

A Simple Case Using GGplot2

张春光

目录

1 基本概念	2
2 映射 Aes 与设定	2
3 几何对象和统计变换	2
3.1 散点图 <code>geom_point</code> 及平滑曲线	3
3.2 折线图 <code>geom_line</code> 和路径图 <code>geom_path</code>	5
3.3 柱状图 <code>geom_bar</code>	7
3.4 直方图及密度曲线 <code>geom_histogram</code>	7
3.5 箱线图 <code>geom_boxplot</code>	9
3.6 树状图	10
3.7 其他常用图形	11
4 图形布局	12
4.1 分面图 <code>facet_grid()</code> 和 <code>facet_wrap</code>	12
4.2 一页多图	13
4.3 复合图	14
5 图形属性	15
5.1 坐标与坐标系	15
5.2 标度与图例	17
6 主题与配色	18
6.1 主题	19
6.2 配色	22
7 数据地图	25
8 Shiny 介绍	28
9 学习帖	28

1 基本概念

GGPLOT2 和 R 基础绘图系统的主要区别在于按照图层形式作图，基本语法为：

$$\text{ggplot}(\text{data}, \text{aes}(x, y)) + \text{geom_xxx}() + \dots + \text{theme}() + \dots$$

语法主要分为以下组件：

组件	说明
数据 Data	长数据
映射 Mapping	数据关联到相应的图形属性
几何对象 Geom	图层控制, 主要阐述图的类型
统计变换 Stats	展示出某个变量的某种统计特征
标度 Scale	控制数据到图形属性的映射
坐标系 Coord	调整坐标
Guides	调整所有 Text
分面 Facet	控制分组绘图的方法和排列形式
主题 Theme	调整不与数据相关的图形属性.

2 映射 Aes 与设定

- 映射: 控制变量到图形属性, 表现为不同属性或同一属性的不同形式如 $\text{ggplot}(\text{data}, \text{aes}(x, y, \text{color} = 'red'))$.
- 设定: 将全部数据设定为同一个图形属性 $\text{ggplot}(\text{data}, \text{aes}(x, y), \text{color} = 'red')$.

3 几何对象和统计变换

- 几何对象 **geom_xxx**: 以图层叠加形式控制图形渲染, 具体表现就是图的类型, 如散点, 直方图。
- 统计变换 **stat_xxx**: 先进行数据的统计变换然后以图层形式表现。

每一个几何对象都有一个默认的统计变换, 并且每一个统计变换都有一个默认的几何对象。因此两者在使用效果上差别不大, 结合使用则更加灵活。

常用组件	说明
geom_abline,geom_hline,geom_vline	辅助线, 水平线, 垂直线
geom_bar,geom_col,stat_count	柱形图
geom_bin2d,stat_bin_2d	二维热力图
geom_histogram,geom_freqpoly,stat_bin	频率分布、直方图
geom_qq,stat_qq	类似 qqnorm
geom_segment,geom_curve	线段和曲线

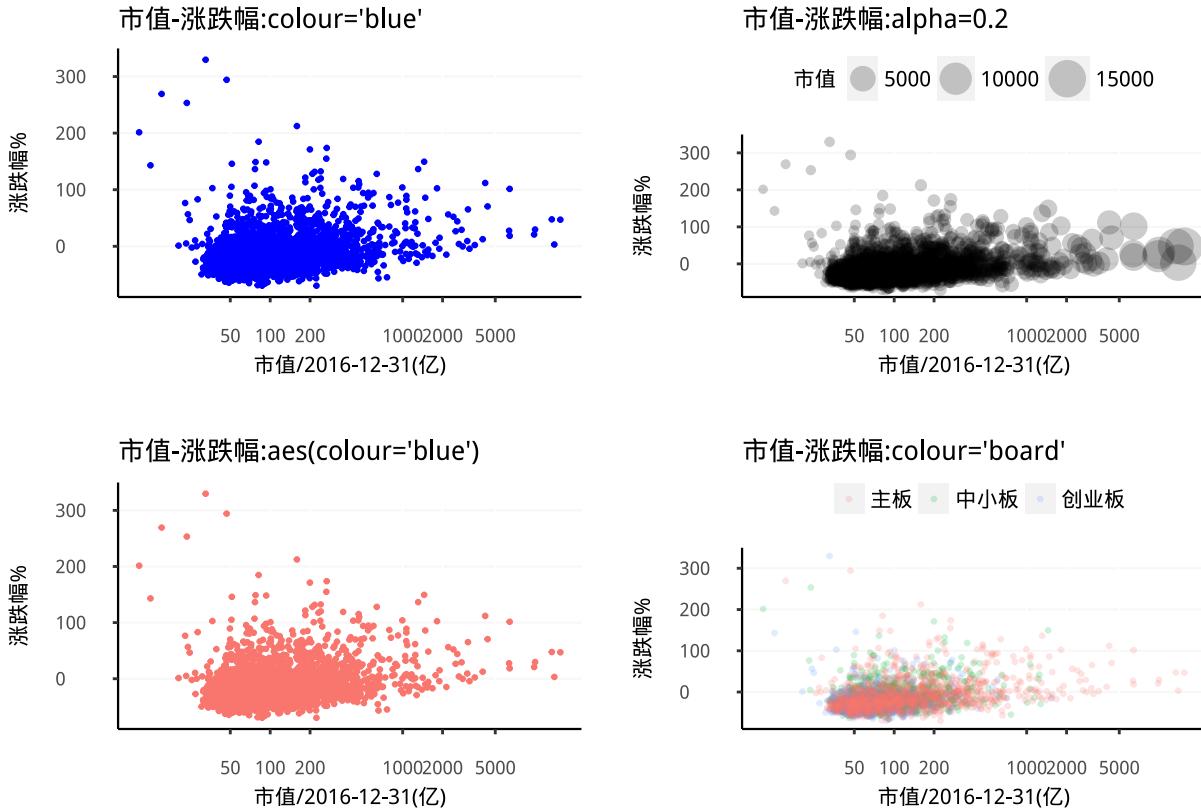
常用组件	说明
geom_point,geom_smooth	散点图, 平滑曲线
geom_label,geom_text	添加标签和文本
其他	官方组件

3.1 散点图 geom_point 及平滑曲线

- 散点图

形式: `geom_point(data, aes(x, y, ...))`, 主要通过 shape,fill,colour,alpha,size 控制点的属性.

```
library(xlsx)
stock<-read.xlsx("examples/stock2017.xlsx",sheetName="2017",header=T,
                  encoding='UTF-8',check.names = FALSE)
require(base);library(plyr);library(lubridate);library(magrittr)
#简单数据离散化
stock <- stock[year(stock$ipo)< 2018,][1:3467,] %>%subset(.,mv17>0&mv18>0)
stock$mv17_d <- cut(stock$mv17,breaks=c(0,50,100,200,500,1000,16000),
                      labels=c('0~50','50~100','100~200','200~500','500~1000','>1000'),
                      right=FALSE,include.lowest=TRUE)
stock$change_d <- cut(stock$change,breaks=c(-100,-20,0,20,50,100,400),
                       labels=c('<-20%','-20~0%','0~20%','20~50%','50~100%','>100%'),
                       right=FALSE,include.lowest=TRUE)
stock$board<- factor(stock$board,levels=c('主板','中小企业板','创业板'),
                      labels=c('主板','中小板','创业板'),ordered=T)
#散点图
library(ggplot2)
p1 <- ggplot(stock,aes(x=mv17,y=change))+labs(x='市值/2016-12-31(亿)',y='涨跌幅%')+theme_os(sd_size = 1)+scale_x_log10(breaks=c(0,50,100,200,1000,2000,5000))#主题设定和坐标转换
p1_1 <- p1+geom_point(size=0.5,colour='blue')+ggtitle("市值-涨跌幅:colour='blue'")#填充为蓝色
p1_2 <- p1+geom_point(aes(colour='blue'),size=0.5)+guides(colour=FALSE)+ggtitle("市值-涨跌幅:aes(colour='blue')")#将蓝色作为变量映射到颜色属性
p1_3 <- p1+geom_point(aes(size=mv17),alpha=0.2)+labs(title='市值-涨跌幅:alpha=0.2',size='市值')#减少重叠的效果
p1_4 <- p1+geom_point(aes(color=board),size=0.5,alpha=0.2)+labs(title="市值-涨跌幅:color='board'",color=NULL)#添加board对散点的颜色映射
```



注意 `p1_1` 与 `p1_2` 的区别, 前者通过属性设定设置为蓝色, 后者将散点颜色映射到 `blue` 变量,R 语言默认配色为红色。

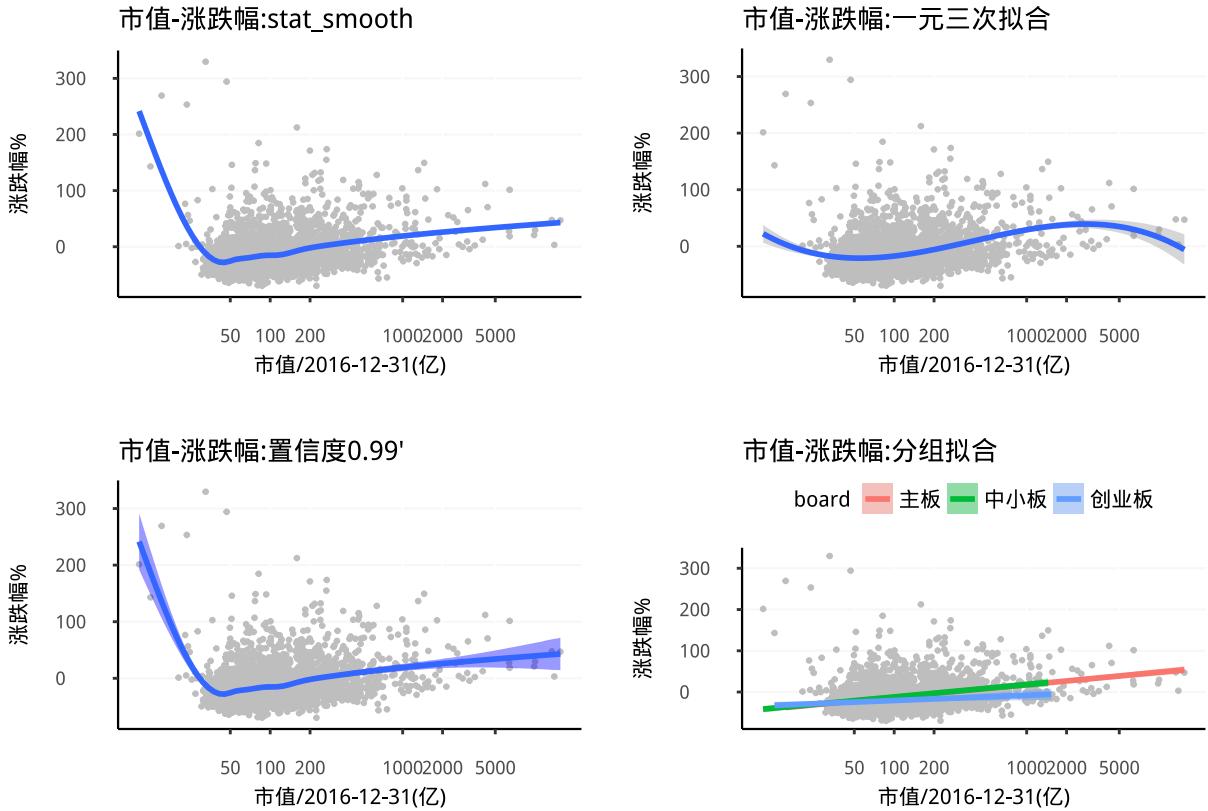
- 平滑曲线-`geom/stat_smooth()`

形式: `stat_smooth(mapping, data, stat =, method =, formula =, se =, span =, level =,)`, 通过调整 `method, formula` 调整拟合方法, `level` 调整置信区间, 'se=F/T' 是否显示误差域.

```

p1_V <- p1+geom_point(size=0.5,color='grey')
p1_V1 <- p1_V+stat_smooth(se=F)+ggtitle("市值-涨跌幅:stat_smooth")
#添加拟合曲线,可以通过method调整,小于1000观测值时默认为loose。此处为gam
p1_V2 <- p1_V+stat_smooth(fill='blue',se=T,level=0.99)+ggtitle("市值-涨跌幅:置信度0.99!")
p1_V3 <- p1_V+stat_smooth(method='lm',formula =y~poly(x,3))+ 
  ggtitle("市值-涨跌幅:一元三次拟合")
p1_V4 <- p1_V+stat_smooth(method='lm',aes(color=board,fill=board))+ 
  ggtitle("市值-涨跌幅:分组拟合")

