

Toggle navigation BZR

- [HOME](#)

- [SEARCH](#)

- [ARCHIVES](#)

- [CATEGORIES](#)

Posted by E-Phone Xu on June 13, 2016

- [TAGS](#)

- [ABOUT ME](#)

目录

- [前言](#)

- [1.用到的包](#)
- [2.数据准备](#)
- [3.数据转换](#)

- [柱形图](#)

- [1.簇状柱形图](#)
- [2.堆积柱形图](#)
- [3.百分比堆积柱形图](#)
- [4.三维簇状柱形图](#)
- [5. 三维堆积柱形图](#)
- [6.三维百分比堆积柱形图](#)
- [7.三维柱形图——用分面替代](#)

- [折线图](#)

- [1.折线图](#)
- [2.堆积折线图](#)
- [3.百分比堆积折线图](#)
- [4.带标记的折线图](#)
- [5.带标记的堆积折线图](#)
- [6.带标记的百分比堆积折线图](#)

- [7.三维折线图 用facet代替](#)
- [饼图](#)
 - [1.饼图](#)
 - [2.三维饼图](#)
 - [3.复合饼图](#)
 - [3.复合条饼图](#)
 - [4.圆环图](#)
 - [5.其他圆形图](#)
- [条形图](#)
 - [1.簇状条形图](#)
 - [2.堆积条形图](#)
 - [3.百分比堆积条形图](#)
 - [4.三维簇状条形图](#)
 - [5.三维堆积条形图](#)
 - [6.三维百分比条形图](#)
 - [7.分面来代替3D](#)
- [面积图](#)
 - [1.面积图](#)
 - [2.堆积面积图](#)
 - [3.百分比堆积面积图](#)
 - [4.三维面积图](#)
 - [5.三维堆积面积图](#)
 - [6.三维百分比堆积面积图](#)
 - [7.分面来代替3D](#)
- [XY散点图](#)
 - [1.散点图](#)
 - [2.带平滑线和数据标记的散点图](#)
 - [3.带平滑线的散点图](#)
 - [4.带直线和数据标记的散点图](#)
 - [5.带直线的散点图](#)

- [6.气泡图](#)
- [7.气泡图](#)
- [股价图](#)
 - [1.盘高-盘低-收盘图](#)
 - [2.开盘-盘高-盘低-收盘图](#)
 - [3.成交量-盘高-盘低-收盘图](#)
 - [4.成交量-开盘-盘高-盘低-收盘图](#)
- [曲面图](#)
 - [1.三维曲面图](#)
 - [2.三维曲面图（框架图）](#)
 - [3.曲面图](#)
 - [4.曲面图（框架图）](#)
- [雷达图](#)
 - [1.雷达图](#)
 - [2.带数据标记的雷达图](#)
 - [3.填充雷达图](#)
- [树状图](#)
- [旭日图](#)
- [直方图](#)
 - [1.直方图](#)
 - [2.排列图](#)
- [箱型图](#)
- [瀑布图](#)
- [漏斗图](#)
 - [1.漏斗图](#)
- [结语](#)

基于ggplot2-2.1.0,不定期补充更新, 更多的资料请见 [ggplot2官方文档](#)。

前言

此文献给熟练EXCEL图表制作的R入门使用者们。

1.用到的包

首先，需要加载做图中可能用到的一些工具包。

```
library(ggplot2) #作图包
library(dplyr) #数据转换包
#>
#> Attaching package: 'dplyr'
#> The following objects are masked from 'package:stats':
#>
#>     filter, lag
#> The following objects are masked from 'package:base':
#>
#>     intersect, setdiff, setequal, union
library(tidyr) #数据转换包
library(splines) #数据差值包
```

2.数据准备

我们需要一些模拟数据，以作为后续作图的必备元素。

```
set.seed(123) #设定随机种子，保证做的图和这里的一样
df <- data.frame(
  var=LETTERS[1:10], #字母A-J
  id=1:10, #数字1-10
  a=runif(10), #10个随机数
  b=runif(10), #10个随机数
```

```

c=runif(10), #10个随机数
stringsAsFactors = F #不转换为因子
)
print(df) #显示数据
#>      var id      a      b      c
#> 1    A  1 0.2875775 0.95683335 0.8895393
#> 2    B  2 0.7883051 0.45333416 0.6928034
#> 3    C  3 0.4089769 0.67757064 0.6405068
#> 4    D  4 0.8830174 0.57263340 0.9942698
#> 5    E  5 0.9404673 0.10292468 0.6557058
#> 6    F  6 0.0455565 0.89982497 0.7085305
#> 7    G  7 0.5281055 0.24608773 0.5440660
#> 8    H  8 0.8924190 0.04205953 0.5941420
#> 9    I  9 0.5514350 0.32792072 0.2891597
#> 10   J 10 0.4566147 0.95450365 0.1471136

```

3.数据转换

上表是我们在excel中常见的数据集形式，但是在R中，由于作图需要，通常会转换为长数据的形式。

```

df1 <- df %>% gather("item",value,-1:-2) %>%
  bind_cols(data.frame(item_id=rep(1:3,each=10))) #使用tidyr和d
print(df1) #显示数据
#>      var id item      value item_id
#> 1    A  1    a 0.28757752      1
#> 2    B  2    a 0.78830514      1
#> 3    C  3    a 0.40897692      1
#> 4    D  4    a 0.88301740      1
#> 5    E  5    a 0.94046728      1
#> 6    F  6    a 0.04555650      1

```

#> 7	G	7	a	0.52810549	1
#> 8	H	8	a	0.89241904	1
#> 9	I	9	a	0.55143501	1
#> 10	J	10	a	0.45661474	1
#> 11	A	1	b	0.95683335	2
#> 12	B	2	b	0.45333416	2
#> 13	C	3	b	0.67757064	2
#> 14	D	4	b	0.57263340	2
#> 15	E	5	b	0.10292468	2
#> 16	F	6	b	0.89982497	2
#> 17	G	7	b	0.24608773	2
#> 18	H	8	b	0.04205953	2
#> 19	I	9	b	0.32792072	2
#> 20	J	10	b	0.95450365	2
#> 21	A	1	c	0.88953932	3
#> 22	B	2	c	0.69280341	3
#> 23	C	3	c	0.64050681	3
#> 24	D	4	c	0.99426978	3
#> 25	E	5	c	0.65570580	3
#> 26	F	6	c	0.70853047	3
#> 27	G	7	c	0.54406602	3
#> 28	H	8	c	0.59414202	3
#> 29	I	9	c	0.28915974	3
#> 30	J	10	c	0.14711365	3

万事俱备后，接下来就可以作图了。 以下图形将严格按照excel 2016版中所有图形的出现顺序，读者可自行对照两者的异同。

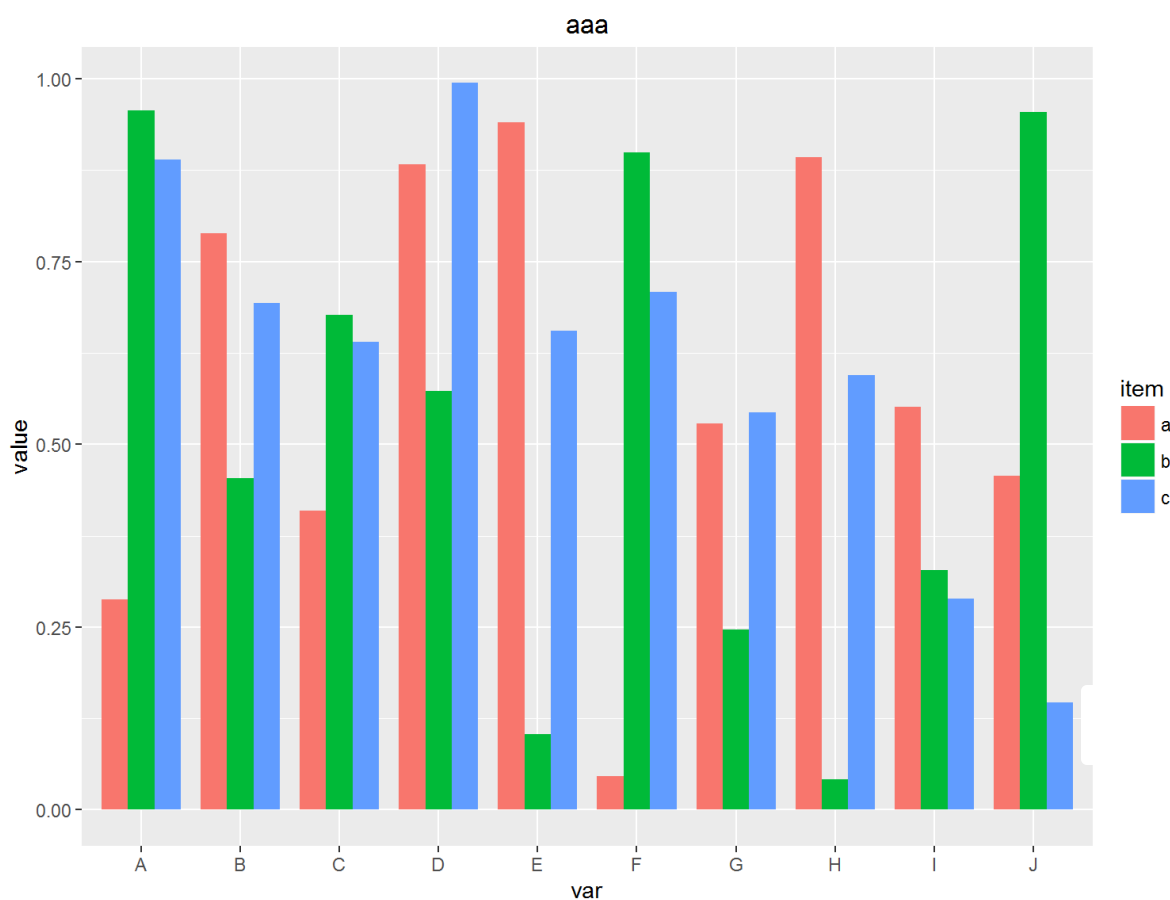
柱形图

一般来说，ggplot的柱形图默认统计方法（stat）是计数（count）的，但是大多数时候，我们更习惯于使用统计完成的数据，因此，在相关的图形参数中必须要增加stat="identity"这一项。

1.簇状柱形图

簇状柱形图的特点是在不同类别中，不同变量的数据是横向排列的，因此需要图形参数中设置position="dodge"

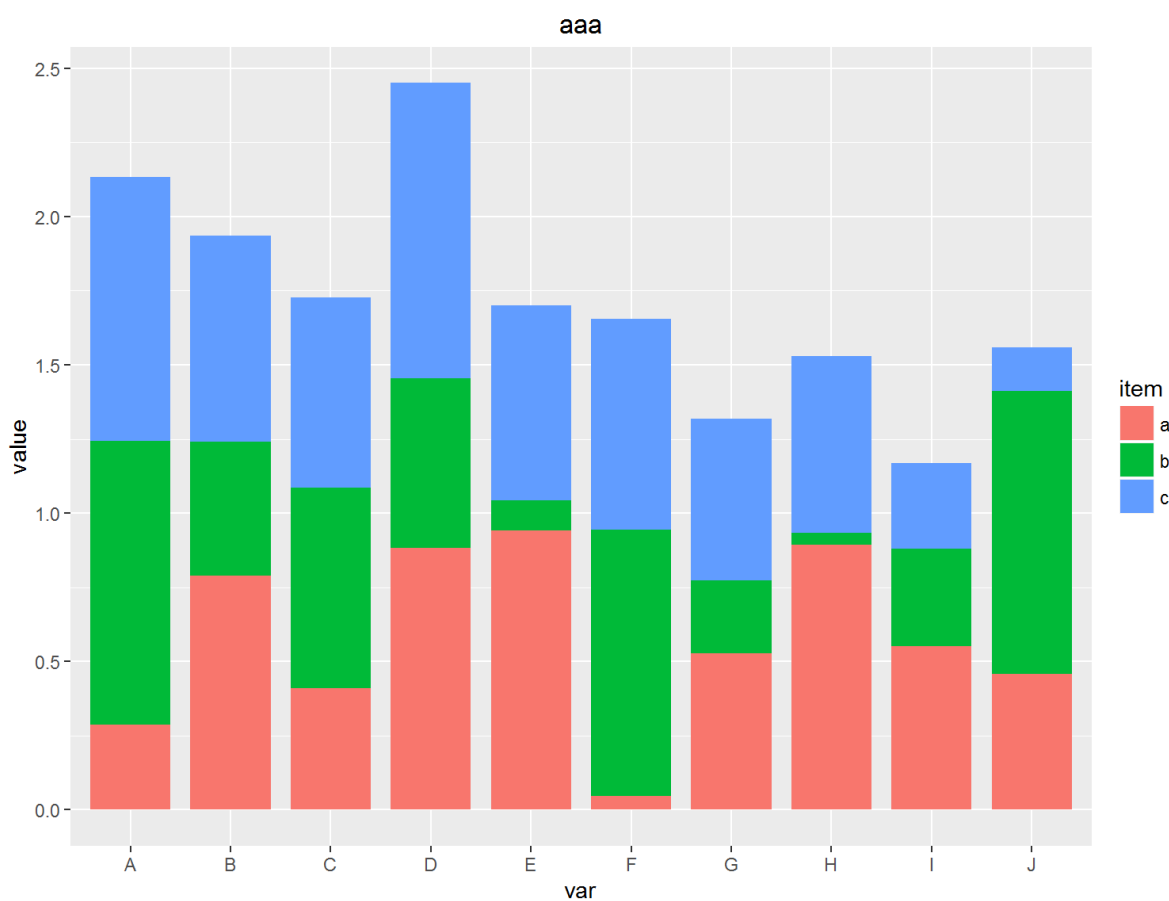
```
ggplot(df1,aes(var,value))+  
  geom_bar(aes(fill=item),stat = "identity",position="dodge",w  
  labs(title="aaa")
```



2.堆积柱形图

堆积柱形图的特点是在不同类别中，不同变量的数据是纵向堆积的，因此需要图形参数中设置position="stack"

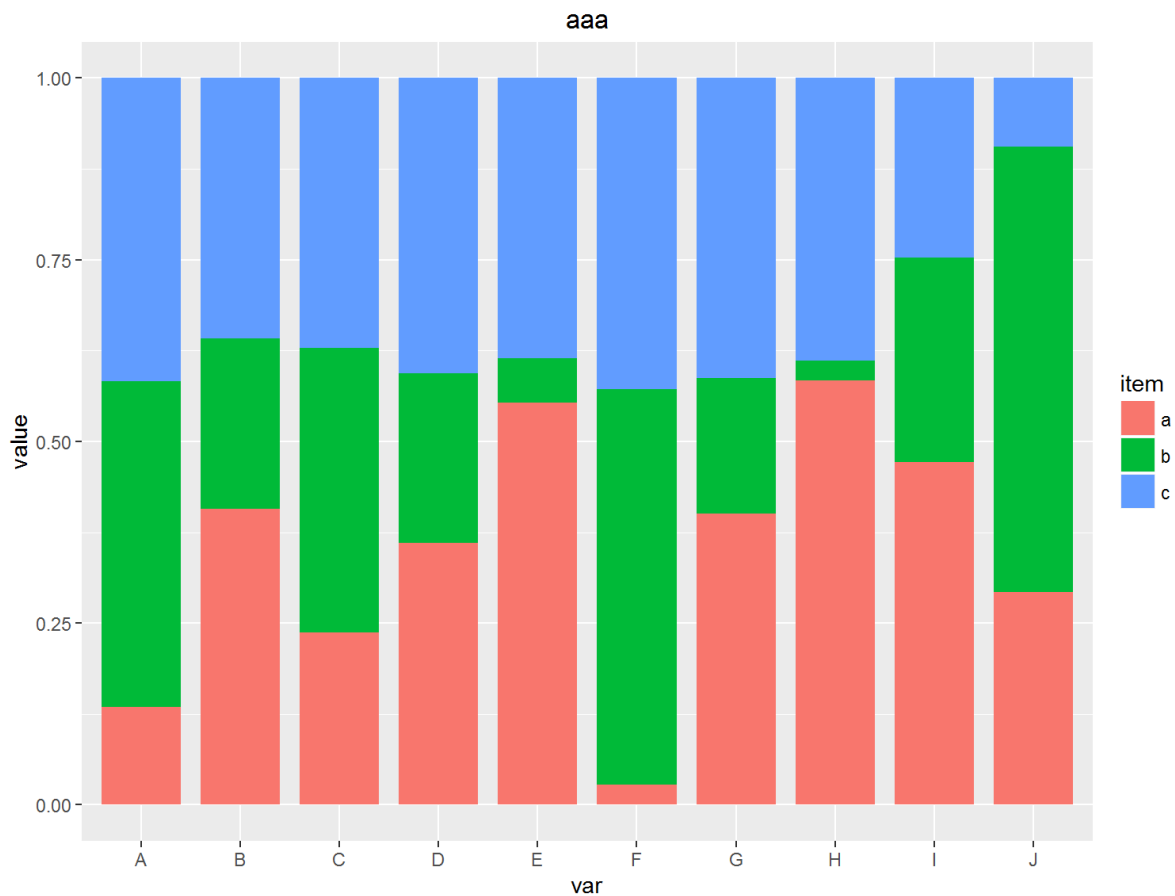
```
ggplot(df1, aes(var, value)) +  
  geom_bar(aes(fill=item), stat = "identity", position="stack", width=0.8, color="black",  
  labs(title="aaa")
```



3.百分比堆积柱形图

百分比堆积柱形图的特点是在不同类别中，不同变量的数据转换为百分比后再纵向堆积，因此需要图形参数中设置position="fill"


```
ggplot(df1, aes(var, value)) +  
  geom_bar(aes(fill=item), stat = "identity", position="fill", width=0.8,   
  labs(title="aaa")
```



4. 三维簇状柱形图

ggplot2中，所有作图均为平面图形，不推荐使用3D的呈现形式，因此此处不考虑作图。

5. 三维堆积柱形图

ggplot2中，所有作图均为平面图形，不推荐使用3D的呈现形式，因此此处不考虑作图。

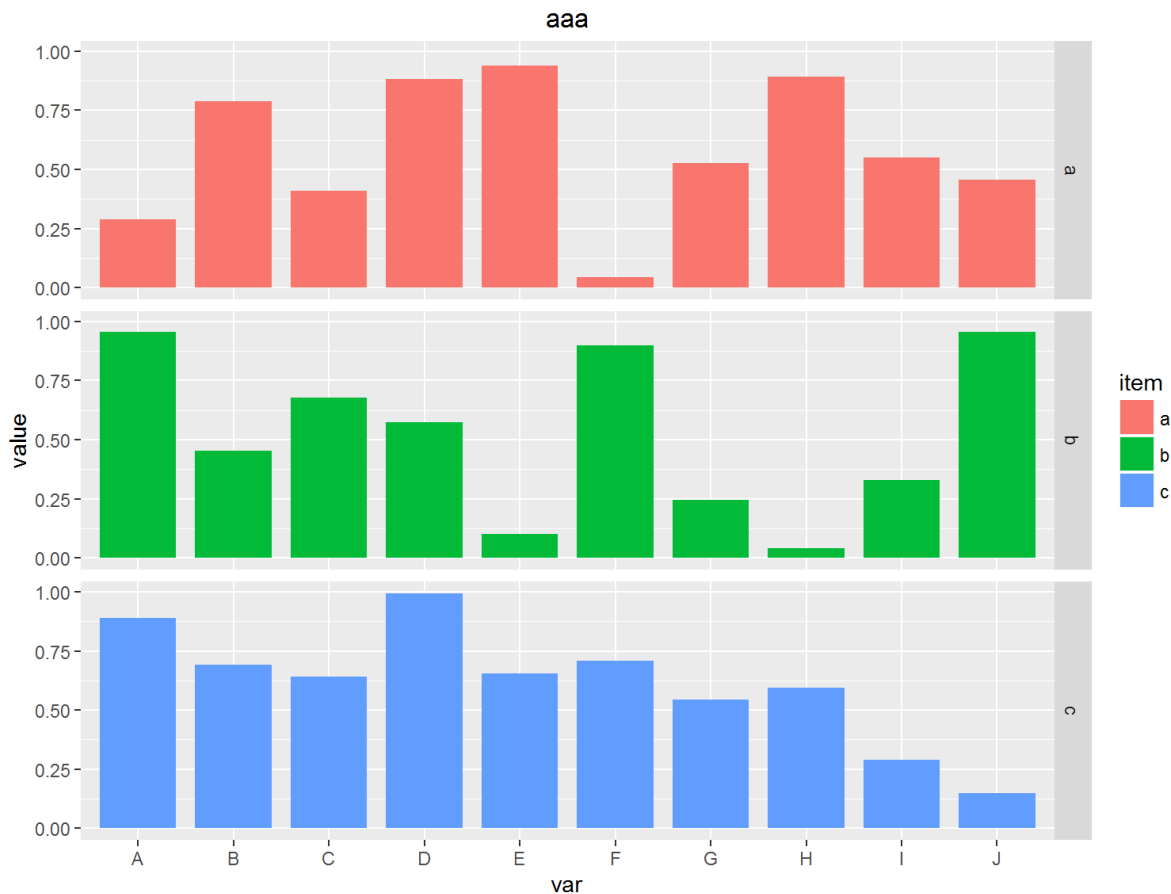
6.三维百分比堆积柱形图

ggplot2中，所有作图均为平面图形，不推荐使用3D的呈现形式，因此此处不考虑作图。

7.三维柱形图——用分面替代

三维柱形图的特定是利用x轴和y轴分别表示两个分类变量，z轴表示具体数值，但是由于3D的遮挡性，所以通常可视化效果欠佳。在ggplot2中，可以用分面（facet_grid）解决这一问题。

```
ggplot(df1, aes(var, value)) +  
  geom_bar(aes(fill=item), stat = "identity", width=0.8) +  
  facet_grid(item~.) +  
  labs(title="aaa")
```



