Contents

利用 ggplot2 绘图

为什么使用 ggplot2

ggplot2 基本要素

- 数据 (Data) 和映射 (Mapping)
- 几何对象 (Geometric)
- 标尺 (Scale)
- 统计变换 (Statistics)
- 坐标系统 (Coordinante)
- 图层 (Layer)
- 分面 (Facet)
- 主题 (Theme)

这里将从这些基本要素对 ggplot2 进行介绍。

数据 (Data) 和映射 (Mapping)

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
data(diamonds)
set.seed(42)
small <- diamonds[sample(nrow(diamonds), 1000), ]
head(small)
```

```
## # A tibble: 6 × 10
##   carat       cut color clarity depth table price     x     y     z
##   <dbl>     <ord> <ord>   <ord> <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.71 Very Good     H     SI1  62.5    60  2096  5.68  5.75  3.57
## 2  0.79   Premium     H     SI1  61.8    59  2275  5.97  5.91  3.67
## 3  1.03     Ideal     F     SI1  62.4    57  6178  6.48  6.44  4.03
## 4  0.50     Ideal     E     VS2  62.2    54  1624  5.08  5.11  3.17
## 5  0.27     Ideal     E     VS1  61.6    56   470  4.14  4.17  2.56
## 6  0.30   Premium     E     VS2  61.7    58   658  4.32  4.34  2.67
```

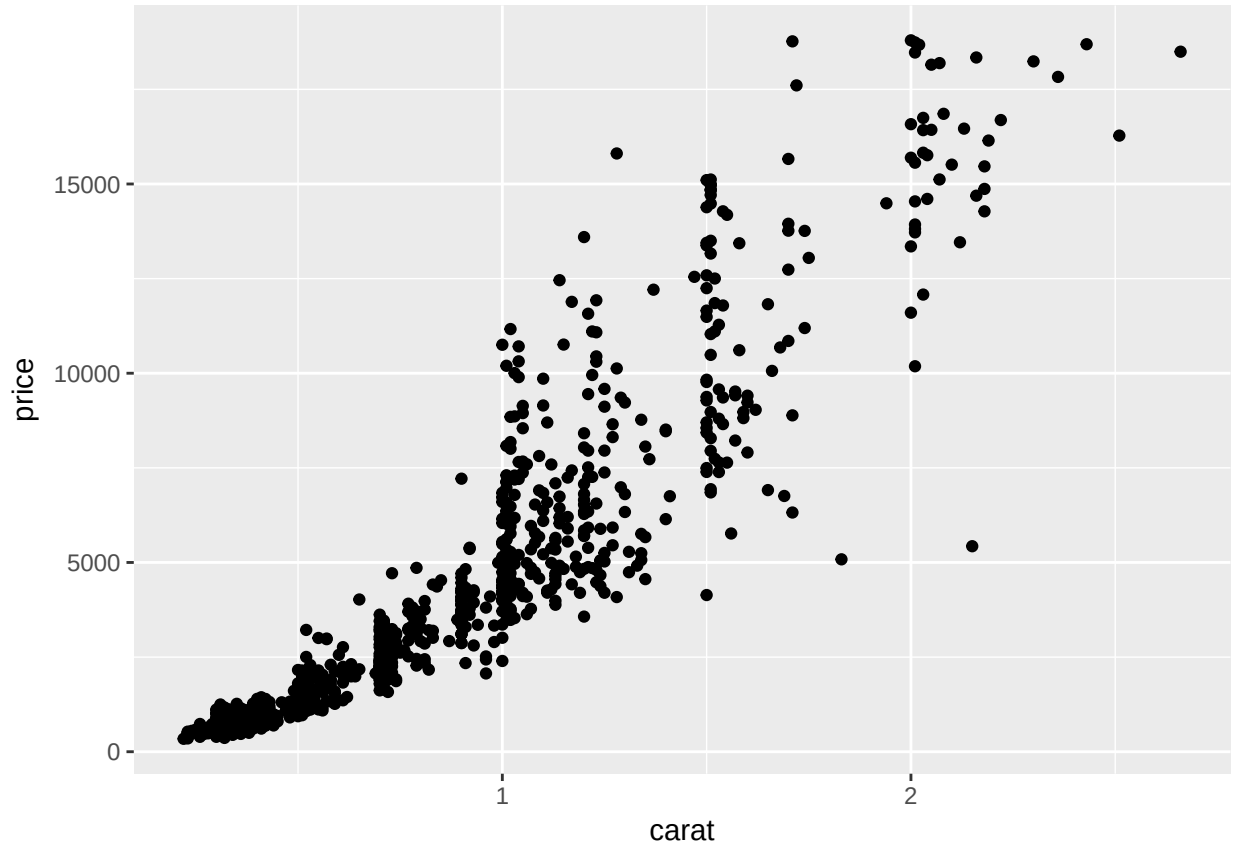
```
summary(small)
```

```
##      carat               cut        color        clarity        depth      
##  Min.   :0.2200   Fair      : 28   D:121   SI1     :258   Min.   :55.20  
##  1st Qu.:0.4000   Good      : 88   E:186   VS2     :231   1st Qu.:61.00  
##  Median :0.7100   Very Good:227   F:164   SI2     :175   Median :61.80  
```