```



注意 `ggplot(aes(x,y))` 函数中指定的是全局变量，而 `geom/stat_xxx()` 中指定的是局部变量，如 `ggplot(stock) + geom_point(aes(x = mv17, y = change)) + stat_smooth()` 因 `stat_smooth` 有没有数据映射发生错误, `geom_xxx` 是添加图层, `stat_xxx` 是作统计变化之后进行映射，结果大致相同。

3.2 折线图 `geom_line` 和路径图 `geom_path`

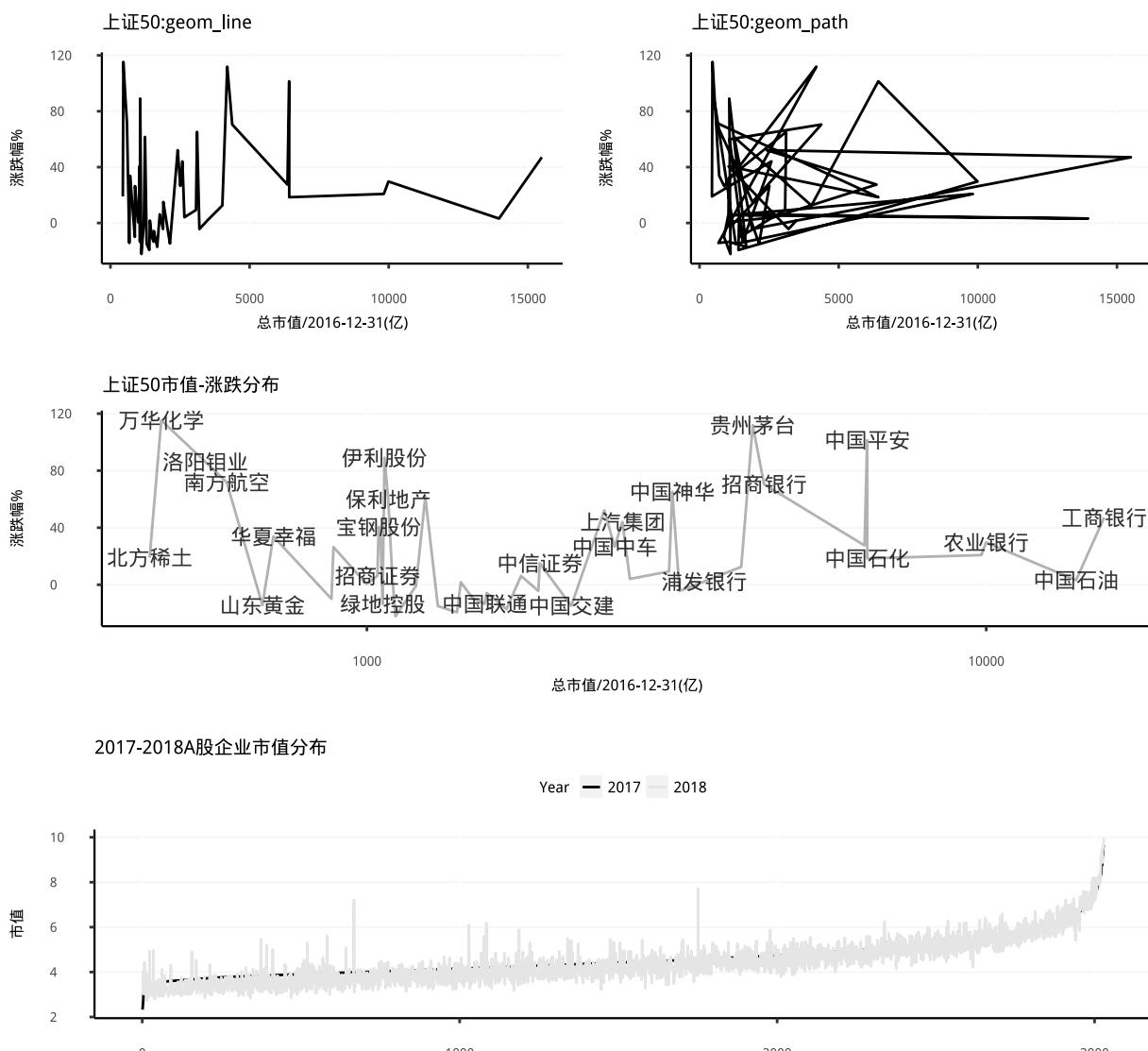
- * 折线图按照横坐标顺序依次连接, 形式为 `geom_line(mapping, data, stat, arrow = arrow(), ...)`:
- * 路径图按观测值出现先后顺序连接:`geom_path`

```
library(xlsx)
SZ50 <- read.xlsx('examples/SZ50.xlsx', sheetName="SZ50", header=T,
                    encoding='UTF-8', check.names = FALSE)
stock_50 <- subset(stock, company %in% SZ50$证券简称)
p2 <- ggplot(stock_50, aes(x=mv17, y=change)) + labs(x='总市值/2016-12-31(亿)', y='涨跌幅%') +
  theme_os(sd_size = 0.75)
p2_1 <- p2 + geom_line(size=0.5) + ggtitle('上证50:geom_line')
p2_2 <- p2 + geom_path(size=0.5) + ggtitle('上证50:geom_path')
p2_3 <- p2 + geom_line(size=0.5, alpha=0.3) + scale_x_log10() + ggtitle('上证50市值-涨跌分布') +
  geom_text(aes(label=company), alpha=0.8, size=3, check_overlap=T)
stock_m <- stock[,c(2,14,15)] %>% dplyr::arrange(., mv17) %>% data.frame(id=seq(1, dim(stock)[1]), .) %>%
```

```

set_colnames(c('id','company',2017,2018))
library(reshape2)
stock_m2 <- melt(stock_m,id.vars = c('id','company'),variable.name ='Year',value.name = 'MarketValue')
p2_4 <- ggplot(stock_m2,aes(x=id,y=log(MarketValue),colour=Year))+geom_line()+
  labs(title='2017-2018A股企业市值分布',x=NULL,y='市值',fill='板市')+
  scale_color_manual(values=c('black','grey90'))+
  theme_os(sd_size = 0.75)
#添加标签, 同geom_label()

```



```

## null device
## 1

```

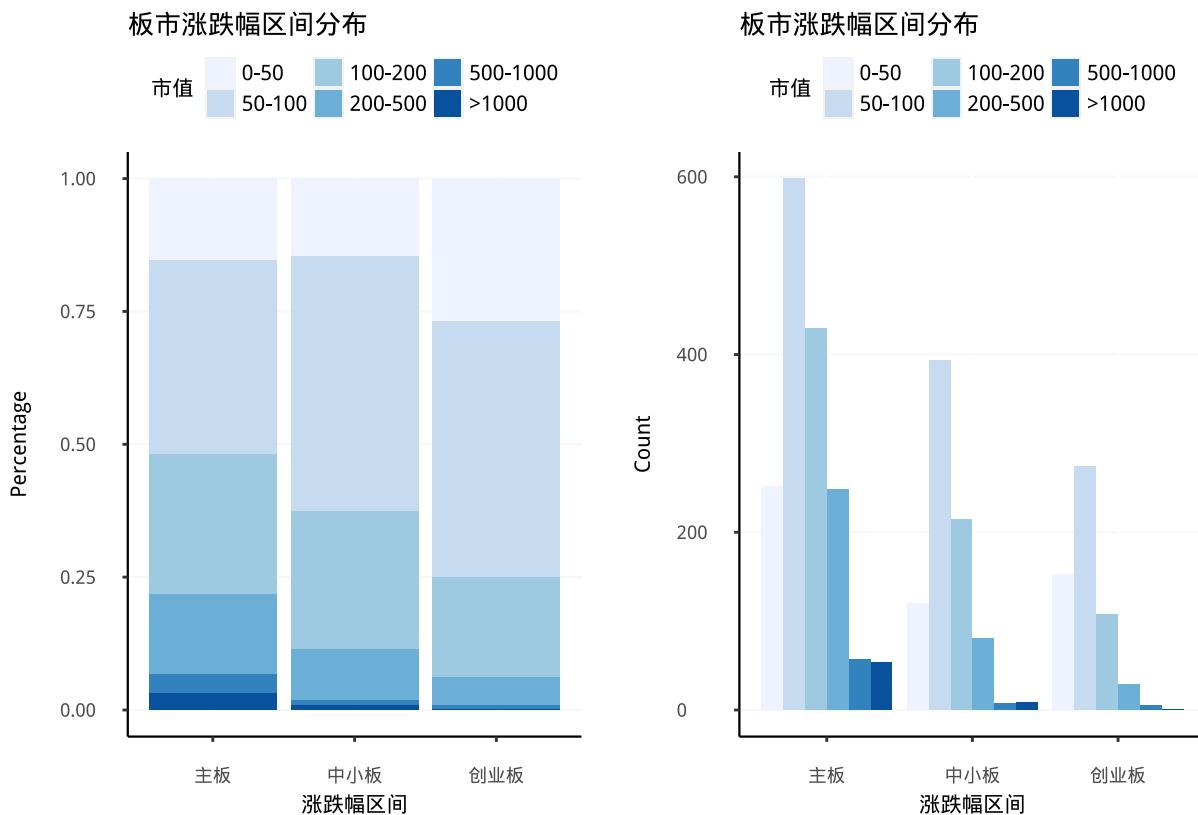
3.3 柱状图 geom_bar

- 柱状图，横坐标为离散型，形式相对简单：

```
geom_boxplot(mapping, data, , position = "dodge", ...)
```

，一般关注柱的摆放方式即可。

```
p3 <- ggplot(stock,aes(x=board))+geom_bar(aes(fill=mv17_d),position='fill')+  
  labs(title='板市涨跌幅区间分布',x='涨跌幅区间',y='Percentage',fill='市值')+  
  theme_os(sd_size=1)+scale_fill_brewer()#scale为颜色填充标度  
p3_1 <- ggplot(stock,aes(x=board))+geom_bar(aes(fill=mv17_d),position='dodge')+  
  labs(title='板市涨跌幅区间分布',x='涨跌幅区间',y='Count',fill='市值')+  
  theme_os(sd_size = 1)+scale_fill_brewer()#scale为颜色填充标度
```