折线图

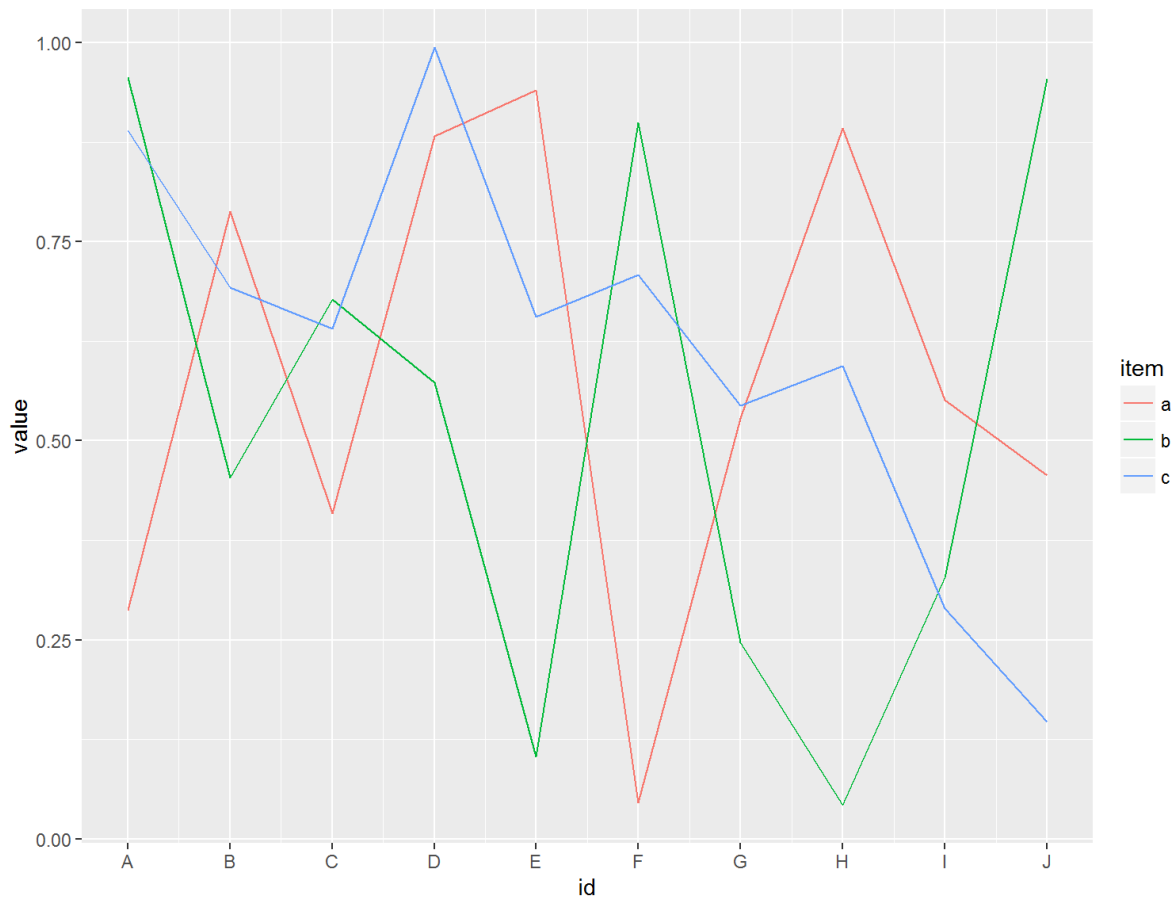
在ggplot2中，自变量为分类变量的时候是不能使用折线图的，因此，需要将分类变量先转换/标记为数字（一开始df中设置id就是这个用处），然后在标签上使用 `scale_x_continuous()` 进行设置后就可以完美解决这一问题。

但是，依然不推荐使用分类变量刻画折线图！！

1.折线图

折线图特点是每个变量先用散点标记，然后相连。因此全部使用默认参数即可。

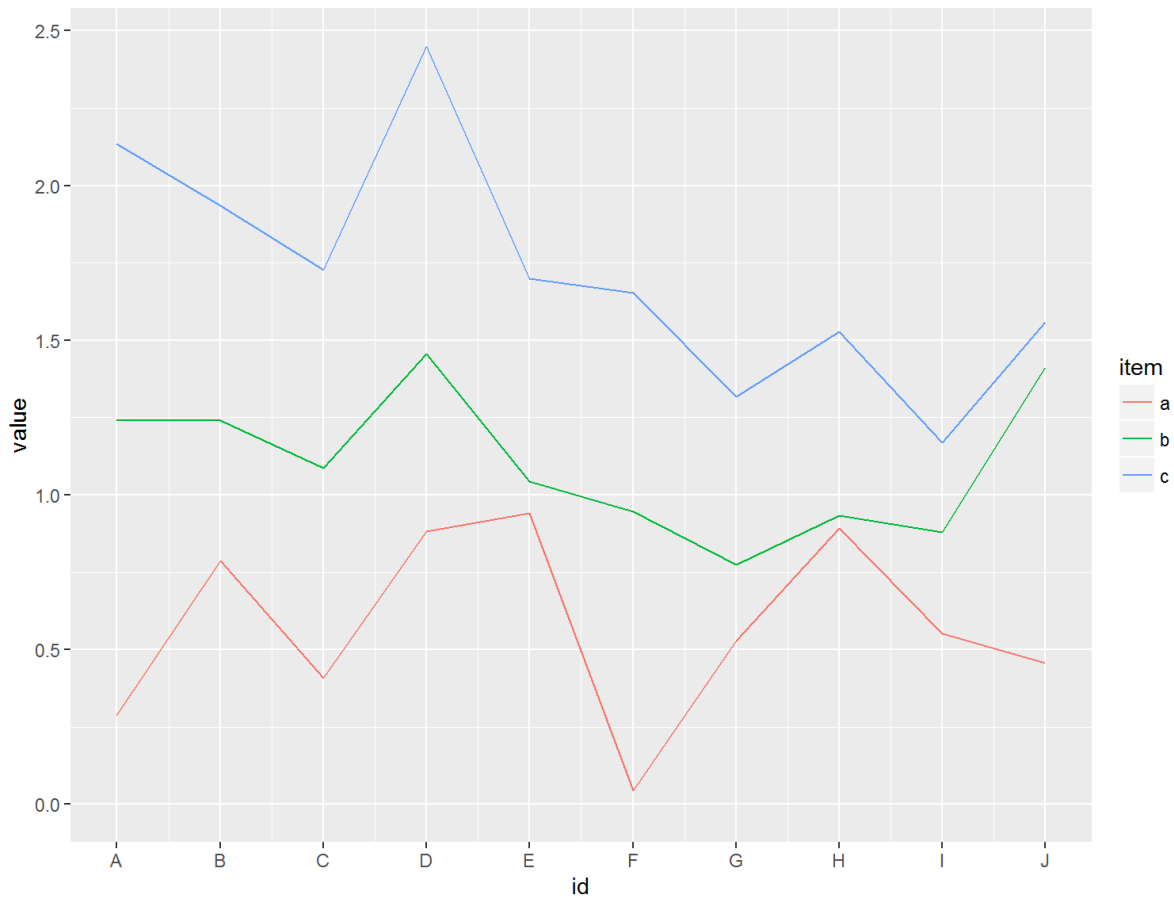
```
ggplot(df1,aes(id,value,colour=item))+
  geom_line()+
  scale_x_continuous(breaks = 1:10,labels = LETTERS[1:10]) #改
```



2.堆积折线图

堆积折线图特点是所有变量上的不同属性是纵向堆积关系，所以只要设置位置参数 `position="stack"` 即可。

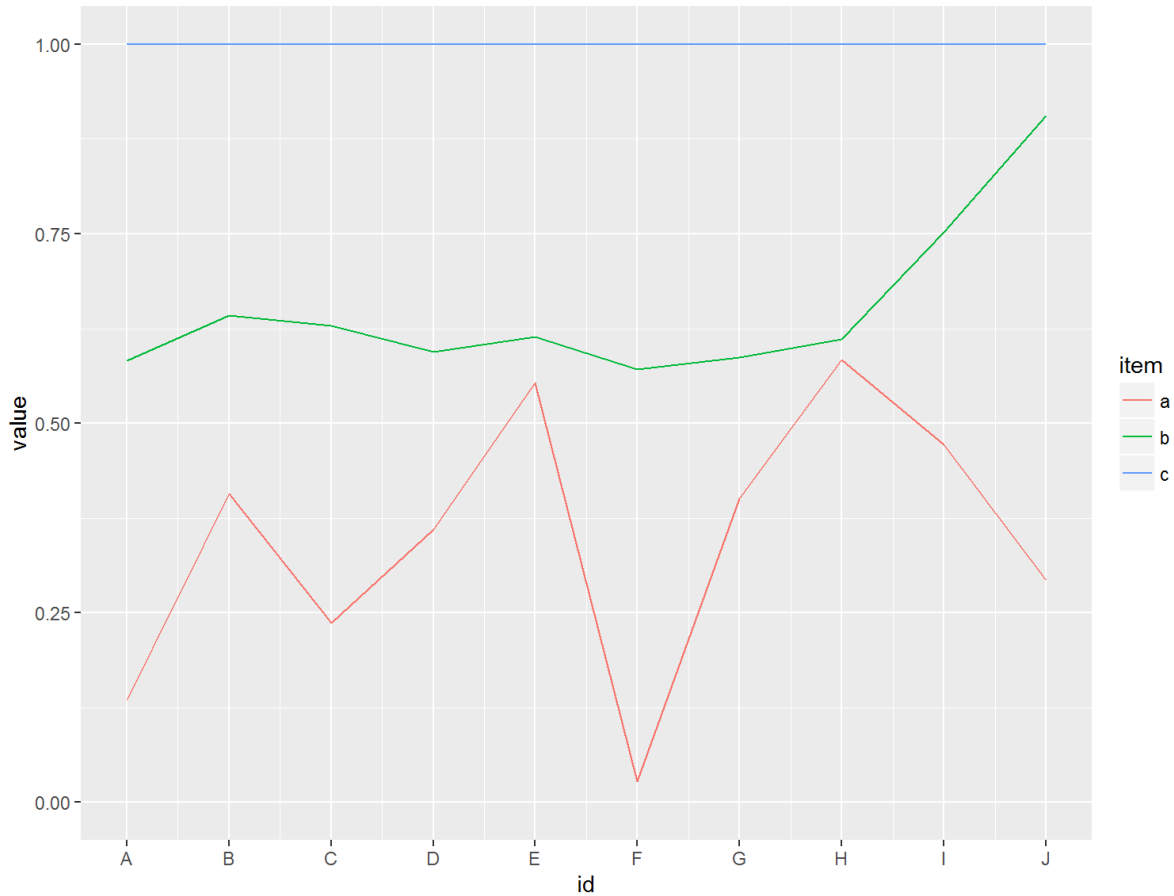
```
ggplot(df1,aes(id,value,colour=item))+
  geom_line(position="stack")+
  scale_x_continuous(breaks = 1:10,labels = LETTERS[1:10])
```



3. 百分比堆积折线图

百分比堆积折线图特点是所有变量上的不同属性转换成百分比后再进行纵向堆积（所以最高那条线必定是直线...），所以只要设置位置参数`position="fill"`即可。

```
ggplot(df1, aes(id, value, colour=item))+  
  geom_line(aes(ymax=0), position="fill")+  
  scale_x_continuous(breaks = 1:10, labels = LETTERS[1:10])
```

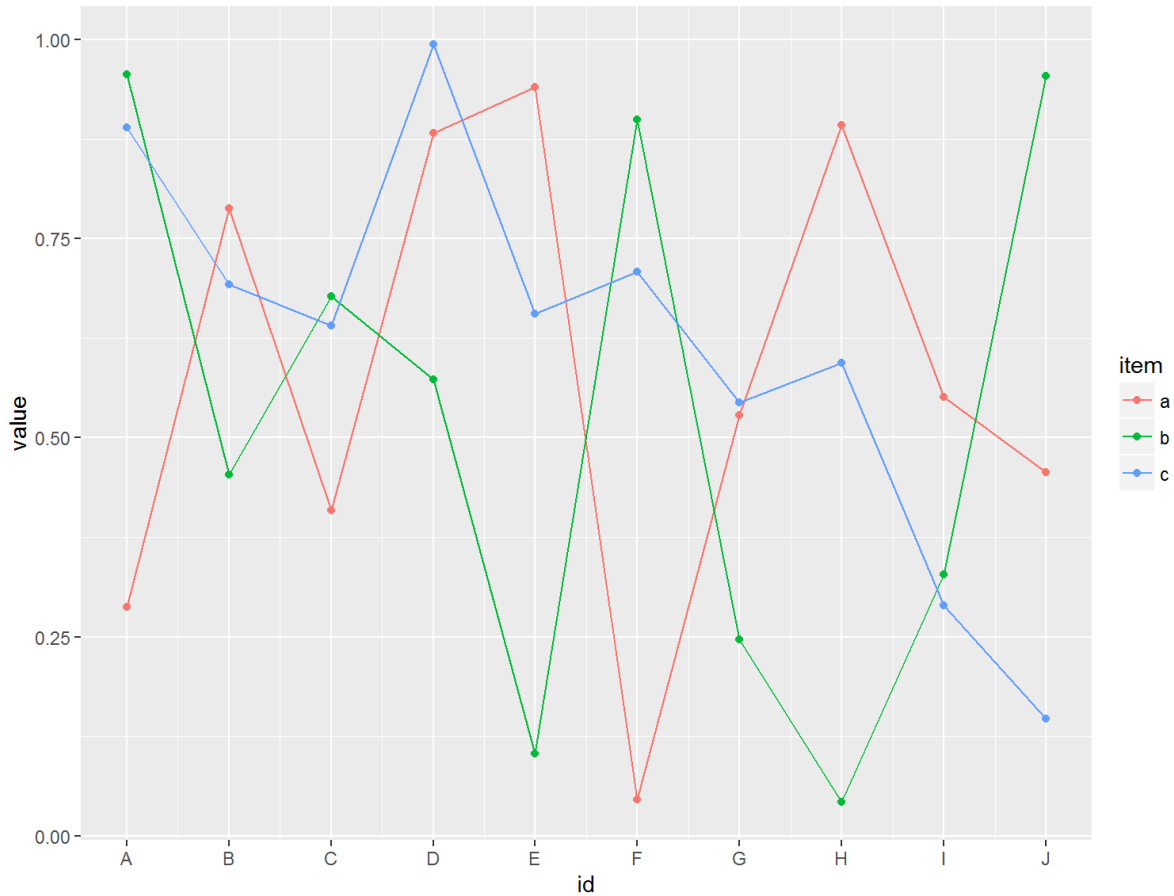


不过，这里需要注意的一点是，在[aes\(\)](#)里必须设置ymax

4.带标记的折线图

带标记的折线图特点是在折线图基础上用了点标记，只要增加散点图就可以实现。

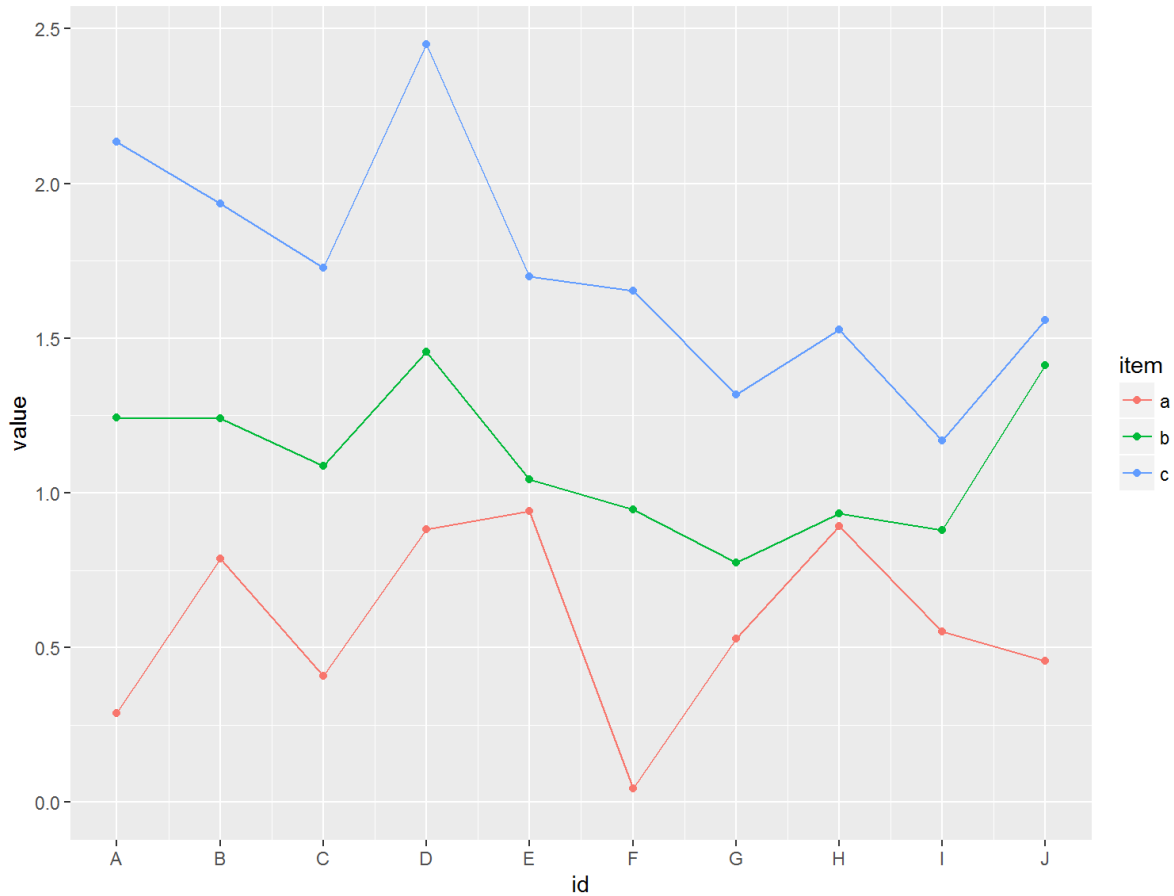
```
ggplot(df1, aes(id, value, colour=item))+  
  geom_line()+  
  geom_point()+  
  scale_x_continuous(breaks = 1:10, labels = LETTERS[1:10],
```



5.带标记的堆积折线图

带标记的堆积折线图特点是在堆积折线图基础上用了点标记，只要增加散点图就可以实现。

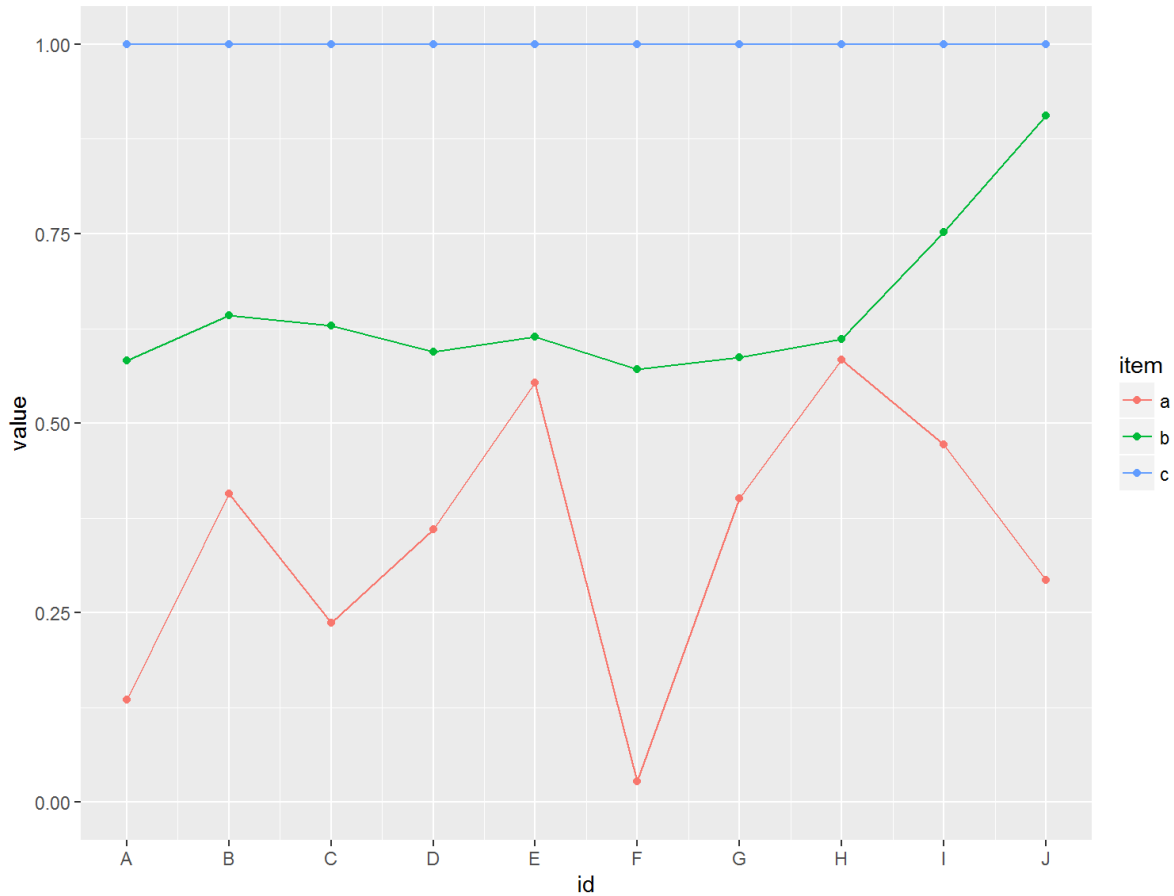
```
ggplot(df1,aes(id,value,colour=item))+
  geom_line(position="stack")+
  geom_point(position="stack")+
  scale_x_continuous(breaks = 1:10,labels = LETTERS[1:10])
```