```
##  Mean   :0.8187   Premium  :257   G:216   VS1    :141   Mean   :61.71
##  3rd Qu.:1.0700   Ideal    :400   H:154   VVS2   : 91   3rd Qu.:62.50
##  Max.   :2.6600                   I:106   VVS1   : 67   Max.   :72.20
##                                   J: 53   (Other): 37
##      table           price             x                y
##  Min.   :50.10   Min.   :  342.0   Min.   :3.850   Min.   :3.840
##  1st Qu.:56.00   1st Qu.:  989.5   1st Qu.:4.740   1st Qu.:4.758
##  Median :57.00   Median : 2595.0   Median :5.750   Median :5.775
##  Mean   :57.43   Mean   : 4110.5   Mean   :5.787   Mean   :5.791
##  3rd Qu.:59.00   3rd Qu.: 5495.2   3rd Qu.:6.600   3rd Qu.:6.610
##  Max.   :65.00   Max.   :18795.0   Max.   :8.830   Max.   :8.870
##
##        z
##  Min.   :2.330
##  1st Qu.:2.920
##  Median :3.550
##  Mean   :3.572
##  3rd Qu.:4.070
##  Max.   :5.580
##
```

画图实际上是把数据中的变量映射到图形属性上。以克拉 (carat) 数为 X 轴变量，价格 (price) 为 Y 轴变量。

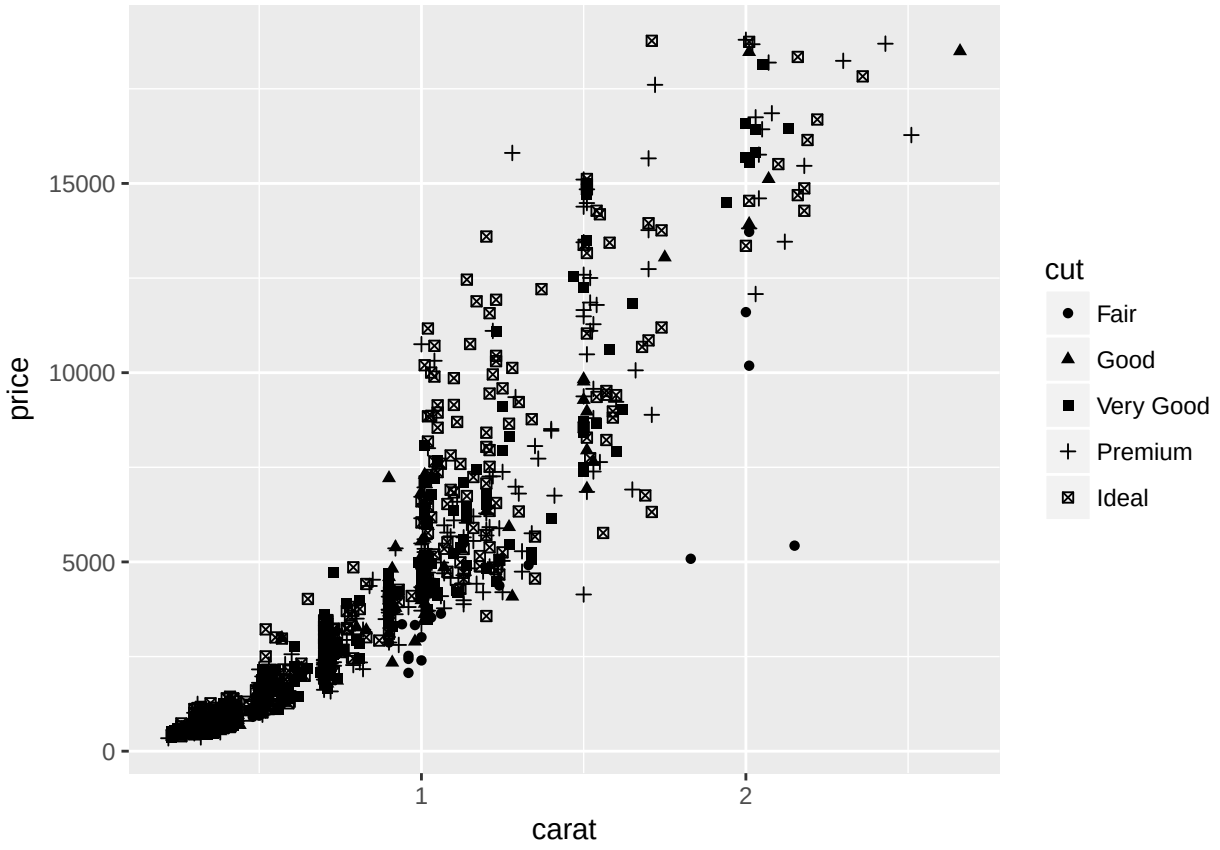
```
p <- ggplot(data = small, mapping = aes(x = carat, y = price))
p + geom_point()
```



上面这行代码把数据映射 XY 坐标轴上，需要告诉 ggplot2，这些数据要映射成什么样的几何对象，这里以散点为例：

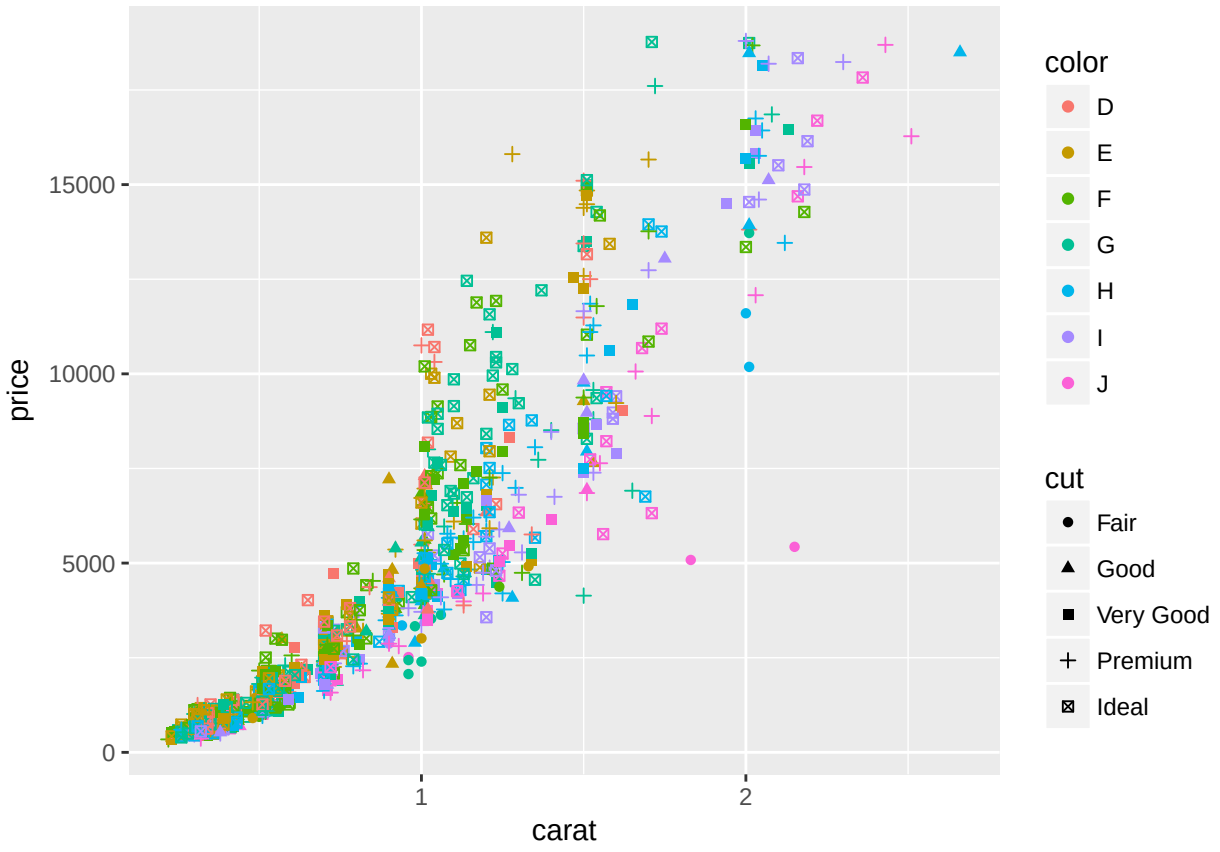
如果想将切工（cut）映射到形状属性。只需要：

```
p <- ggplot(data=small, mapping=aes(x=carat, y=price, shape=cut))
p+geom_point()
```



再比如我想将钻石的颜色（color）映射颜色属性：

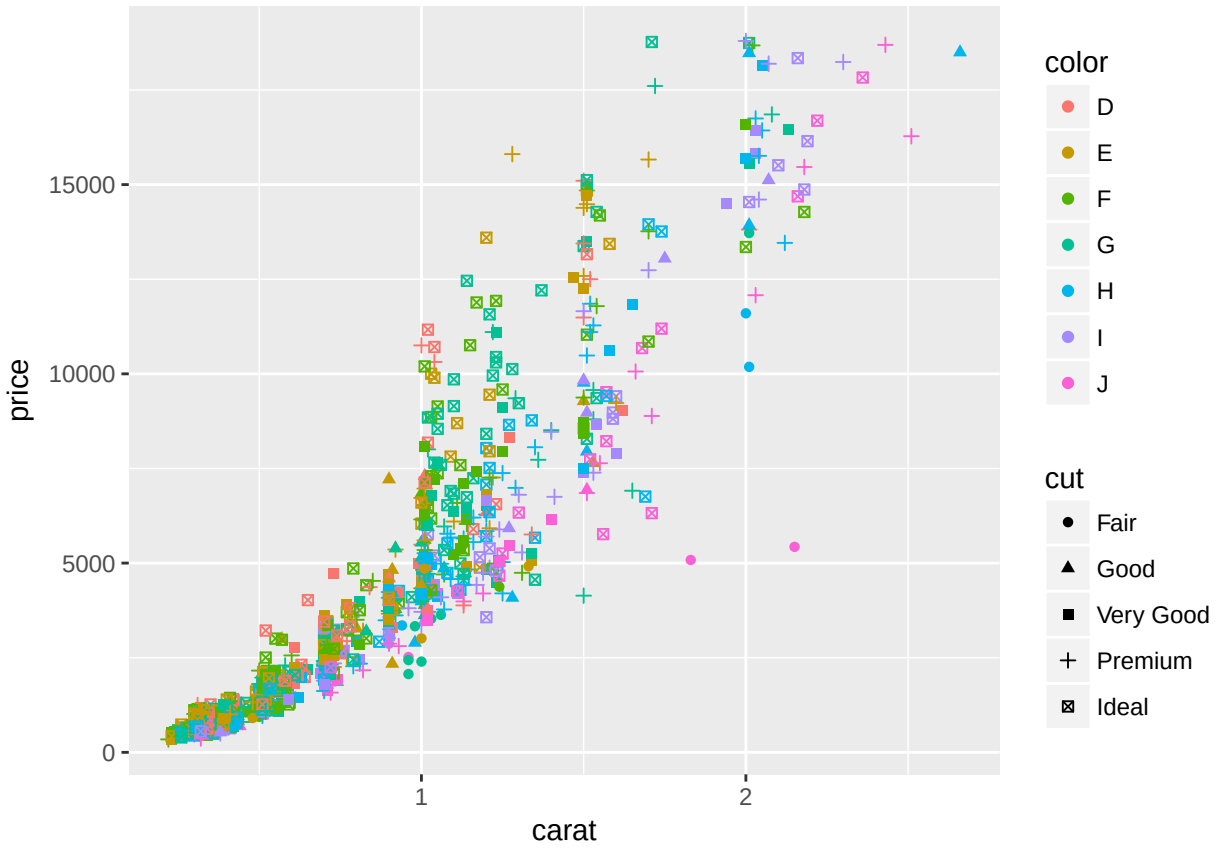
```
p <- ggplot(data=small, mapping=aes(x=carat, y=price, shape=cut, colour=color))
p+geom_point()
```



几何对象（Geometric）

在上面的例子中，各种属性映射由 ggplot 函数执行，只需要加一个图层，使用 geom_point() 告诉 ggplot 要画散点，于是所有的属性都映射到散点上。geom_point() 完成的就是几何对象的映射，ggplot2 提供了各种几何对象映射，如 geom_histogram 用于直方图，geom_bar 用于画柱状图，geom_boxplot 用于画箱式图等等。不同的几何对象，要求的属性会有些不同，这些属性也可以在几何对象映射时提供，比如上一图，也可以用以下语法来画：

