

1、Why use `ggplot2`

`ggplot2`是我见过最 human friendly 的画图软件，这得益于 Leland Wilkinson 在他的著作《The Grammar of Graphics》中提出了一套图形语法，把图形元素抽象成可以自由组合的成分，Hadley Wickham 把这套想法在 R 中实现。

为什么要学习 `ggplot2`，可以参考 `ggplot2: 数据分析与图形艺术的序言` (btw: 在序言的最后，我被致谢了)。

Hadley Wickham 也给出一堆理由让我们说服自己，我想再补充一点，Hadley Wickham 是学医出身的，做为学生物出身的人有什么理由不支持呢?)

`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。

2、数据 (Data) 和映射 (Mapping)

下面以一份钻石的数据为例，这份数据非常大，随机取一个子集来画图。

```


1 require(ggplot2)
2 data(diamonds)
3 set.seed(42)
4 small <-
  diamonds[sample(nrow(diamonds),
1000), ]
5 head(small)
1 ##      carat      cut
  color clarity depth table
  price   x     y     z
2 ## 49345  0.71 Very
  Good     H     SI1  62.5    60  2096
  5.68 5.75 3.57
3 ##
  50545  0.79  Premium    H     SI1  61.8    59  2275
  5.97 5.91 3.67
4 ##
  15434  1.03    Ideal    F     SI1  62.4    57  6178
  6.48 6.44 4.03
5 ##
  44792  0.50    Ideal    E     VS2  62.2    54  1624
  5.08 5.11 3.17
6 ##
  34614  0.27    Ideal    E     VS1  61.6    56  470
  4.14 4.17 2.56
7 ##
  27998  0.30  Premium    E     VS2  61.7    58  658
  4.32 4.34 2.67
1 summary(small)
0 ##      carat      cut      color      clarity      depth
1
02 ## Min.    :0.220    Fair      :
  28 D:121  SI1    :258  Min.    :55.2
03 ## 1st Qu.:0.400    Good      :
  88 E:186  VS2    :231  1st
  Qu.:61.0
04 ## Median :0.710    Very
  Good:227  F:164  SI2    :175  Median :61.8
0 ## Mean    :0.819    Premium  :257  G:216  VS1    :141  Mean    :61.
5 7
06 ## 3rd
  Qu.:1.070    Ideal    :400  H:154  VVS2    :
  91 3rd Qu.:62.5
07 ## Max.    :2.660
  I:106  VVS1    :

```

```

67   Max.    :72.2
08   ##                                     J:
53   (Other): 37
09   ##      table      price      x      y
10   ## Min.    :50.1   Min.    : 342   Min.    :3.85   Min.    :3.84
11   ## 1st
   Qu.:56.0   1st
   Qu.: 990   1st
   Qu.:4.74   1st
   Qu.:4.76
12   ## Median :57.0   Median :
2595   Median :5.75   Median :5.78
13   ## Mean    :57.4   Mean    :
4111   Mean    :5.79   Mean    :5.79
14   ## 3rd
   Qu.:59.0   3rd
   Qu.:
5495   3rd
   Qu.:6.60   3rd
   Qu.:6.61
15   ## Max.    :65.0   Max.    :18795   Max.    :8.83   Max.    :8.87
16   ##
17   ##      z
18   ## Min.    :2.33
19   ## 1st
   Qu.:2.92
20   ## Median :3.55
21   ## Mean    :3.57
22   ## 3rd
   Qu.:4.07
23   ## Max.    :5.58
24   ##

```

 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。

```

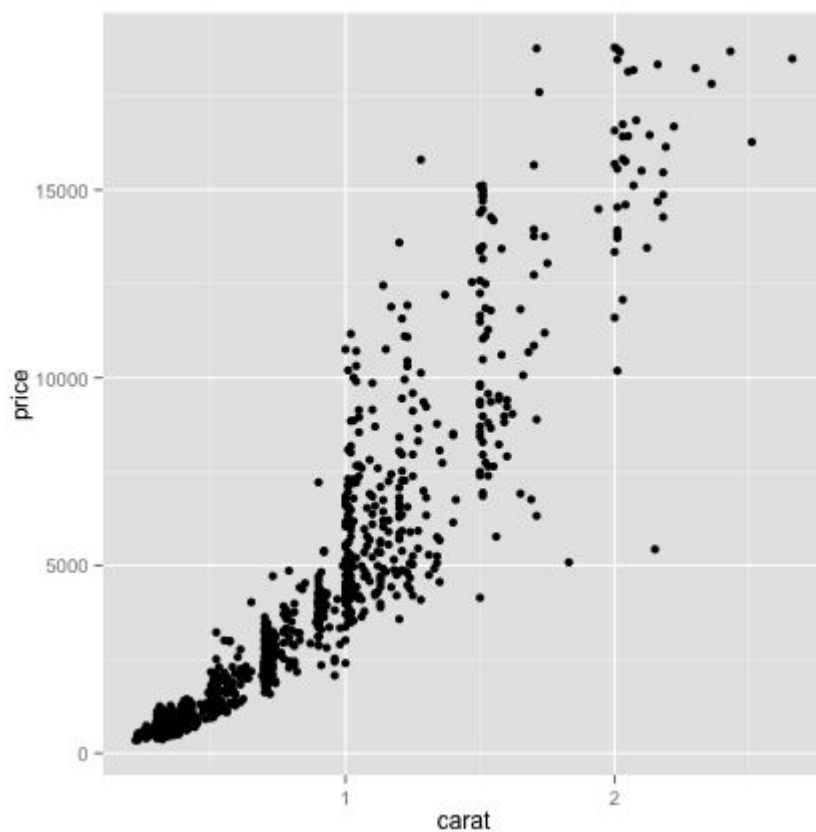
1   p <-
   ggplot(data
   = small,
   mapping =
   aes(x =
   carat, y =
   price))

```

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，下面以散点为例：

```
1 | p  
  | + geom_point()
```



