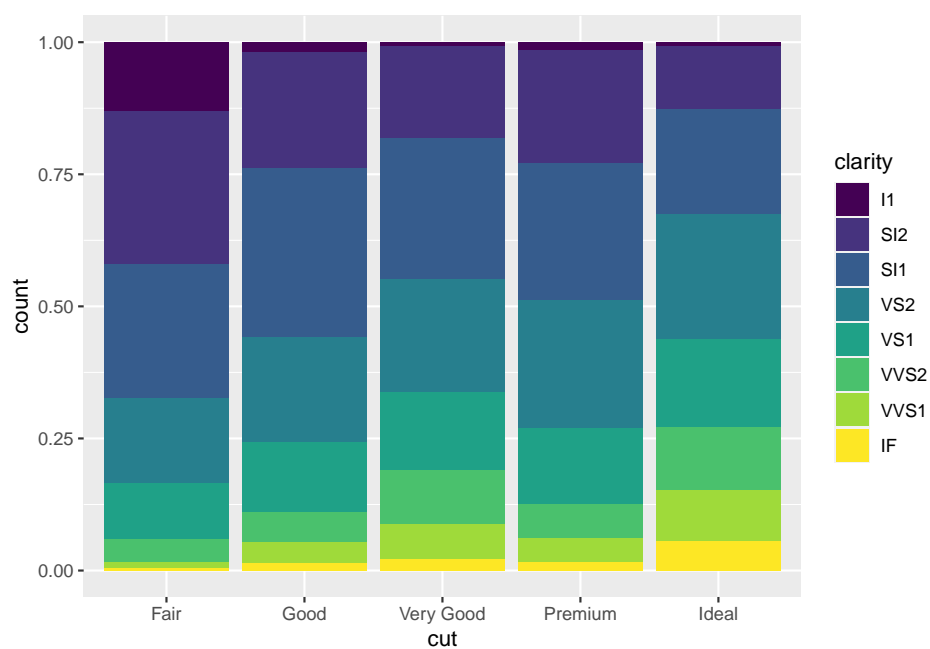这种堆叠是由 **position** 来进行位置调换的的，如果不想要这样的也可以选择 **identity，dodge** 或者 **fill**

position="identity" 会把每个对象的准确量表现在那，但是不大适用于条形图，因为会覆盖，我们可以通过设置 alpha 或者 fill=NA 来让变透明

```
ggplot(
    data = diamonds,
    mapping = aes(x = cut, fill = clarity)
)+
  geom_bar(alpha = 3/5, position = "identity")
```
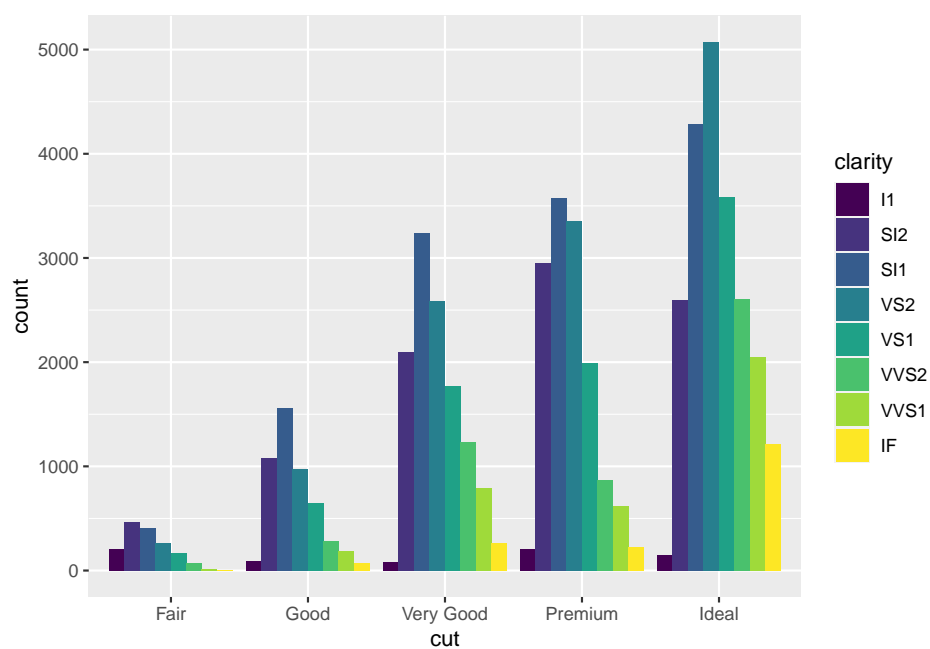
position ="fill" 可以轻松观察比例

```r
