统计制图第二次课堂作业

中央财经大学
王思雨

一、多变量数据和高维数据的展示

➢ 数据准备与预处理

```
library("corrplot")
library("car")
library("psych")
library("ggplot2")
library("devtools")
library("foreign")
library("ggradar")
library("reshape2")
library("ggthemes")
library("RColorBrewer")
library("lubridate")
library("lattice")
library("plotrix")
library("knitr")        #读取所在分析中应用的包
setwd("D:/大数据作业/可视化/第二次作业/")  #设置工作路径。
data=read.table("dailyprice.txt",head=TRUE,sep="\t")  #读取数据集 dailyprice，设置间隔符为换行符
mydata1=data[1:1132,]    #截取 2003/01/02 一天的 1132 只股票成交记录作为截面数据
write.csv(mydata1,"D:/大数据作业/可视化/mydata1.csv",row.names = TRUE)
                          #输出 csv 文件，为 2003/01/02 日一天所有股票交易情况
mydata1 <- na.omit(mydata1) #去除空值
#构造对总市值构造分类变量，利用 summary 确定分位数，来合理平均的分为四个档次
shizhi<-vector(mode="numeric",length=0)
summary(mydata1)
for(i in 1:dim(mydata1)[1])
{
  if(mydata1$mkt_cap[i]<=1.685e+09){
    shizhi[i]=4
  }
  if(mydata1$mkt_cap[i]<=2.297e+09 &&mydata1$mkt_cap[i]>1.685e+09){
    shizhi[i]=3
  }
  if(mydata1$mkt_cap[i]<=3.732e+09 &&mydata1$mkt_cap[i]>2.297e+09){
    shizhi[i]=2
  }
  if(mydata1$mkt_cap[i]>3.732e+09 ){
    shizhi[i]=1
```

```
  }
}
shizhi=as.character(shizhi)              #将 shizhi 变为 character 变量
mydata1=data.frame(mydata1,shizhi)       #将新列 shizhi 分类变量合并到 mydata1 矩阵中形成新的分类变量
```

➢ 气泡图

```
mydata2<-mydata1[1:100,] #由于数据量太大，气泡相互重叠因此截取一个 100 行的矩阵
ggplot(mydata2,aes(x=log(mkt_freeshares),y=log(amt),size=mkt_cap))+geom_point(shape=21,colour="black",fill="lightblue")+xlab("log(自由流通市值)")
        +ylab("log(成交额)") +ggtitle('流通市值与成交额气泡图')
        +theme(plot.title = element_text(hjust = 0.5,
                family="myFont",size=18,color="red"),
                panel.background=element_rect(fill='mistyrose',color='black'))
```