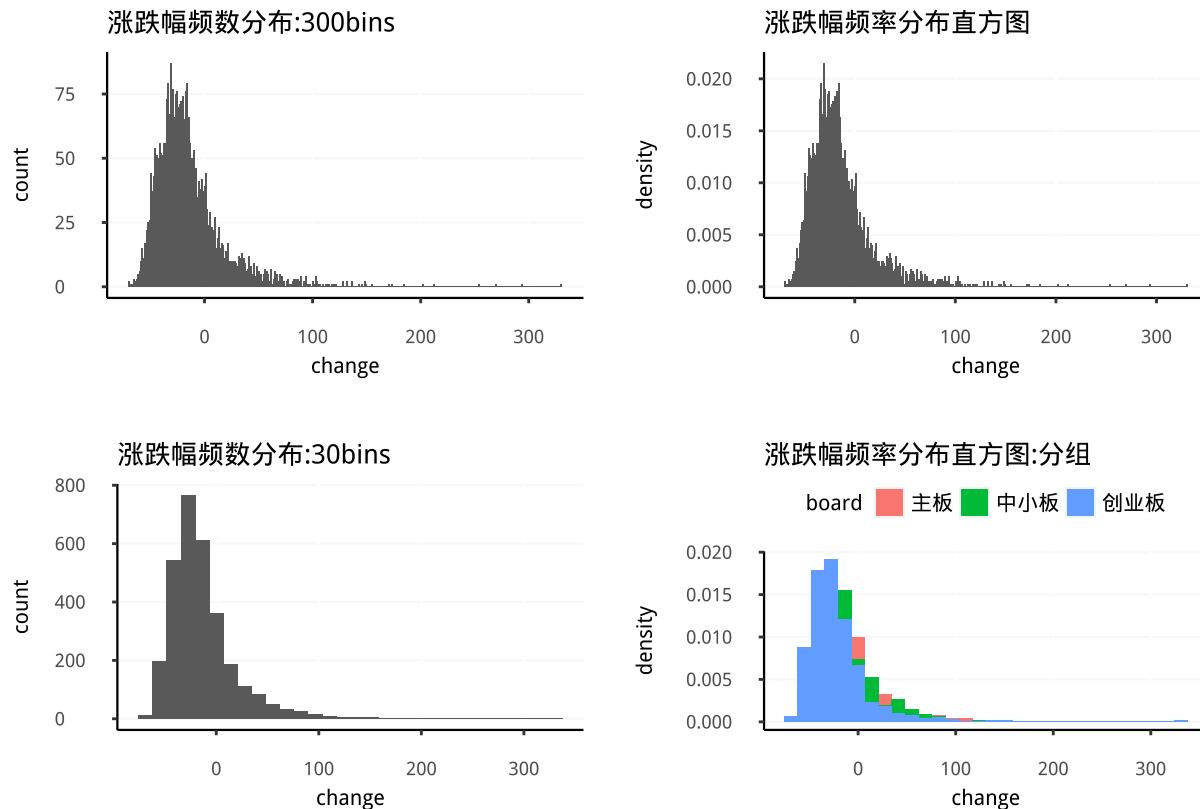
3.4 直方图及密度曲线 geom_histogram

- 直方图：可以理解为连续变量的柱状图，函数形式: `geom_boxplot(mapping, data, , position, bins = , binwidth =)`。一般关注 bins 和 binwidth 两个参数来设置直方个数。

```

p4 <- ggplot(data=stock,aes(x=change))+theme_os()
p4_1 <- p4+geom_histogram(bins = 300)+ggtitle('涨跌幅频数分布:300bins')#bins和binwidth控制组个数和
p4_2 <- p4+geom_histogram(bins = 30)+ggtitle('涨跌幅频数分布:30bins')
p4_3 <- p4+stat_bin(aes(y=..density..),bins=300)+ggtitle('涨跌幅频率分布直方图')#默认为Count,可以按
p4_4 <- p4+stat_bin(aes(y=..density..,fill=board),position = 'identity')+ggtitle('涨跌幅频率分布直方图:分组')##可观察三者分布差异

```



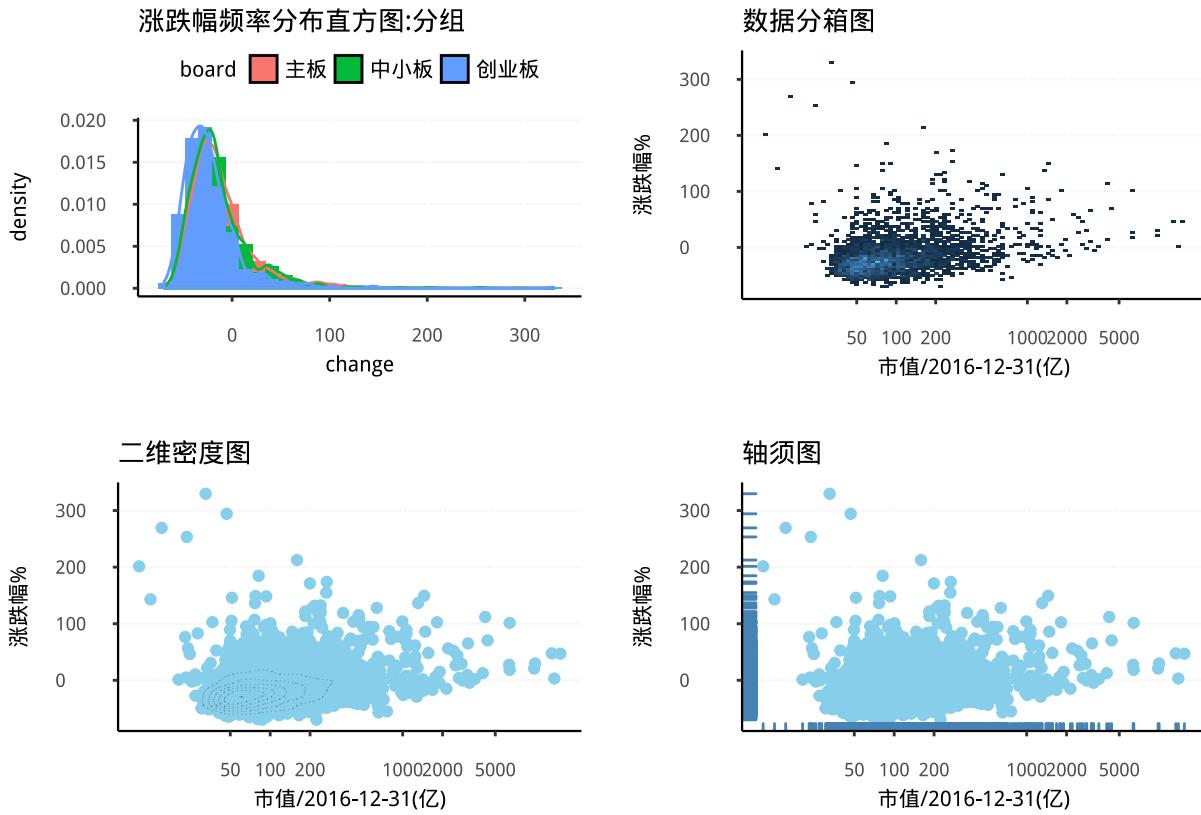
- 密度曲线

一维密度曲线 `geom_density`, 二维包括 `geom_density2d`, `geom_bin2d`, `geom_tile()`.

```

p4_V1 <- p4_4+geom_density(aes(colour=board),alpha=0.5)+guides(colour=FALSE)
p4_V2 <- p1+geom_point(colour='skyblue')+stat_density2d(geom='tile')+ggtitle('二维密度图')
p4_V3 <- p1+stat_bin2d(bins=100)+ggtitle('数据分箱图')+theme(legend.position = 'none')
p4_V4 <- p1+geom_point(colour='skyblue')+geom_rug(color='steelblue')+ggtitle('轴须图')#边际分布

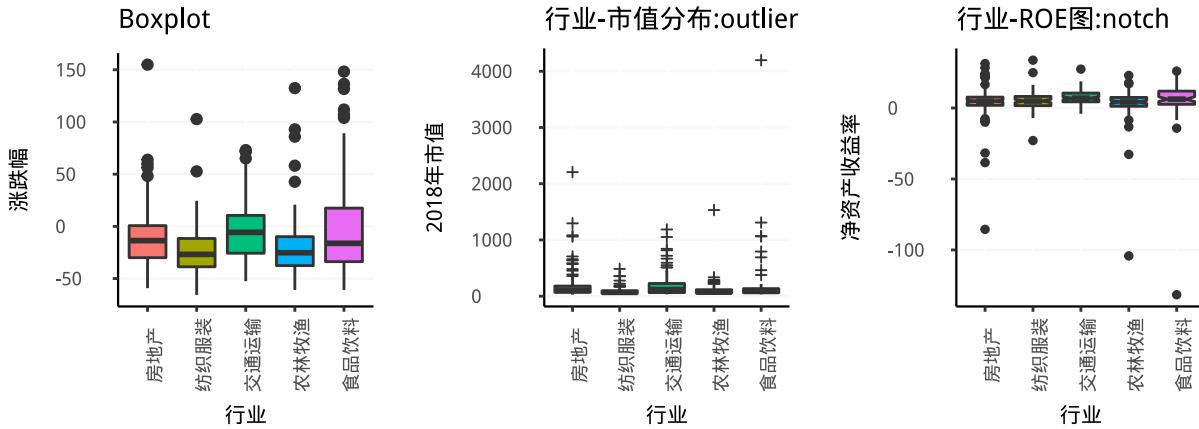
```



3.5 箱线图 geom_boxplot

```
geom_boxplot(mapping, data, stat, position = "dodge", ..., outlier.xxx, notch, notchwidth)
```

```
p5 <- ggplot(subset(stock, industry %in% c('纺织服装', '食品饮料', '房地产', '交通运输', '农林牧渔')),
             aes(x=industry)) + labs(x='行业', fill=NULL) + theme_os() %+ replace%
  theme(axis.text.x = element_text(angle = 90)) + guides(fill=FALSE)
p5_1 <- p5 + geom_boxplot(aes(y=change, fill=industry)) + ylab('涨跌幅') + ggtitle('Boxplot')
p5_2 <- p5 + geom_boxplot(aes(y=mv17, fill=industry), outlier.size=1, outlier.shape = 3) +
  ylab('2018年市值') + ggtitle('行业-市值分布:outlier') #改变outliers图形属性
p5_3 <- p5 + geom_boxplot(aes(y=roe, fill=industry), notch = T, outlier.size=1) +
  ylab('净资产收益率') + ggtitle('行业-ROE图:notch') #改变box形状凸显特征
```