```
p <- ggplot(small)
p+geom_point(aes(x=carat, y=price, shape=cut, colour=color))
```

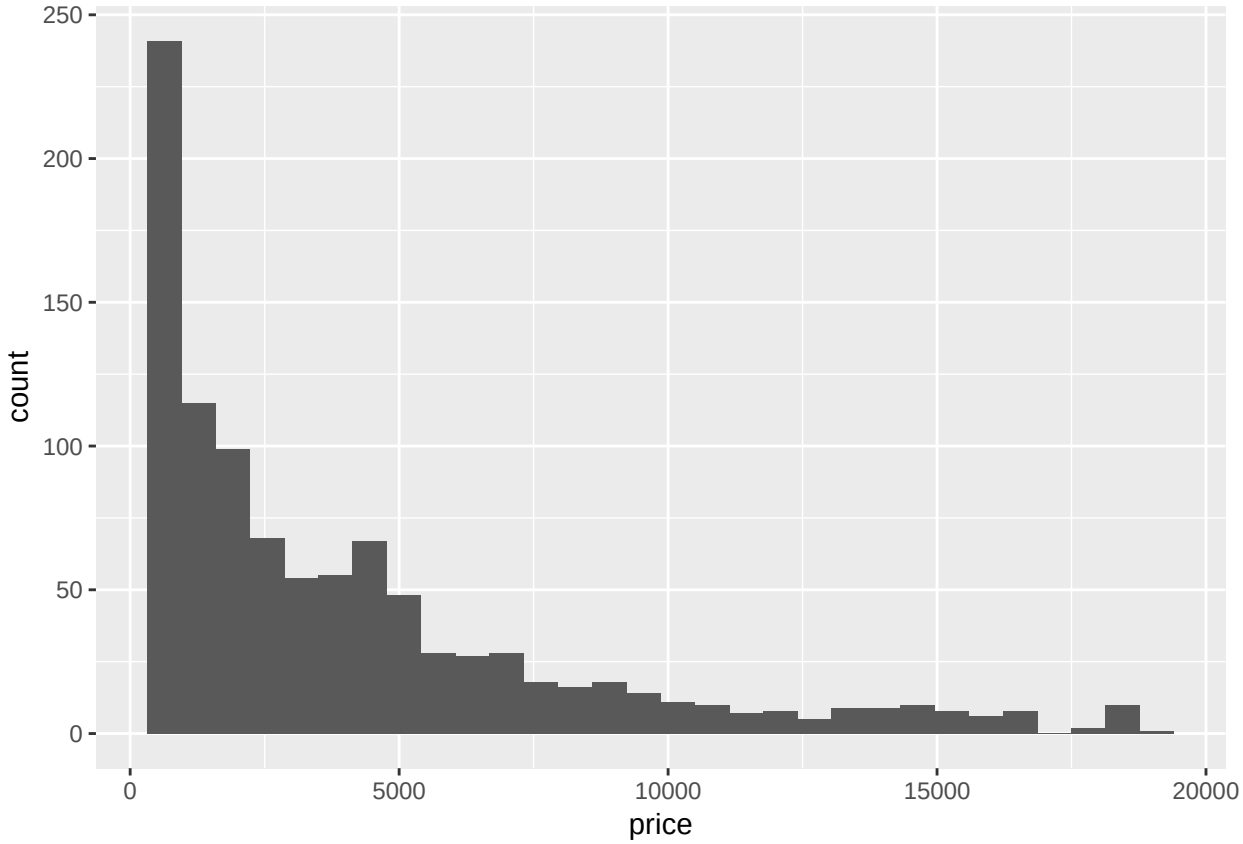


直方图

直方图最容易，提供一个 x 变量，画出数据的分布。

```
ggplot(small)+geom_histogram(aes(x=price))
```

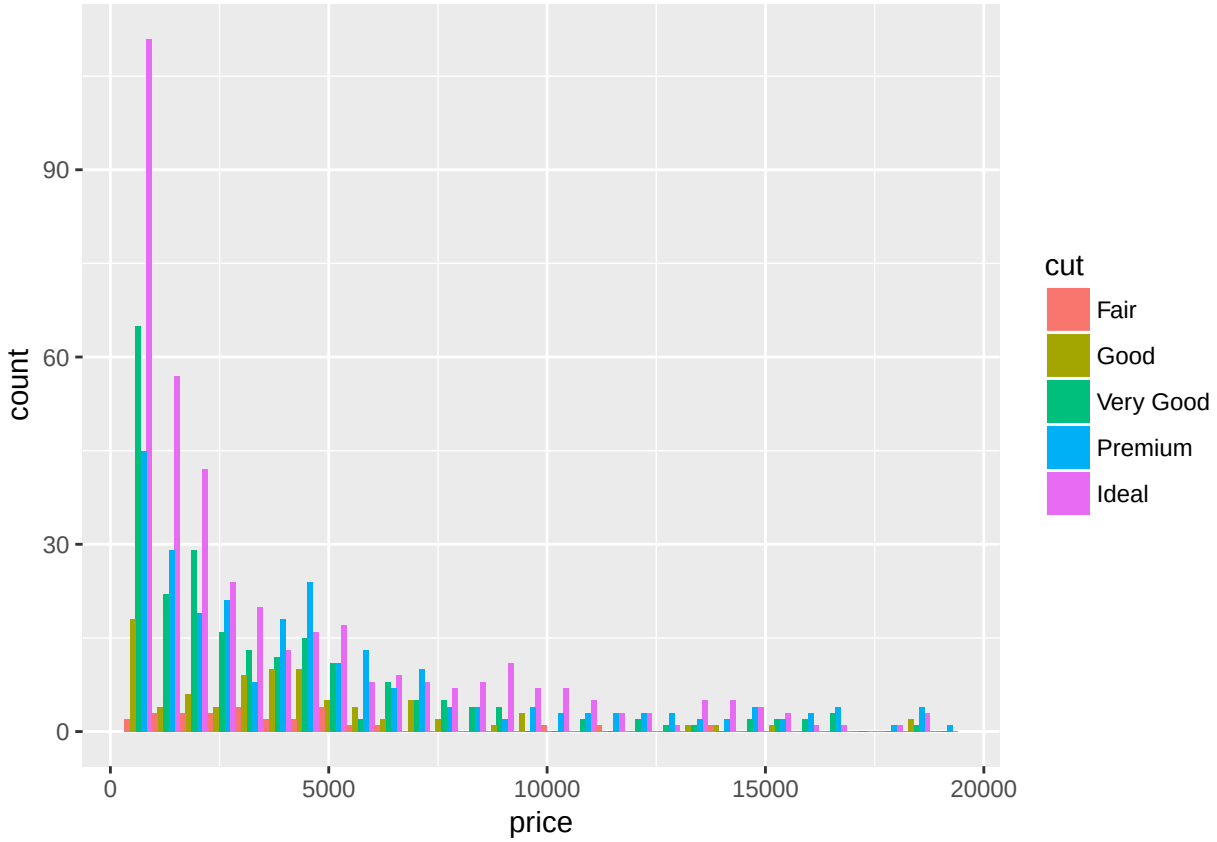
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



同样可以根据另外的变量给它填充颜色，比如按不同的切工：

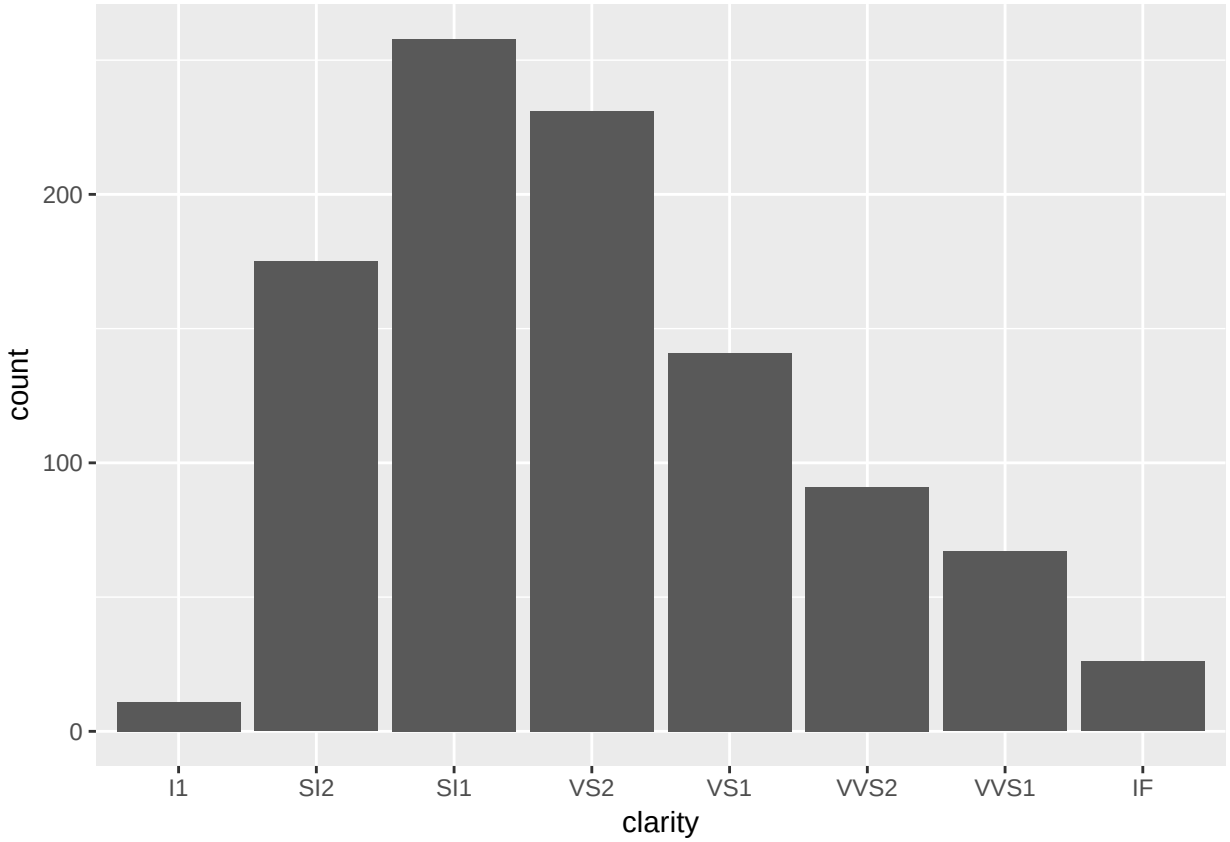
```
ggplot(small)+geom_histogram(aes(x=price, fill=cut), position="dodge")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