几何对象将在下面的小节介绍，这一节，关注的是数据和图形属性之间的映射。

如果想将切工 (cut) 映射到形状属性。只需要：

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

再比如我想将钻石的颜色 (color) 映射颜色属性：

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

3、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：

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

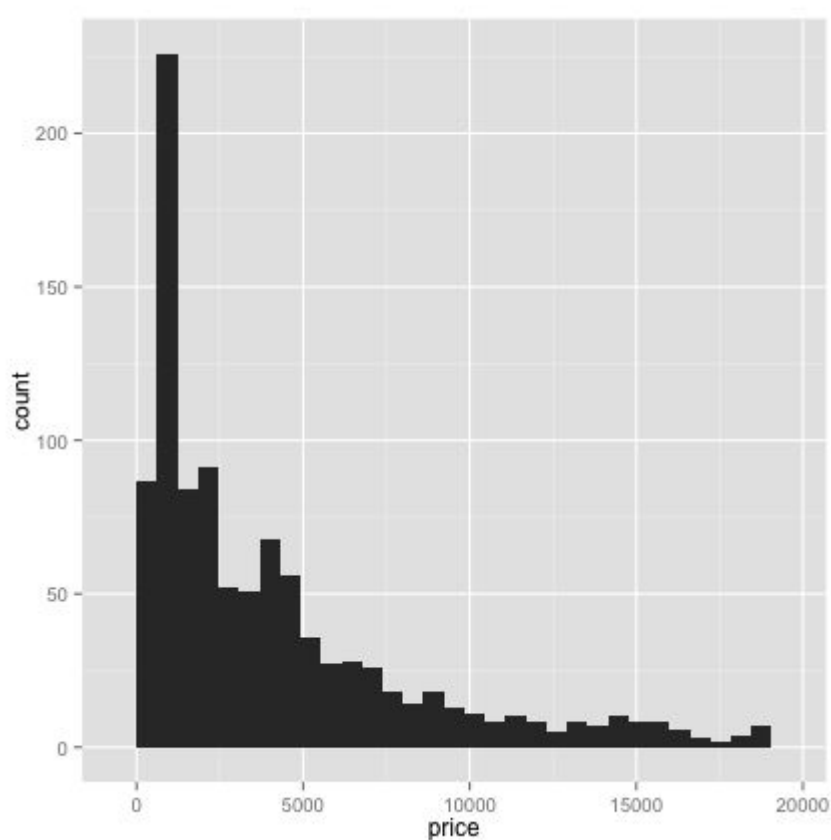
`ggplot2` 支持图层，我通常把不同的图层中共用的映射提供给 `ggplot` 函数，而某一几何对象才需要的映射参数提供给 `geom_xxx` 函数。

这一小节我们来看一下各种常用的几何对象。

直方图

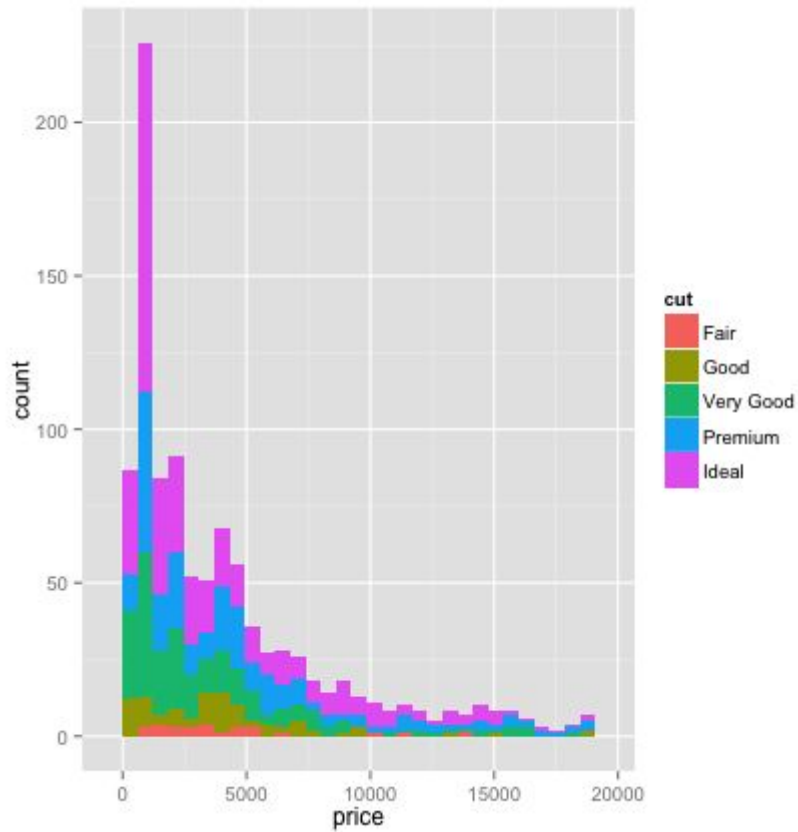
直方图最容易，提供一个 `x` 变量，画出数据的分布。

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



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：

```
1 | ggplot(small)+geom_histogram(aes(x=price,  
  fill=cut))
```