6.带标记的百分比堆积折线图

带标记的百分比堆积折线图特点是在堆积折线图基础上用了点标记，只要增加散点图就可以实现。

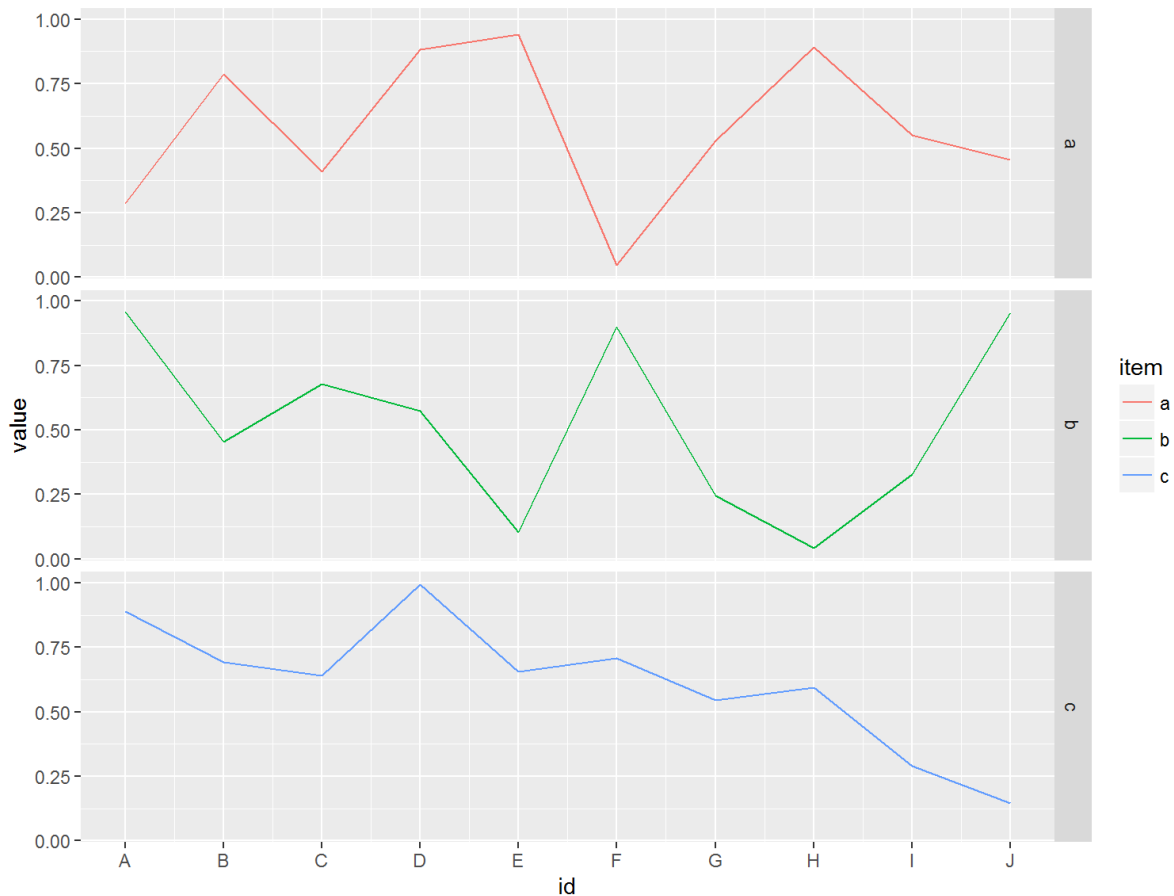
```
ggplot(df1,aes(id,value,colour=item))+
  geom_line(aes(ymax=0),position="fill")+
  geom_point(aes(ymax=0),position="fill")+
  scale_x_continuous(breaks = 1:10,labels = LETTERS[1:10])
```

7.三维折线图_用facet代替

三维折线图特点是利用x轴和y轴分别表示两个分类变量，z轴表示具体数值，但是由于3D的遮挡性，所以通常可视化效果欠佳。在ggplot2中，可以用分面（facet_grid）解决这一问题。

```
ggplot(df1, aes(id, value, colour=item))+  
  facet_grid(item~.)+  
  geom_line()+  
  scale_x_continuous(breaks = 1:10, labels = LETTERS[1:10])
```



饼图

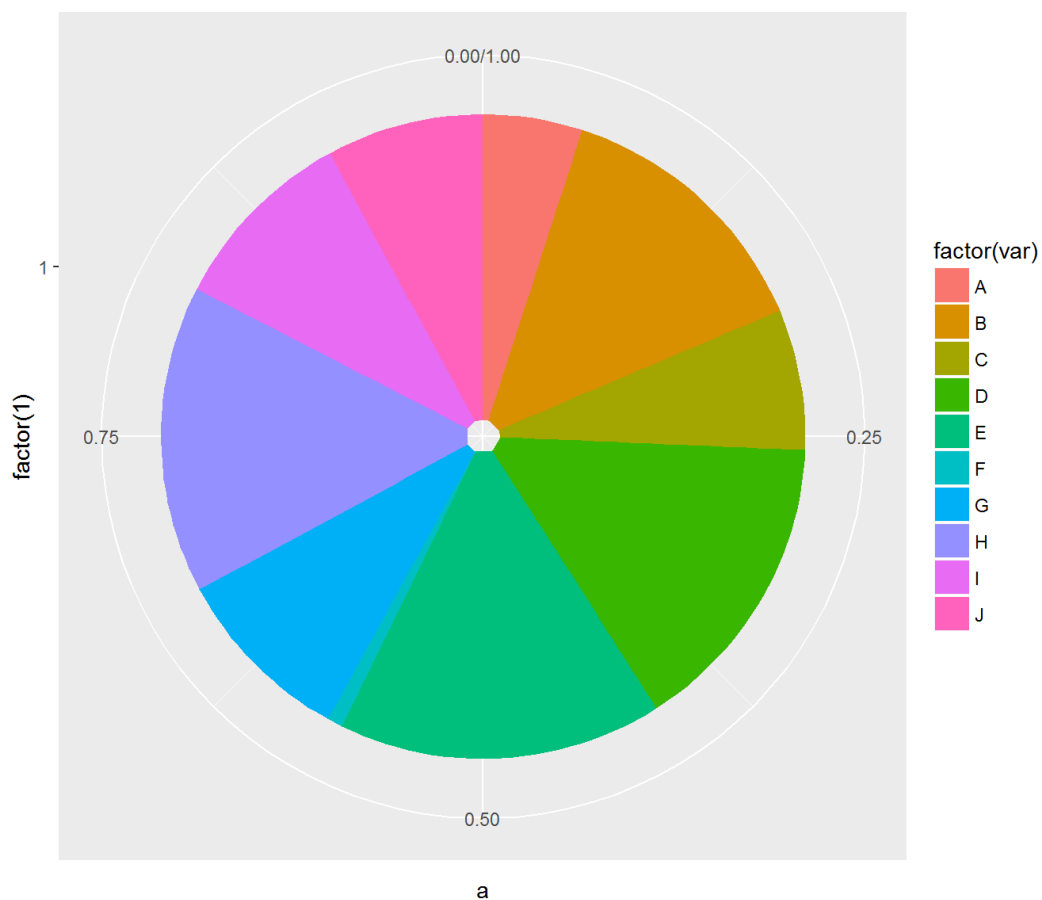
在ggplot2中，饼图和柱形图关系密切，其实就是一般的柱形图进行直角坐标轴到极坐标轴的转换即可。函数为`coord_polar()` 但是，同样的，大多数人诟病饼图圆形面积不如柱形图的高低直观，因此并不推荐使用饼图。

1.饼图

饼图的特点是使用圆形的面积大小表示数据的占比份额。因此一般的饼图只要`x`~`y`~`z`极坐标变换即可。

```
ggplot(df,aes(x = factor(1),a,fill=factor(var)))+  
  geom_bar(stat="identity",position="fill")+
```

```
coord_polar(theta="y")
```



需要注意的是，因为是对y轴进行转换，因此x轴长短需要一致，统一设置为某一值即可（本文使用`x=factor(1)`）。另外，需要注意的是，饼图通常是表示百分比，所以需要设置参数`position="fill"`。

2. 三维饼图

三维不做图。

3. 复合饼图

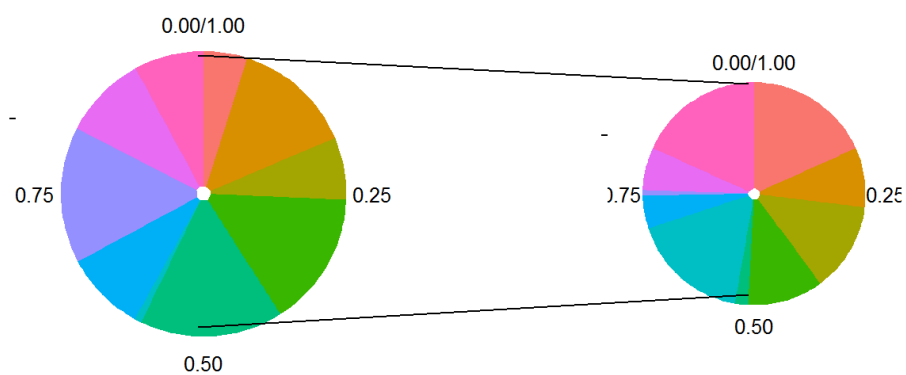
复合饼图的特点是，当分类很多或者有一个分类含有子分类时，可以使用另一个图形将其中的内容显示出来。

```
df_tmp<-data.frame(x=1,y=1) #准备画布所需数据
base <- ggplot(df_tmp,aes(x,y))+
  geom_blank()+
  theme_void()+
  xlim(c(0,2))+
  ylim(c(0,2)) #准备画布
base + annotation_custom(
  grob = ggplotGrob(
    ggplot(df,aes(x = "",a,fill=factor(var)))+
    geom_bar(stat="identity",position="fill",
             show.legend = F)+
    labs(x=NULL,y=NULL)+
    coord_polar(theta="y")+
    theme_classic()),
  xmin =0,xmax=1,
  ymin=0.5,ymax = 1.5)+ # 图形1
  annotation_custom(
    grob = ggplotGrob(
      ggplot(df,aes(x = "",b,fill=factor(var)))+
      geom_bar(stat="identity",
               position="fill",
               show.legend = F)+
      labs(x=NULL,y=NULL)+
      coord_polar(theta="y")+
      theme_classic()),
    xmin =1.1,xmax=1.9,
    ymin=0.6,ymax = 1.4)+ #图形2
  annotate("segment",x=0.5,xend=1.5,
```

```

y=0.69,yend=0.77)+ #线条1
annotate("segment",x=0.5,xend=1.5,
y=1.35,yend=1.28) #线条2

```



有心的读者应该可以看得出来，目前这一版本只是一个demo，仍然有很多需要改进的地方。但是，实际上用这个图只是为了抛砖引玉，只要你有想象力和创造力，ggplot2可以很自由的帮你实现很多图表。

3.复合条饼图

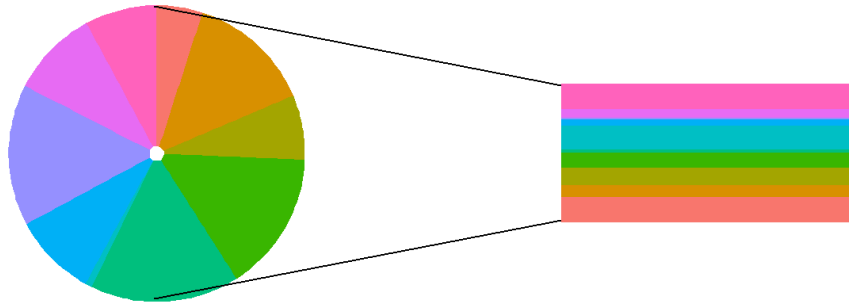
复合条饼图的特点是，当分类很多或者有一个分类含有子分类时，可以使用另一个图形将其中的内容显示出来。

```

base <- ggplot(df_tmp,aes(x,y))+geom_blank()+theme_void()+
  xlim(c(0,2))+ylim(c(0,2))
base + annotation_custom(
  grob = ggplotGrob(
    ggplot(df,aes(x = "",a,fill=factor(var)))+
      geom_bar(stat="identity",
        position="fill",
        show.legend = F)+
      labs(x=NULL,y=NULL)+
      coord_polar(theta="y")+theme_void()),
  xmin =0,xmax=1,
  ymin=0.5,ymax = 1.5)+
  annotation_custom(
    grob = ggplotGrob(
      ggplot(df,aes(x = "",b,fill=factor(var)))+
        geom_bar(stat="identity",
          position="fill",
          show.legend = F)+
        labs(x=NULL,y=NULL)+theme_void()),
    xmin =1.2,xmax=1.8,ymin=0.8,ymax = 1.2)+
  annotate("segment",x=0.5,xend=1.24,y=0.64,yend=0.84)+
  annotate("segment",x=0.5,xend=1.24,y=1.38,yend=1.18)+
  labs(title="aaa")

```

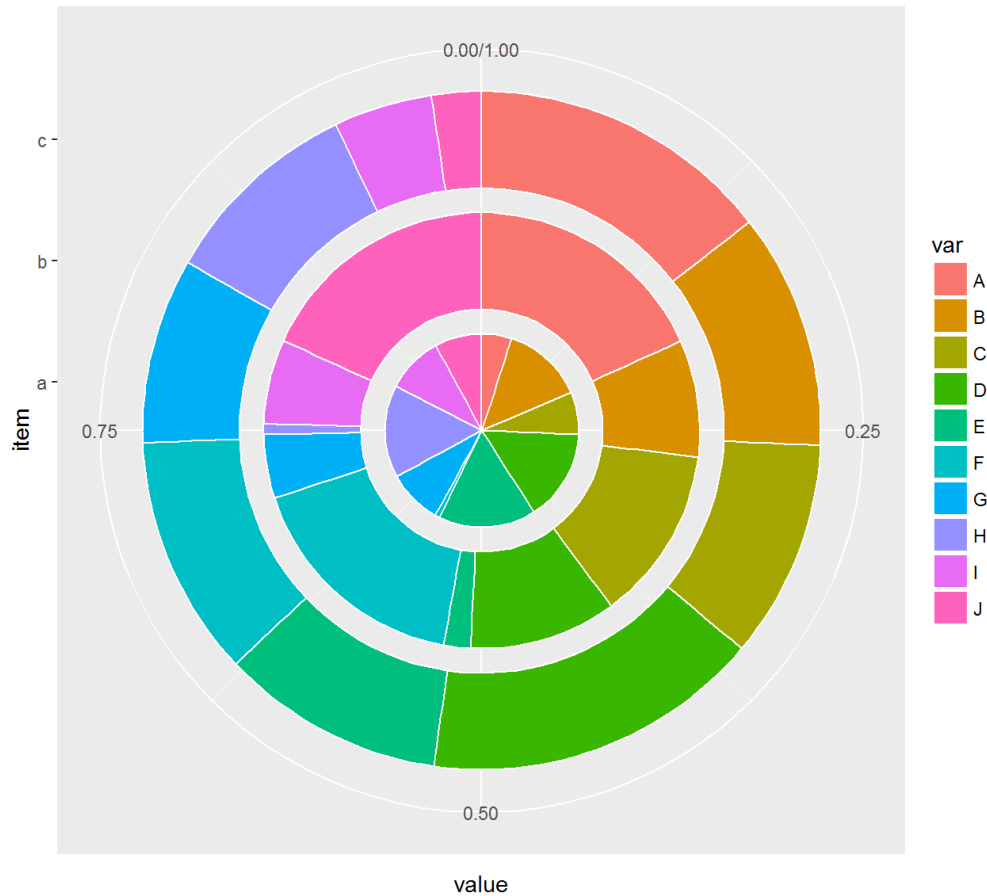
aaa



4.圆环图

圆环图的特点是每一个圆环表示一个变量，其中不同颜色表示不同的属性。

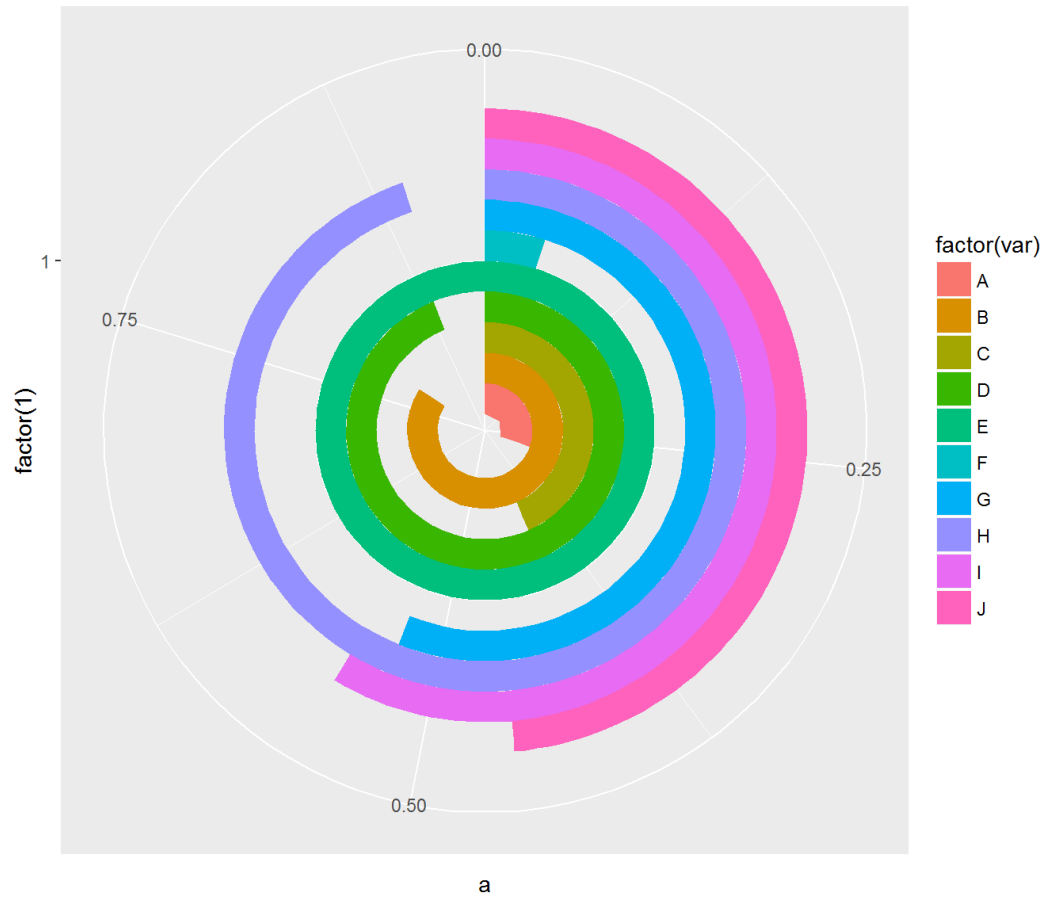
```
ggplot(df1,aes(x = item,value,fill=var))+  
  geom_bar(stat="identity",position="fill",width=0.8,colour="w")+  
  coord_polar(theta="y")
```



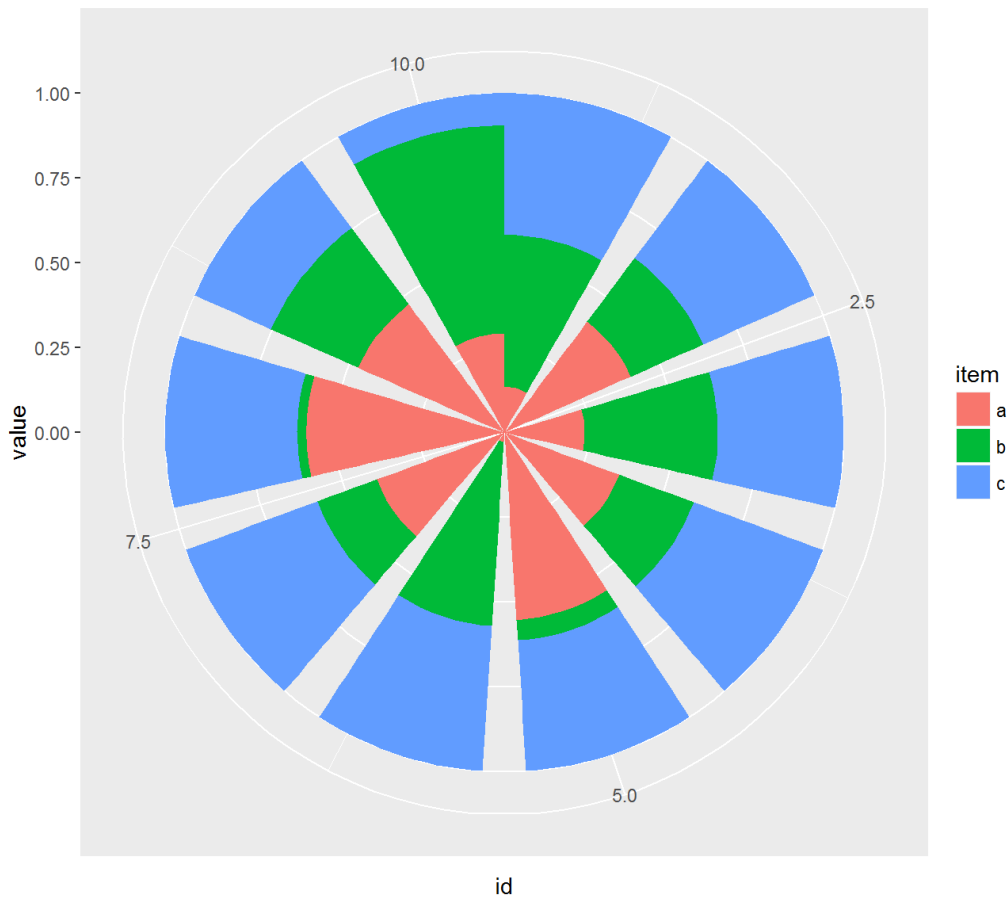
5.其他圆形图

这里不得不说一下，ggplot2的代码demo还是很赞的，用coord_polar可以创造出更多图形.