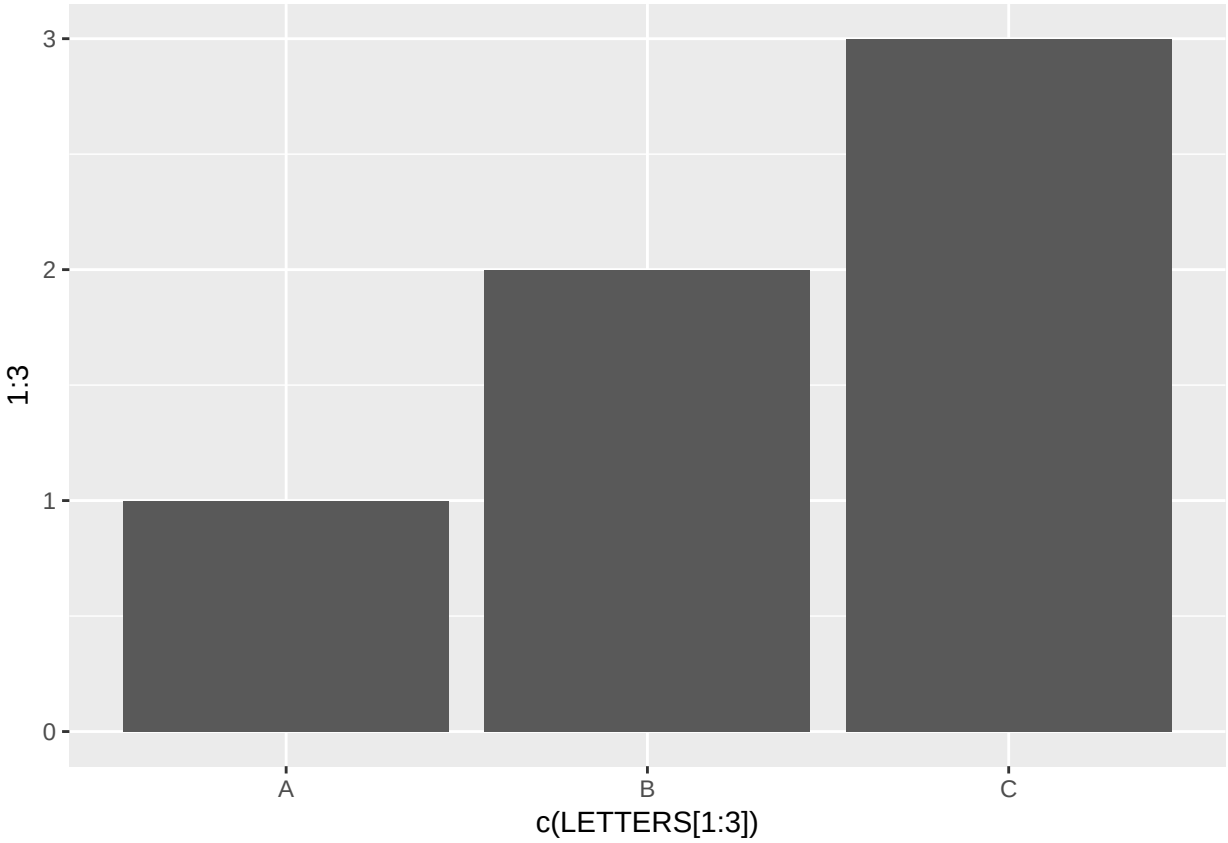
柱状图柱状图非常适合于画分类变量。在这里以透明度（clarity）变量为例。按照不同透明度的钻石的数目画柱状图。

```
ggplot(small)+geom_bar(aes(x=clarity))
```



柱状图两个要素，一个是分类变量，一个是数目，也就是柱子的高度。数目在这里不用提供，因为 ggplot2 会通过 x 变量计算各个分类的数目。当然你想提供也是可以的，通过 stat 参数，可以让 geom_bar 按指定高度画图，比如以下代码：

```
ggplot()+geom_bar(aes(x=c(LETTERS[1:3]),y=1:3), stat="identity")
```

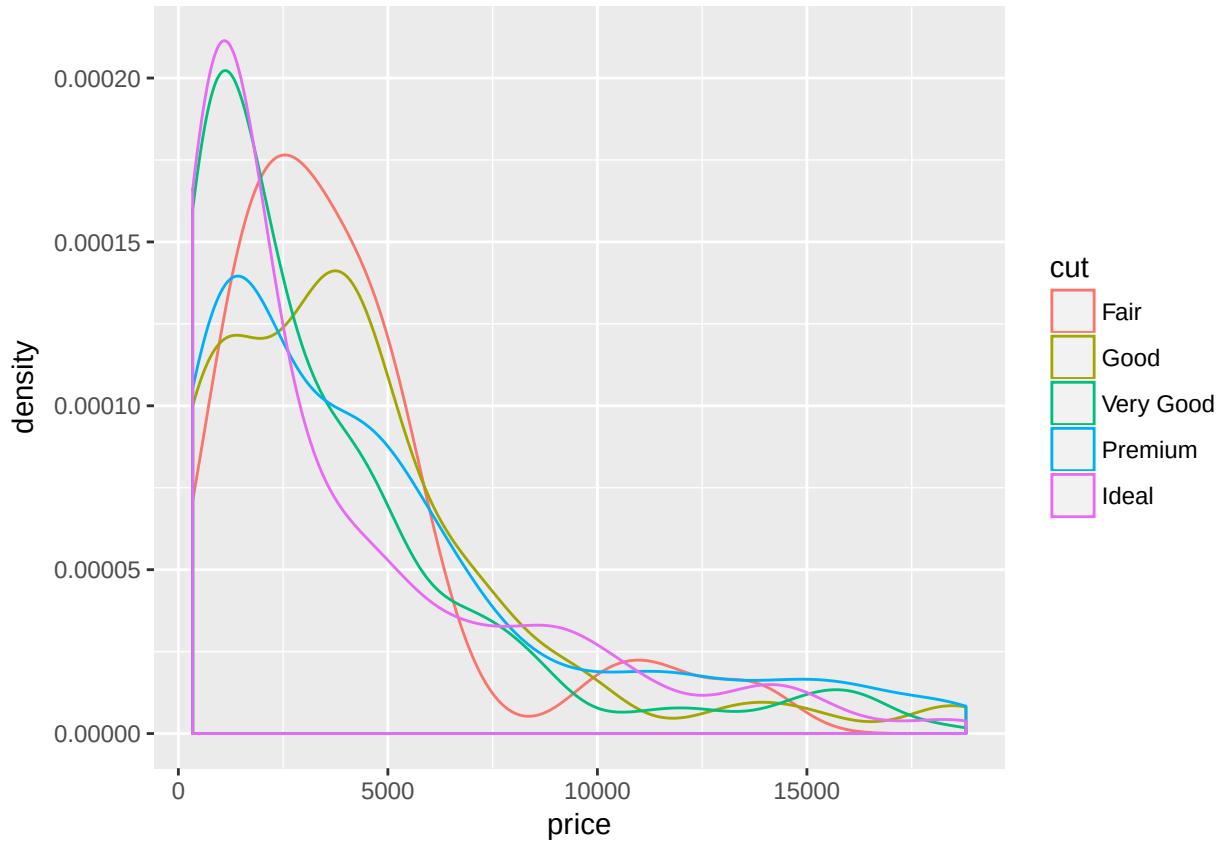


柱状图和直方图是很像的，直方图把连续型的数据按照一个个等长的分区（bin）来切分，然后计数，画柱状图。而柱状图是分类数据，按类别计数。我们可以用前面直方图的参数来画 side-by-side 的柱状图，填充颜色或者按比例画图，它们是高度一致的。柱状图是用来表示计数数据的，但在生物界却被经常拿来表示均值，加上误差来表示数据分布，这可以通常图层来实现。