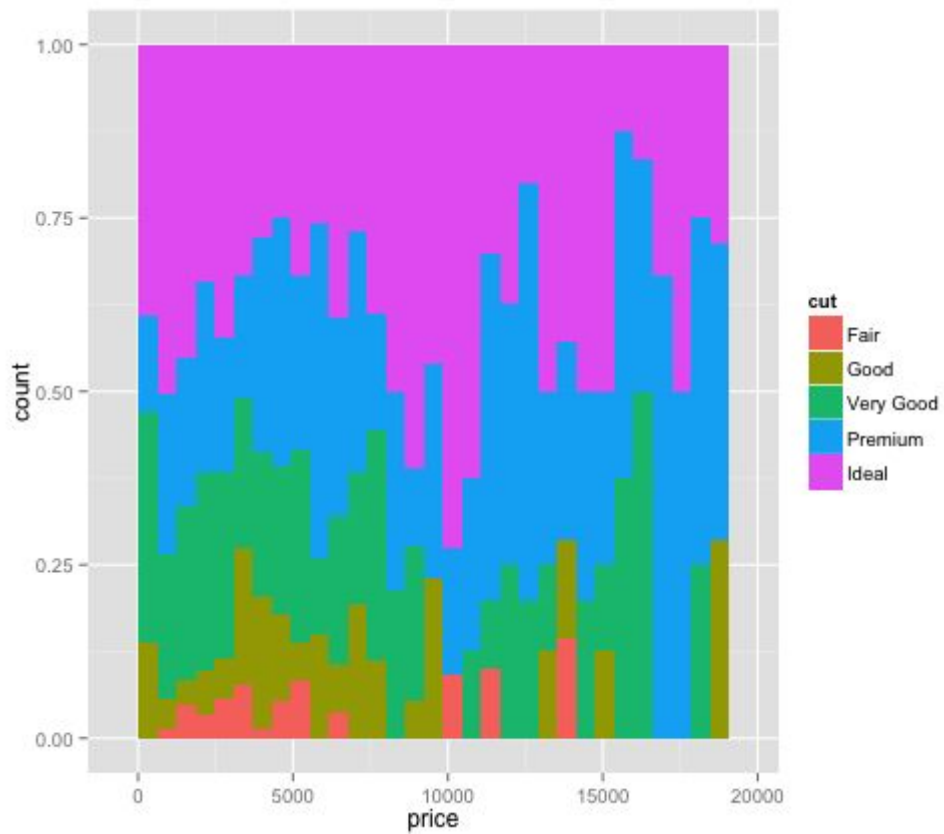


也可以将其分开，side-by-side 地画直方图。

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

还可以使用 position="fill"，按照相对比例来画。

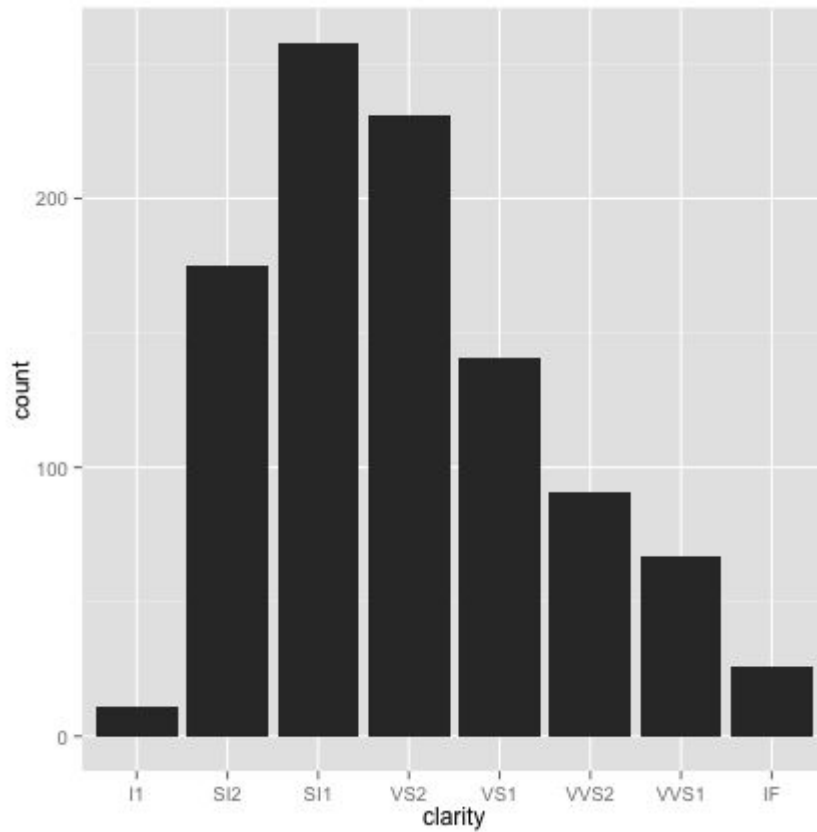
```
1 | ggplot(small)+geom_histogram(aes(x=price,  
  | fill=cut), position="fill")
```

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。

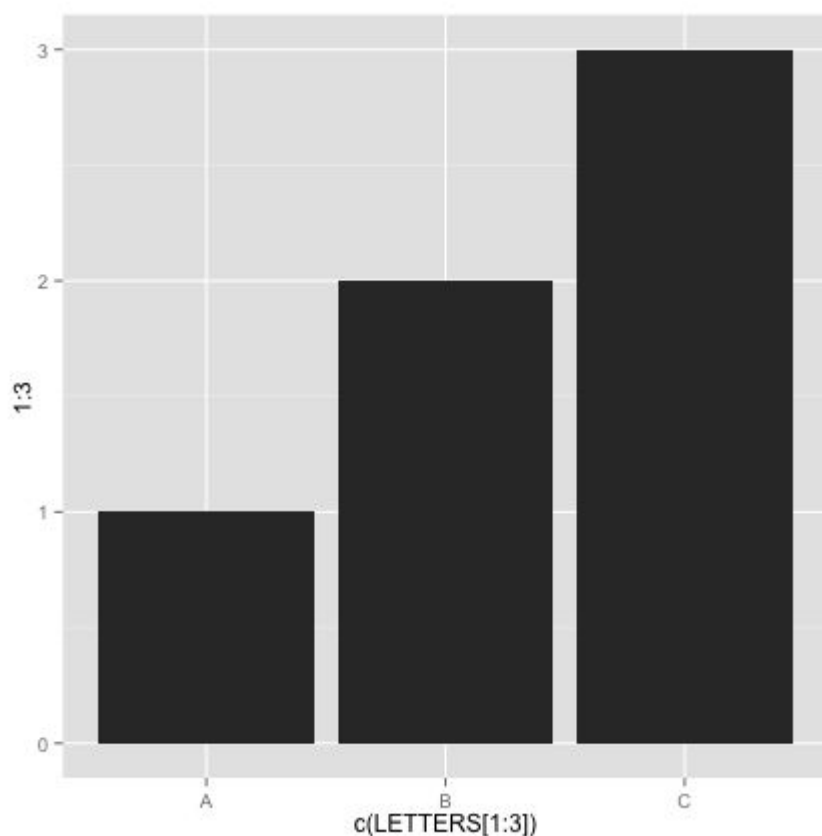
```
1 | ggplot(small)+geom_bar(aes(x=clarity))
```



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：

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



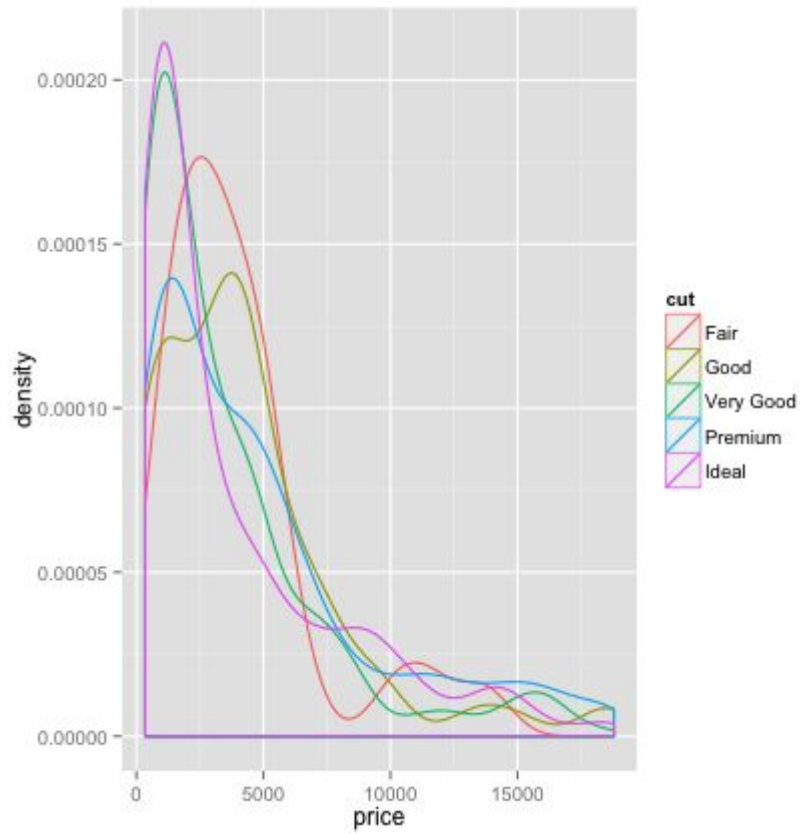
柱状图和直方图是很像的，直方图把连续型的数据按照一个个等长的分区（bin）来切分，然后计数，画柱状图。而柱状图是分类数据，按类别计数。我们可以用前面直方图的参数来画 side-by-side 的柱状图，填充颜色或者按比例画图，它们是高度一致的。