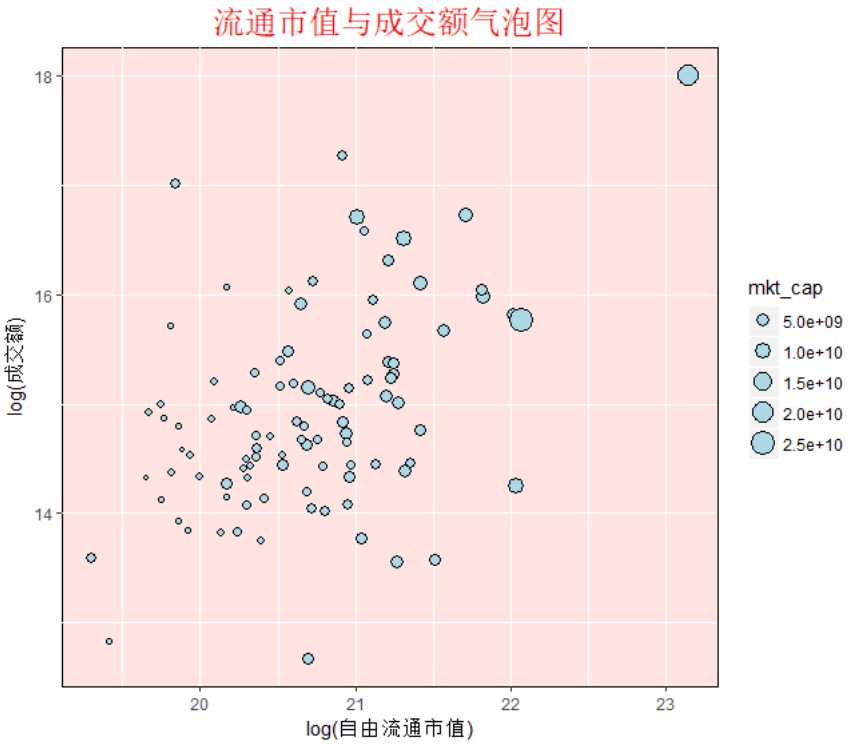

图 1-1　气泡图

此气泡图利用了采取了数据集中 2003 年 1 月 2 日截面数据的前 100 个股票的成交记录，利用“自由流通市值”的对数和“成交额”的对数值作为横纵坐标并将股票市值作为气泡大小的填充。如图所示。

➢ 散点图矩阵

```
mydata2=mydata2[,-c(1,18,19)]
scatterplotMatrix(mydata2[,1:6],main = "提取变量的散点图矩阵")
#利用 scatterplotmatrix 函数 设置各个变量的分布图与轴须图
```

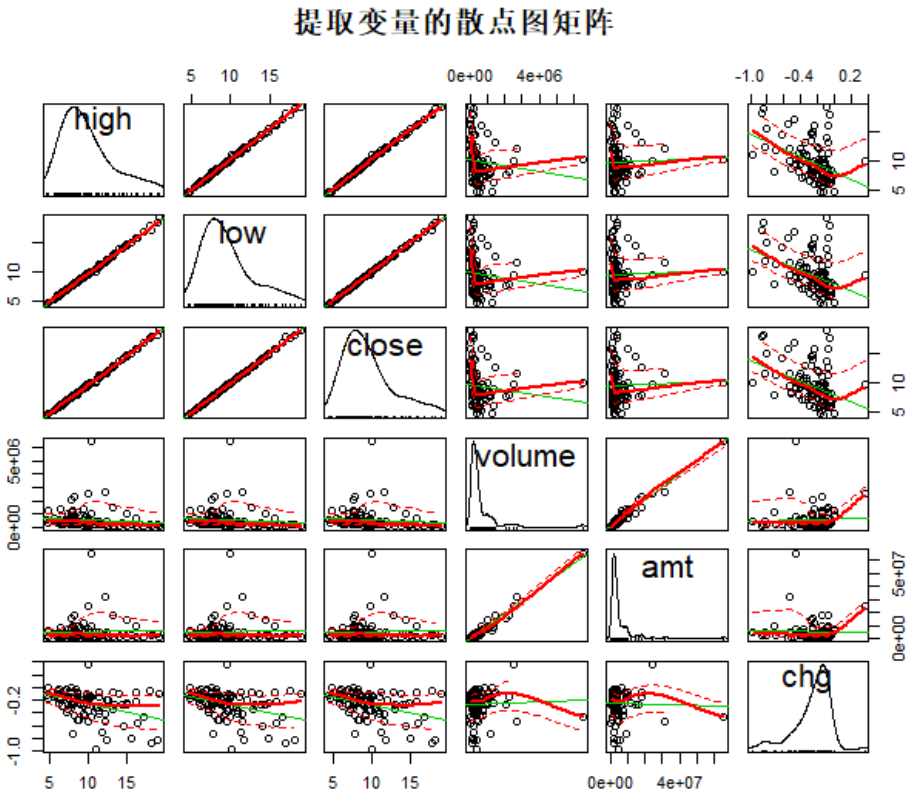

图 1-2 提取变量的散点图矩阵

此散点图矩阵应用了数据集中 2003 年 1 月 2 日截面数据的前 100 个股票的成交记录，将开盘价，成交量，收盘价，成交额等变量的之间的相关关系以及将各变量的分布凸显出来，还绘制了拟合曲线。

➢ 相关系数图

```
mydata2=mydata2[,1:12]          #提取前 12 列变量做相关系数矩阵图
mycor = cor(mydata2, use = "everything")  #计算相关系数矩阵
corrplot(mycor, order = "hclust", addrect = 3, rect.col = "black", tl.cex = 0.7,
tl.col = "black", type = "lower")
#绘制相关系数图，并调节参数并根据 order=hclust 采用相关系数衡量距离，通过聚类算法将指标聚为相关系数较为相近的聚成 3 类
```