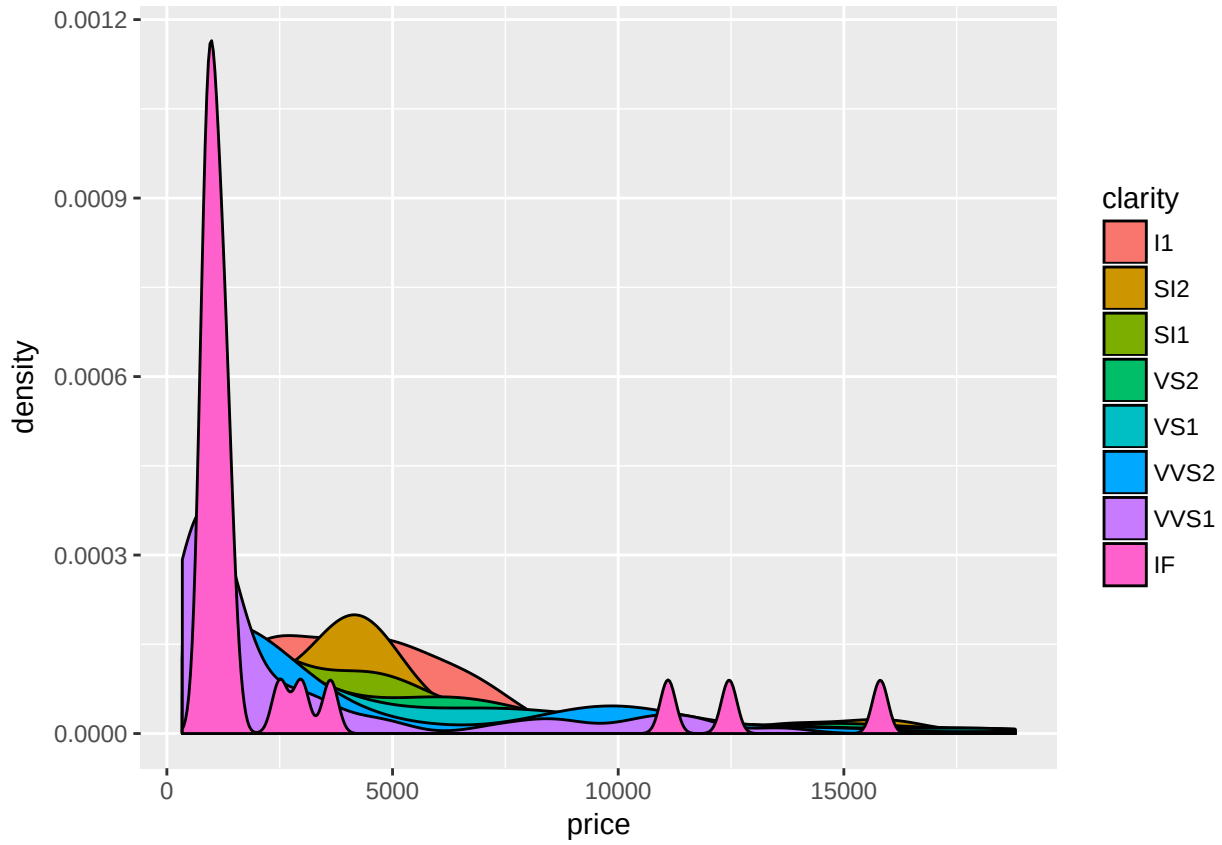
密度函数图

说到直方图，就不得不说密度函数图，数据和映射和直方图是一样的，唯一不同的是几何对象，geom_histogram 告诉 ggplot 要画直方图，而 geom_density 则说我们要画密度函数图，在我们熟悉前面语法的情况下，很容易画出：

```
ggplot(small)+geom_density(aes(x=price, colour=cut))
```




```
ggplot(small)+geom_density(aes(x=price,fill=clarity))
```

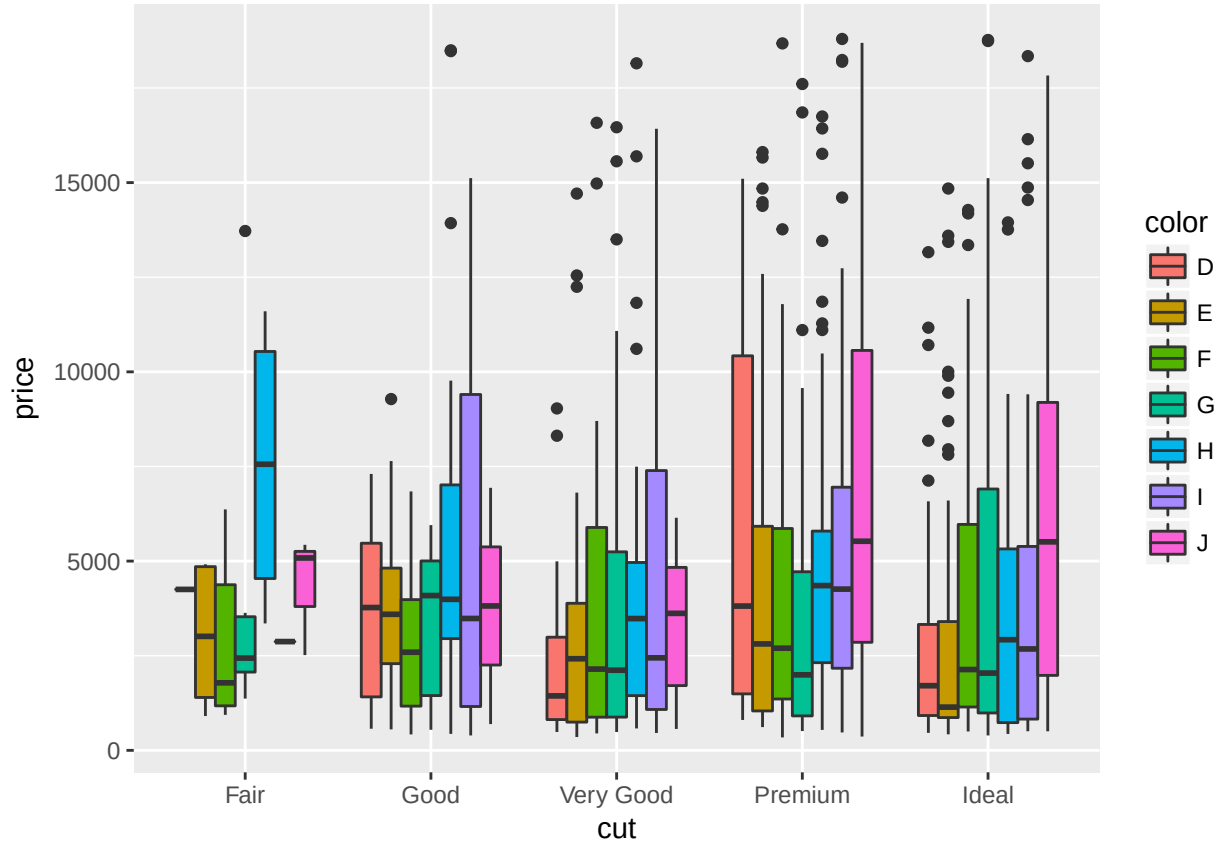


colour 参数指定的是曲线的颜色，而 fill 是往曲线下面填充颜色。

箱式图

数据量比较大的时候，用直方图和密度函数图是表示数据分布的好方法，而在数据量较少的时候，比如很多的生物实验，很多时候大家都是使用柱状图 +errorbar 的形式来表示，不过这种方法的信息量非常低，这种情况推荐使用 boxplot。

```
ggplot(small)+geom_boxplot(aes(x=cut, y=price,fill=color))
```