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，我将在图层一节中给出实例。

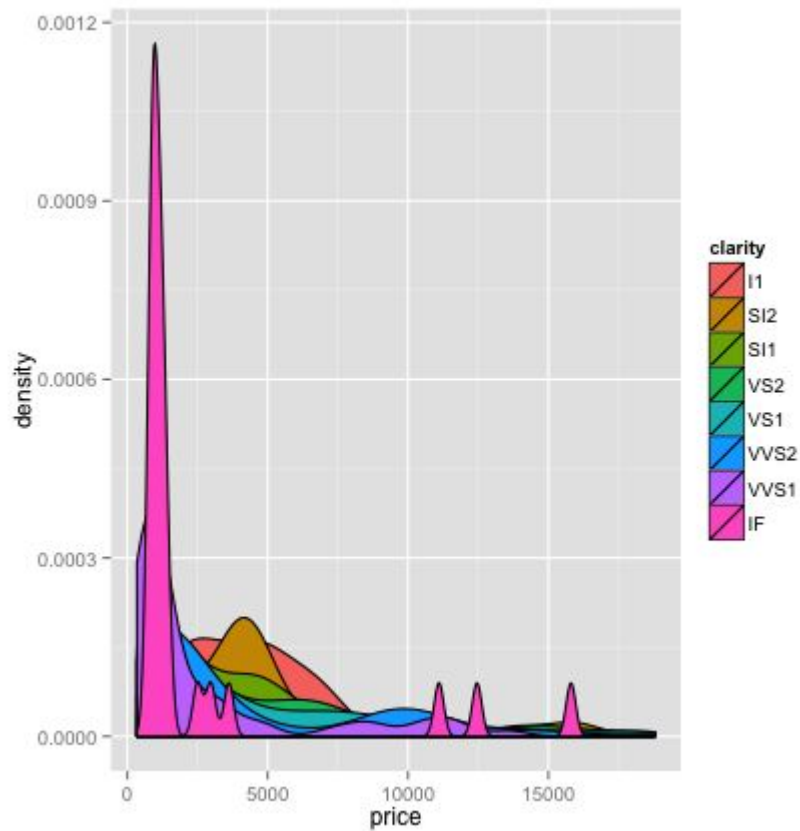
密度函数图

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：

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



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



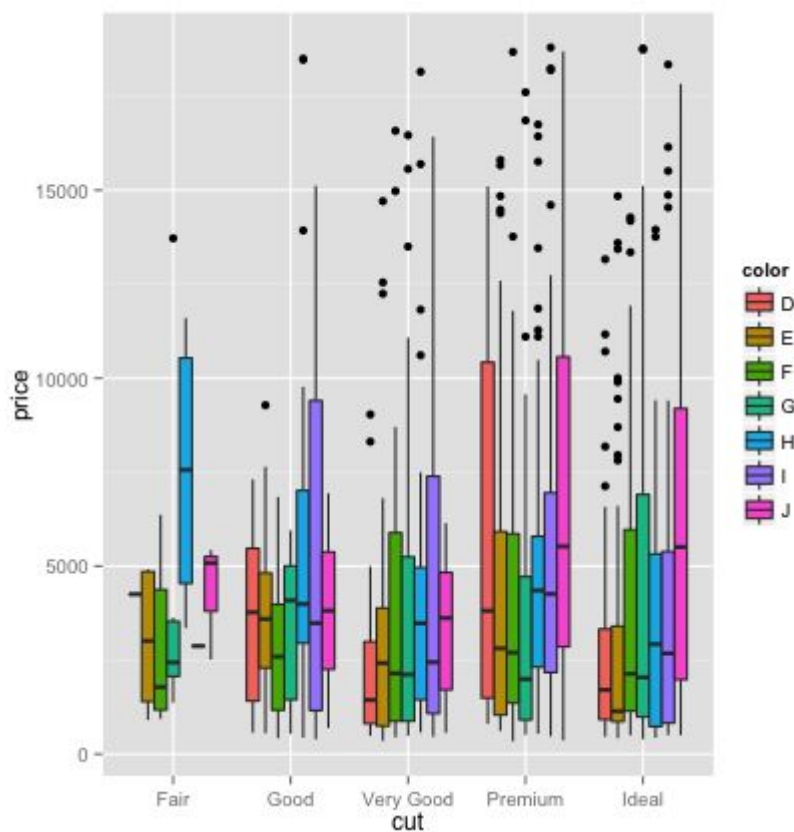
`colour` 参数指定的是曲线的颜色，而 `fill` 是往曲线下面填充颜色。

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，被 **Nature Methods** 吐槽，这种情况推荐使用 `boxplot`。

```
1 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。

```
01 geom_abline    geom_area  
02 geom_bar      geom_bin2d  
03 geom_blank    geom_boxplot
```

```

04 geom_contour    geom_crossbar
05 geom_density    geom_density2d
06 geom_dotplot    geom_errorbar
07 geom_errorbarh    geom_freqpoly
08 geom_hex        geom_histogram
09 geom_hline      geom_jitter
10 geom_line       geom_linerange
11 geom_map        geom_path
12 geom_point      geom_pointrange
13 geom_polygon    geom_quantile
14 geom_raster     geom_rect
15 geom_ribbon     geom_rug
16 geom_segment    geom_smooth
17 geom_step       geom_text
18 geom_tile       geom_violin
19 geom_vline

```

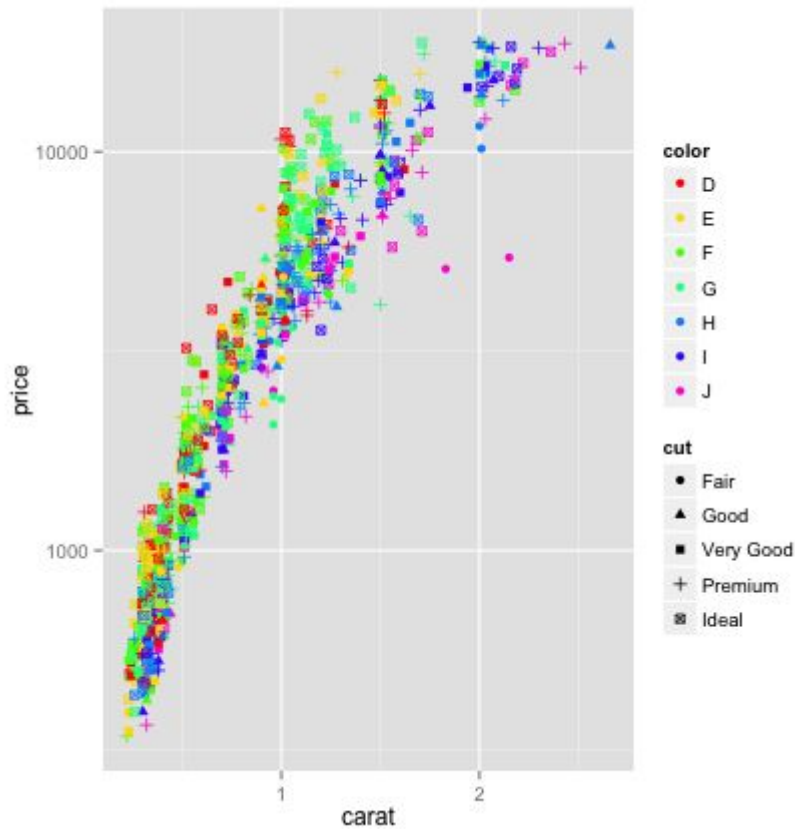
4、标尺 (Scale)

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。