3.6 树状图

树状图一般用于数据的层次结构，大多数用于公司业务分析，此处用于行业市值结构分析。

```

library(ggplot2)
library(treemapify)
library(tweenr)
library(gganimate)
library(RColorBrewer)

stock_ind <- ddply(stock,.(industry),summarize,
                    mv_tot17=sum(mv17,na.rm = T),
                    mv_tot18=sum(mv18,na.rm = T),
                    pe_17=weighted.mean(pe17,na.rm = T),
                    pe_18=weighted.mean(pe18,na.rm = T),
                    counts=length(company))

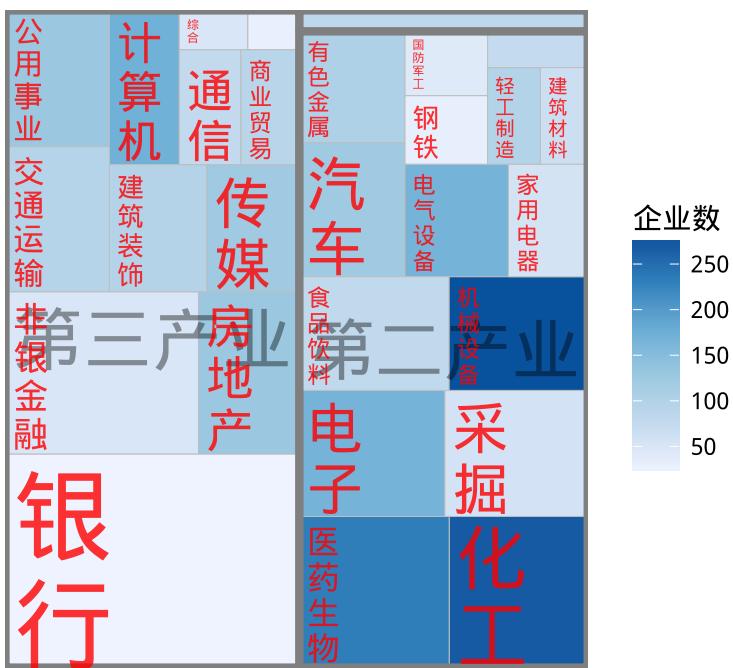
stock_ind$type <- c(2,3,2,2,3,2,3,2,3,2,2,2,3,2,2,3,3,1,2,2,3,2,3,3,2,3,2,3)
stock_ind$type <- factor(stock_ind$type,levels=c('1','2','3'),
                           labels=c('第一产业','第二产业','第三产业'))

ggplot(stock_ind,aes(area=mv_tot18,label=industry,subgroup=type))+
  geom_treemap(aes(fill=counts))+ 
  geom_treemap_subgroup_border()+
  geom_treemap_subgroup_text(fontface="italic",colour="black",
                             place="centre",alpha=0.4)+ 
  geom_treemap_text(fontface='italic',colour='red',
                    place='bottomleft',grow=T,reflow = T,alpha=0.8)+ 
  scale_fill_distiller('企业数',palette = 'Blues',direction = 1,breaks=seq(50,250,50))+ 
  labs(title='2018上市企业市值行业分布',
       captions='注:面积代表行业市值,颜色深度与行业企业数成正比')

```

```
theme(plot.caption=element_text(hjust=0,size=10,color='grey'))
```

2018上市企业市值行业分布

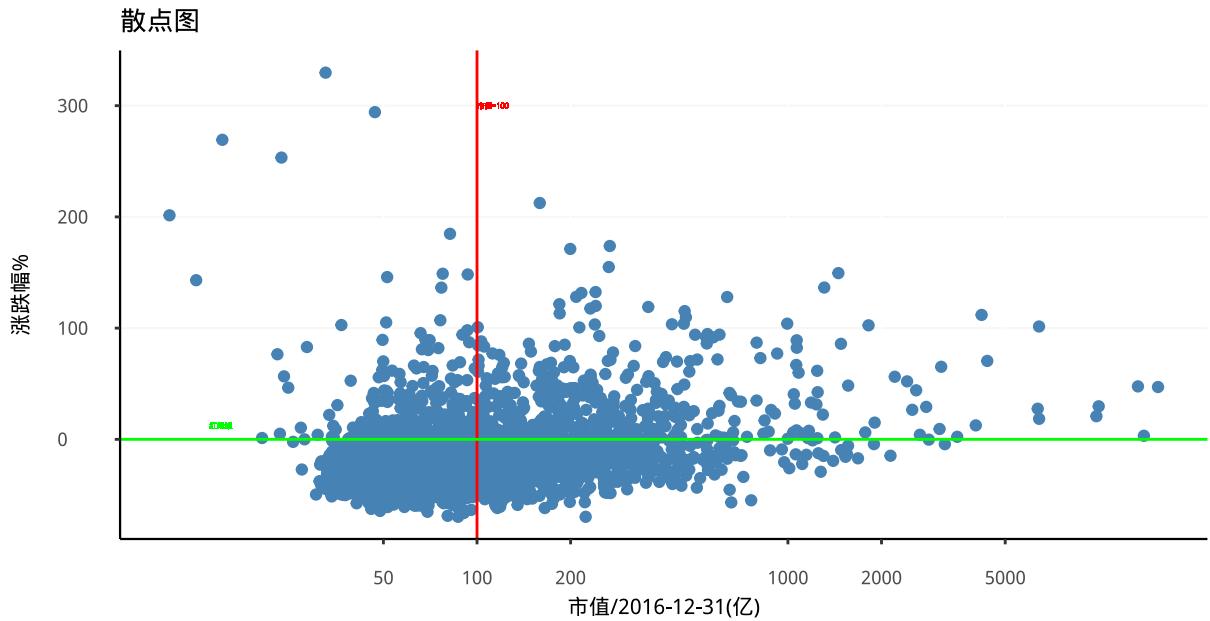


注:面积代表行业市值,颜色深度与行业企业数成正比

3.7 其他常用图形

* 辅助工具 `geom_hline`,`geom_vline`,`geom_abline`,`geom_text` 等。

```
p6_1 <- p1+geom_point(colour='steelblue')+ggtitle('散点图')+
  geom_vline(xintercept =100,color='red')+
  geom_text(aes(x=100,y=300),label='市值=100',color='red',size=1,hjust=0)+
  geom_hline(yintercept =0,colour='green')+
  geom_text(aes(x=15,y=10),color='green',label='红绿线',vjust=0,size=1)
p6_1
```



4 图形布局

常用图形布局可以分为三类，包括分页图，多图和复合图。

4.1 分面图 facet_grid() 和 facet_wrap.

分面图就是将分组数据以并列图呈现中，是比较实用的工具。按照排列方式分为网格形 `facet_grid()` 和 `facet_wrap()` 两种类型。

- `facet_grid`: 生成一个 2 维的面板网格，行和列分类通过变量定义。`facet_grid(var1 var2, margins = FALSE, scales = "fixed", space = "fixed", ...)`#var1 行分类，var2 列分类。scales 控制并列图的横纵坐标是否固定，网格线作业。
- `facet_wrap`: 生成一个一维的并列图，由于排列顺序，因此结果呈现二维形式。`facet_wrap(var1 + var2 + ..., ncol = , scales = "fixed", strip.position = "top")`,dir 控制摆放顺序,strip 控制标签位置,ncol 控制列数量，流水线作业。

```
stock_c <- subset(stock, stock$province %in% c('福建省', '广东省'))
p7_1 <- ggplot(stock_c, aes(x=change)) + theme_os() +
  labs(title='闽粤涨跌分布:网格分面', x='涨跌幅', y='频率', fill=NULL) +
  geom_density(aes(y=..density.., fill=province), color='white') +
  facet_grid(.~province)#行分类~列分类，此处可替换成facet_wrap(~province, ncol=2)
p7_2 <- ggplot(stock_c, aes(x=change, fill=province)) + theme_os() +
  labs(title='闽粤涨跌分布:双变量网格分面', x='涨跌幅', y='频率', fill=NULL) +
```

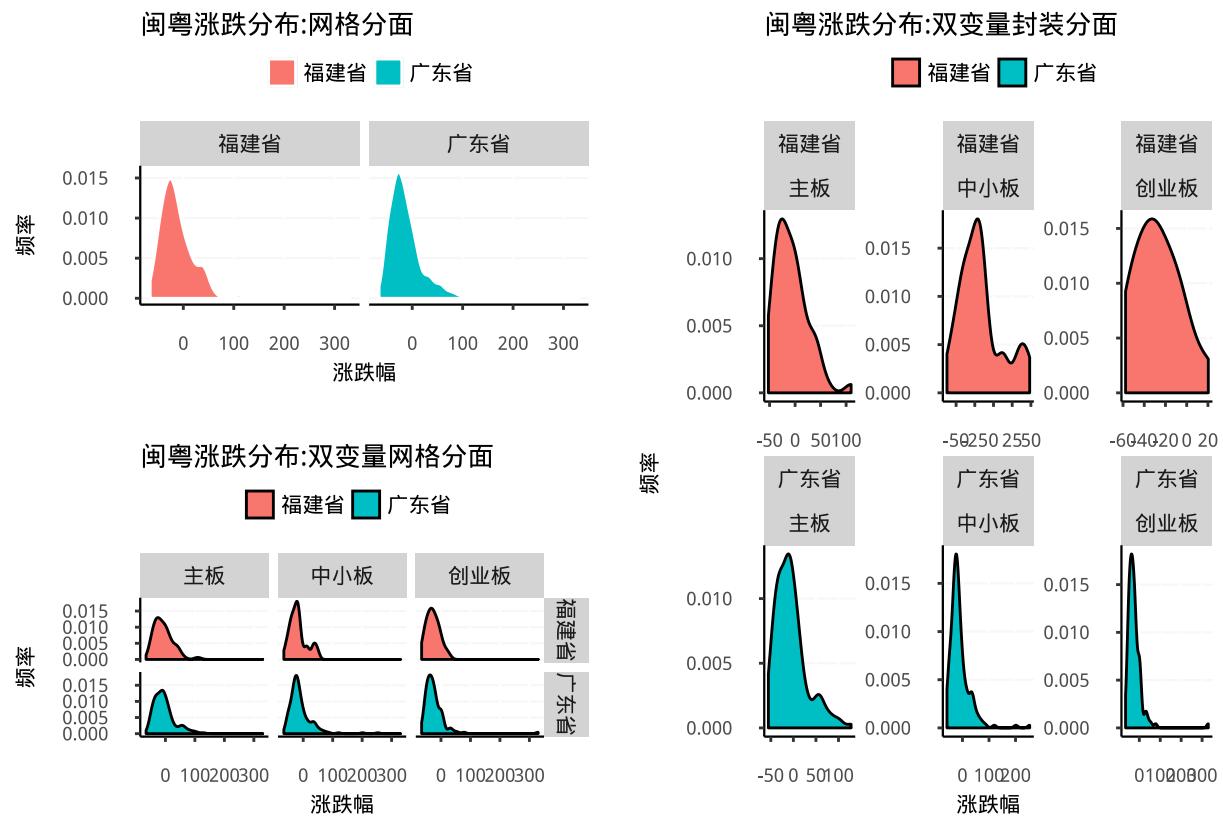
```

geom_density(aes(y=..density..))+
facet_grid(province~board)

p7_3 <- ggplot(stock_c,aes(x=change,fill=province))+theme_os()+
labs(title='闽粤涨跌分布:双变量封装分面',x='涨跌幅',y='频率',fill=NULL)+  

geom_density(aes(y=..density..))+
facet_wrap(~province+board,ncol=3,dir = "h",scales='free')#dir控制按行或列摆放

```



4.2 一页多图

一页多图按字面意思就是将图形组合放在一幅图中，常用以下两种方式：下述两种方式产生结果同折线图一页。

- Rmisc::multiplot(plotlist,cols=...,layout=...). 核心就是通过矩阵来控制你的图形结构。

```

#结果图与geom_line中的一节相同
library(Rmisc)
multiplot(p2_1,p2_2,p2_3,p2_4,layout=matrix(c(1,1,2,2,3,3,3,3,4,4,4,4),nrow=3,byrow=T))

```

- grid::viewport()

```

require(grid)
vplayout <- function(x,y){
  viewport(layout.pos.row=x,layout.pos.col=y)
}
grid.newpage()
pushViewport(viewport(layout=grid.layout(3,4)));
print(p2_1, vp=vplayout(1,1:2));
print(p2_2, vp=vplayout(1,3:4))
print(p2_3, vp=vplayout(2,1:4))
print(p2_4, vp=vplayout(3,1:4))
dev.off()
#结果图与geom_line中的一节相同