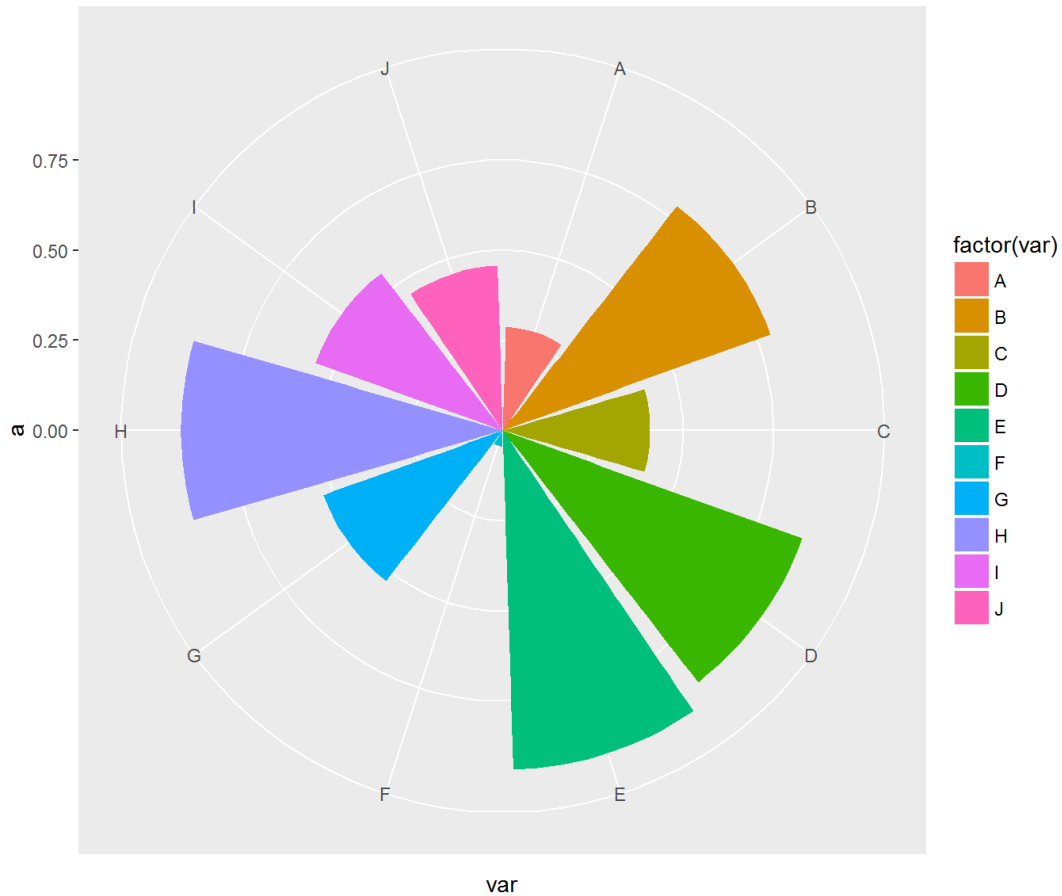
```
ggplot(df,aes(x = factor(1),a,fill=factor(var)))+  
  geom_bar(stat="identity",position="dodge")+  
  coord_polar(theta="y")
```

```
ggplot(df1, aes(x = id, value, fill=item))+
  geom_bar(stat="identity", position="fill", width=0.8)+
  coord_polar(theta="x")
```



```
ggplot(df, aes(x = var, a, fill=factor(var)))+  
  geom_bar(stat="identity")+  
  coord_polar(theta="x")
```



条形图

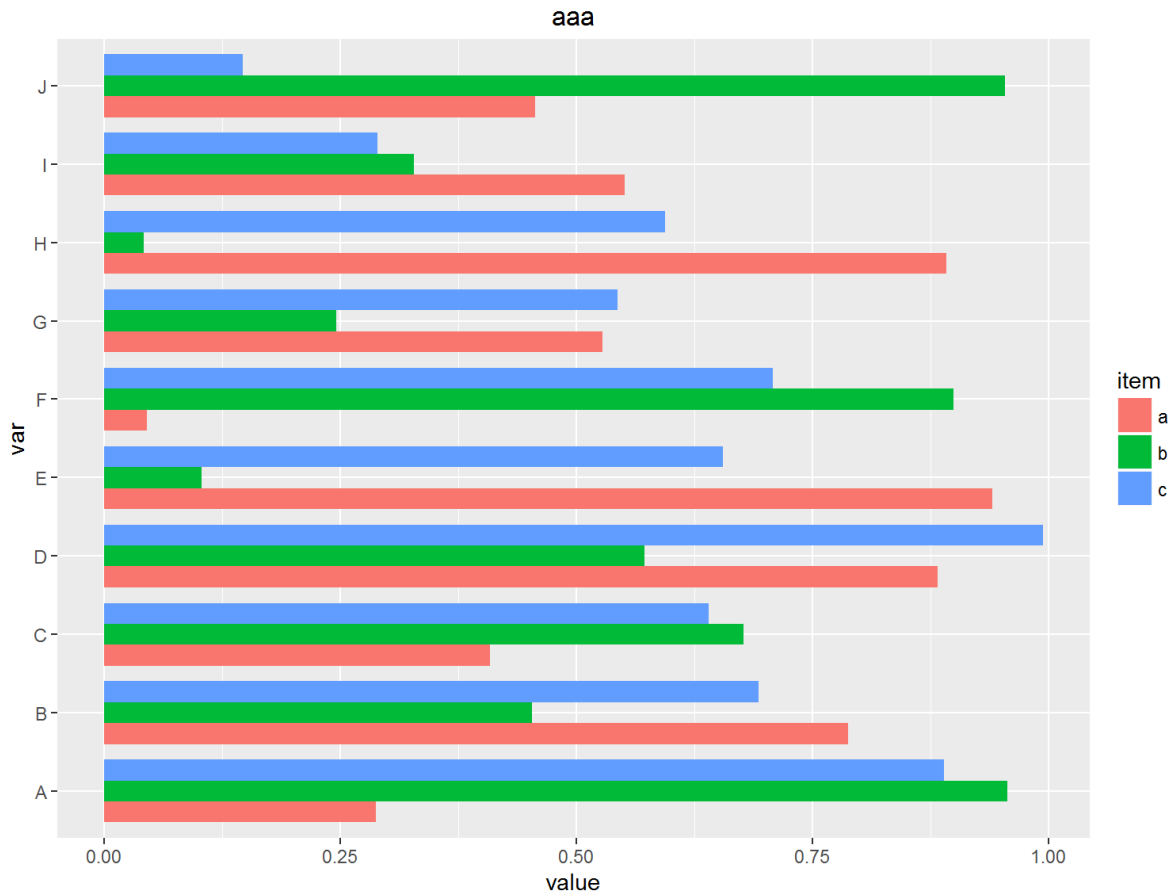
条形图就是横过来的柱形图，所以，只要一个`coord_flip()`即可。需要注意的是，柱形图通常用于分类较少的情况下，分类较多且分类的名字特别长的时候，建议使用条形图。

1.簇状条形图

簇状条形图就是簇状柱形图转一下。

```
ggplot(df1, aes(var, value)) +
  geom_bar(aes(fill=item), stat = "identity", position="dodge", w
```

```
labs(title="aaa")+  
coord_flip()
```



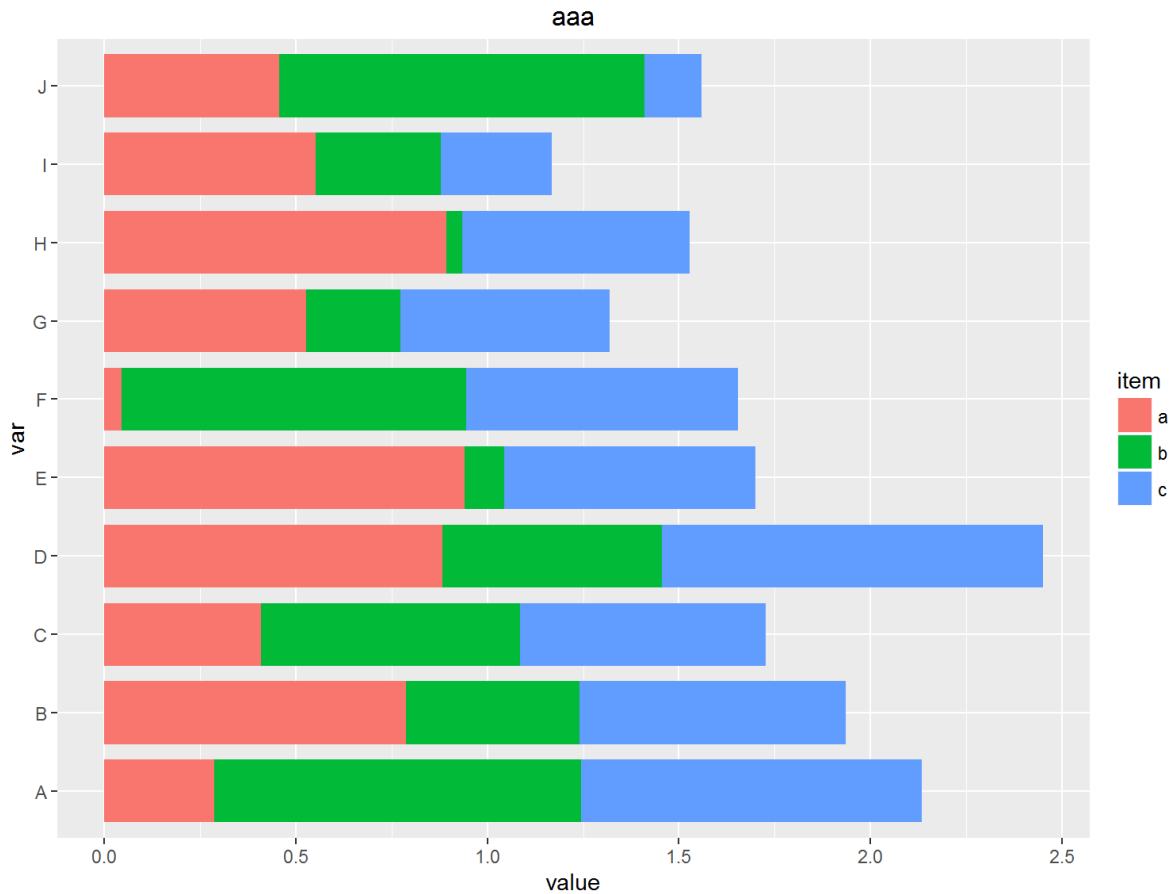
有心的小伙伴应该注意到了，excel做条形图的时候常常需要逆序坐标轴，但是这里没有做这个处理，就当做一个小小的BUG吧，交给读者自行处理。

2.堆积条形图

堆积条形图就是堆积柱形图转一下。

```
ggplot(df1,aes(var,value))+  
  geom_bar(aes(fill=item),stat = "identity",position="stack",w
```

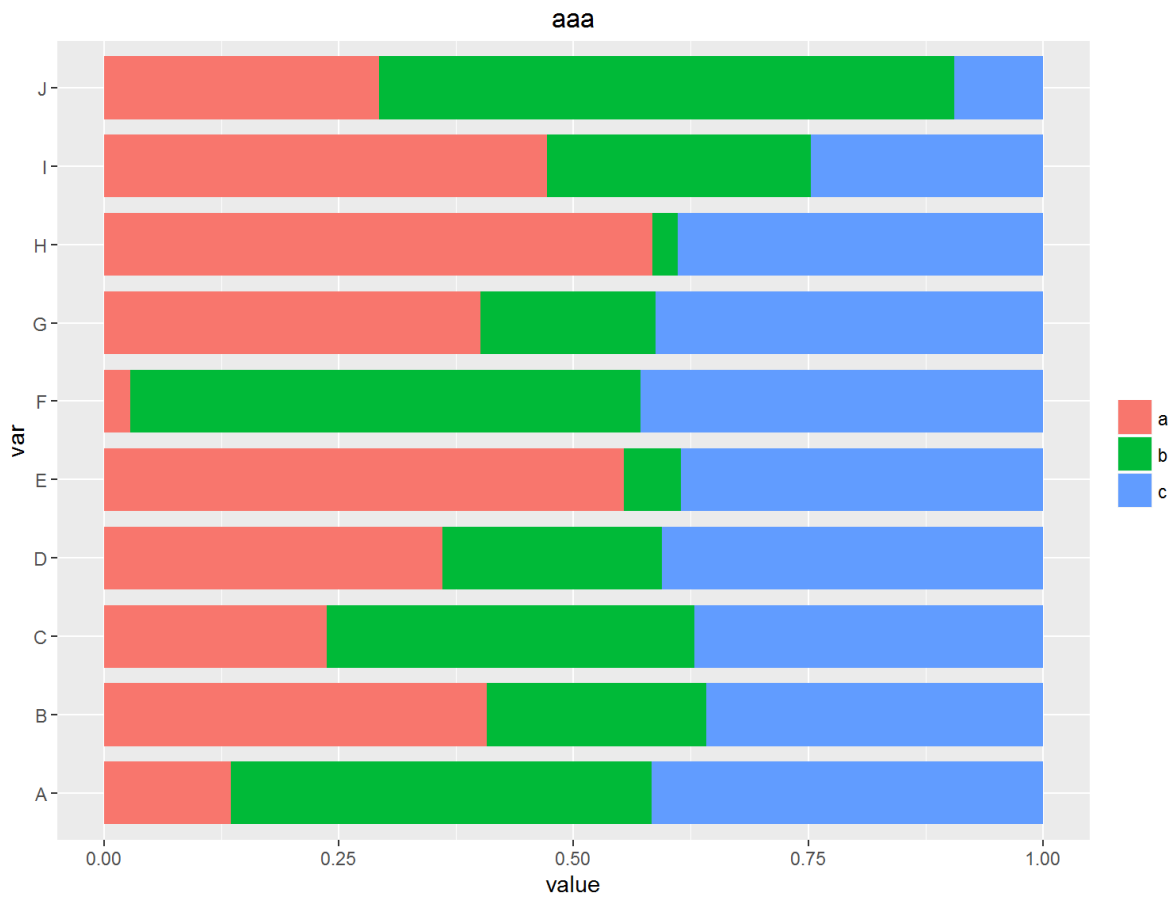
```
labs(title="aaa")+
coord_flip()
```



3.百分比堆积条形图

百分比堆积条形图就是百分比堆积柱形图转一下。

```
ggplot(df1,aes(var,value))+
  geom_bar(aes(fill=item),stat = "identity",position="fj      wi
  labs(title="aaa",fill="")+
  coord_flip()
```



4.三维簇状条形图

3D就不做了

5.三维堆积条形图

3D就不做了

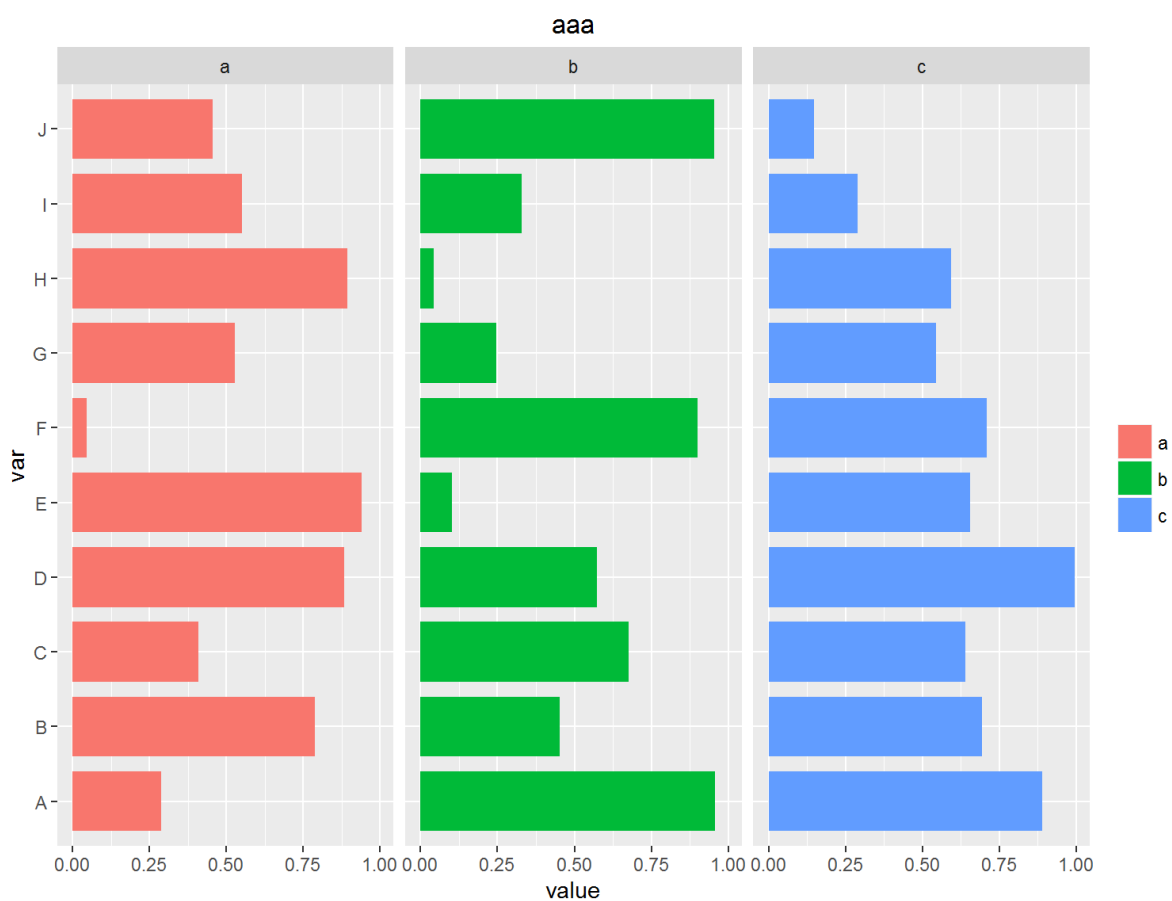
6.三维百分比条形图

3D就不做了

7.分面来代替3D

在柱形图里可以使用分面来代替3D柱形图，条形图一样可以

```
ggplot(df1, aes(var, value)) +  
  geom_bar(aes(fill=item), stat = "identity", position="dodge", width=0.8) +  
  labs(title="aaa", fill="") +  
  coord_flip() +  
  facet_grid(.~item)
```



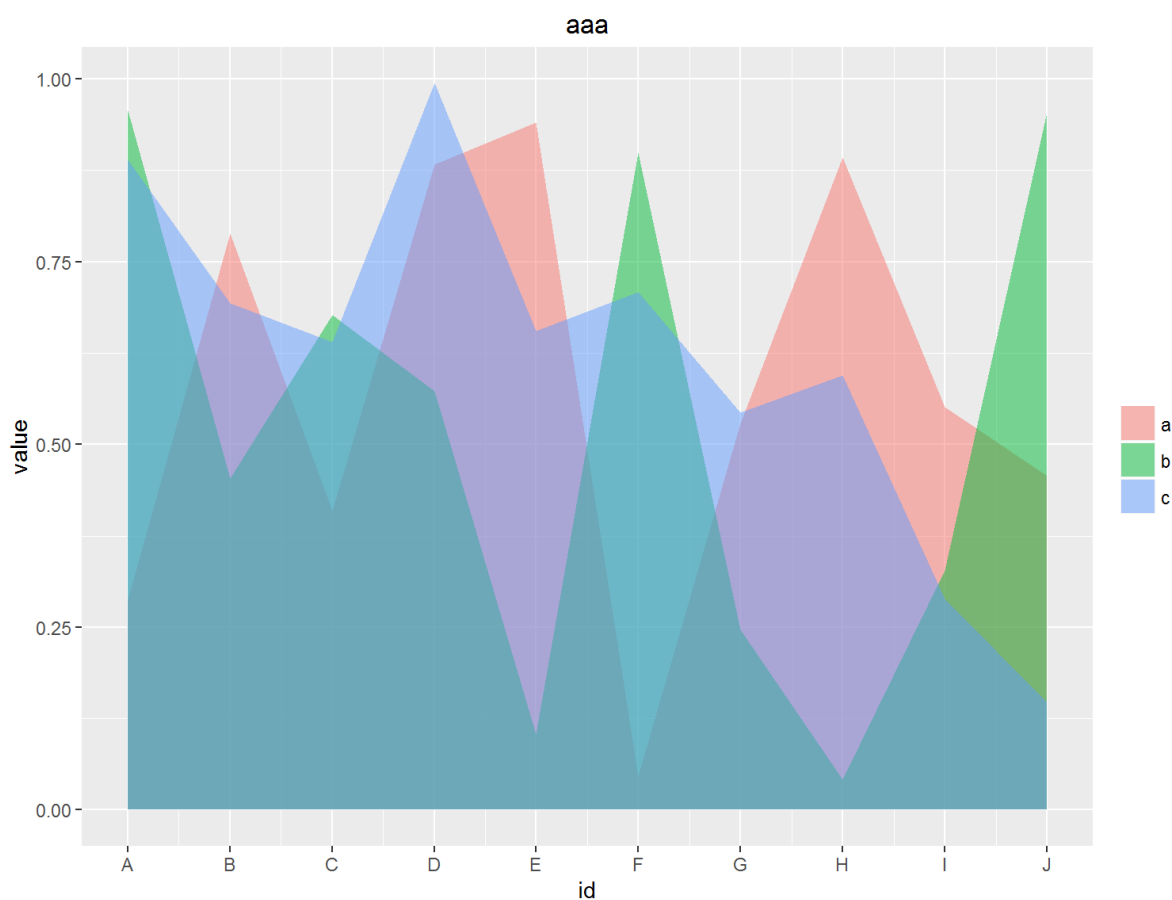
面积图

面积图就是一条折线图，然后用颜色把下面的区域全部覆盖了。通常情况下，当一条折线太单调的时候，使用面积图是不错的选择。

1.面积图

面积图的特点就是会覆盖所有该数据以下的区域。

```
ggplot(df1, aes(id, value)) +  
  geom_area(aes(fill=item), position=position_dodge(width = 0),  
            alpha=0.5) +  
  labs(title="aaa", fill="") +  
  scale_x_continuous(breaks = 1:10, labels = LETTERS[1:10])
```

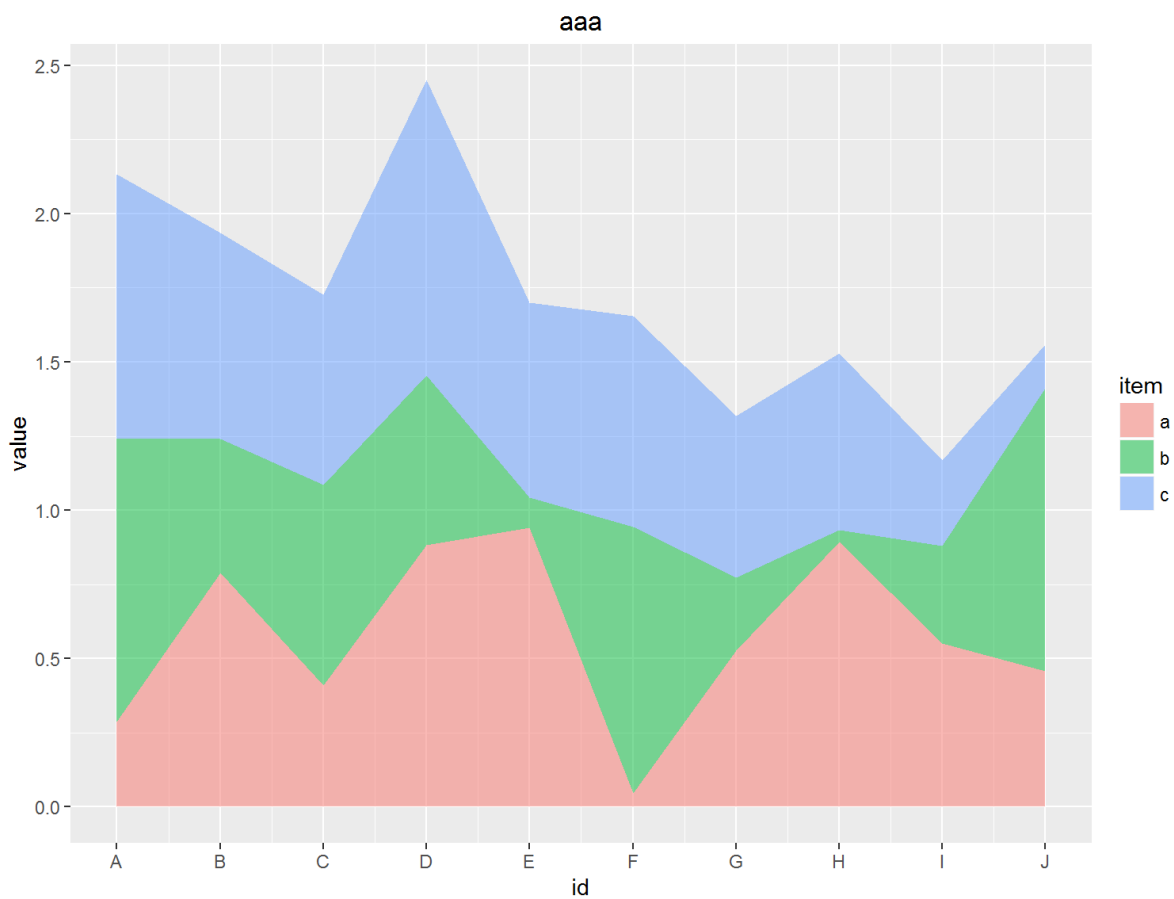


这里需要注意的是，位置函数需要特别定义`position=position_dodge(width = 0)`。另外，多分类的时候，应当设置透明度，如`alpha=0.5`。