```

1 ggplot(small)+geom_point(aes(x=carat, y=price, shape=cut,
  colour=color))+scale_y_log10()+scale_colour_manual(values=rainbow(7))

```

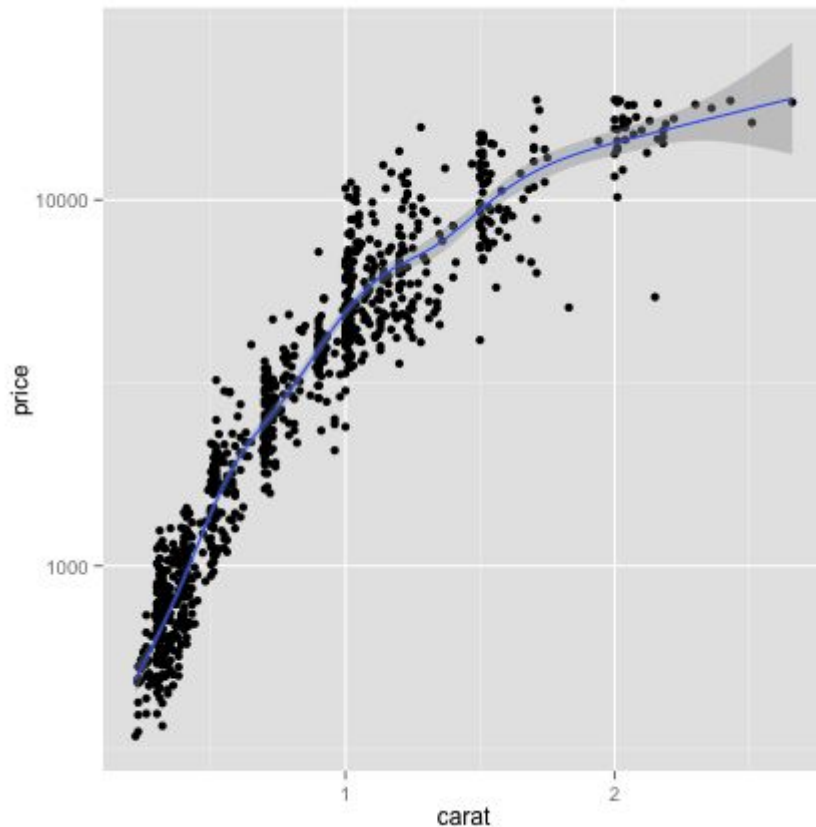


以数据（Data）和映射（Mapping）一节中所画散点图为例，将Y轴坐标进行log10变换，再自己定义颜色为彩虹色。

5、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。

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




这里就不按颜色、切工来分了，不然 [ggplot](#) 会按不同的分类变量分别做回归，图就很乱，如果需要这样做，我们可以使用分面，这个将在后面介绍。

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：

1	<code>stat_abline</code>	<code>stat_contour</code>	<code>stat_identity</code>	<code>stat_summary</code>
2	<code>stat_bin</code>	<code>stat_density</code>	<code>stat_qq</code>	<code>stat_summary2d</code>
3	<code>stat_bin2d</code>	<code>stat_density2d</code>	<code>stat_quantile</code>	<code>stat_summary_hex</code>
4	<code>stat_bindot</code>	<code>stat_ecdf</code>	<code>stat_smooth</code>	<code>stat_unique</code>
5	<code>stat_binhex</code>	<code>stat_function</code>	<code>stat_spoke</code>	<code>stat_vline</code>
6	<code>stat_boxplot</code>	<code>stat_hline</code>	<code>stat_sum</code>	<code>stat_ydensity</code>

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` 函数的默认统计参数。

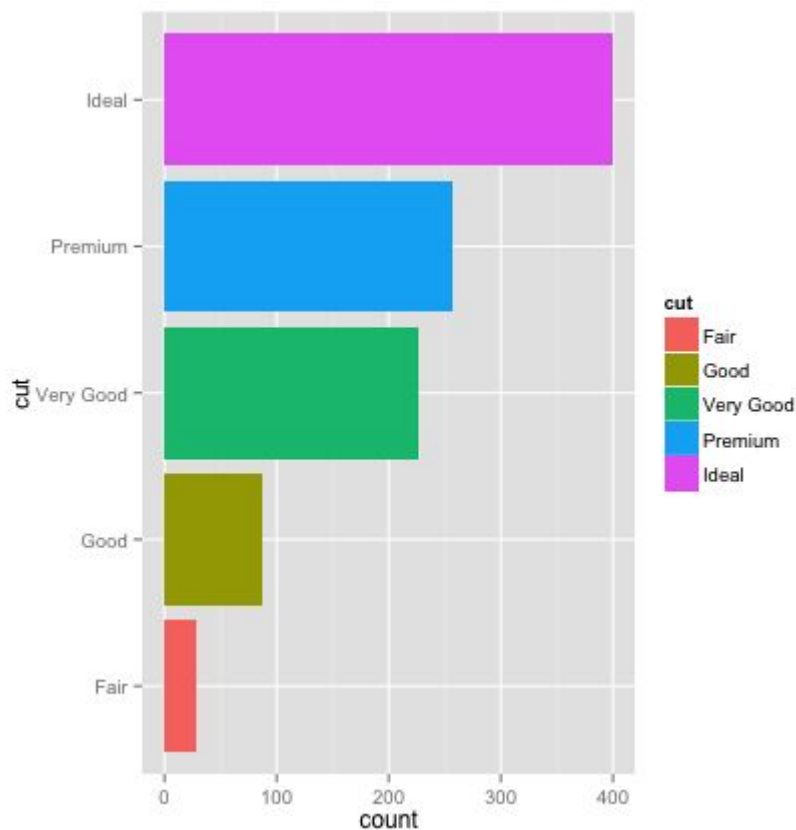
比如我在 [Proteomic investigation of the interactome of FMNL1 in hematopoietic cells unveils a role in calcium-dependent membrane plasticity](#) 的图一中，就把 boxplot 的中位线替换成了平均值来作图。

6、坐标系 (Coordinante)

坐标系控制坐标轴，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。

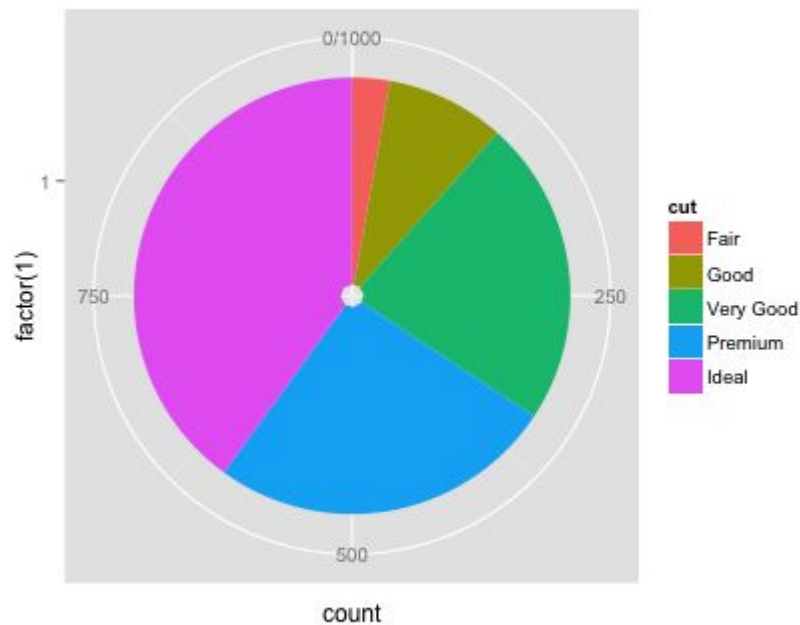
坐标轴翻转由 `coord_flip()` 实现