ggplot(data = diamonds) + geom_bar(
        mapping = aes(x = cut, fill = clarity),
        position = "fill"
    )
```
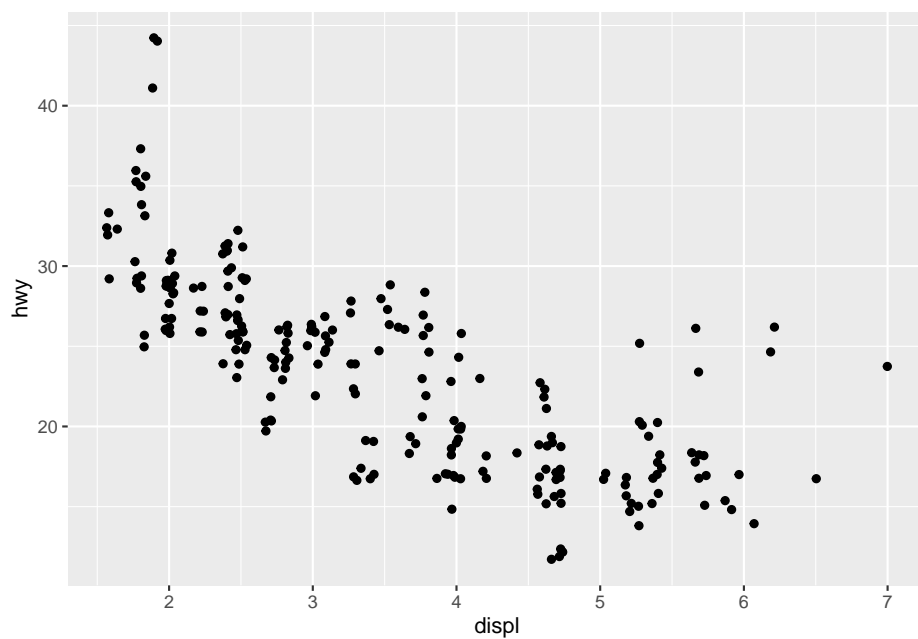
position="dodge" 变成多个条形

```r
ggplot(data = diamonds) + geom_bar(
        mapping = aes(x = cut, fill = clarity),
        position = "dodge"
    )
```

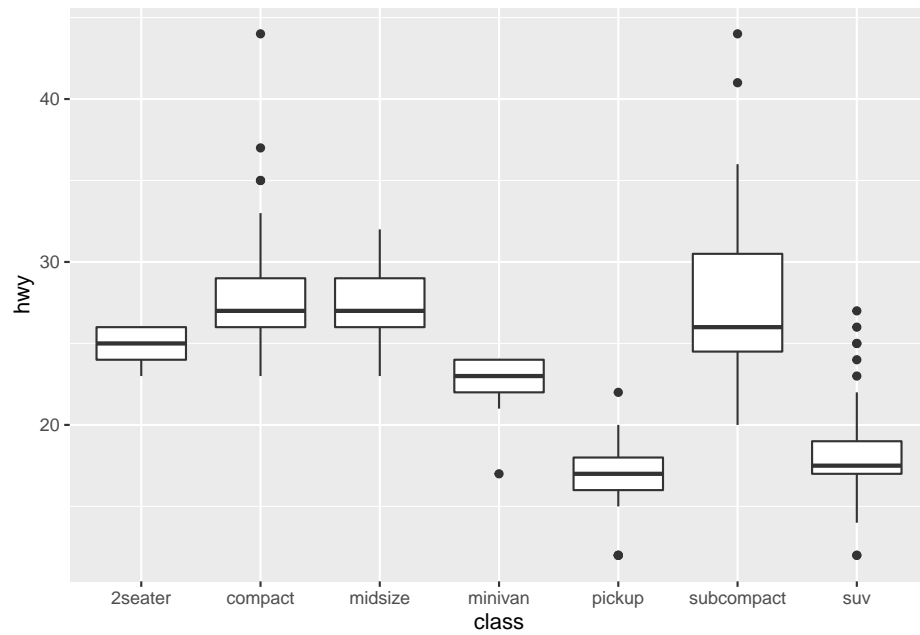另外也有一种不适合条形图但适合散点图的调整方法，散点图中经常会有点重合，我们可以加上一个小扰动：