2.堆积面积图

堆积面积图特点是会一层一层覆盖上去

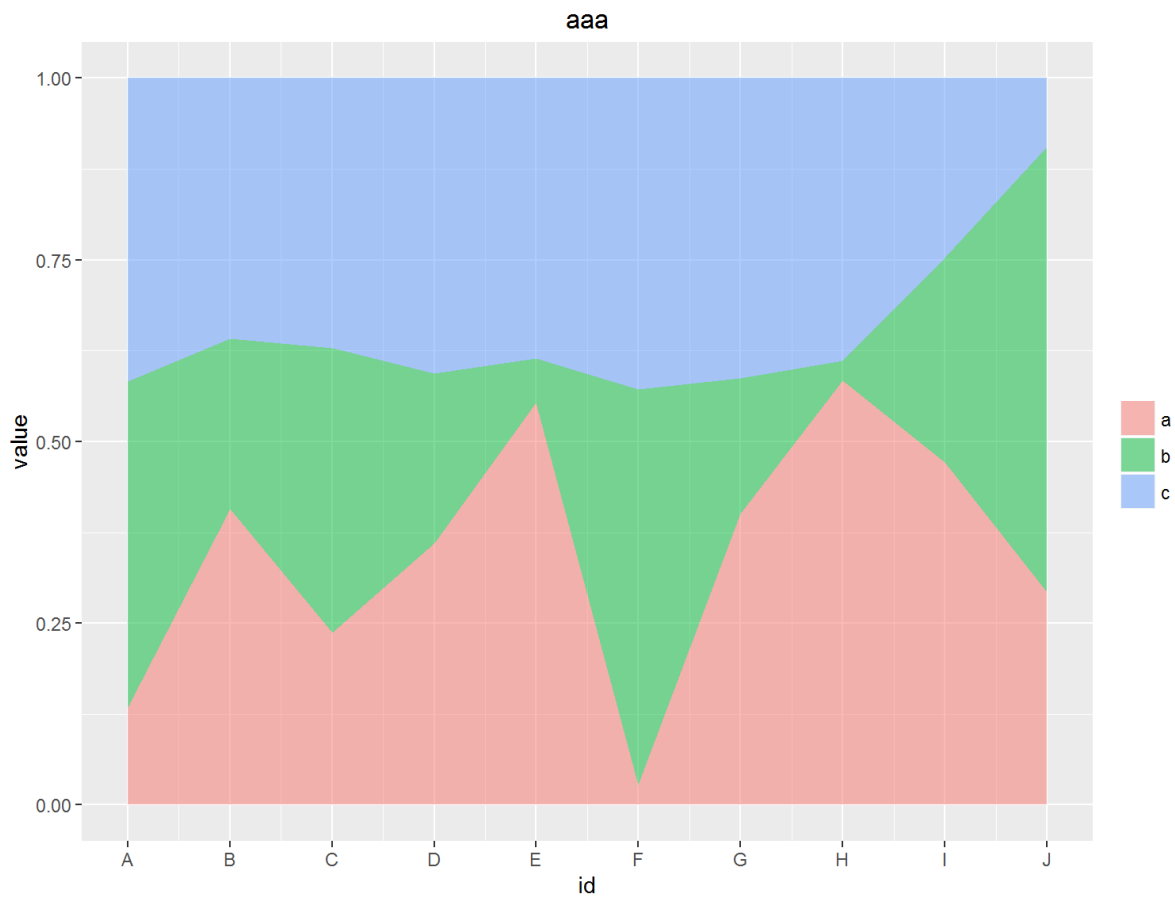
```
ggplot(df1, aes(id, value)) +  
  geom_area(aes(fill=item), alpha=0.5) +  
  labs(title="aaa") +  
  scale_x_continuous(breaks = 1:10, labels = LETTERS[1:10])
```



3.百分比堆积面积图

百分比堆积面积图特点是先把数据转换为百分比，然后一层一层覆盖上去

```
ggplot(df1,aes(id,value))+
  geom_area(aes(fill=item),position="fill",alpha=0.5)+
  labs(title="aaa",fill="")+
  scale_x_continuous(breaks = 1:10,labels = LETTERS[1:10])
```



4.三维面积图

3D不做图

5.三维堆积面积图

3D不做图

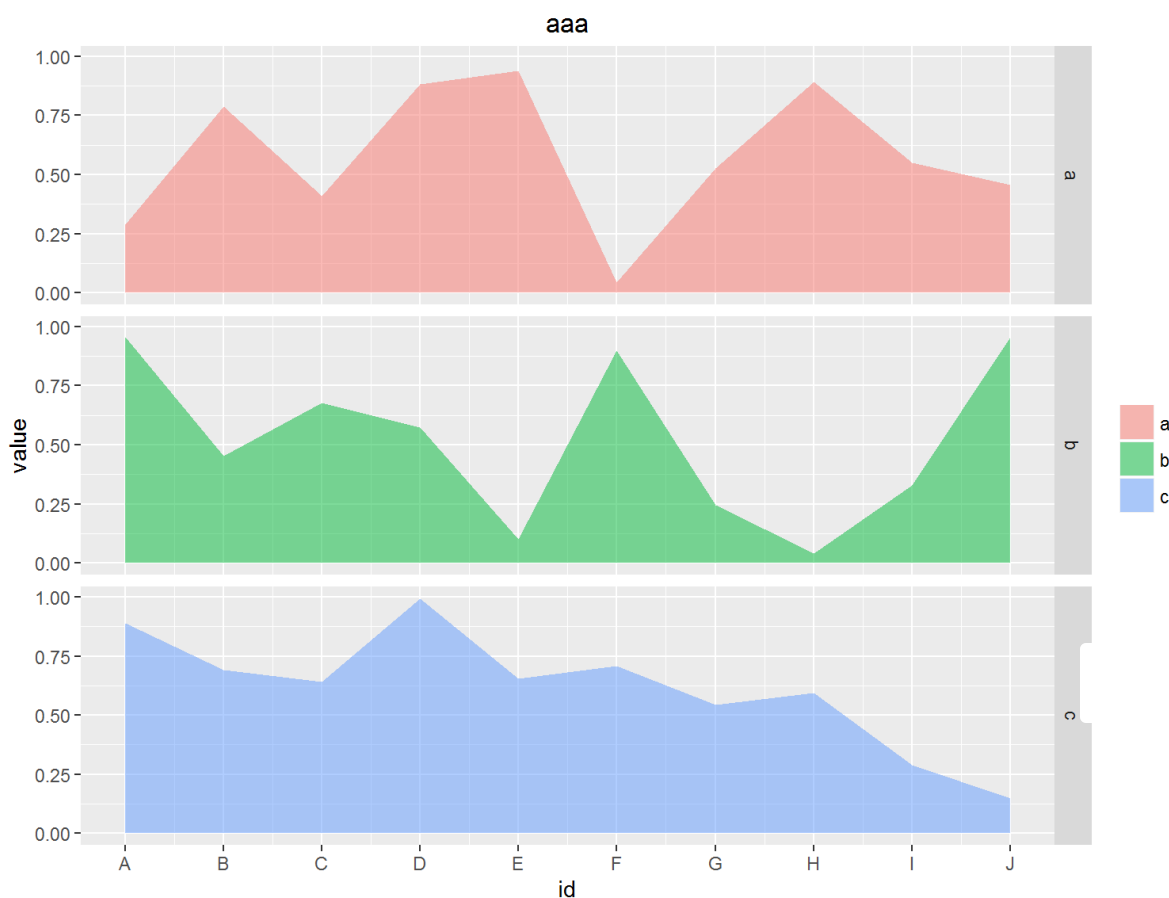
6.三维百分比堆积面积图

3D不做图

7.分面来代替3D

如果觉得面积图用透明度也一样不是很清楚的话，那不妨用分面。

```
ggplot(df1, aes(id, value)) +  
  geom_area(aes(fill=item), position="stack", alpha=0.5) +  
  labs(title="aaa", fill="")+  
  scale_x_continuous(breaks = 1:10, labels = LETTERS[1:10]) +  
  facet_grid(item~.)
```



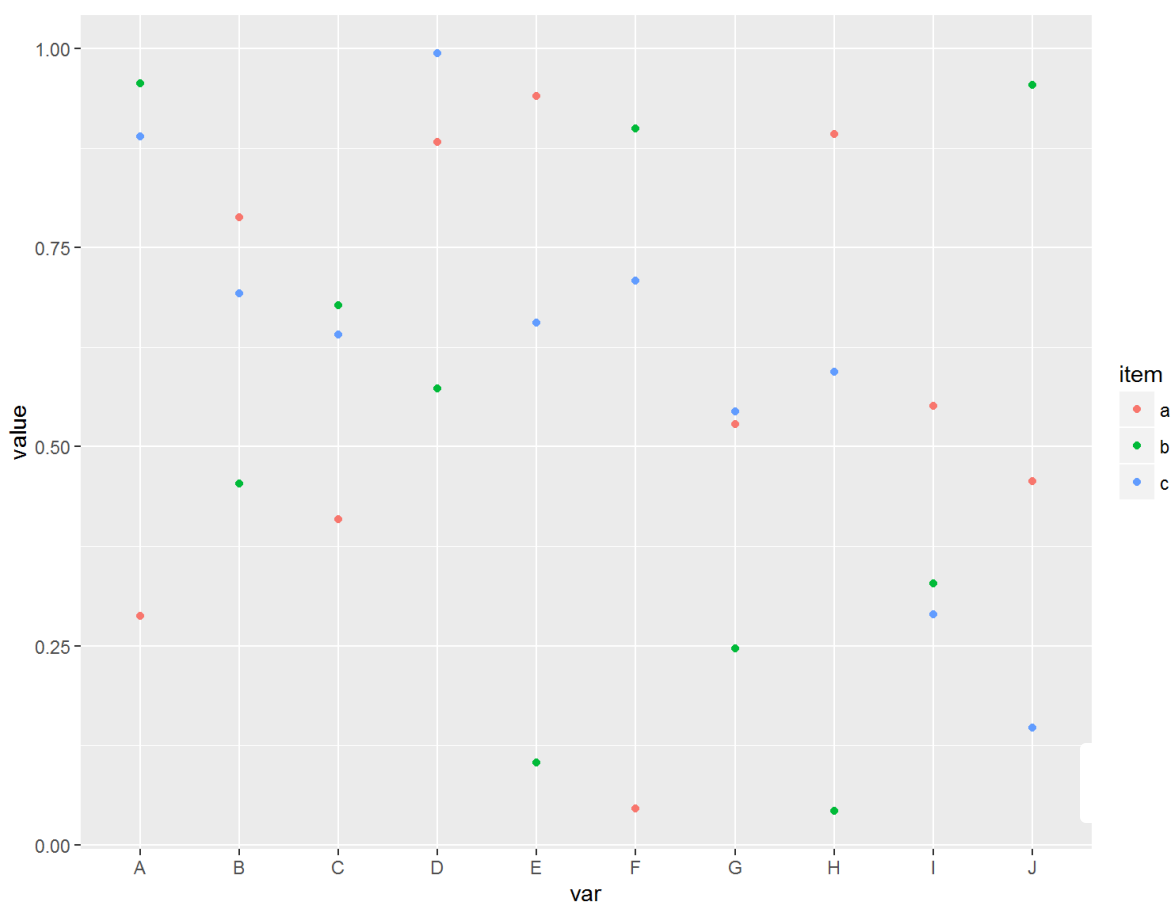
XY散点图

怎么说呢，散点图基本上来说是所有图形的基础图形，所以一直以来对散点图还是有很深的感情。说到散点图本身，其实就是知道x和y坐标，然后把点放上去的一种图形。

1.散点图

散点图的特点是，已知x和y的数值，然后把点放上去。

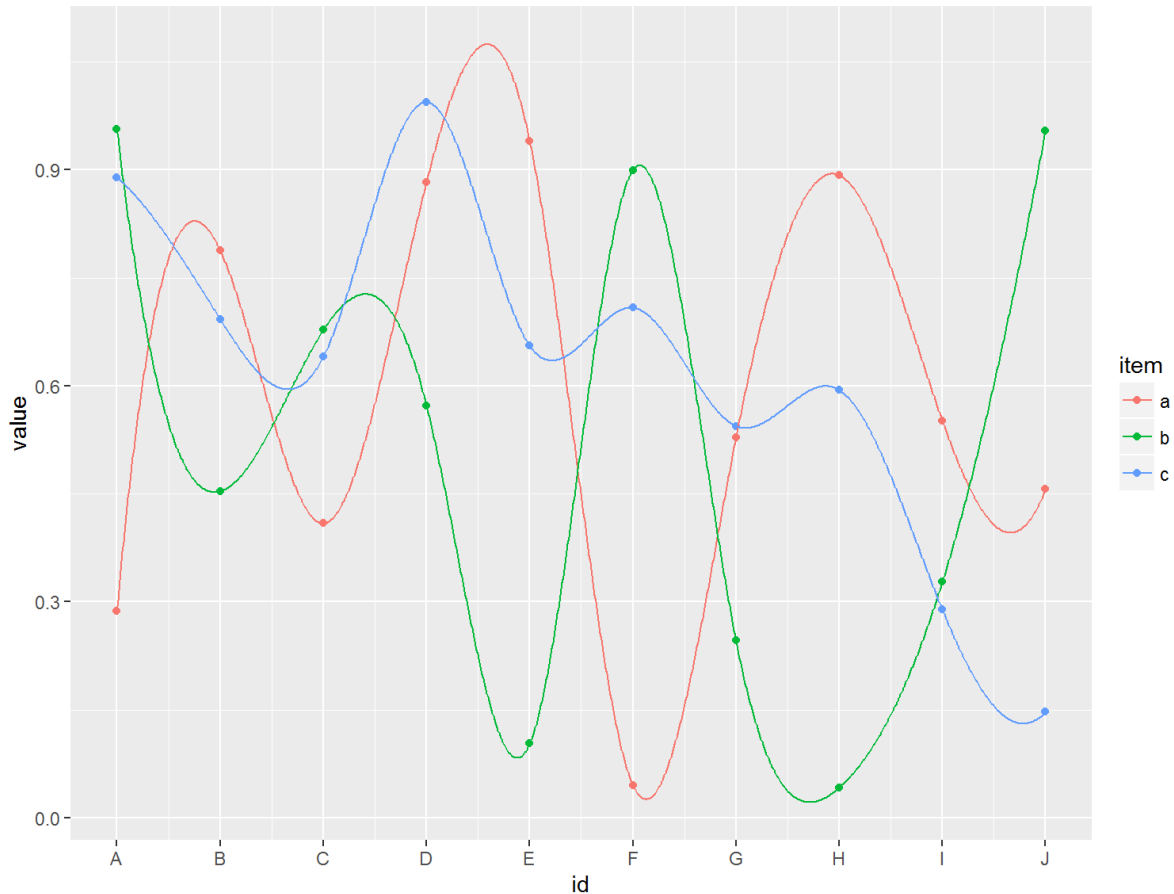
```
ggplot(df1,aes(var,value))+  
  geom_point(aes(colour=item))
```



2.带平滑线和数据标记的散点图

带平滑线和数据标记的散点图的特点是，已知x和y的数值，然后把点放上去。然后用平滑的曲线连起来。

```
df1_a<-df1 %>% filter(item=="a") %>% select(value) %>%  
  unlist %>% spline(,1000) %>% as.data.frame()  
df1_b<-df1 %>% filter(item=="b") %>% select(value) %>%  
  unlist %>% spline(,1000) %>% as.data.frame()  
df1_c<-df1 %>% filter(item=="c") %>% select(value) %>%  
  unlist %>% spline(,1000) %>% as.data.frame()  
  
df1_sp<-bind_rows(df1_a,df1_b,df1_c) %>%  
  mutate(item=rep(letters[1:3],each=1000))  
  
ggplot()+  
  geom_point(data=df1,aes(id,value,colour=item))+  
  geom_line(data=df1_sp,aes(x,y,colour=item))+  
  scale_x_continuous(breaks = 1:10,labels = LETTERS[1:10])
```

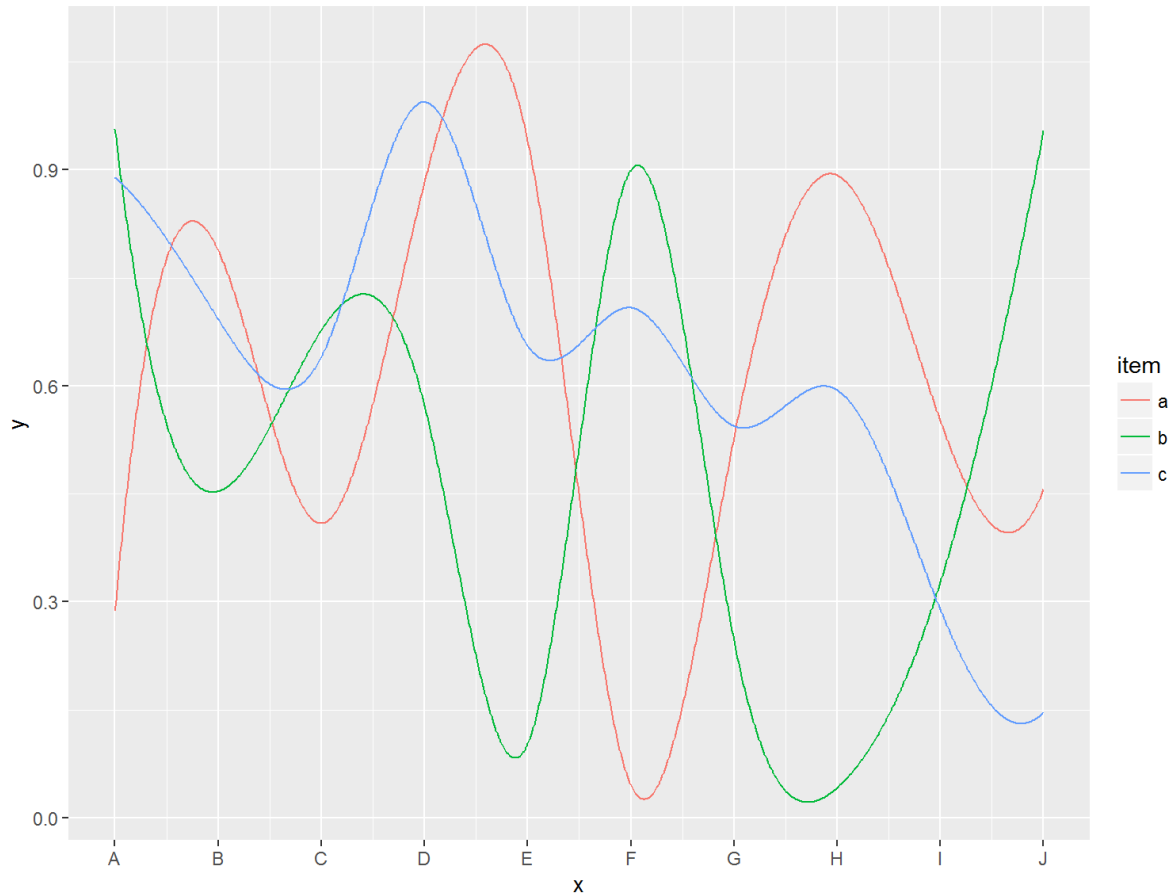


这里需要注意的是，平滑线的应用方式并不是ggplot2自带的geom_smooth函数,而是借用splines包里面的spline函数，在原先10个点的基础上，创造了1000个差值点，用于平滑。这里其实出现了一些重复操作可以用函数封装，如果有自动化需求的读者，可自行编写。

3.带平滑线的散点图

带平滑线的散点图的特点是，已知x和y的数值，直接用平滑的曲线连起来。

```
ggplot()+  
  geom_line(data=df1_sp,aes(x,y,colour=item))+  
  scale_x_continuous(breaks = 1:10,labels = LETTERS[1:10])
```