```
1 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：

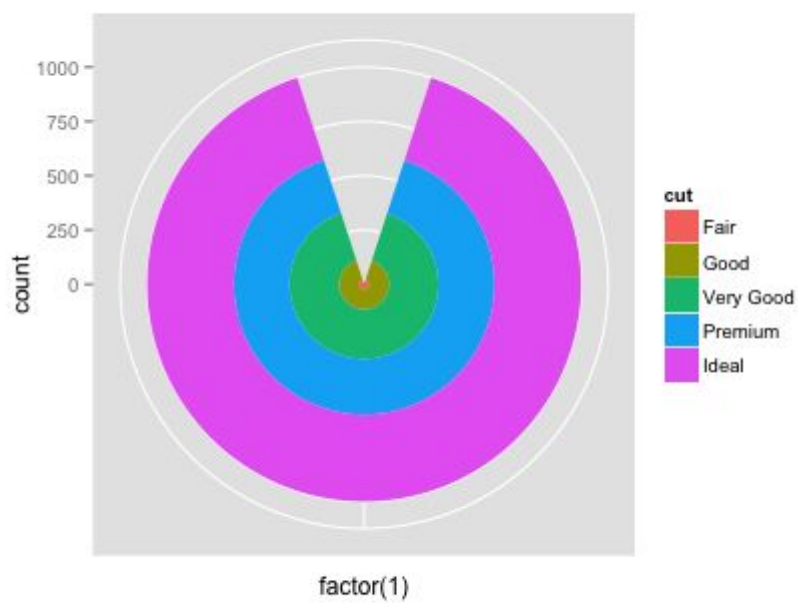
```
1 ggplot(small)+geom_bar(aes(x=factor(1),
  fill=cut))+coord_polar(theta="y")
```



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）

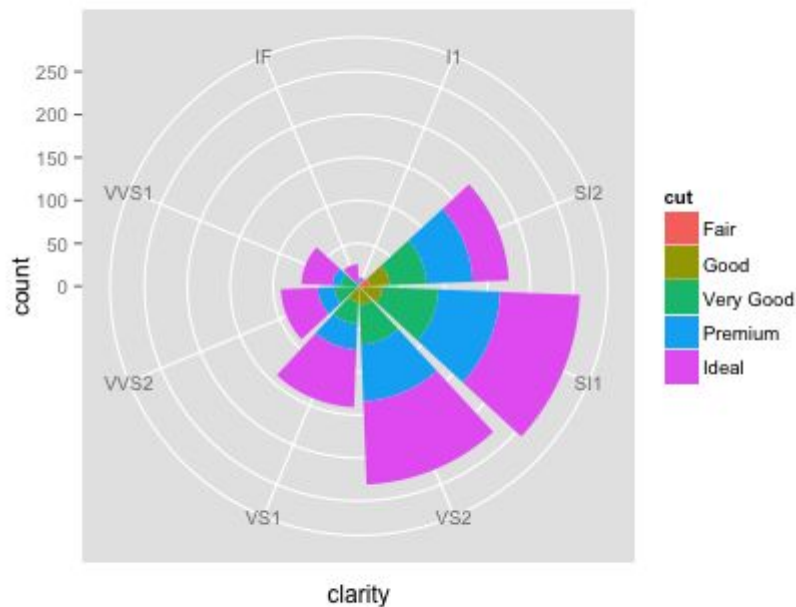
还可以画靶心图：

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



以及风玫瑰图(windrose)

```
1 | ggplot(small)+geom_bar(aes(x=clarity,  
| fill=cut))+coord_polar()
```



7、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。

```

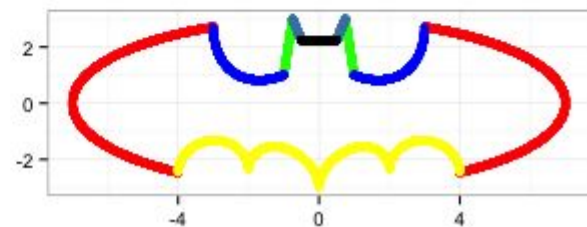
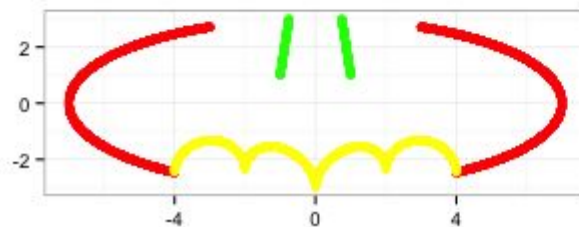
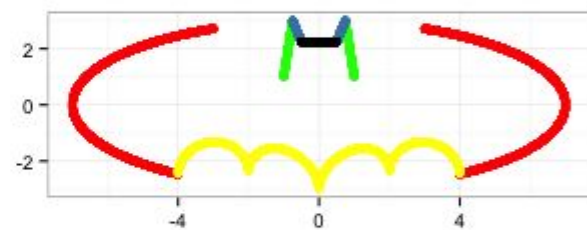
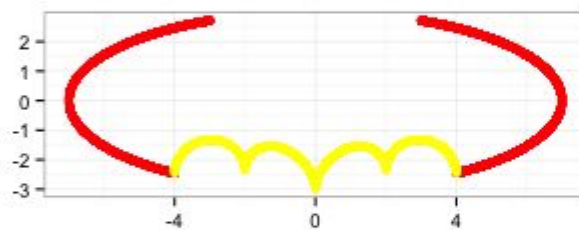
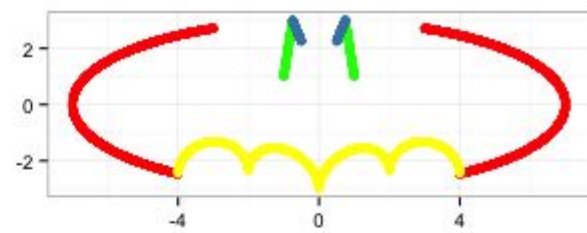
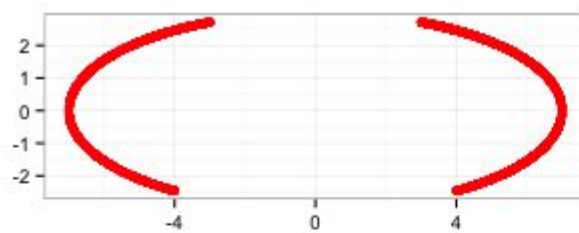
01   require(ggplot2)
02   f1data.frame(x=x,y=y)
03   d
04   -3*sqrt(33)/7,]
05   return(d)
06   }

```

```

07 x1data.frame(x2=x2,
08 y2=y2)
09 p2data.frame(x3=x3,
10 y3=y3)
11 p3data.frame(x4=x4,y4=y4)
12 p4data.frame(x5=x5,y5=y5)
13 p5data.frame(x6=x6,y6=y6)
14 p6