```

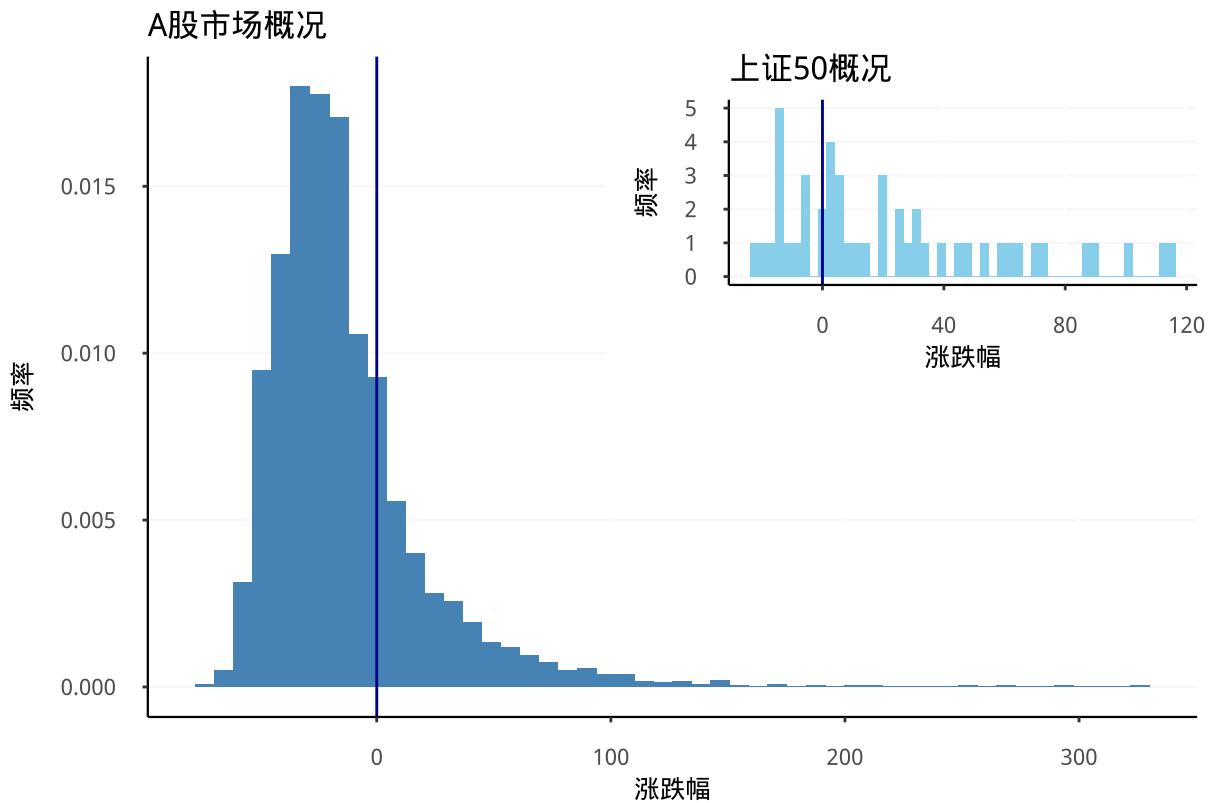
4.3 复合图

复合图和一页多图类似, 用 `viewport` 确定子图位置 `print(plot,viewport(x,y,width,height,just =))`

```

p8 <- ggplot(stock,aes(x=change),alpha=0.2,size=0.2) +
  labs(title='A股市场概况',x='涨跌幅',y='频率') +
  geom_histogram(aes(y=..density..),bins=50,fill='steelblue') +
  geom_vline(xintercept = 0,color='darkblue')+theme_os(sd_size = 1.2)
p8_1 <- ggplot(stock_50,aes(x=change),alpha=0.2,size=0.2) +
  labs(title='上证50概况',x='涨跌幅',y='频率') +
  geom_histogram(aes(),bins=50,fill='skyblue') +
  geom_vline(xintercept = 0,color='darkblue')+theme_os(sd_size = 1.2)
p8
library(grid)
vp <- viewport(width=0.5,height=0.45,just=c('left','bottom'))#用viewport函数指定子图的大小的位置
print(p8_1, vp=vp)

```

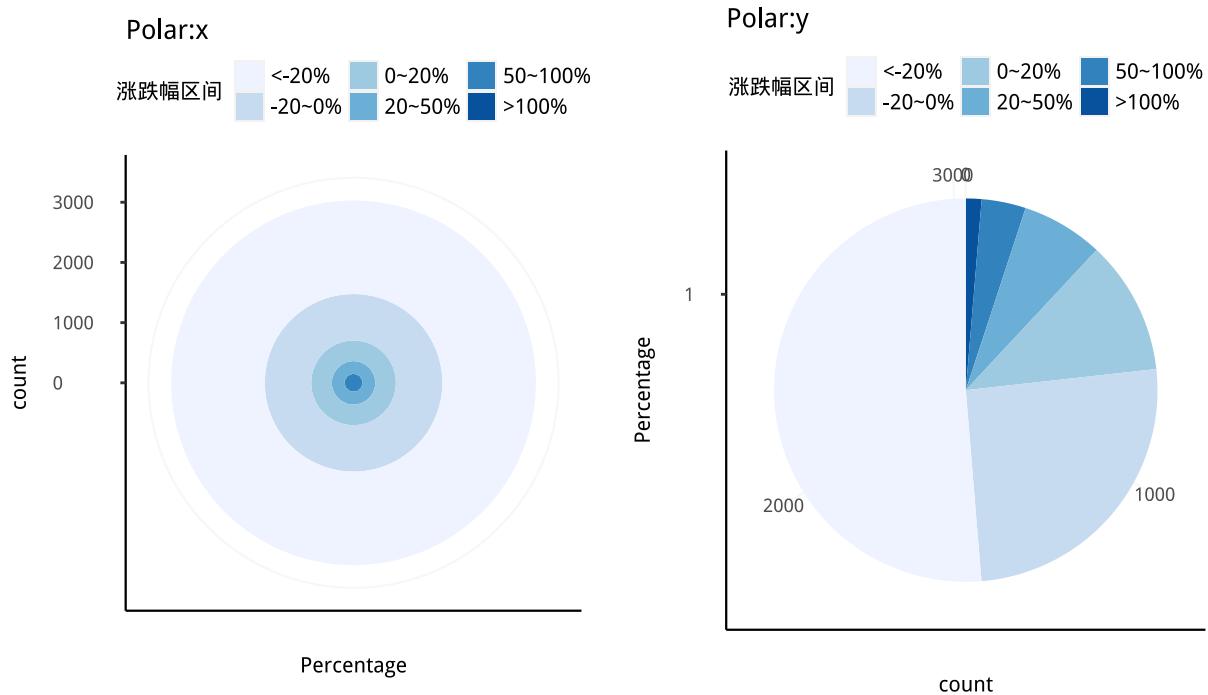


5 图形属性

5.1 坐标与坐标系

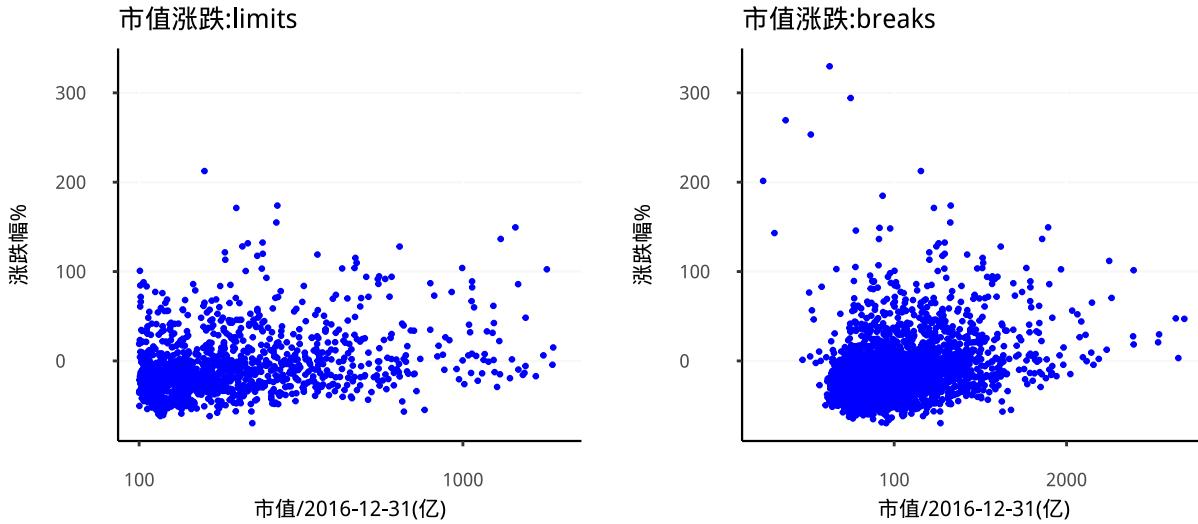
- 交换 x,y 轴,coord_flip
- 控制 x,y 轴比例,coord_fixed(ratio=1)
- 极坐标 coord_polar(theta = “x/y”,start=,)# 选择 x,y 一者作为角度;
- 坐标转换, coord_trans(x='log10',y='log10',...)

```
p9 <- ggplot(stock,aes(x=factor(1),fill=change_d))+geom_bar(width = 1)+  
  ggtitle('柱形图')+theme_os()+labs(x='Percentage',fill='涨跌幅区间')  
#width=1是为了x映射为角度时能够在转化为极坐标时候形成闭合  
p9_1 <- p9+coord_polar(theta='x')+ggtitle('Polar:x')+scale_fill_brewer()#将x映射为角度  
p9_2 <- p9+coord_polar(theta='y')+ggtitle('Polar:y')+scale_fill_brewer()#将y映射为角度  
library(Rmisc)  
multiplot(p9_1,p9_2,cols=2)
```



- x,y 轴值域,coord_cartesian(xlim=,ylim=), 可以简写为 xlim(),ylim(),expand 控制 x,y 轴溢出。要注意 limits 和 breaks 的区别。

```
p9_3 <- p1_1
p9_4 <- p9_3+scale_x_log10(limits=c(100,2000))+ggtitle('市值涨跌:limits')#控制x的顺序和个数
p9_5 <- p9_3+scale_x_log10(breaks=c(100,2000))+ggtitle('市值涨跌:breaks')#分割点
library(Rmisc)
multiplot(p9_4,p9_5,cols=2)
```