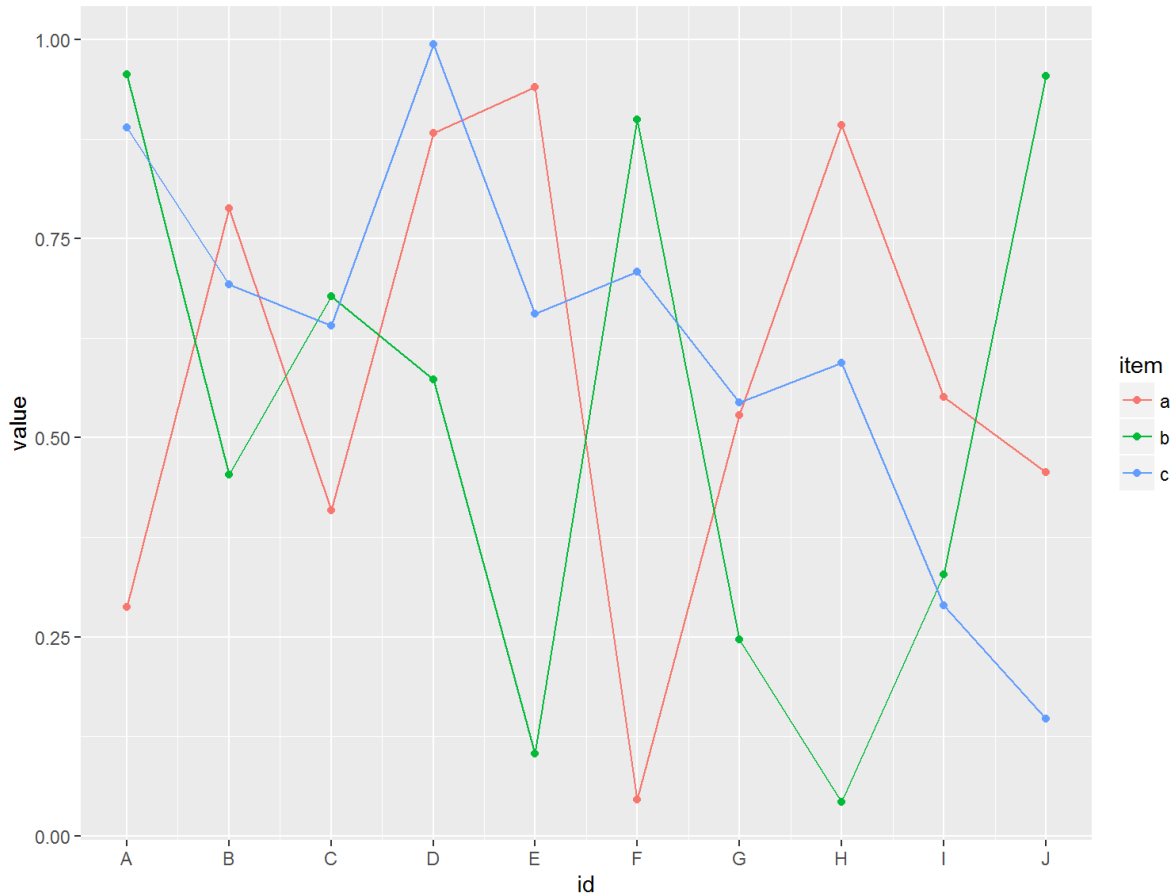


其实从某种意义上来说，应该算是线图了。

4.带直线和数据标记的散点图

带直线和数据标记的散点图的特点是，已知x和y的数值，然后把点放上去，再用线段把他们连起来。

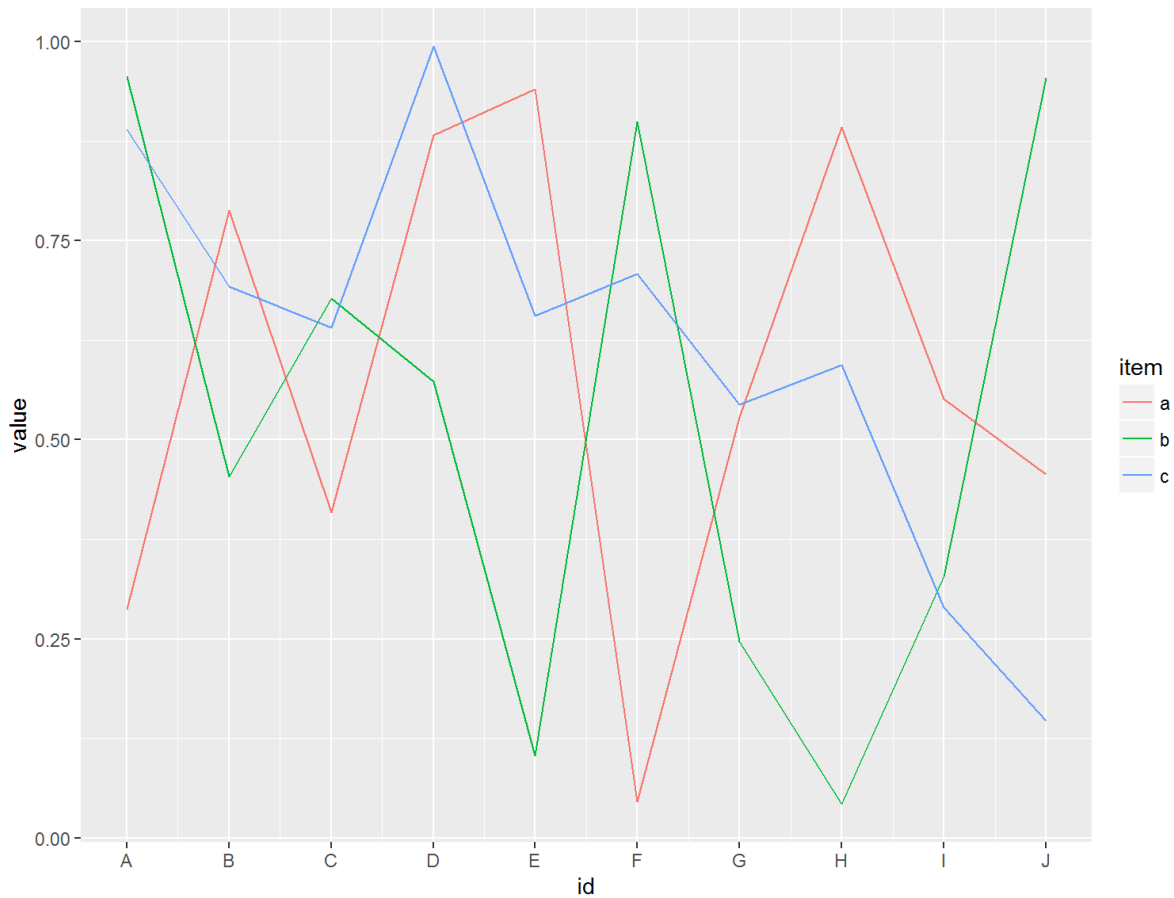
```
ggplot(df1, aes(id, value, colour=item))+  
  geom_point()+  
  geom_line()+  
  scale_x_continuous(breaks = 1:10, labels = LETTERS[1:10])
```



5.带直线的散点图

带直线的散点图的特点是，已知x和y的数值，直接用线段把他们连起来。。

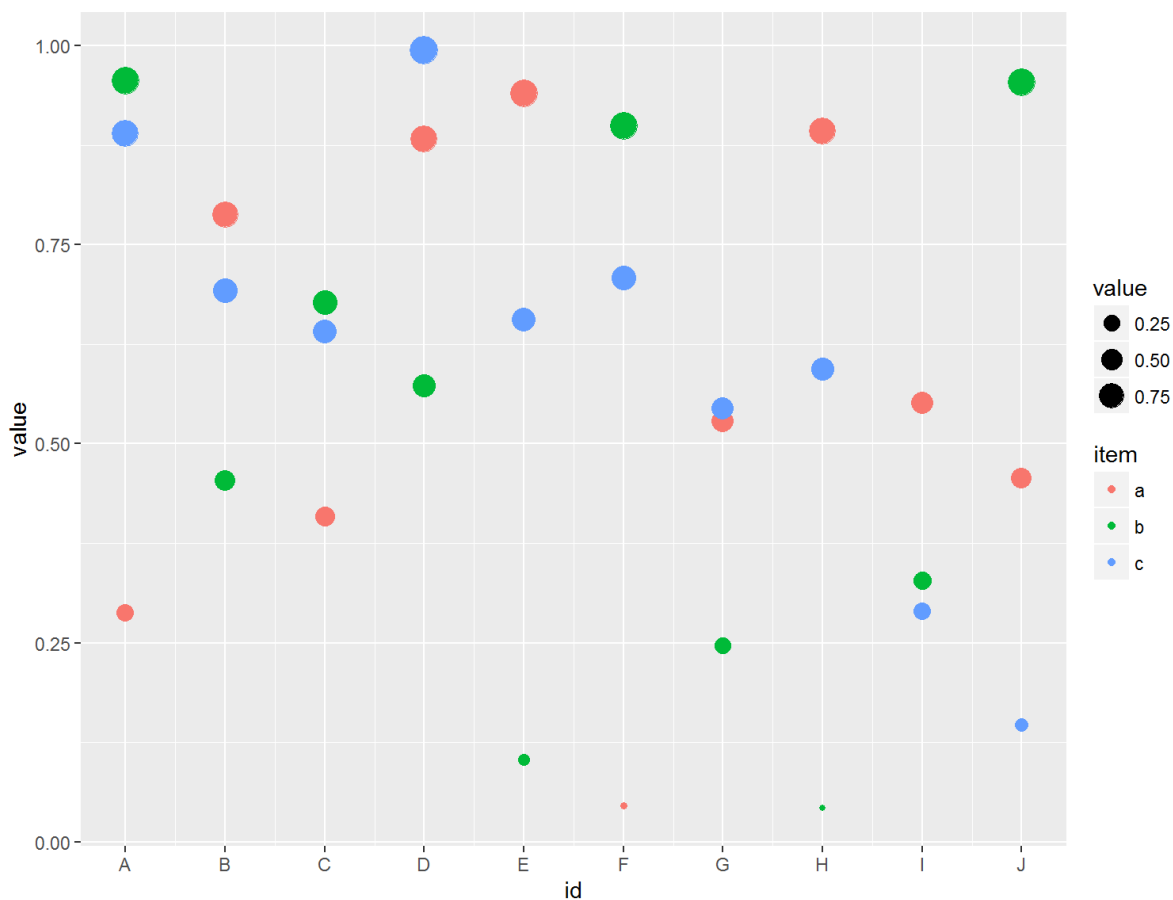
```
ggplot(df1,aes(id,value,colour=item))+  
  geom_line()+  
  scale_x_continuous(breaks = 1:10,labels = LETTERS[1:10])
```

6.气泡图

气泡图的特点是，已知x和y的数值，把他们点出来，然后把他们的大小用不同的圆标记。

```
ggplot(df1,aes(id,value,colour=item))+  
  geom_point(aes(size=value))+  
  scale_x_continuous(breaks = 1:10,labels = LETTERS[1:10])
```



7.气泡图

3D不做图

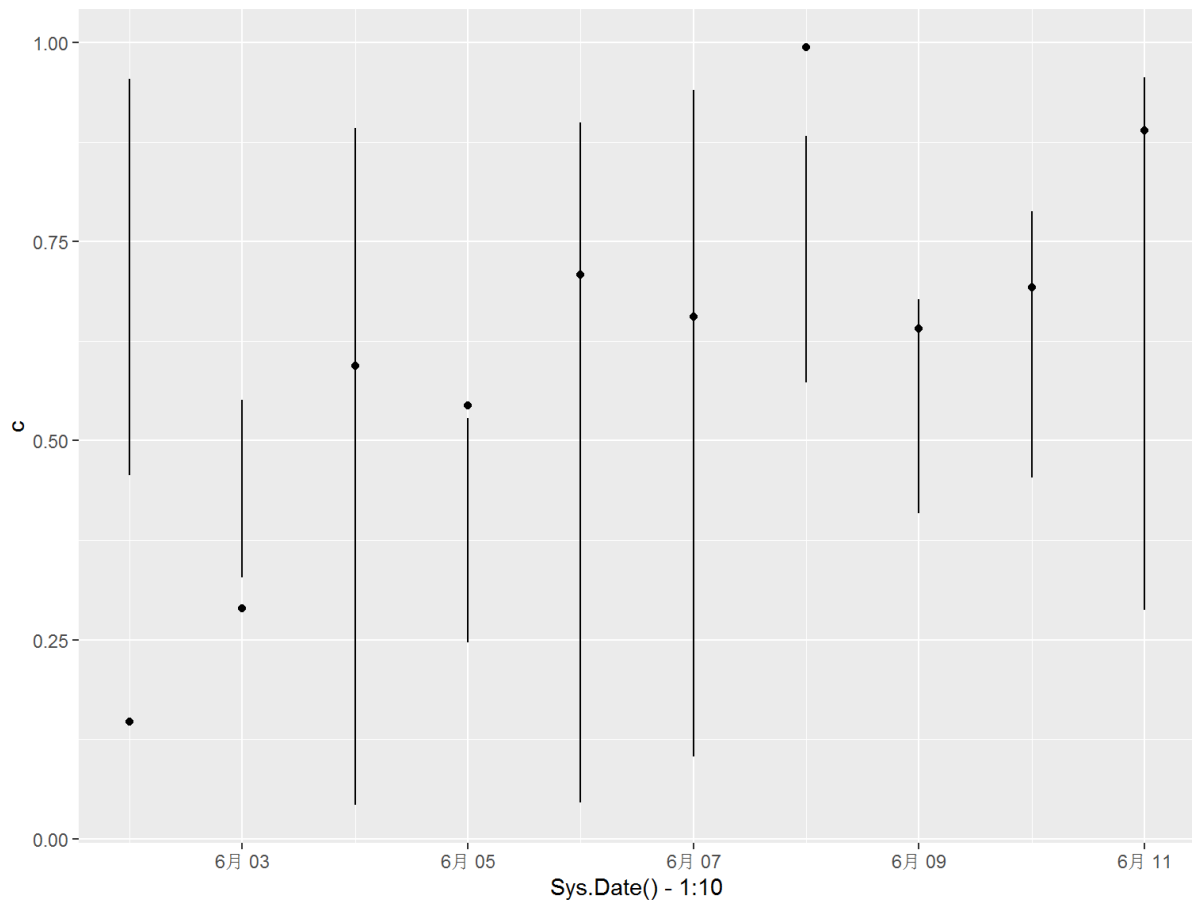
股价图

其实觉得在excel里面出现股价图是很奇怪的一件事，因为这个不算是基础图形，更偏向业务。这里也就不着重介绍了。

1.盘高-盘低-收盘图

已知盘高-盘低-收盘图情况下做的图形。

```
ggplot(df)+  
  geom_point(aes(Sys.Date()-1:10,c))+  
  geom_linerange(aes(Sys.Date()-1:10,ymin=a,ymax=b))
```



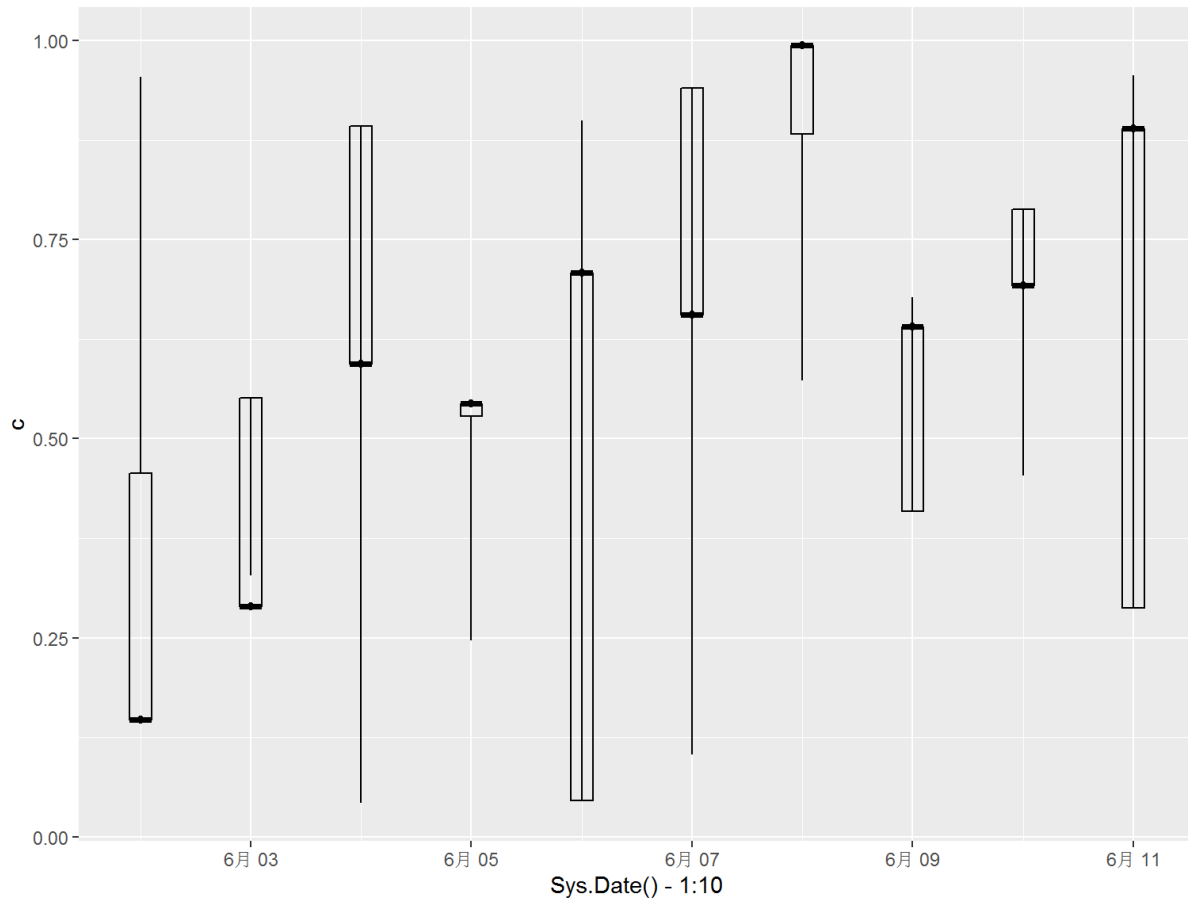
作图的关键是geom_linerange()这个函数

2.开盘-盘高-盘低-收盘图

已知开盘-盘高-盘低-收盘图情况下做的图形。

```
ggplot(df)+  
  geom_point(aes(Sys.Date()-1:10,c))+
```

```
geom_linerange(aes(Sys.Date()-1:10,ymin=a,ymax=b))+
geom_crossbar(aes(Sys.Date()-1:10,c,ymin=a,ymax=c),width=0.2)
```



作图的关键是geom_crossbar()这个函数

3.成交量-盘高-盘低-收盘图

已知成交量-盘高-盘低-收盘图情况下做的图形。但是由于ggplot2开发者不推荐使用副坐标轴，因此采用分面的形式。参考网站：

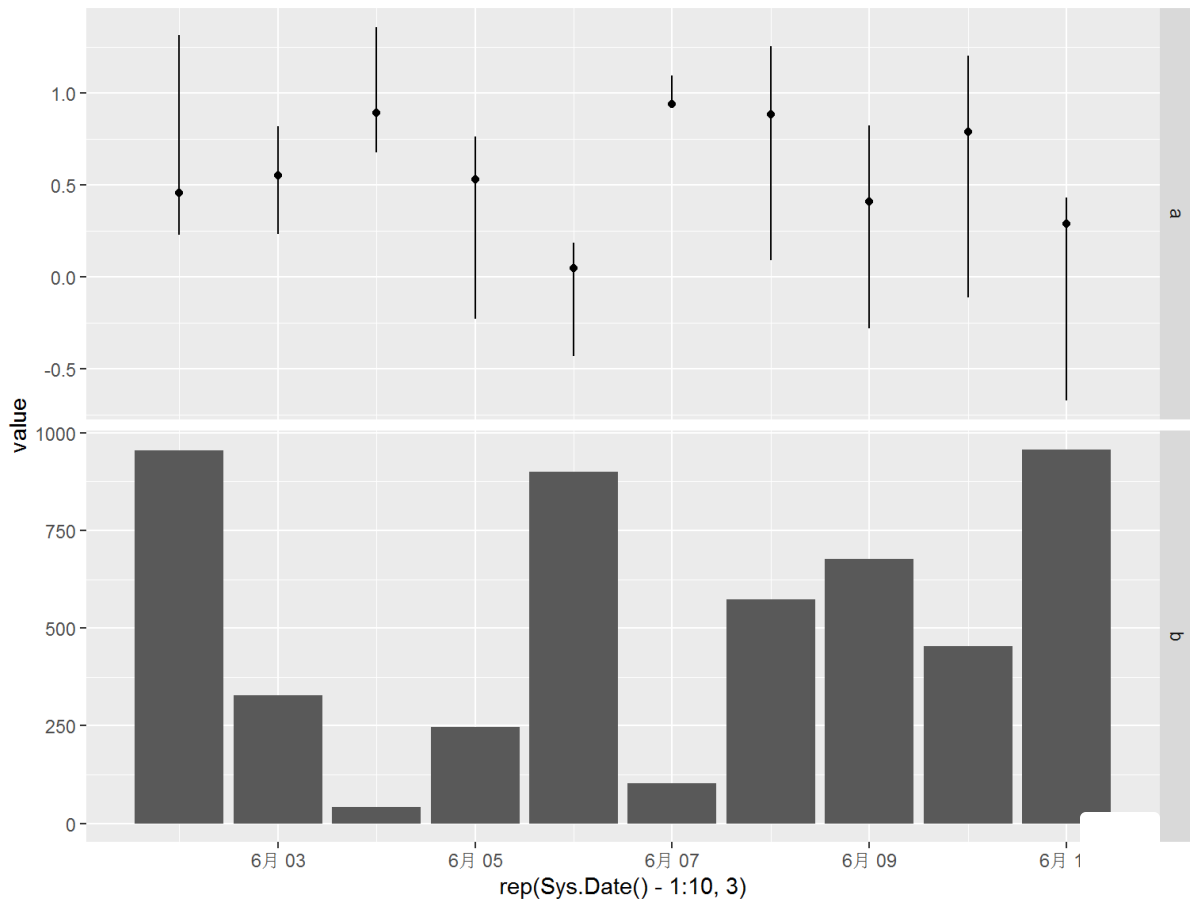
<https://github.com/hadley/ggplot2/wiki/Align-two-plots-on-a-page>

```
ggplot(data = filter(df1,item != "c"),
       aes(rep(Sys.Date()-1:10,3),value))+
```

```

facet_grid(item~., scale="free")+
  geom_point(data = filter(df1,item == "a"),
            aes(Sys.Date()-1:10,value))+
  geom_linerange(data = filter(df1,item == "a"),
               aes(Sys.Date()-1:10,value,
                   ymin=value-runif(10),
                   ymax=value+runif(10)))+
  geom_bar(data = filter(df1,item == "b"),
          aes(Sys.Date()-1:10,value*1000),
          stat="identity")

```

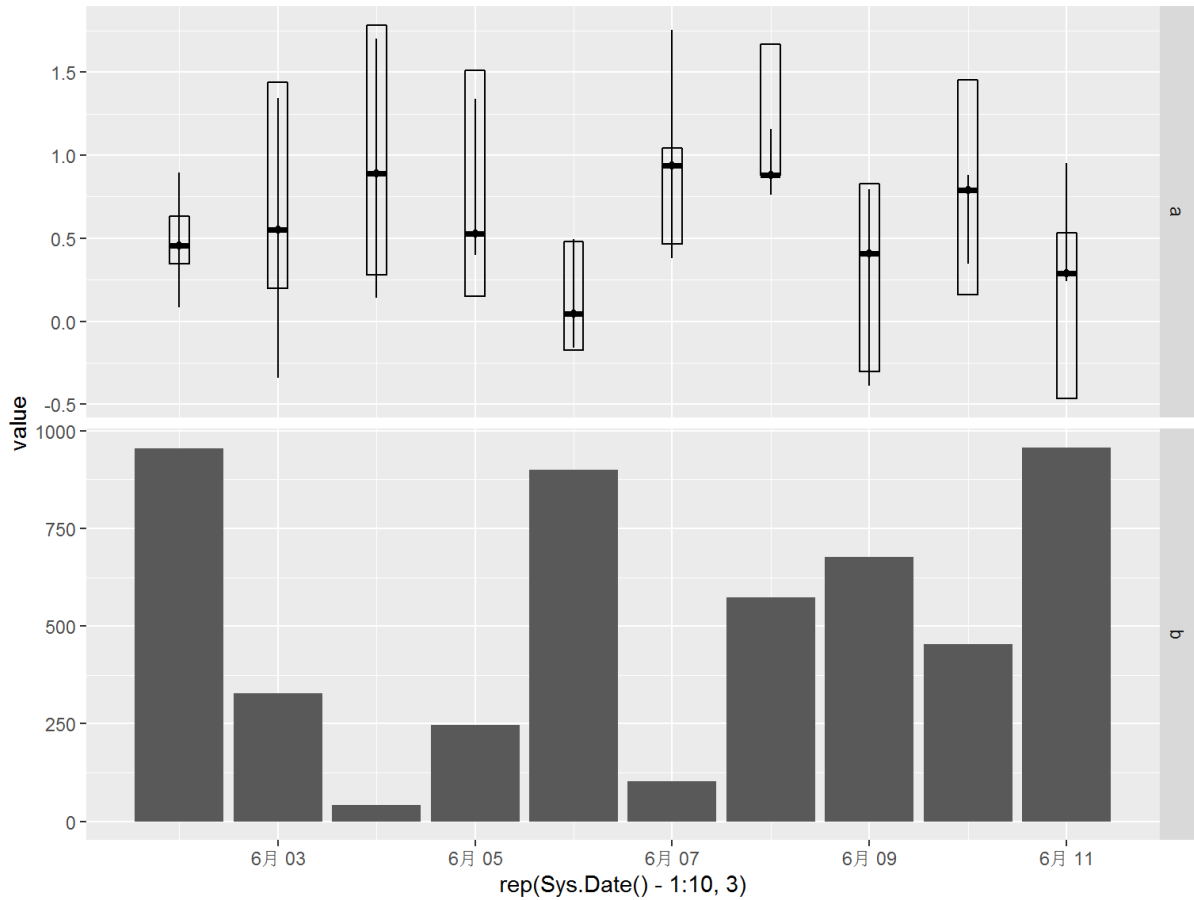


这里给我们一个很重要的思路，在分面图形中，可以存在完全不同的图表形式。

4.成交量-开盘-盘高-盘低-收盘图

已知成交量-开盘-盘高-盘低-收盘图情况下做的图形。

```
ggplot(data = filter(df1,item != "c"),
       aes(rep(Sys.Date()-1:10,3),value))+
  facet_grid(item~.,scale="free")+
  geom_point(data = filter(df1,item == "a"),
            aes(Sys.Date()-1:10,value))+
  geom_linerange(data = filter(df1,item == "a"),
                aes(Sys.Date()-1:10,value,
                    ymin=value-runif(10),
                    ymax=value+runif(10)))+
  geom_crossbar(data = filter(df1,item == "a"),
                aes(Sys.Date()-1:10,value,
                    ymin=value-runif(10),
                    ymax=value+runif(10)),
                width=0.2)+
  geom_bar(data = filter(df1,item == "b"),
           aes(Sys.Date()-1:10,value*1000),
           stat="identity")
```