```



下面再以生物界中常用的柱状图+误差图为例，展示 [ggplot2](#) 非常灵活的图层。以我 2011 年发表的文章 [Phosphoproteome profile of human lung cancer cell line A549](#) 中的 westernblot 数据为例。

```

1 Normaldata.frame(V=c("Normal", "Cancer"),
2 mean=m, sd=s)
3 d$V

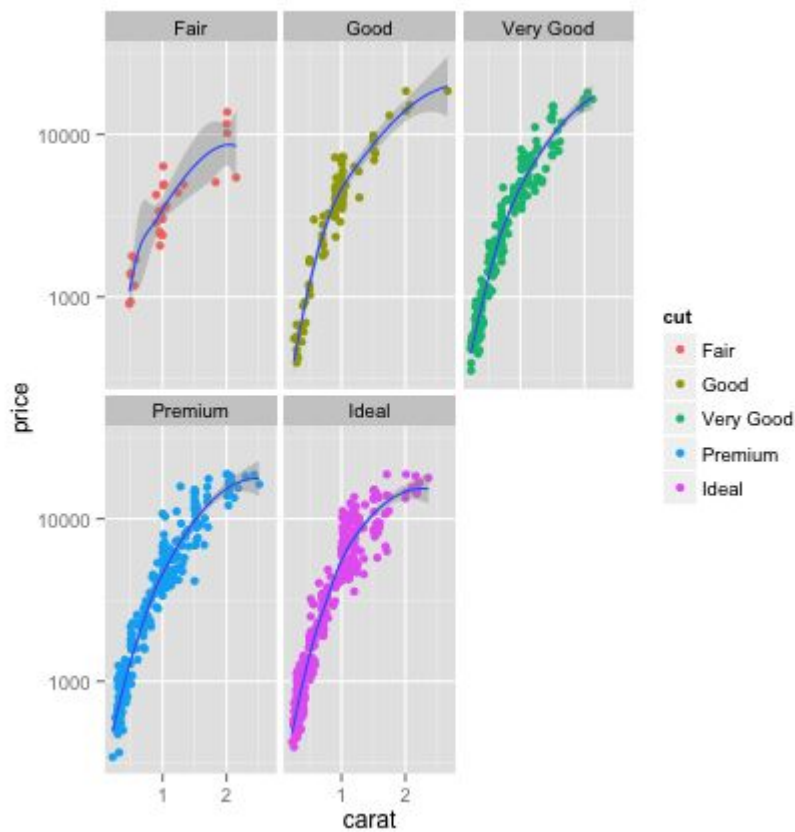
```

8、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。

```
1 | ggplot(small, aes(x=carat,
  y=price))+geom_point(aes(colour=cut))+scale_y_log10()
  +facet_wrap(~cut)+stat_smooth()
```

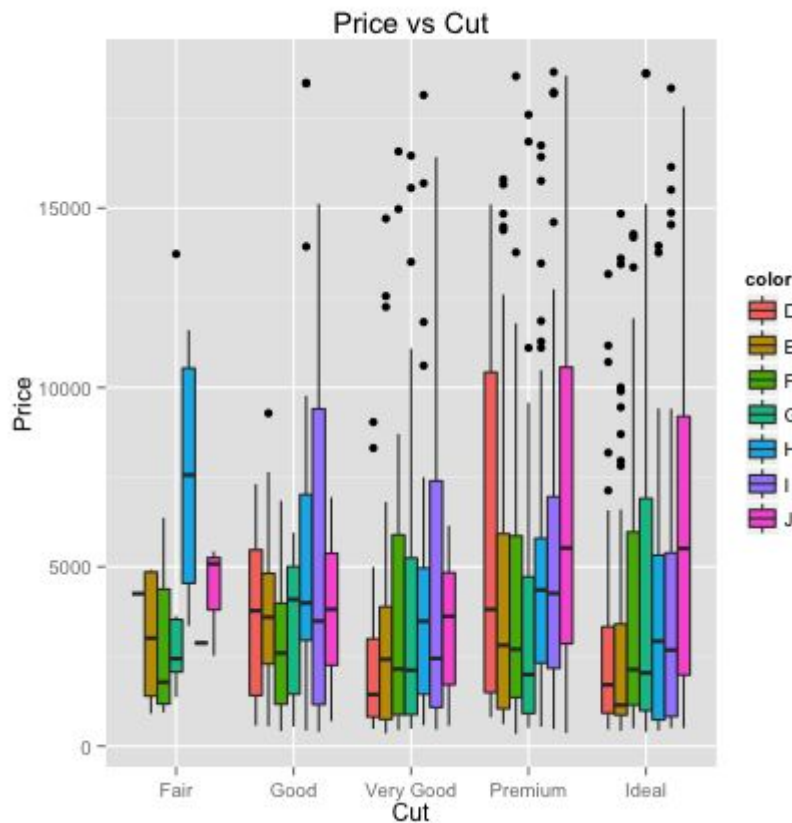


9、主题 (Theme)

通过 ggplot 画图之后，我们可能还需要对图进行定制，像 title, xlab, ylab 这些高频需要用到的，自不用说，ggplot2 提供了 ggtitle(), xlab()和 ylab()来实现。

比如：

```
1 | p
```



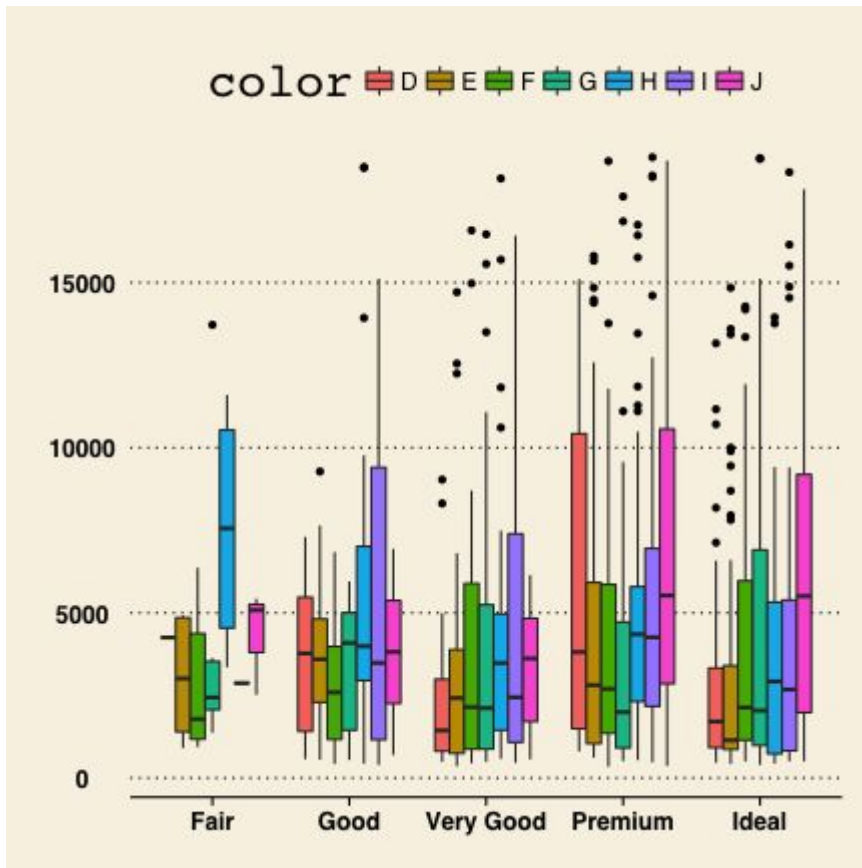
但是这个远远满足不了需求，我们需要改变字体，字体大小，坐标轴，背景等各种元素，这需要通过 `theme()` 函数来完成。