5.2 标度与图例

官方图例组件

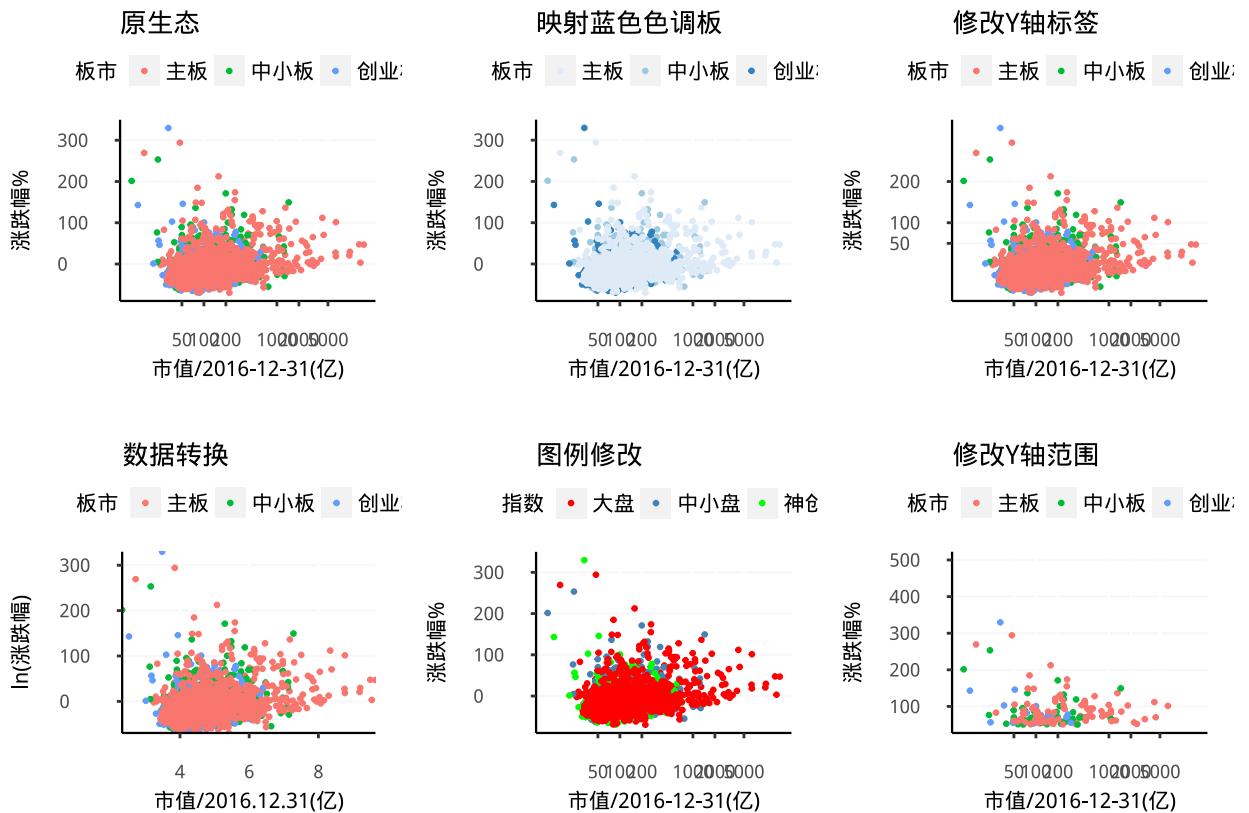
标度是数据到图形属性的映射，包括 color,lty,fill,position,shape 等类型，主要参数：names 控制标题，breaks 控制显示的标签，values 控制图例颜色，labels 控制图例标签名称等。按照函数主要包括颜色标度，位置标度，手动标度和同一型标度，主要形式为 scale_ 图形属性 _ 标度如 scale_fill_gradient,scale_x_continuous,scale_colour_manual 等。也可以通过 guides(标度 =guide_legend(属性 =)),theme(legend.xxx=) 方式来修改。

```

p10 <- p1+geom_point(aes(colour=board),size=0.5)+theme_os()+
  labs(title='原生态',colour='板市')
p10_1 <- p10+scale_colour_brewer()+labs(title='映射蓝色色调板',colour='板市');
#颜色修改，离散型:scale_xxx_hue/brewer/manual,连续型:scale_xxx_gradient/2/n
p10_2 <- p10+scale_colour_manual('指数',values=c('red','steelblue','green'),
  labels=c('大盘','中小盘','神创'))+
  labs(title='图例修改',colour='板市')
#自定义颜色、图例,guide=F控制是否显示图例
p10_3 <- p10+scale_y_continuous(breaks=c(50,100,200),minor_breaks = c(75,150),
  labels=c(50,100,200))+ggtitle('修改Y轴标签')
p10_4 <- p10+scale_y_continuous(limits=c(50,500))+ggtitle('修改Y轴范围')
#limits控制显示的范围, breaks显示刻度
p10_5 <- ggplot(stock,aes(x=log(mv17),y=change))+
  geom_point(aes(color=board),size=0.5)+
  scale_x_continuous(expand=c(0,0))+ 
  labs(title='数据转换',y='ln(涨幅)',x='市值/2016.12.31(亿)',colour='板市')+
  scale_y_continuous(expand = c(0,0))+theme_os()
#注意两者的标签不同,原因在于一个改变数据一个改变标度,且前者转化数据导致数据丢失,expand控制溢出

```

```
library(Rmisc)
multiplot(p10,p10_1,p10_3,p10_5,p10_2,p10_4,layout=matrix(c(1,2,3,4,5,6),nrow=2,byrow=T))
```



以 fill 为例常用图例操作

说明

guides(colour/fill/shape/size=FALSE)	移除图例, 其中 fill 为图例所映射的变量
scale_fill_discrete(limits=)	修改图例项目的顺序
guides(fill=guide_legend(reverse=TRUE))	反转图例顺序 (属性 fill)
labs(fill=),guides(fill=guide.legend(title=))	设置图例标题 (属性 fill)
scale_fill_discrete(limits=,labels=)	修改图例文本
theme(legend.position="bottom"/c(x,y))	修改图例的位置/置于图片中
theme(legend.xxx=)	修改图例外观

6 主题与配色

主题主要围绕图形元素文本类、线条类、矩形、空白参数修改

- element_rect()
- element_rect(fill, colour, size, linetype, color, inherit.blank)

- element_line(colour,size,linetype,lineend,color,arrow,inherit.blank)
- element_text(family,face,colour,size,hjust,vjust,angle,lineheight,color,margin,debug,inherit.blank)

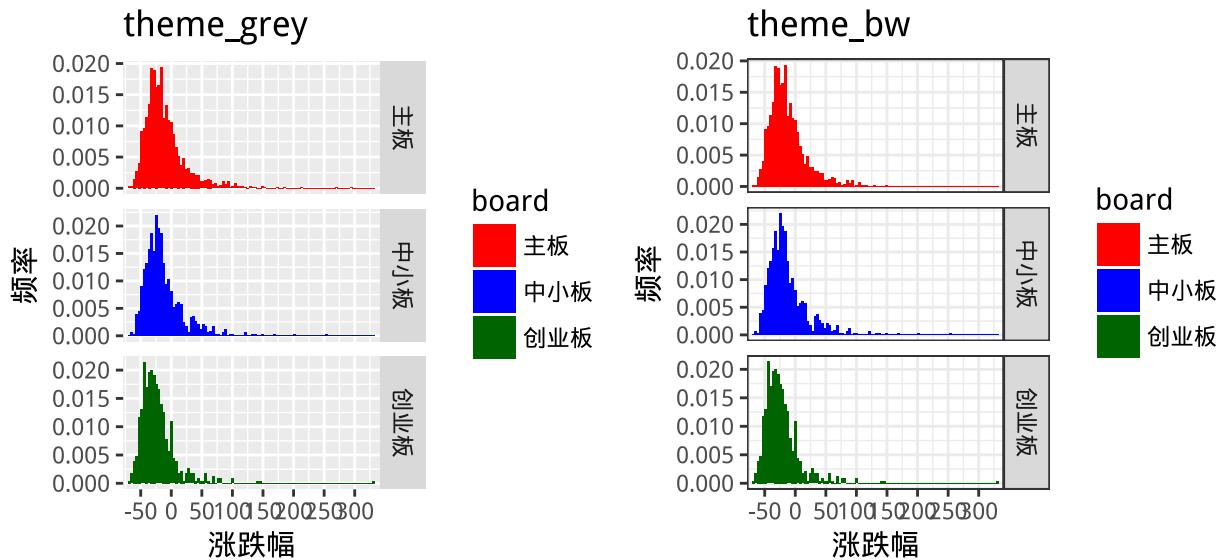
6.1 主题

6.1.1 默认主题

GGPLOT2 默认主题为 theme_bw() 和 theme_grey()。

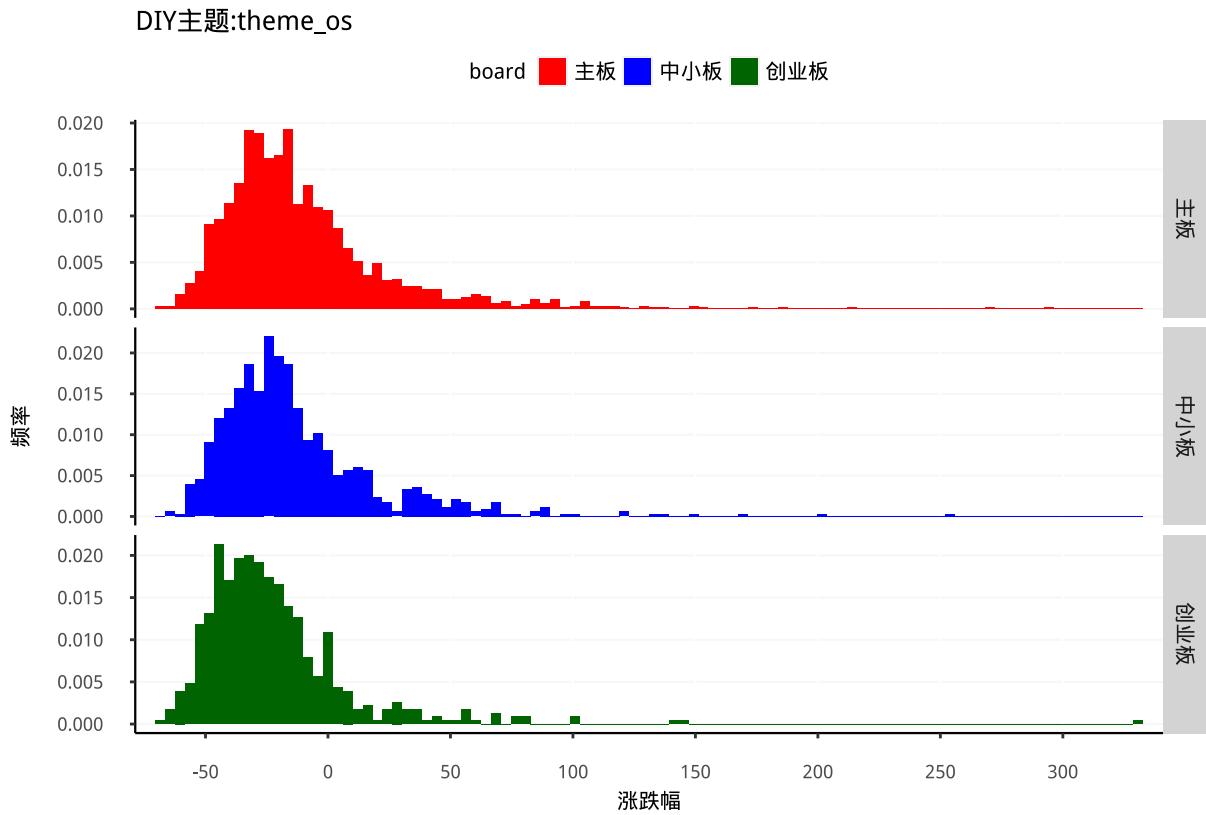
```
library(scales)
p11_grey <- ggplot(stock,aes(x=change,fill=board))+ #分组并列柱形图
  geom_histogram(aes(y=..density..),bins=100)+facet_grid(board~.,scales='free_y')+ 
  labs(title='theme_grey',x='涨跌幅',y='频率',fill='板市')+ 
  scale_fill_manual('board',values=c('red','blue','darkgreen'))+ 
  scale_x_continuous(breaks=seq(-100,400,50),expand=c(0.02,0.02))#控制x轴溢出
p11_white <- p11_grey+theme_bw()#默认白色主题
```

```
library(Rmisc)
multiplot(p11_grey,p11_white,cols=2)
```



6.1.2 自定义主题

```
p11_DIY <- p11_grey+theme_os()+ggtitle('DIY主题:theme_os');p11_DIY
```