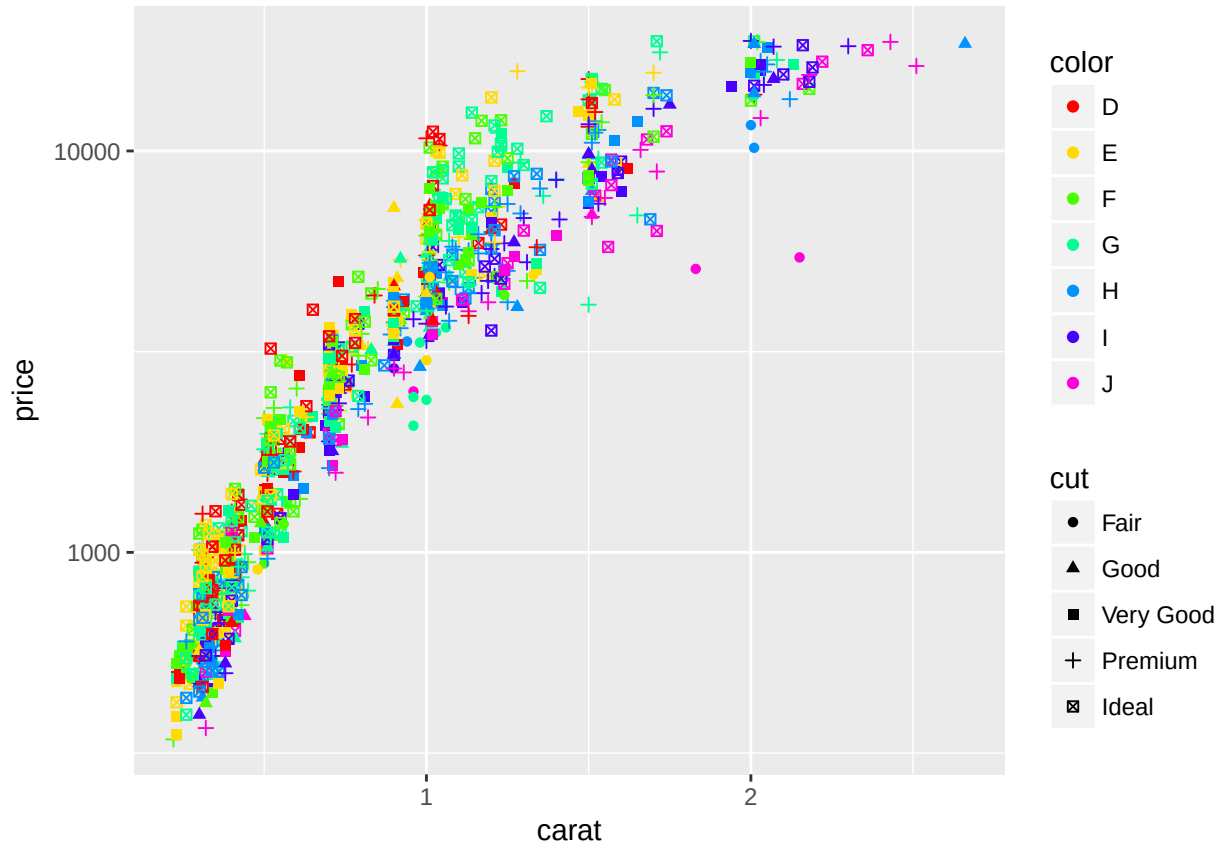


geom_boxplot 将数据映射到箱式图上，上面的代码，我们应该很熟悉了，按切工 (cut) 分类，对价格 (price) 变量画箱式图，再分开按照 color 变量填充颜色。ggplot2 提供了很多的 geom_xxx 函数，可以满足我们对各种图形绘制的需求。geom_abline geom_area
geom_bar geom_bin2d geom_blank geom_boxplot
geom_contour geom_crossbar geom_density geom_density2d
geom_dotplot geom_errorbar geom_errorbarh geom_freqpoly
geom_hex geom_histogram geom_hline geom_jitter
geom_line geom_linerange geom_map geom_path
geom_point geom_pointrange geom_polygon geom_quantile
geom_raster geom_rect geom_ribbon geom_rug
geom_segment geom_smooth geom_step geom_text
geom_tile geom_violin geom_vline

标尺 (Scale)

前面我们已经看到了，画图就是在做映射，不管是映射到不同的几何对象上，还是映射各种图形属性。这一小节介绍标尺，在对图形属性进行映射之后，使用标尺可以控制这些属性的显示方式，比如坐标刻度，可能通过标尺，将坐标进行对数变换；比如颜色属性，也可以通过标尺，进行改变。

```
ggplot(small)+geom_point(aes(x=carat, y=price, shape=cut, colour=color))+scale_y_log10()+s
```



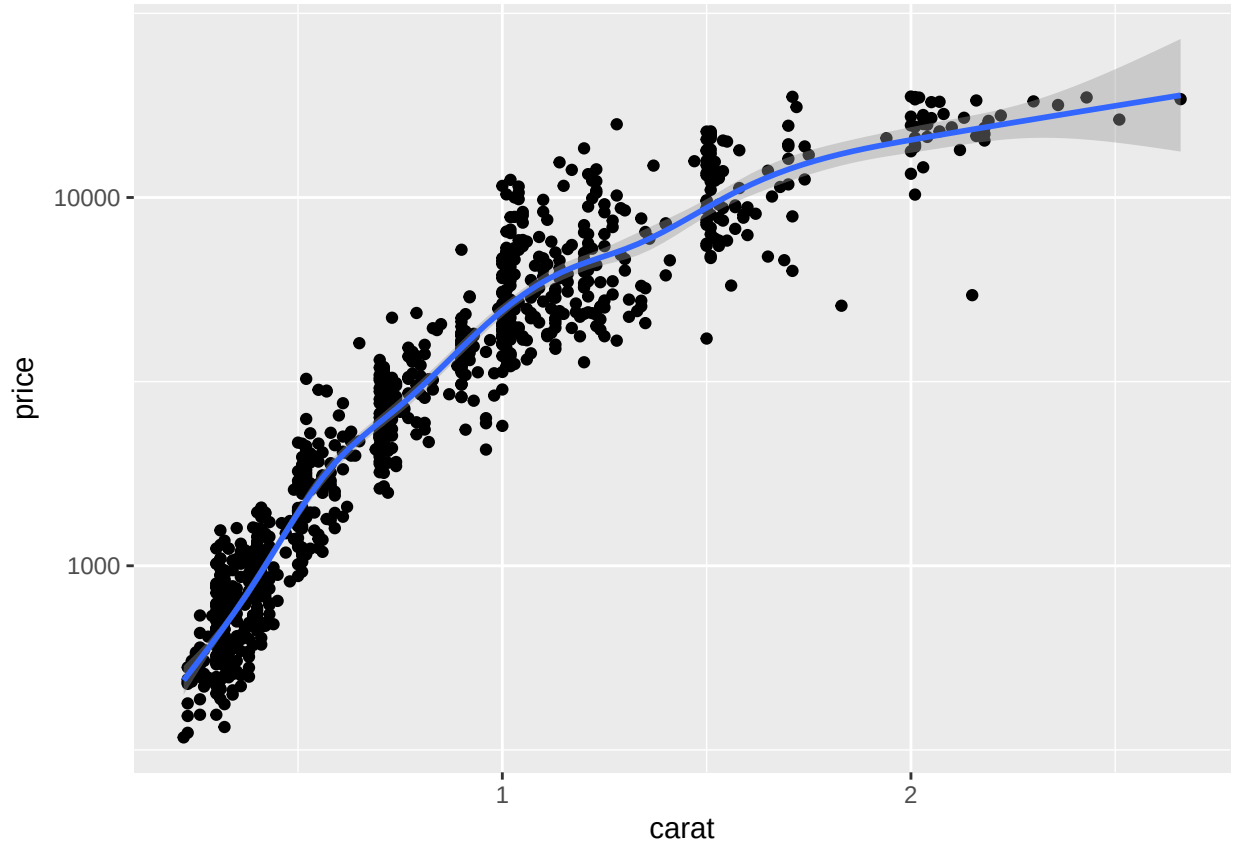
以数据（Data）和映射（Mapping）一节中所画散点图为例，将 Y 轴坐标进行 log10 变换，再自己定义颜色为彩虹色。

统计变换（Statistics）

统计变换对原始数据进行某种计算，然后在图上表示出来，例如对散点图上加一条回归线。

```
ggplot(small, aes(x=carat, y=price))+geom_point()+scale_y_log10()+stat_smooth()
```

```
## `geom_smooth()` using method = 'gam'
```