`ggplot2` 提供一些已经写好的主题，比如 `theme_grey()` 为默认主题，我经常用的 `theme_bw()`

为白色背景的主题，还有 `theme_classic()` 主题，和 `R` 的基础画图函数较像。

另外 `ggthemes` 包提供了一些主题可供使用，包括：

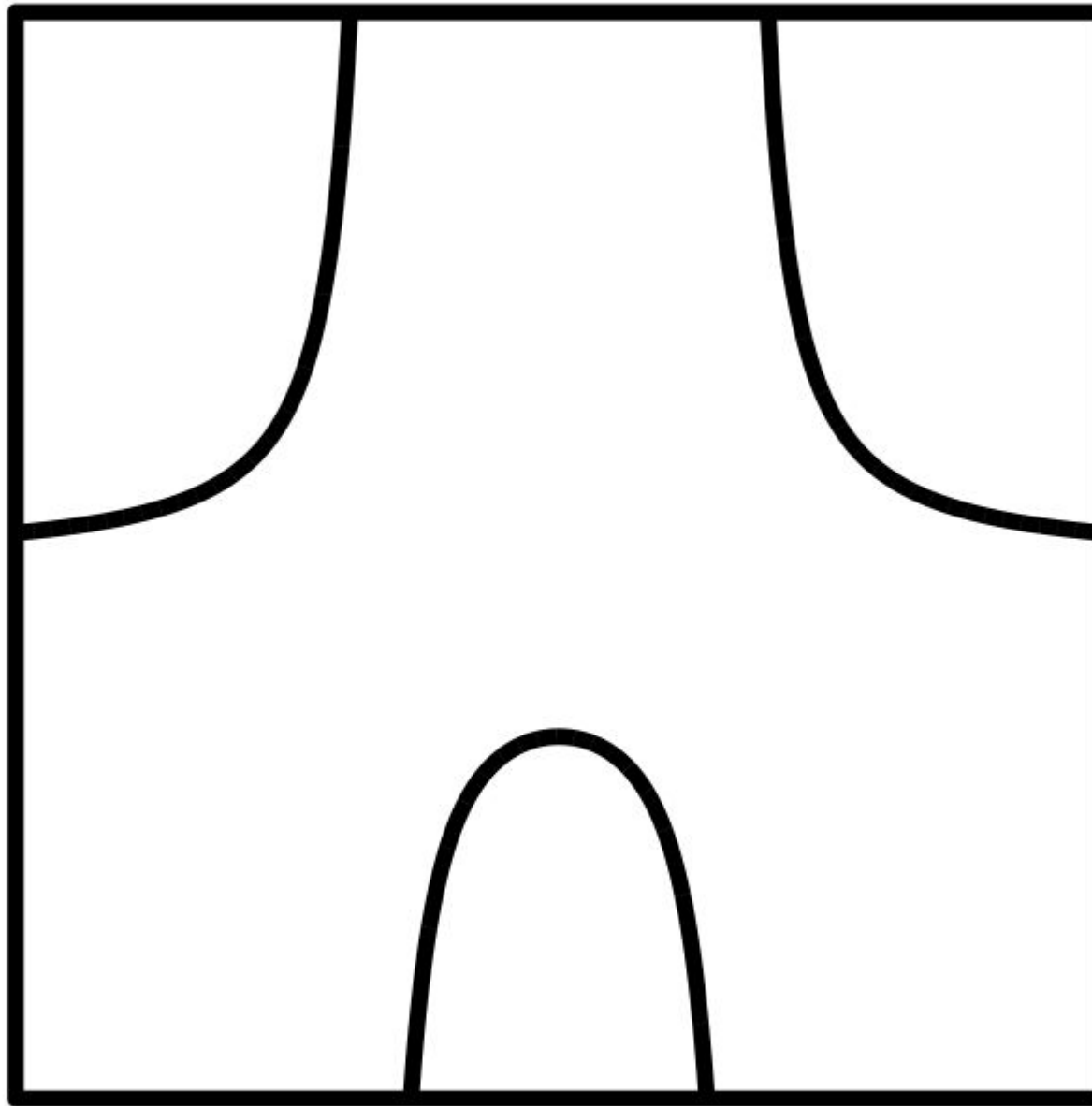
```
1 theme_economist
  theme_economist_white
2 theme_wsj      theme_excel
3 theme_few      theme_foundation
4 theme_igray    theme_solarized
5 theme_stata    theme_tufte
1 require(ggthemes)
2 p +
  theme_wsj()
```



在 2013 年发表的文章 [Putative cobalt- and nickel-binding proteins and motifs in Streptococcus pneumoniae](#) 中的图 3 就是使用 `theme_stata` 来画的。

至于如何改变这些元素，我觉得我之前[画囫字的博文](#)可以做为例子：

```
1 fdata.frame(x=x,y=y)
2
3 p
```

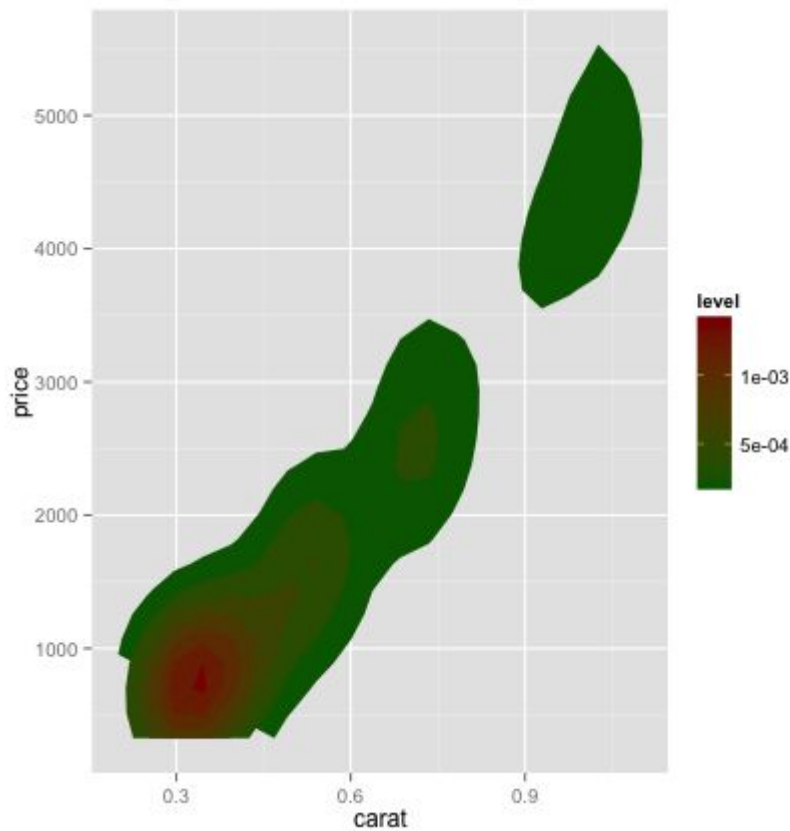
详细的说明，可以参考?theme 的帮助文档。

10、二维密度图

在这个文档里，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。

```
1 | ggplot(diamonds, aes(carat, price))+ stat_density2d(aes(fill
```

```
= ..level..),
geom="polygon")+ scale_fill_continuous(high='darkred',low='darkgreen'
)
```



11、ggplot2 实战

果壳知性里有帖子介绍了个猥琐邪恶的曲线，引来无数宅男用各种工具来画图，甚至于 3D 动态

图都出来了。这里用 ggplot2 来画。3D 版本请[猛击此处](#)。

```
1 fdata.frame(x=c(x1,x2,x3),
2             y=rep(y,3),
3             type=rep(LETTERS[1:3],
4                     each=length(y)))
5 p
```

再来一个蝴蝶图，详见《Modern Applied Statistics with S-PLUS》第一章。

```
1 theta data.frame(x=radius*sin(theta),
2                 y=radius*cos(theta))
3 ggplot(dd, aes(x,
4               y))+geom_path()+theme_null()+xlab("")+ylab("")
```