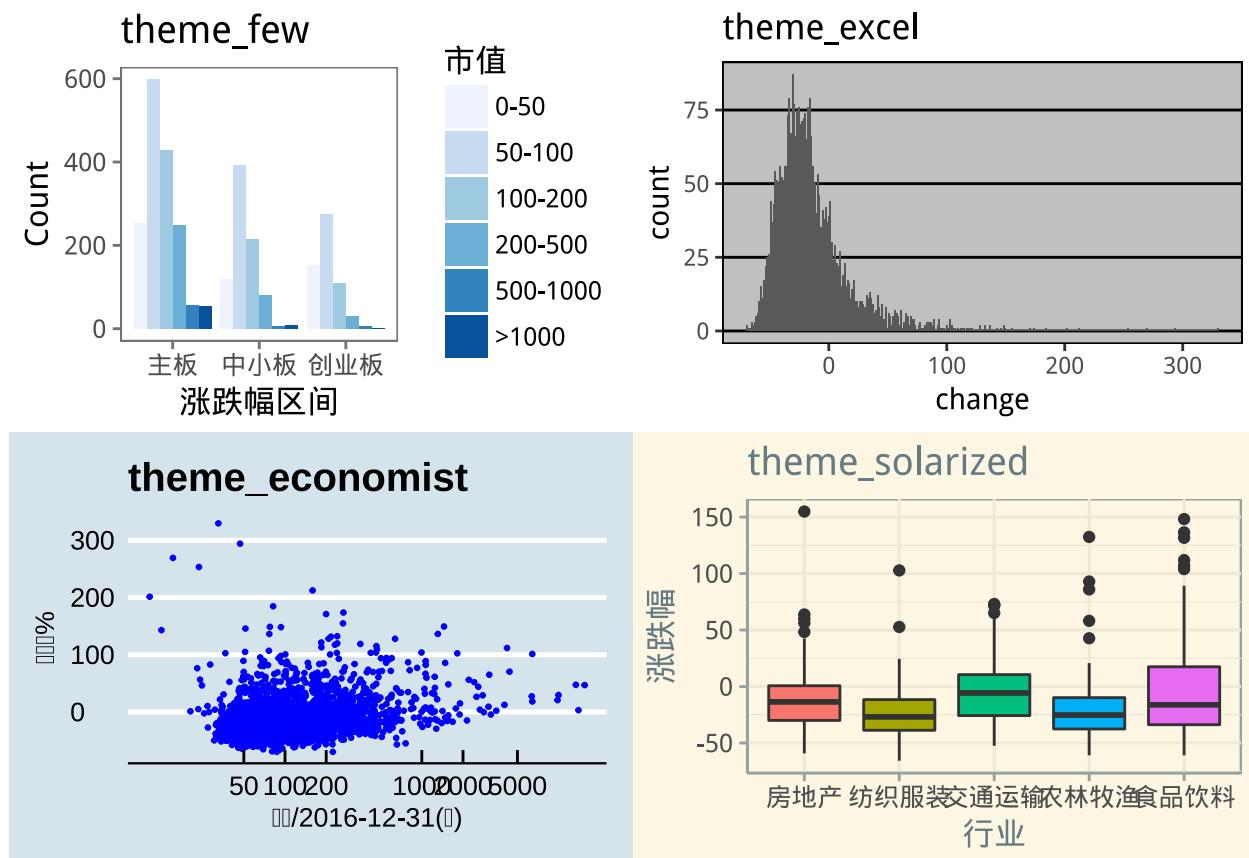
6.1.3 套用主题模板

- `ggtheme`:

[ggtheme 的模板说明](#)

ggtheme 主题	说明
<code>theme_base</code>	ggplot 默认设置
<code>theme_calc</code>	LibreOffice Calc 图表
<code>theme_few</code>	简洁型
<code>theme_economist</code>	经济类主题
<code>theme_economist_white</code>	经济类主题
<code>theme_excel</code>	类似经典 excel 图表
<code>theme_hc</code>	Highcharts JS
<code>theme_map()</code>	一个简洁的地图主题
<code>theme_pander</code>	pander 的默认主题
<code>theme_stata</code>	基于 Stata graph schemes 的主题
<code>theme_tufte</code>	基于数据墨水最大化无边界, 轴线, 网格

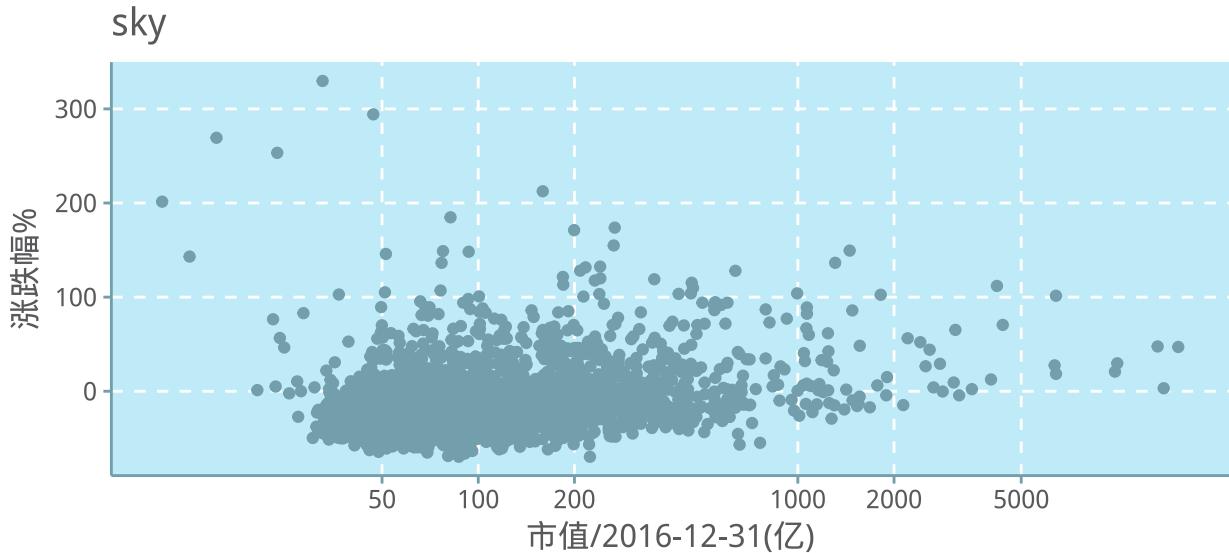
```
library(ggthemes)
p11_1 <- p3_1+theme_few()+ggtitle('theme_few')+scale_color_few()
p11_2 <- p1_1+theme_economist()+ggtitle('theme_economist')+scale_color_economist()
p11_3 <- p4_1+theme_excel()+ggtitle('theme_excel')+scale_color_excel()
p11_4 <- p5_1+theme_solarized()+ggtitle('theme_solarized')+scale_color_solarized()
multiplot(p11_1,p11_2,p11_3,p11_4,cols=2)
```



- ggthemr:

关于 ggthemr 的说明

```
#在引用时。先申明主题
library(ggthemr)
p11_v1 <- ggplot(stock,aes(x=mv17,y=change))+  
  labs(x='市值/2016-12-31(亿)',y='涨跌幅%')+  
  scale_x_log10(breaks=c(0,50,100,200,1000,2000,5000))+geom_point()  
ggthemr('sky')  
p11_v1 +ggtitle('sky')
```



```
ggthemr_reset()
```

一般套用模板的结果都会产生一些不大衔接的地方，可以通过查看主题模板的函数 `themexxx` 来修改，适用于自己的图片。

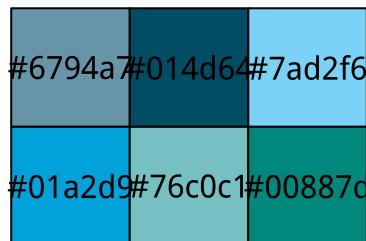
6.2 配色

- 调用主题包中的颜色模板，模式为 `scale_fill/colour_theme(params.)`

```
library(ggthemes)
library(scales)
economist_pal()(6)#查看主题的配色代码
```

```
## [1] "#6794a7" "#014d64" "#7ad2f6" "#01a2d9" "#76c0c1" "#00887d"
```

```
show_col(economist_pal()(6))#查看颜色代码的颜色
```



- 调用 RColorBrewer 包中的配色卡，[RColorBrewer](#)网站有丰富的配色方案。

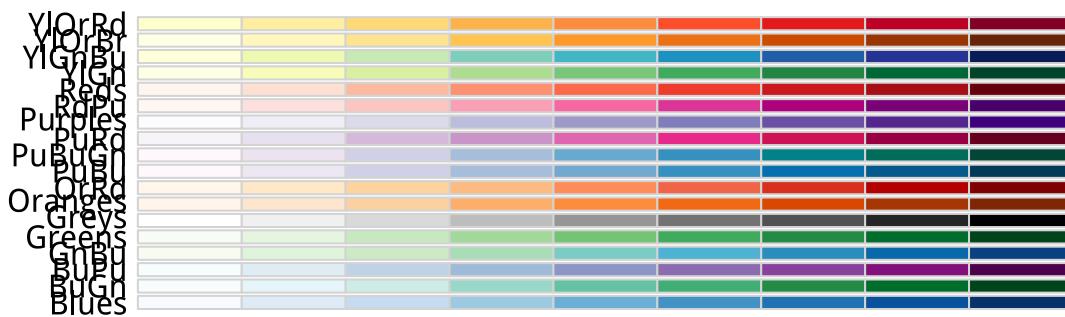
```
library(RColorBrewer)  
rownames(brewer.pal.info)[1:5] #查看所有配色板名称
```

```
## [1] "BrBG" "PiYG" "PRGn" "PuOr" "RdBu"
```

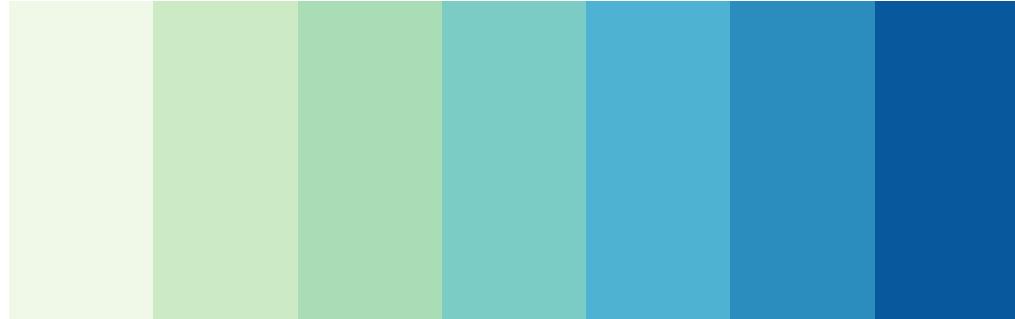
```
head(brewer.pal.info) #查看配色板信息
```

##	maxcolors	category	colorblind
## BrBG	11	div	TRUE
## PiYG	11	div	TRUE
## PRGn	11	div	TRUE
## PuOr	11	div	TRUE
## RdBu	11	div	TRUE
## RdGy	11	div	FALSE

```
display.brewer.all(type='seq')#展示所有单色配色板，修改为div, qual, all查看其他类型
```



```
display.brewer.pal(7, 'GnBu')#展示GnBu调色板颜色
```