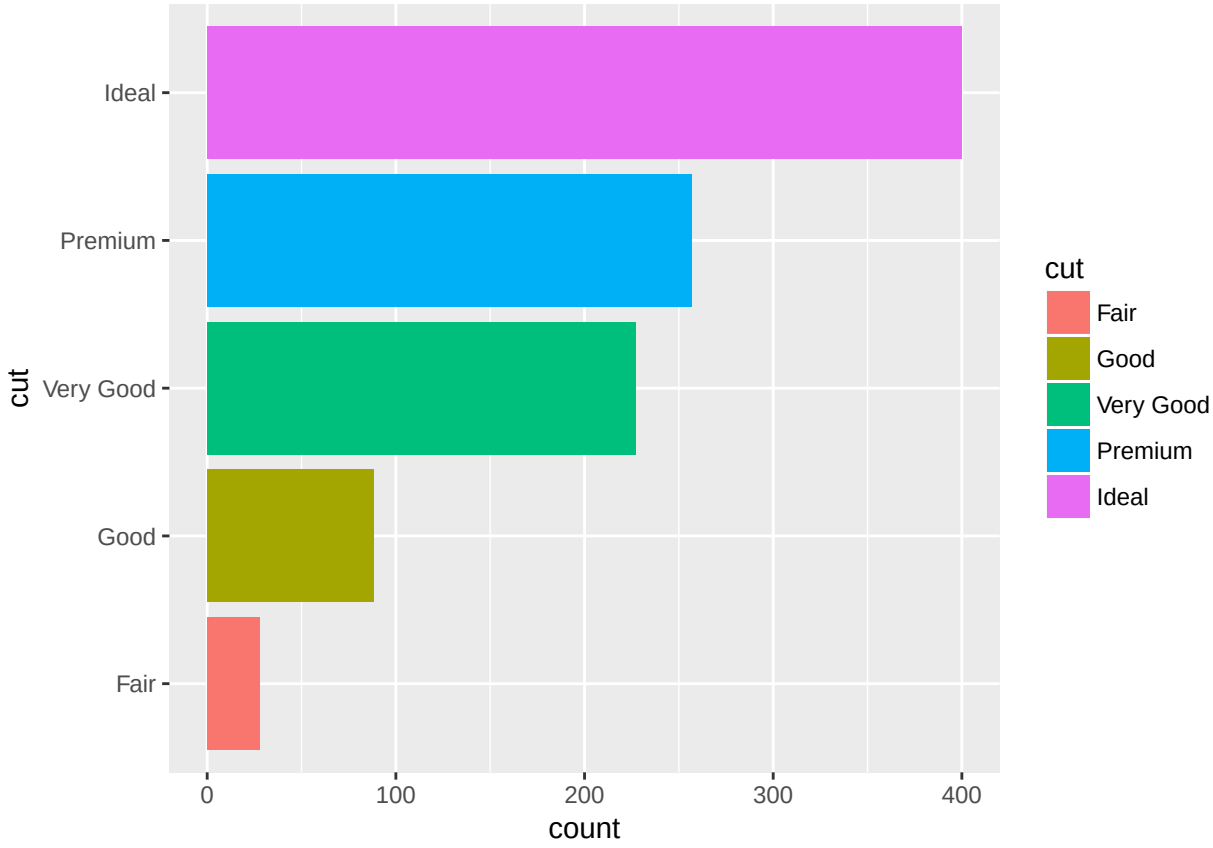


这里，aes 所提供的参数，就通过 ggplot 提供，而不是提供给 geom_point，因为 ggplot 里的参数，相当于全局变量，geom_point() 和 stat_smooth() 都知道 x,y 的映射，如果只提供给 geom_point()，则相当于是局部变量，geom_point 知道这种映射，而 stat_smooth 不知道，当然你再给 stat_smooth 也提供 x,y 的映射，不过共用的映射，还是提供给 ggplot 好。ggplot2 提供了多种统计变换方式：stat_abline stat_contour stat_identity stat_summary stat_bin stat_density stat_qq stat_summary2d stat_bin2d stat_density2d stat_quantile stat_summary_hex stat_bindot stat_ecdf stat_smooth stat_unique stat_binhex stat_function stat_spoke stat_vline stat_boxplot stat_hline stat_sum stat_ydensity 统计变换是非常重要的功能，我们可以自己写函数，基于原始数据做某种计算，并在图上表现出来，也可以通过它改变 geom_xxx 函数画图的默认统计参数。

坐标系统（Coordinante）

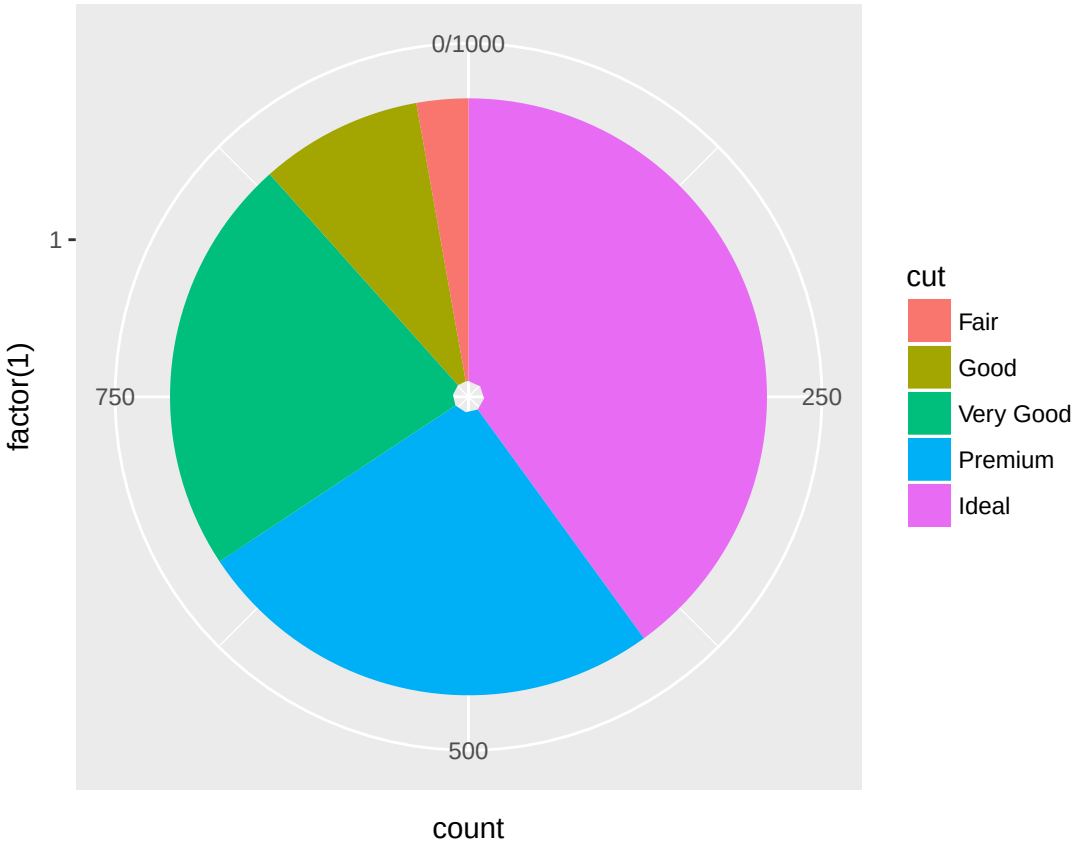
坐标系统控制坐标轴，可以进行变换，例如 XY 轴翻转，笛卡尔坐标和极坐标转换，以满足我们的各种需求。坐标轴翻转由 coord_flip() 实现

```
ggplot(small)+geom_bar(aes(x=cut, fill=cut))+coord_flip()
```



而转换成极坐标可以由 coord_polar() 实现：

```
ggplot(small)+geom_bar(aes(x=factor(1), fill=cut))+coord_polar(theta="y")
```



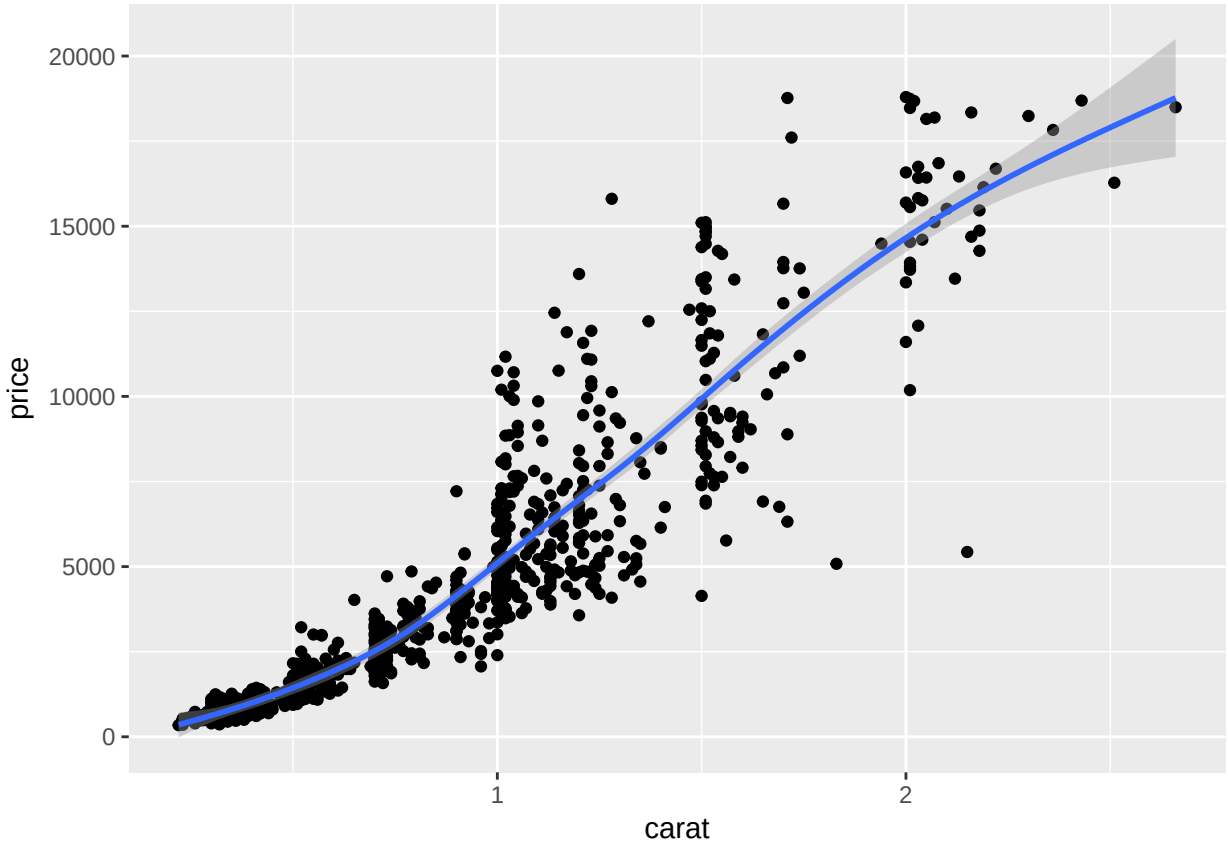
这也是为什么之前介绍常用图形画法时没有提及饼图的原因，饼图实际上就是柱状图，只不过是使用极坐标而已，柱状图的高度，对应于饼图的弧度，饼图并不推荐，因为人类的眼睛比较弧度的能力比不上比较高度（柱状图）