曲面图

曲面图是个很有趣的三维图形，一般用于实地图形的表现，比如山脉。不过由于ggplot2本身不支持三维呈现，所以这里不做三维曲面图，主要针对二维。

1.三维曲面图

3D不做图

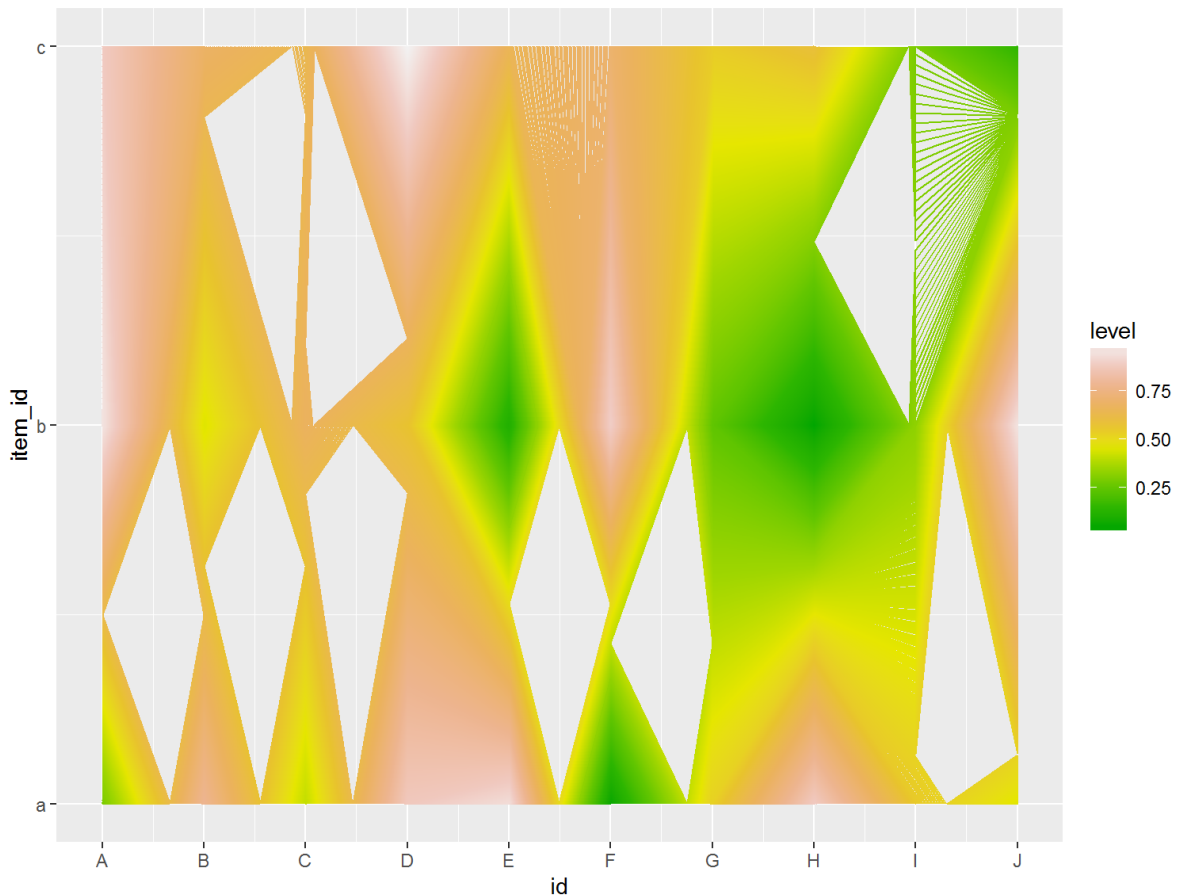
2.三维曲面图（框架图）

3D不做图

3.曲面图

曲面图的特点是，用颜色填充高低点。

```
ggplot(df1,aes(id,item_id))+  
  geom_contour(aes(z=value,colour=..level..),  
               binwidth=0.001)+  
  scale_colour_gradientn(colours = terrain.colors(10))+  
  scale_x_continuous(breaks = 1:10,labels = LETTERS[1:10])+  
  scale_y_continuous(breaks = 1:3,labels = letters[1:3])
```



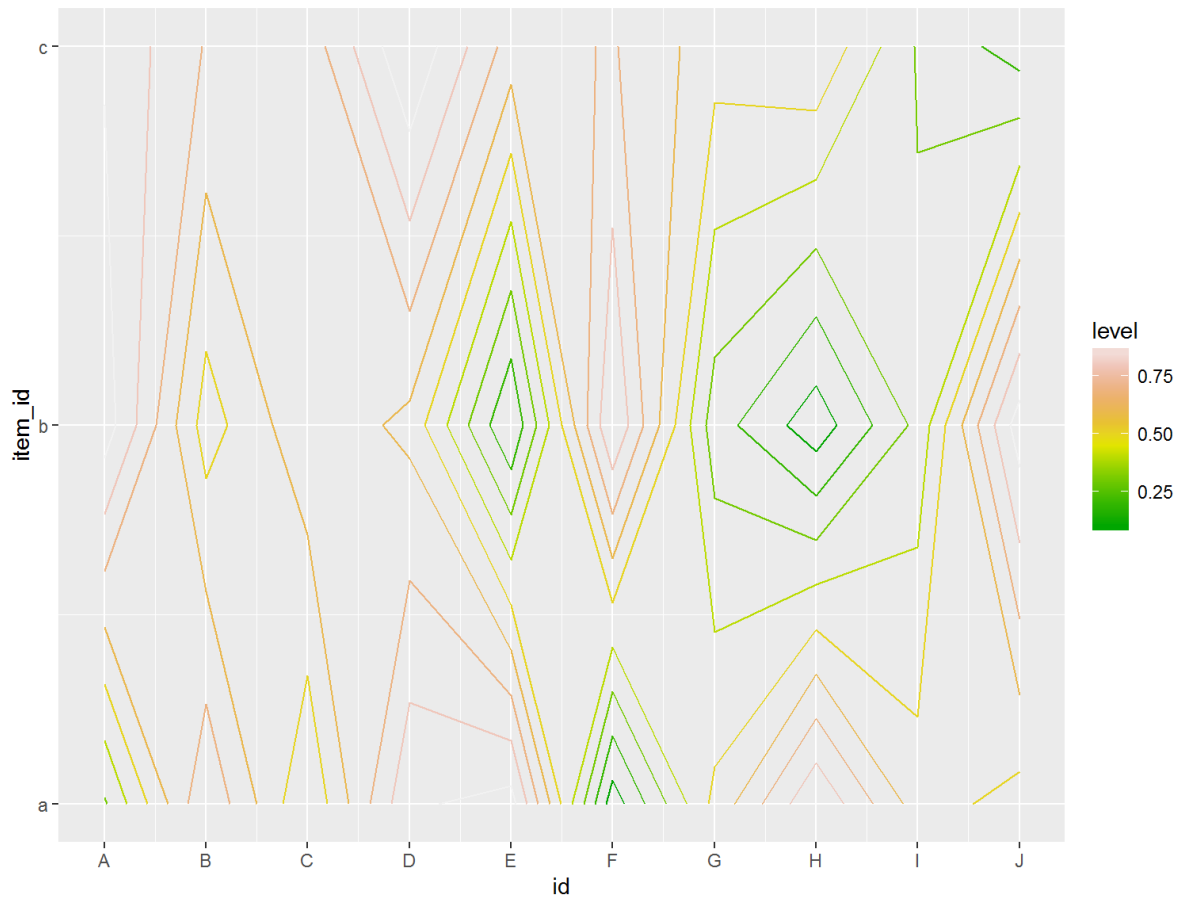
这里其实用了一个比较取巧的办法，就是尽可能多的画线（binwidth=0.0001）来填充颜色，但是速度就非常慢。

另外，在graphics包里面的contour()函数也可以做这个图表

4.曲面图（框架图）

曲面图（框架图）的特点是，用不同颜色的等高线进行作图。

```
ggplot(df1,aes(id,item_id))+  
  geom_contour(aes(z=value,colour=..level..),  
               binwidth=0.1)+  
  scale_colour_gradientn(colours = terrain.colors(10))+  
  scale_x_continuous(breaks = 1:10,labels = LETTERS[1:10])+  
  scale_y_continuous(breaks = 1:3,labels = letters[1:3])
```



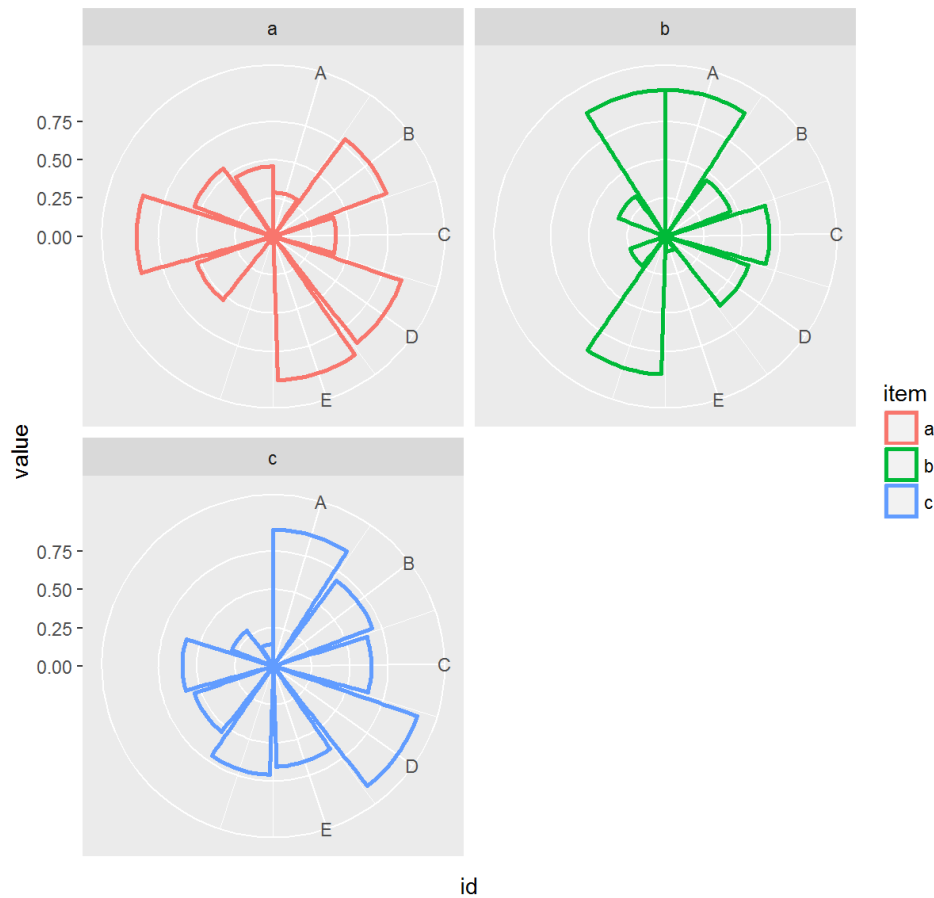
雷达图

雷达图作图比较复杂，因此考虑使用风向玫瑰图代替

1.雷达图

雷达图特点是，使用圆形的图形加上折线展现每个维度的高低情况。

```
ggplot(df1, aes(id, value, colour=item))+  
  geom_bar(stat="identity",  
           position="dodge",  
           fill="transparent",  
           size=1)+  
  coord_polar()+  
  scale_x_continuous(breaks = 1:5, labels = LETTERS[1:5])+  
  facet_wrap(~item, nrow=2)
```

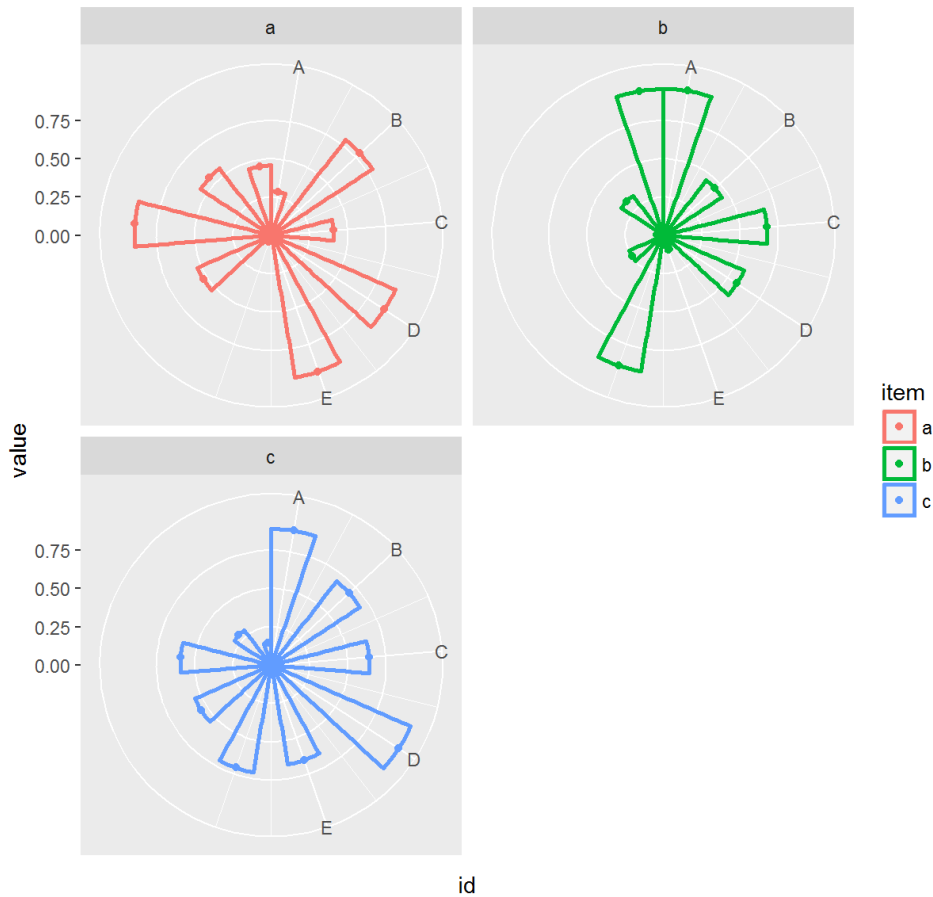


2.带数据标记的雷达图

带数据标记的雷达图特点是，使用圆形的图形加上折线展现每个维度的高低情况，然后用点标记出来。

```
ggplot(df1,aes(id,value,colour=item))+
  geom_bar(stat="identity",
    position="dodge",
    fill="transparent",
    size=1,
    width=0.5)+
  geom_point()+
  coord_polar()+
```

```
scale_x_continuous(breaks = 1:5, labels = LETTERS[1:5])+  
facet_wrap(~item, nrow=2)
```

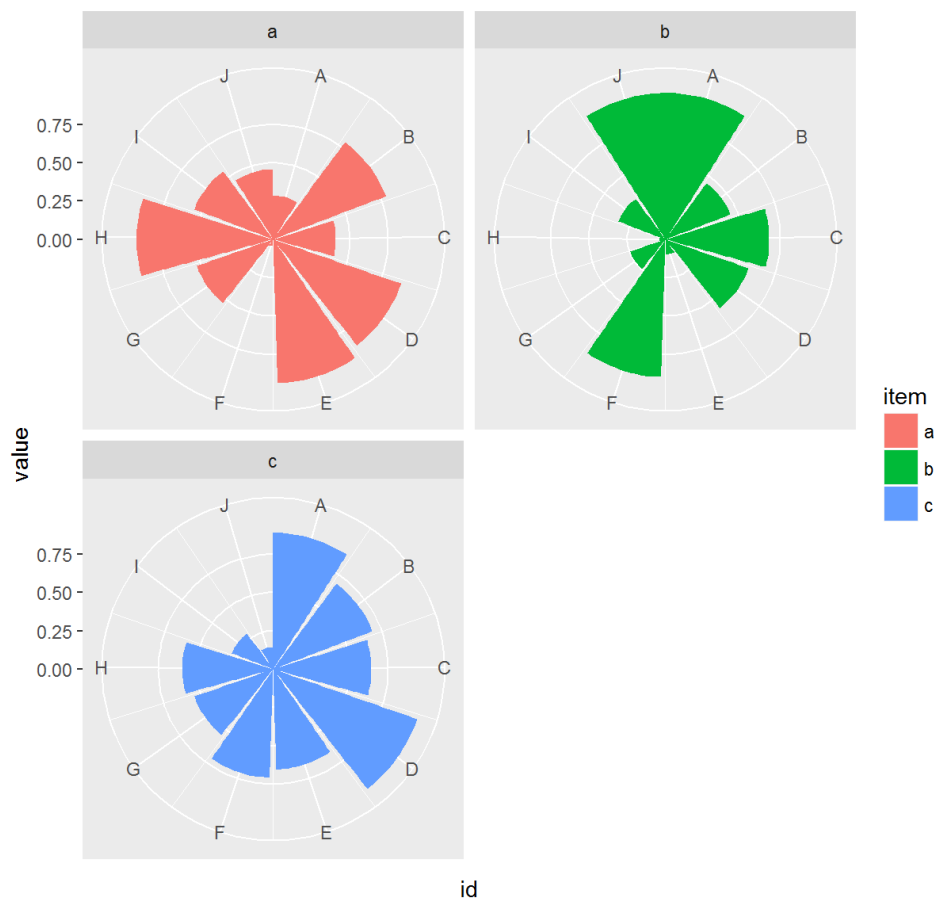


3.填充雷达图

填充雷达图特点是，使用圆形的图形加上折线展现每个维度的高低情况，然后里面填充色即可。

```
ggplot(df1, aes(id, value))+  
  geom_bar(aes(fill=item), stat="identity",  
           position="dodge")+  
  scale_x_continuous(breaks = 1:10, labels = LETTERS[1:10])+
```

```
coord_polar()+  
facet_wrap(~item,nrow=2)
```



树状图

树状图没有找到相应的图形，可以考虑使用其他包实现。

旭日图

旭日图没有找到相应的图形，可以考虑使用其他包实现。不过目前看来，旭日图的实用性有待验证

直方图

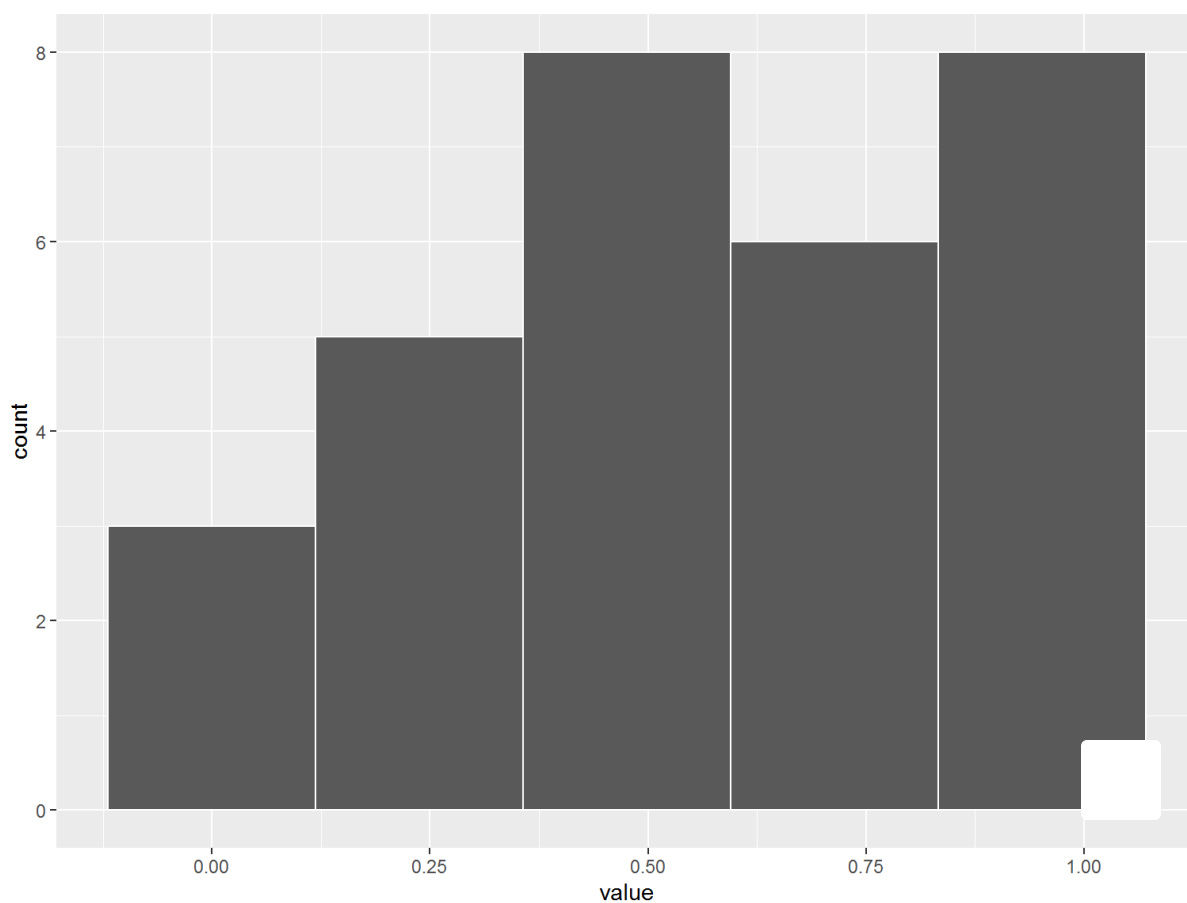
直方图是一种统计图形，是excel2016新加的，但是需要选取原始数据才能做图，所以并不推荐在excel上直接使用。