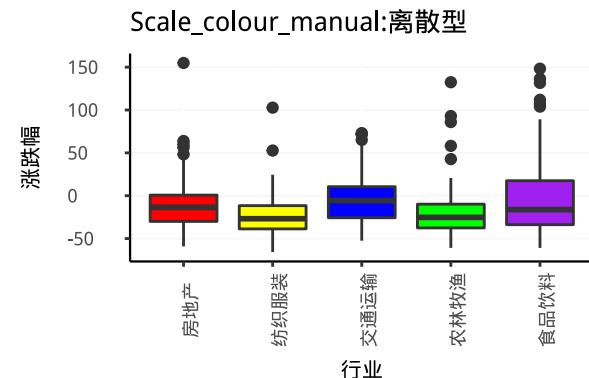
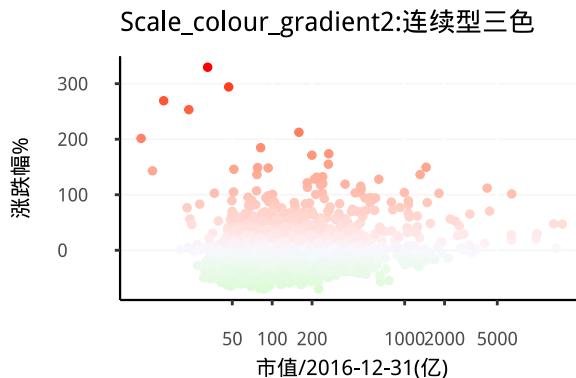
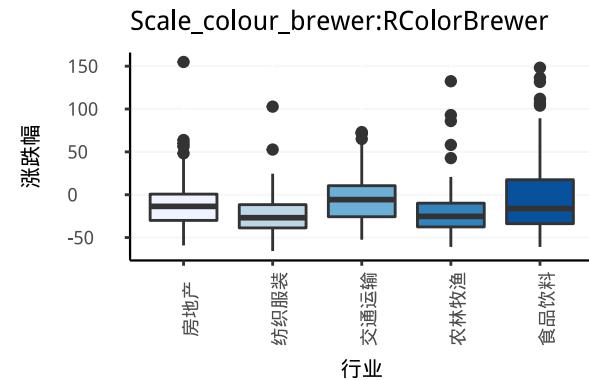
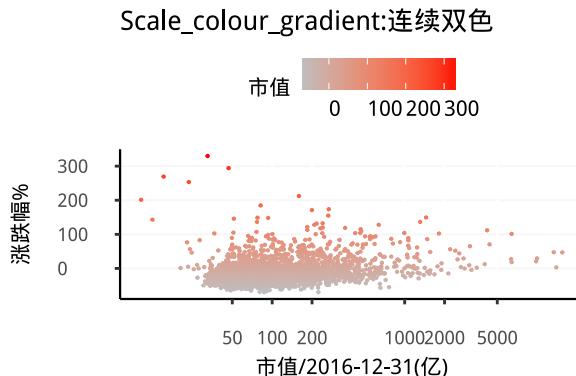


GnBu (sequential)

```
#也利用scale包查看颜色:show_col(brewer_pal(palette = 'Reds')(6))
```

- 手动加载颜色,
- 离散型: scale_colour/fill_manual()
- 连续型: scale_colour/fill_gradient/2/n()

```
p12 <- p1+geom_point(aes(colour=change),size=0.1)+  
  ggtitle('Scale_colour_gradient:连续双色')+  
  scale_colour_gradient('市值',low = "grey",high = "red")+theme_os()  
p12_1 <- p1+geom_point(aes(colour=change),size=1)+  
  ggtitle('Scale_colour_gradient2:连续型三色')+  
  theme_os() +scale_colour_gradient2(low="green",mid="#F8F8FF",high="red")+guides(colour=F)  
p12_2 <- p5_1+scale_fill_brewer(palette = 1)+ggtitle('Scale_colour_brewer:RColorBrewer')  
p12_3 <- p5_1+scale_fill_manual(values = c('#FF0000','#FFFF00','#0000FF','#00FF00','#A020F0'))+  
  ggtitle('Scale_colour_manual:离散型')  
multiplot(p12,p12_1,p12_2,p12_3,cols=2)
```



7 数据地图

数据地图主要分为 shp 和 json 两类，以 rgdal 包读取,ggplot2 用 geom_polygon()/geom_sf() 画图。

```
library(maptools);library(rgdal);library(magrittr)
map_data_c <- rgdal::readOGR('DataWarehouse-master/Rstudy/CHN_adm/bou2_4p.shp')#适用json和shp

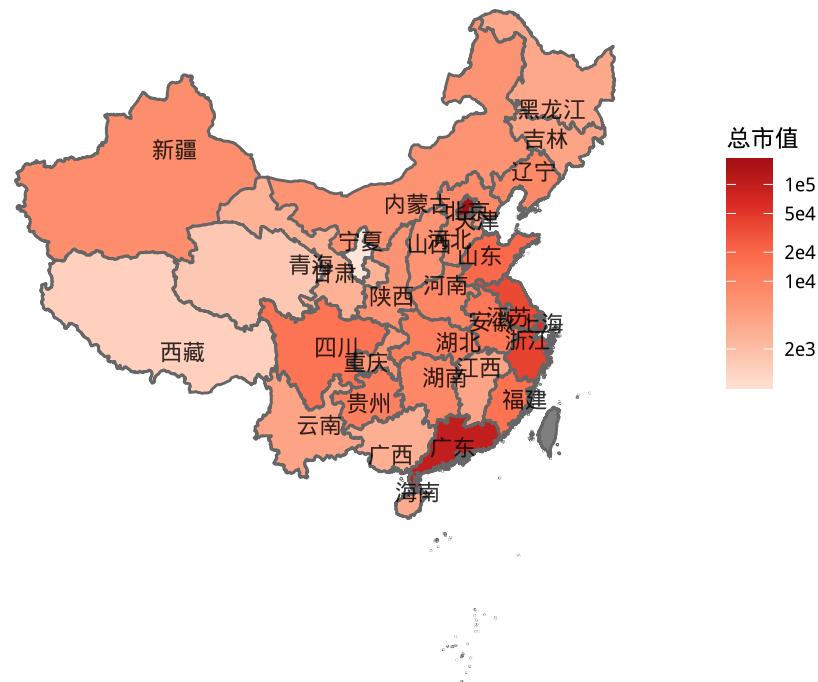
## OGR data source with driver: ESRI Shapefile
## Source: "DataWarehouse-master/Rstudy/CHN_adm/bou2_4p.shp", layer: "bou2_4p"
## with 925 features
## It has 7 fields
## Integer64 fields read as strings: BOU2_4M_ BOU2_4M_ID

p_map <- ggplot(map_data_c,aes(x=long,y=lat,group=group))+  

  geom_polygon(fill='white',colour='grey')+coord_map('polyconic')#绘制中国地图轮廓
city_df <- read.csv('DataWarehouse-master/Rstudy/CHN_adm/chinaprovincecity.csv')%>%
  set_colnames(c('NAME','city','long','lat','index','class'))
china_amd <- map_data_c@data %>% data.frame(.,id=seq(0:924)-1)#省份信息
```

```
map_df_c <- fortify(map_data_c) %>% plyr::join(.,china_amd,type='full')#地图转化为数据框
library(plyr)
stock_mv <- ddply(stock,.province,summarize,mv_total=sum(mv18))%>%
  set_colnames(c('NAME','mv_total'))#各省上市企业总市值
stock_c <- data.frame(table(stock$province)) %>% set_colnames(c('NAME','count')) %>%
  plyr::join(stock_mv,.,by='NAME')#各省上市企业数目
library(stringr)
map_df_c$NAME <- map_df_c$NAME%>%as.character(.)%>%str_sub(.,1,2)%>%
  str_replace(.,'黑龙','黑龙江')%>%str_replace(.,'内蒙','内蒙古')%>%as.factor()
stock_c$NAME <- stock_c$NAME%>%as.character(.)%>%str_sub(.,1,2)%>%
  str_replace(.,'黑龙','黑龙江')%>%str_replace(.,'内蒙','内蒙古')%>%as.factor()
city_df$NAME <- city_df$NAME%>%as.character(.)%>%str_sub(.,1,2)%>%
  str_replace(.,'黑龙','黑龙江')%>%str_replace(.,'内蒙','内蒙古')%>%as.factor()
map_df <- plyr::join(map_df_c,stock_c,by='NAME',type='full')
cit_df <- plyr::join(stock_c,city_df[,1:4],by='NAME',type='inner')
p_map1 <- ggplot(map_df,aes(x=long,y=lat,fill=log(mv_total))) +
  geom_polygon(aes(group=group),colour="grey40")+
  scale_fill_distiller('总市值',breaks=c(7.6,9.21,9.90,10.82,11.51),
    labels=c('2e3','1e4','2e4','5e4','1e5'),
    palette = 'Reds',direction = 0)+#指定渐变填充色，可使用RGB
  coord_map("polyconic")+ggtitle('中国上市公司总市值各省份分布')+
  geom_text(aes(x=long,y=lat,label=NAME),size=3,colour='black',fontface='bold',
    alpha=0.8,data=cit_df,check_overlap = F)+
  theme_map() %+replace% theme(legend.position = c(0.9,0.4),legend.text.align=1,
    plot.title=element_text(margin = margin(t=10),
    hjust=0.5,face='bold',size=rel(1.5)))
p_map1
```

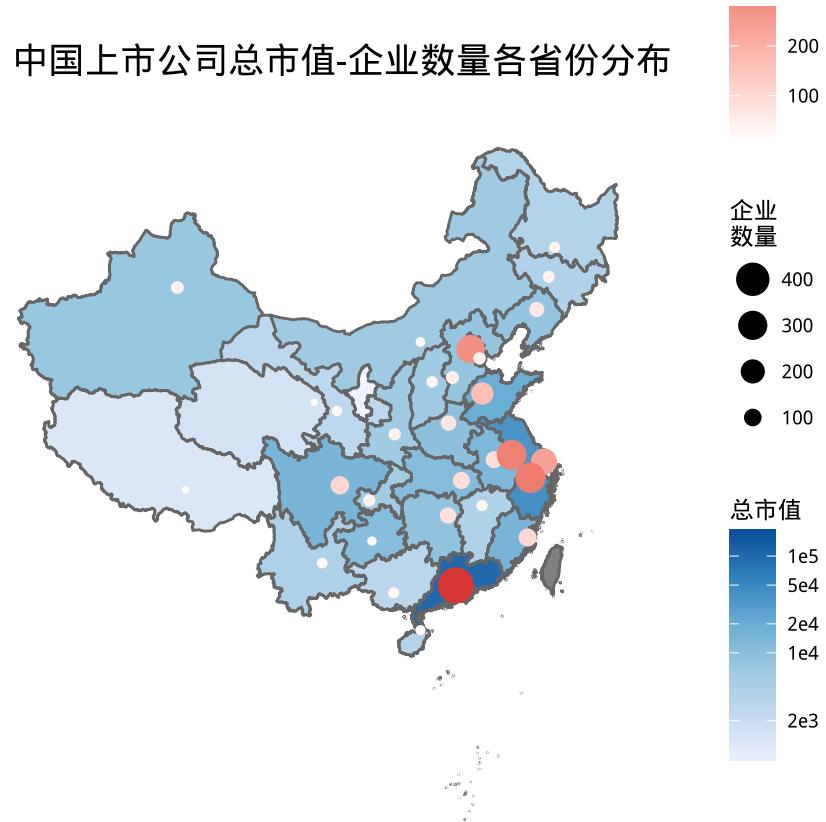
中国上市公司总市值各省份分布



```

p_map2 <- ggplot(map_df,aes(x=long,y=lat,fill=log(mv_total)),colour='white')+  
  geom_polygon(aes(group=group),colour='grey40')+  
  scale_fill_distiller('总市值',breaks=c(7.6,9.21,9.90,10.82,11.51),  
    labels=c('2e3','1e4','2e4','5e4','1e5'),  
    palette = 'Blues',direction = 0)+  
  geom_point(aes(x=long,y=lat,size=count,colour=count),shape=16,data=cit_df)+  
  scale_colour_gradient('企业\n数量',low='white',high='#D73434')+  
  scale_size_area('企业\n数量',max_size=6,guide=guide_legend(reverse = T))+  
  coord_map("polyconic")+ggtitle('中国上市公司总市值-企业数量各省份分布')+  
  theme_map() %+replace% theme(legend.position = c(0.9,0.1),legend.text.align=1,  
    plot.title=element_text(margin = margin(t=10),  
    hjust=0.5,face='bold',size=rel(1.5)))  
p_map2

```



```
## 在线地图
```

8 Shiny 介绍

9 学习帖