图层（Layer）

photoshop 流行的原因在于 PS 3.0 时引入图层的概念，ggplot 的牛 B 之处在于使用 + 号来叠加图层，这堪称是泛型编程的典范。在前面散点图上，我们已经见识过，加上了一个回归线拟合的图层。有了图层的概念，使用 ggplot 画起图来，就更加得心应手。做为图层的一个很好的例子是蝙蝠侠 logo，batman logo 由 6 个函数组成，在下面的例子中，我先画第一个函数，之后再加一个图层画第二个函数，不断重复这一过程，直到六个函数全部画好。

```
require(ggplot2)
p <- ggplot(small, aes(carat, price))
p + geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'gam'
```

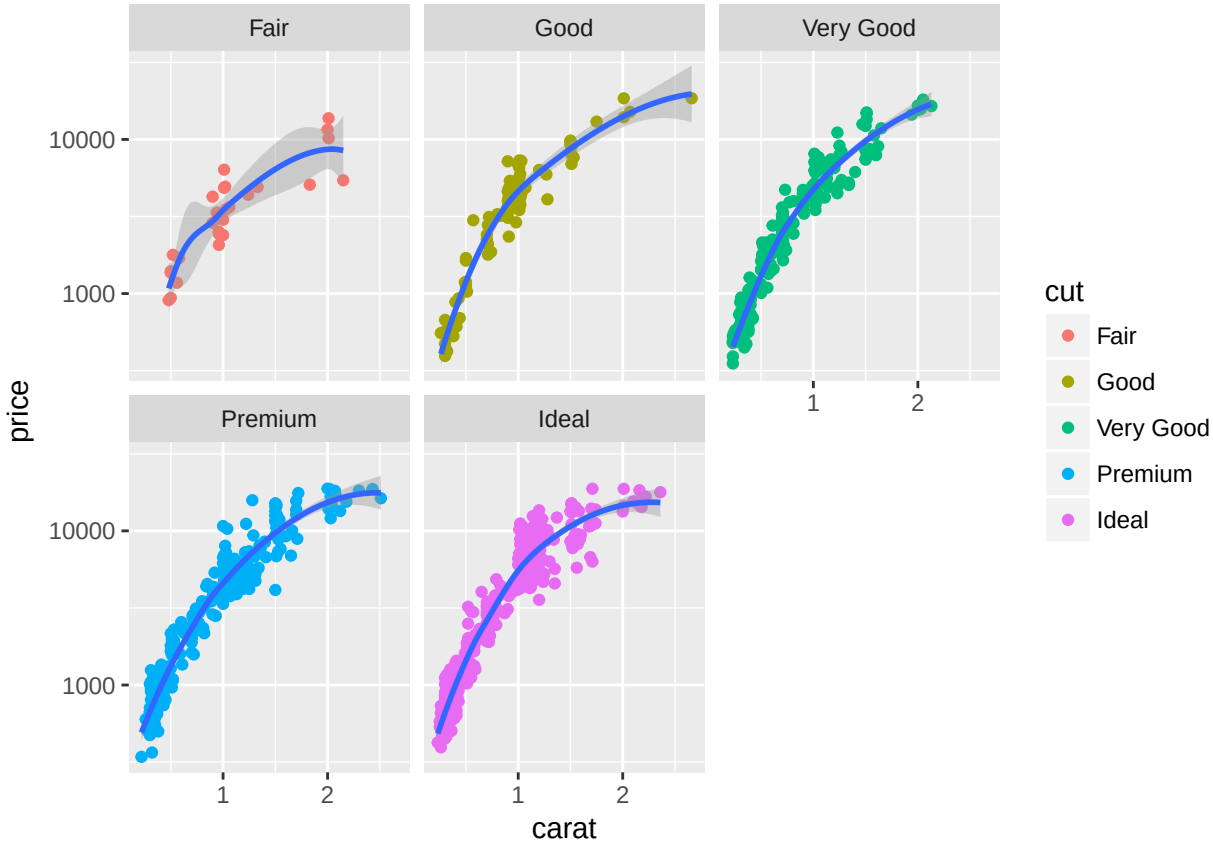


分面（Facet）

分面可以让我们按照某种给定的条件，对数据进行分组，然后分别画图。在统计变换一节中，提到如果按切工分组作回归线，显然图会很乱，有了分面功能，我们可以分别作图。

```
ggplot(small, aes(x=carat, y=price))+geom_point(aes(colour=cut))+scale_y_log10() +facet_wra
```

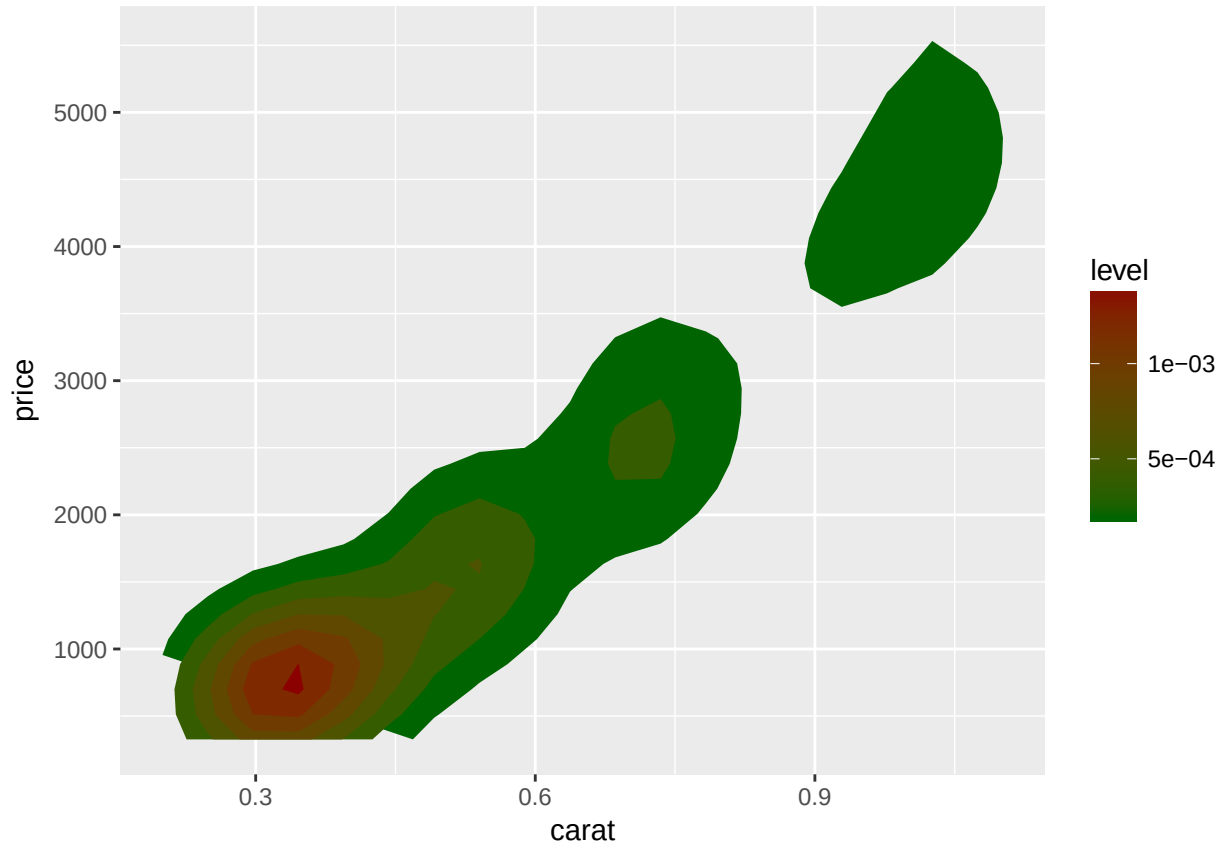
```
## `geom_smooth()` using method = 'loess'
```



二维密度图

为了作图方便，我们使用 diamonds 数据集的一个子集，如果使用全集，数据量太大，画出来散点就糊了，这种情况可以使用二维密度力来呈现。

```
ggplot(diamonds, aes(carat, price))+ stat_density2d(aes(fill = ..level..), geom="polygon")
```


price
5000
4000
3000
2000
1000
0.3
0.6
0.9
carat
level
1e−03
5e−04