```
ggplot(data = mpg) + geom_point(
    mapping = aes(x = displ, y = hwy),
    position = "jitter"
)
```
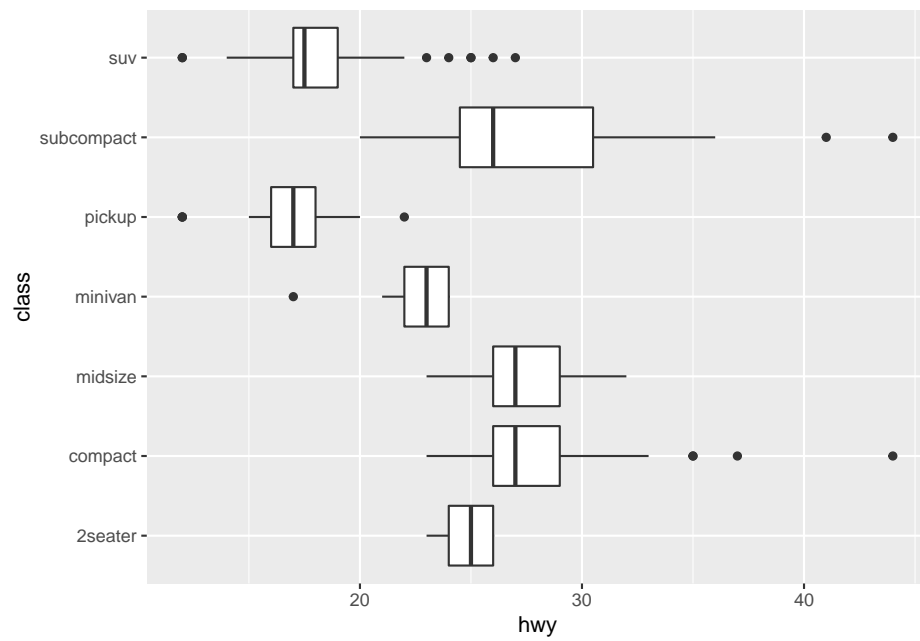
0.6 坐标系

coord_flip() 用来交换 x 轴和 y 轴，在画水平箱图以及长标签很有用。

```
ggplot(data = mpg, mapping = aes(x = class, y = hwy)) +
  geom_boxplot()
```
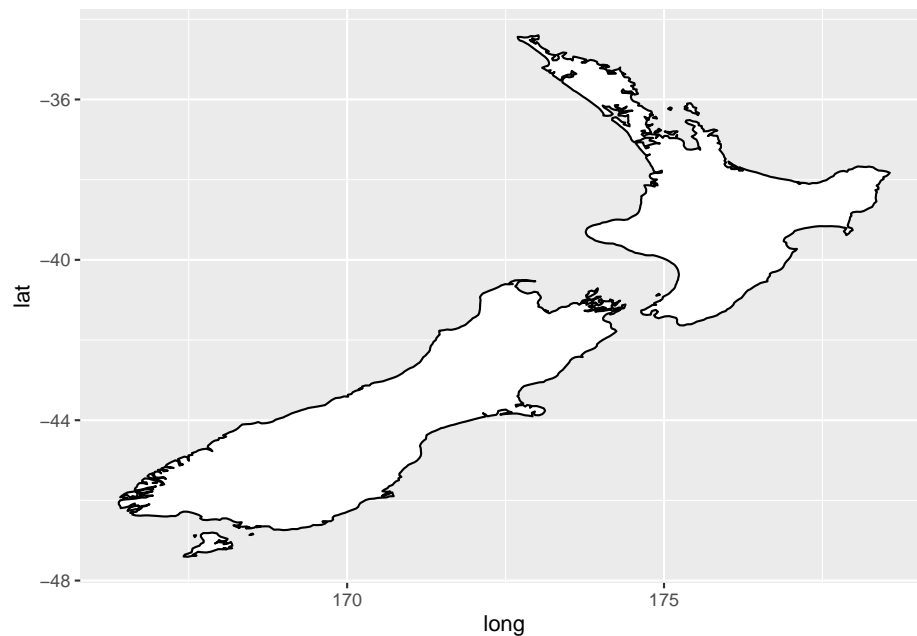
```r
ggplot(data = mpg, mapping = aes(x = class, y = hwy)) +
  geom_boxplot() +