在ggplot里，则相对自由度较高，可设置的变量更多。

1.直方图

直方图特点是把数据先划分区间，然后从小到大排序，最后用柱状图的形式表现。

```
ggplot(df1,aes(value))+  
  geom_histogram(bins=5,colour="white")
```

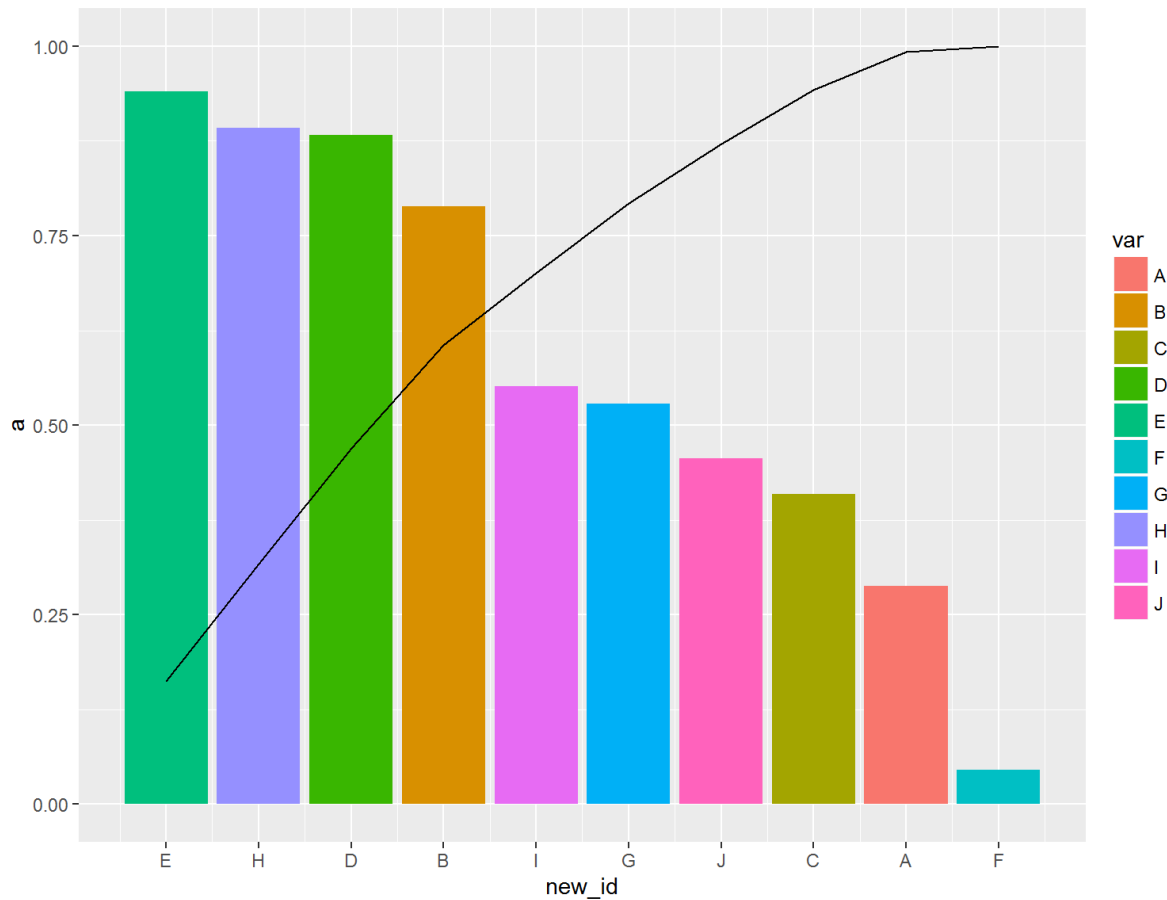


数据量比较小，所以无法体现直方图的用处。

2.排列图

排列图，也叫帕累托图，特点是把数据从大到小排列后，并计算每个数据块的百分比占比，然后两图合一。

```
df_tmp2<-df %>% select(1:3) %>%  
  arrange(a) %>%  
  mutate(per = a/sum(a)) %>%  
  arrange(desc(a)) %>%  
  mutate(new_id = 1:10)%>%  
  mutate(per = cumsum(per))  
  
ggplot(df_tmp2)+  
  geom_bar(aes(new_id,a,fill=var), stat="identity")+  
  geom_line(aes(new_id,per))+  
  scale_x_continuous(breaks = 1:10,  
                     labels = df_tmp2$var)
```



这边一定要注意，排列后也不能直接用，需要重新使用排序。

箱型图

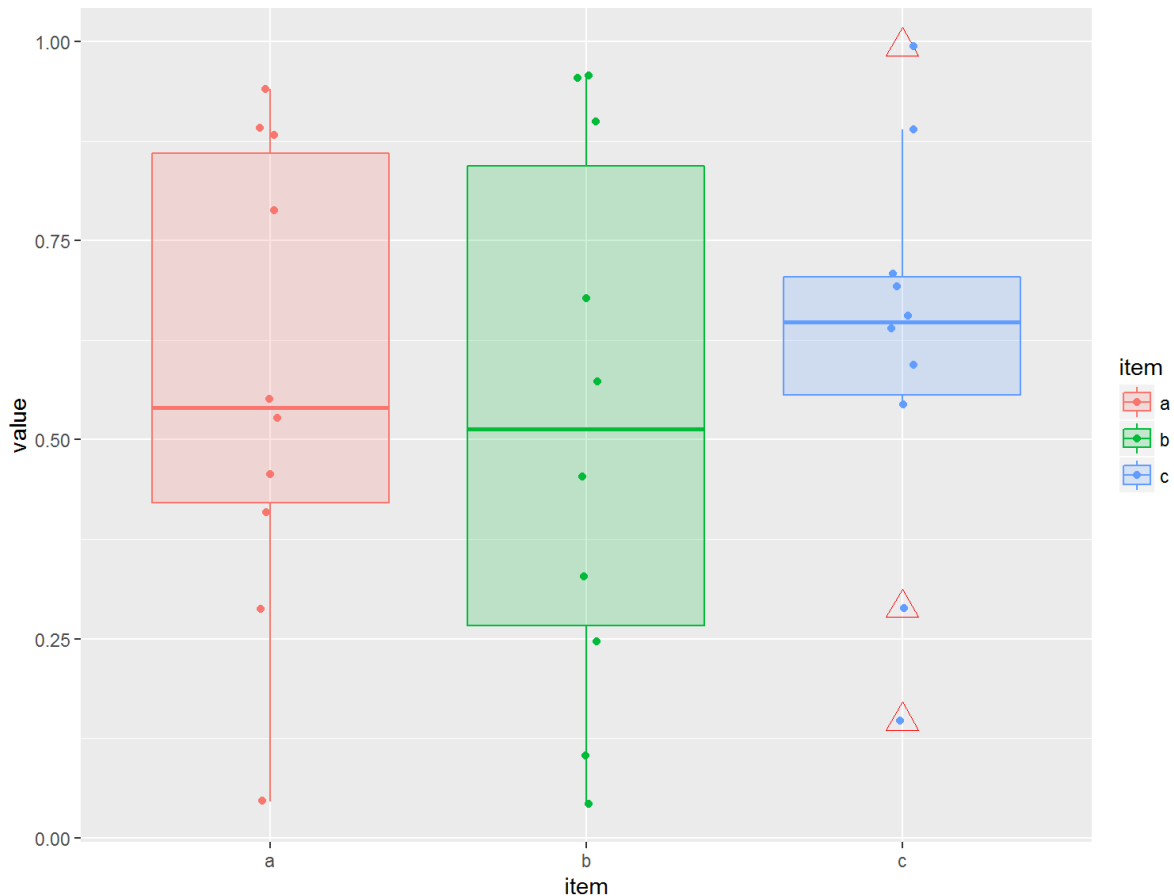
箱型图是一种统计图形，是excel2016新加的，但是需要选取原始数据才能做图，所以并不推荐在excel上直接使用。在ggplot里，则相对自由度较高，可设置的变量更多。

箱型图特点是，使用分位数表示数据的离散和集中趋势。

```
ggplot(df1,aes(item,value,colour=item))+  
  geom_boxplot(aes(fill=item),alpha=0.2,  
    outlier.colour = "red",  
    outlier.shape = 2,
```



```
outlier.size = 5,  
coef=1.5)+  
geom_jitter(width = 0.1)
```



本图中由于没有异常值，所以没有标记，但是可以通过coef调整。另外，图上还是可以增加一个平均值点，读者可以自行摸索。

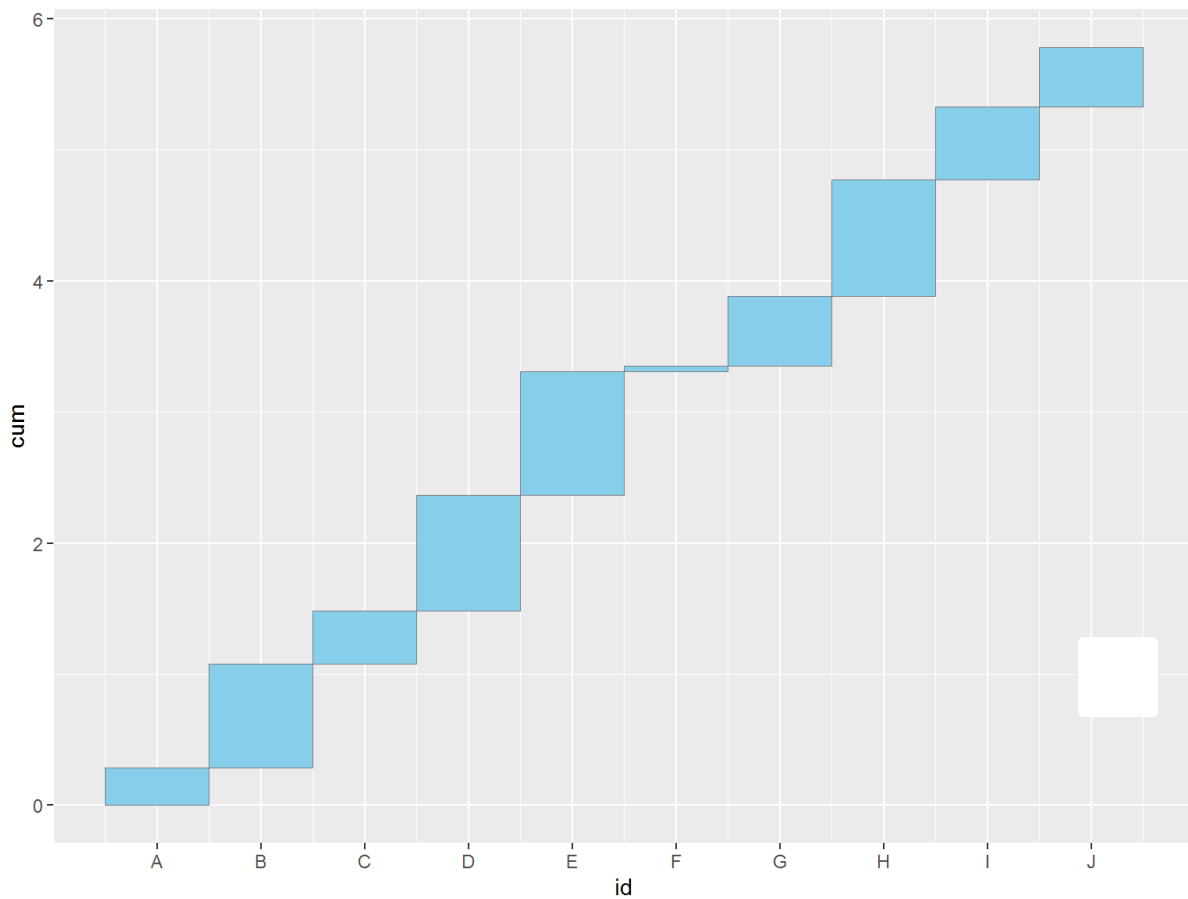
瀑布图

瀑布图是一种代表图形涨跌趋势的图表，是excel2016新加的，但是需要再计算。在ggplot2里，没有现成的图表，需要转换。

瀑布图的特点是后一个柱子和前一个柱子有增长或下降的关系。

```
df_tmp3 <- df %>% select(1:3) %>%
  mutate(cum=cumsum(a),
         low=lag(cum,default = 0))

ggplot(df_tmp3,aes(id,cum))+
#  geom_step(colour="grey50")+
  geom_crossbar(aes(ymin=low,ymax=cum),
               size=0,
               fill="skyblue",
               colour="grey50",
               width=1)+
  scale_x_continuous(breaks = 1:10,
                     labels = LETTERS[1:10])
```



本例中是采用增长的图表，如果需要下降的趋势读者可以自行尝试。

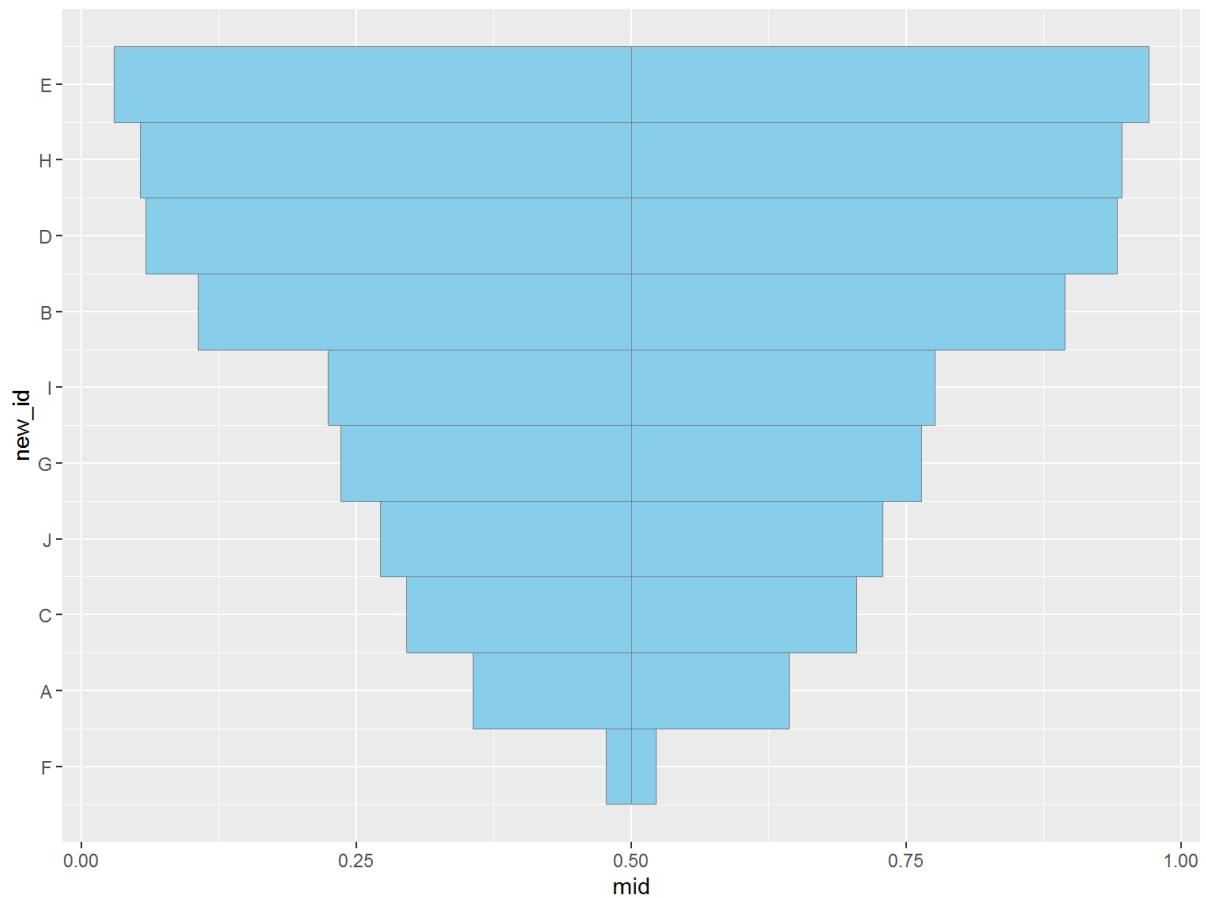
漏斗图

漏斗图其实和一般的柱形图或条形图类似，只是其实点变成了中间。在ggplot2里，没有现成的图表，需要转换。

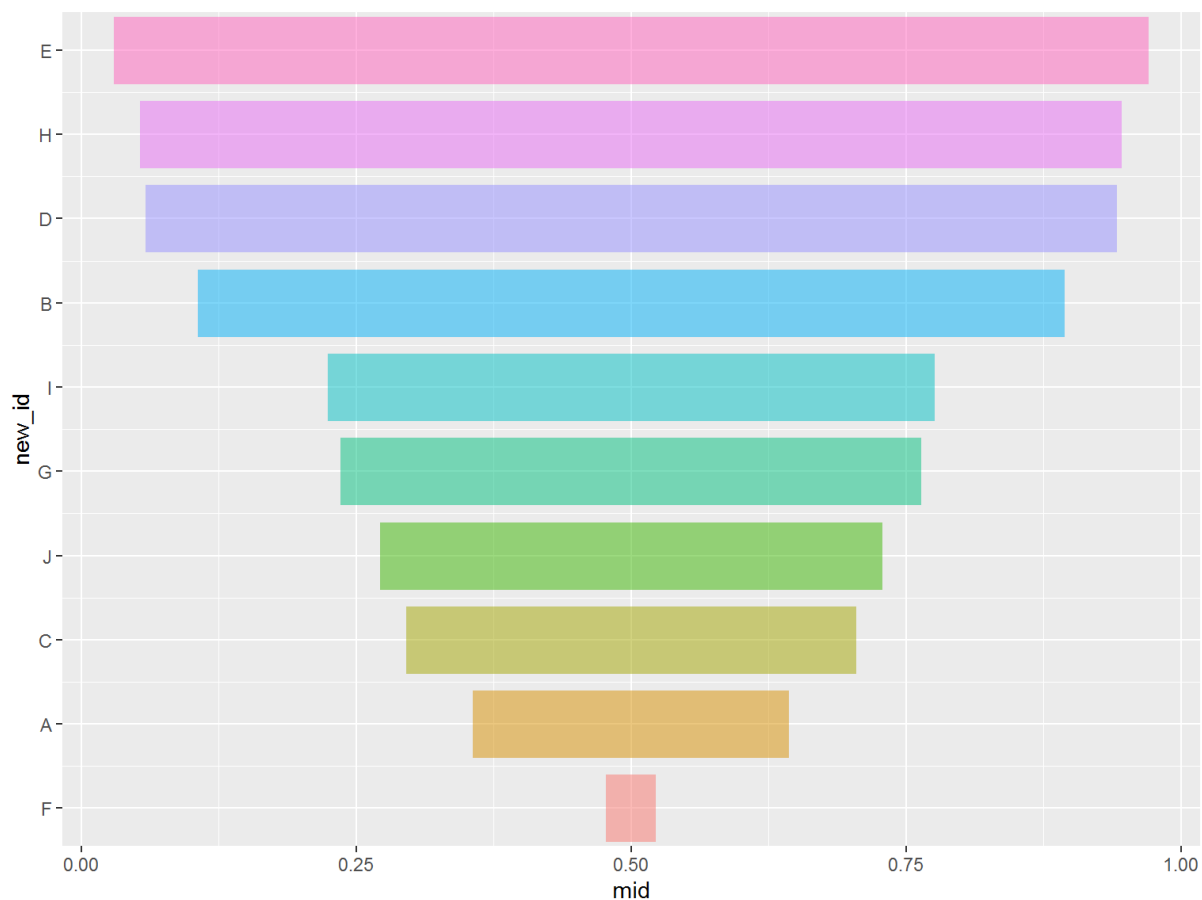
1.漏斗图

箱型图特点是，使用分位数表示数据的离散和集中趋势。

```
df_tmp4<-df %>% select(1:3) %>%  
  arrange(a) %>%  
  mutate(new_id=1:10,  
         ymin = (1-a)/2,  
         ymax = a+(1-a)/2,  
         mid = 0.5)  
  
ggplot(df_tmp4,aes(new_id,mid))+  
  # geom_step(colour="grey50")+  
  geom_crossbar(aes(ymin=ymin,ymax=ymax),  
               size=0,  
               fill="skyblue",  
               colour="grey50",  
               width=1)+  
  scale_x_continuous(breaks = 1:10,  
                     labels = df_tmp4$var)+  
  coord_flip()
```



```
ggplot(df_tmp4, aes(new_id, mid)) +
  geom_linerange(aes(ymin=ymin, ymax=ymax,
                     colour=factor(new_id)),
                size=15,
                alpha=0.5, show.legend = F) +
  scale_x_continuous(breaks = 1:10,
                     labels = df_tmp4$var) +
  coord_flip()
```



以上提供了两种漏斗图的作图方式，请读者自行选择。

结语

本文实际上是一个尝试，作为刚刚读完ggplot2操作文档的作者来说，需要练手，所以可能有很多地方还不成熟。希望是抛砖引玉的一个文章。

其实要说为什么要和excel比较的话，其实是看中ggplot2更自由的作图风格和非常多的参数设置。

不得不说，实际上对于excel用惯的作者来说，还有很多可以调整的参数有待开发，比如，坐标轴、图例、数据标签等等等等，这个就需要后续另行思考和实践了。

[E-Phone Xu](#) / 2016-06-13 / - views

Published under[\(CC\) BY-NC-SA 3.0 CN](#).

-
- - [R \(12\)](#)
 -
 - [ggplot2 \(3\)](#)
-

[← PREVIOUS](#)

[Git进阶之一](#)

论数 //参与数

[// NEXT →](#)

评 [Rmd生成中文Beamer](#)

友情链接

- [R学习社区](#)
- [Hux-模板来源](#)
- [谢益辉](#)
- [落园](#)
- [刘思喆](#)
- [邱怡轩](#)
- [JRQ 西南交大](#)
- [MenBin 东财应统](#)