  coord_flip()
```
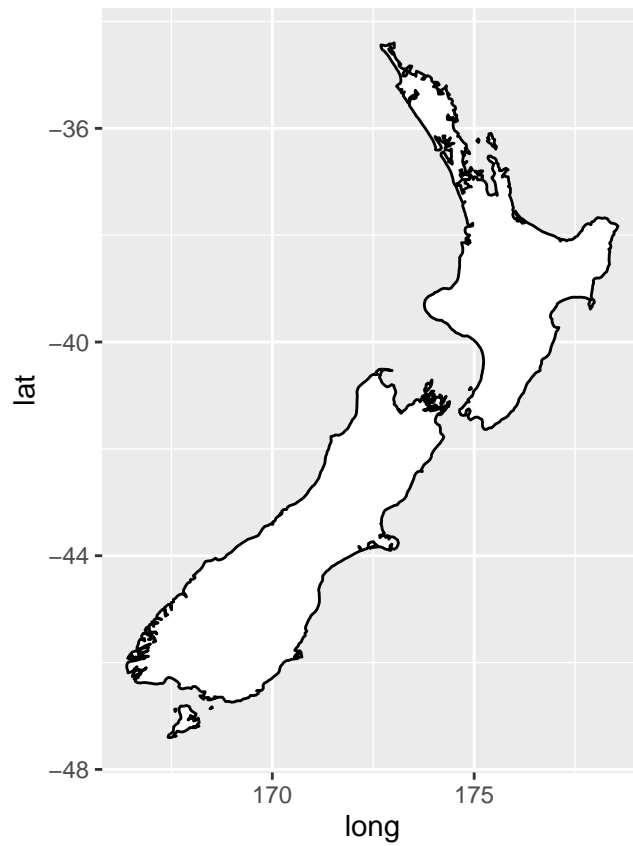
coord_quickmap() 为地图设置合适的比例:

```
nz <- map_data("nz")
ggplot(nz, aes(long, lat, group = group)) +
  geom_polygon(fill = "white", color = "black")
```
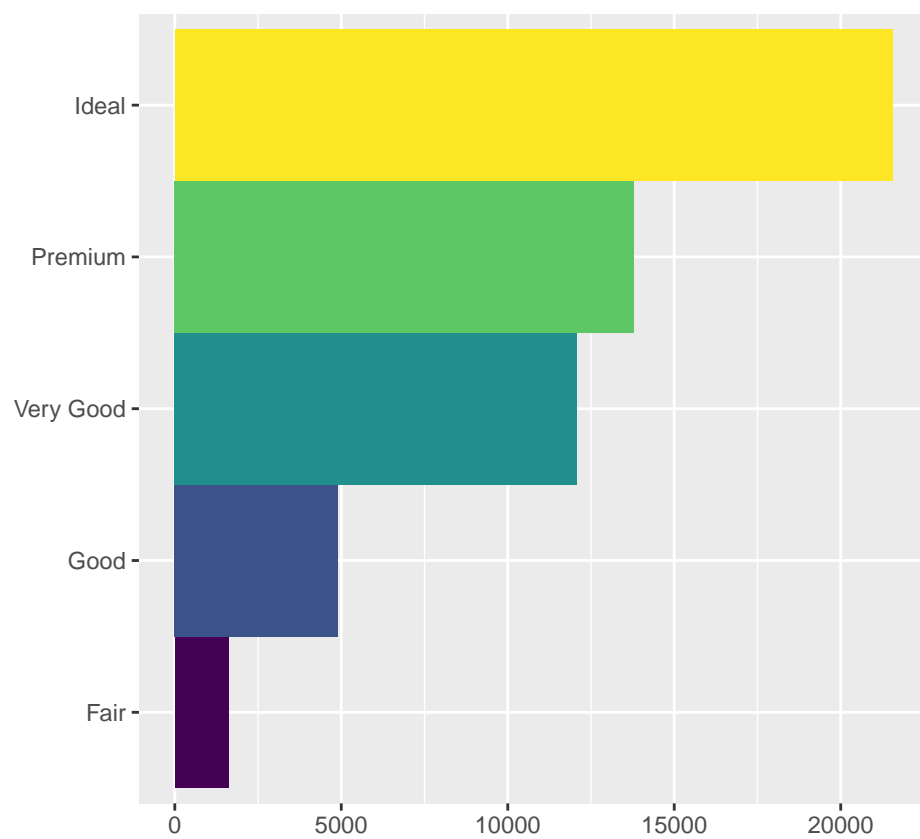


```
ggplot(nz, aes(long, lat, group = group)) +
  geom_polygon(fill = "white", color = "black") +
  coord_quickmap()
```
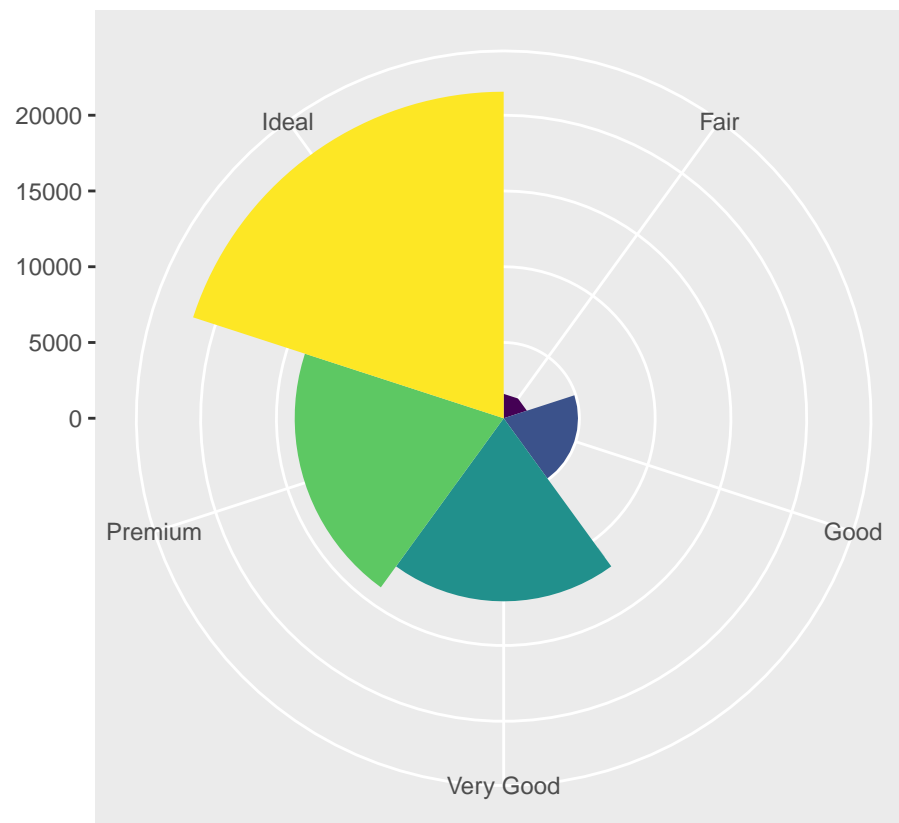
coord_polar() 用来得到极坐标，可以揭示出条形图和鸡冠图的关系：

```r
bar <- ggplot(data = diamonds) +
  geom_bar(
   mapping = aes(x = cut, fill = cut),
   show.legend = FALSE,
   width = 1
  )+
  theme(aspect.ratio = 1) +
  labs(x = NULL, y = NULL)
  bar + coord_flip()
```

```
bar + coord_polar()
```

0.7 图层分层语法

ggplot(data=)+

(

```
mapping =aes(<MAPPINGS),

stat=<STAT>,

position=<POSITION>
```

)+

+