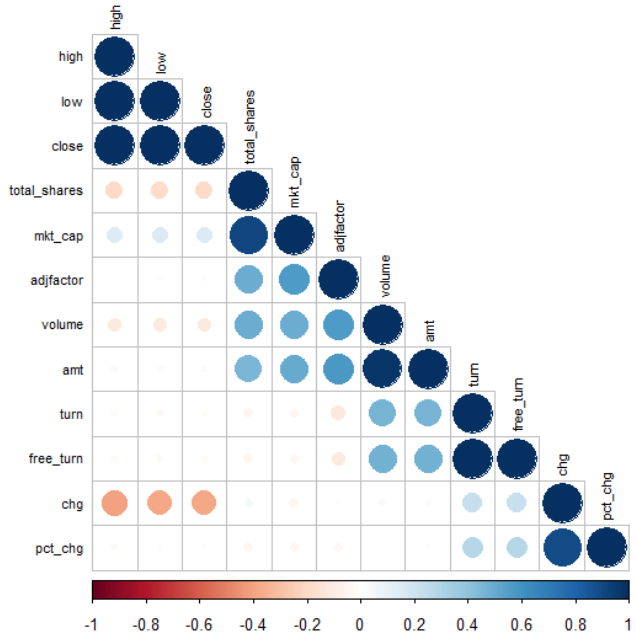

图 1-3　相关系数矩阵

```
ggpairs(mydata2[,1:8])          #利用 ggpairs 函数绘制数据集前 8 个变量的散点图矩阵和相关系数矩阵
```

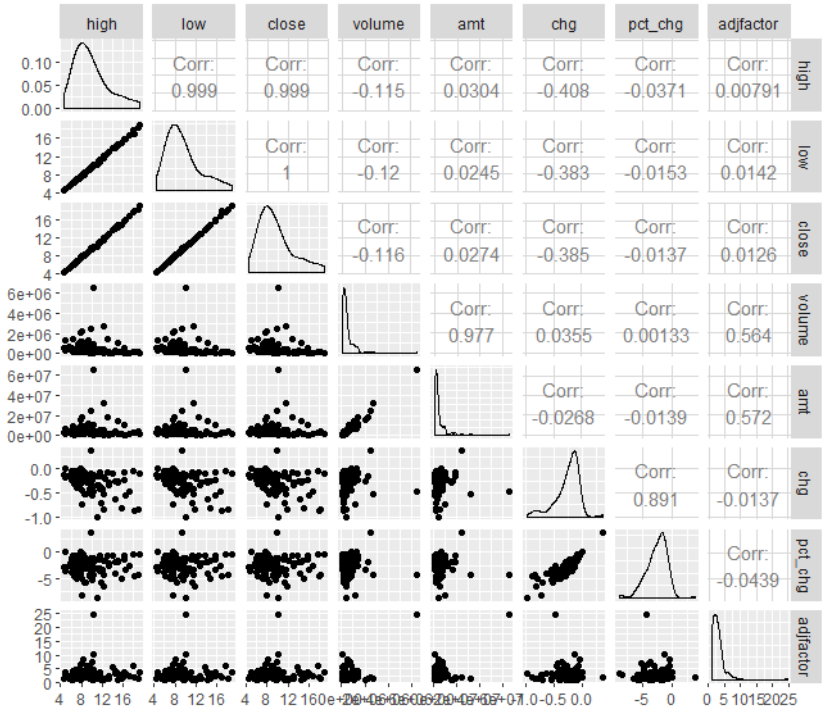

图 1-4 相关系数矩阵以及散点图分布

两个相关系数散点图矩阵应用了数据集中 2003 年 1 月 2 日截面数据的前 100 个股票的成交记录，将开盘价，成交量，收盘价，成交额等变量的之间的相

关关系以及将各变量的分布凸显出来，并计算了相关系数。图 1-3 利用圆圈的颜色深浅表示相关关系大小并将相关关系大变量聚类到一起，图 1-4 则是显示了变量之间的散点图，以及各变量的分布和相关系数。

➢ 密度图

```
##绘制核密度估计图
#生成几何对象
p<-ggplot(mydata1,aes(x=log(mkt_freeshares),y=log(amt)))
#默认等高线图
p+geom_point()+stat_density2d(aes(colour=..level..))
  +xlab("log(自由流通市值)") + ylab("log(成交额)")
  +ggtitle('流通市值与成交额密度线图')
  +theme(plot.title = element_text(hjust = 0.5,
              family="myFont",size=18,color="red"),
              panel.background=element_rect(fill='thistle2',color='black'))
```

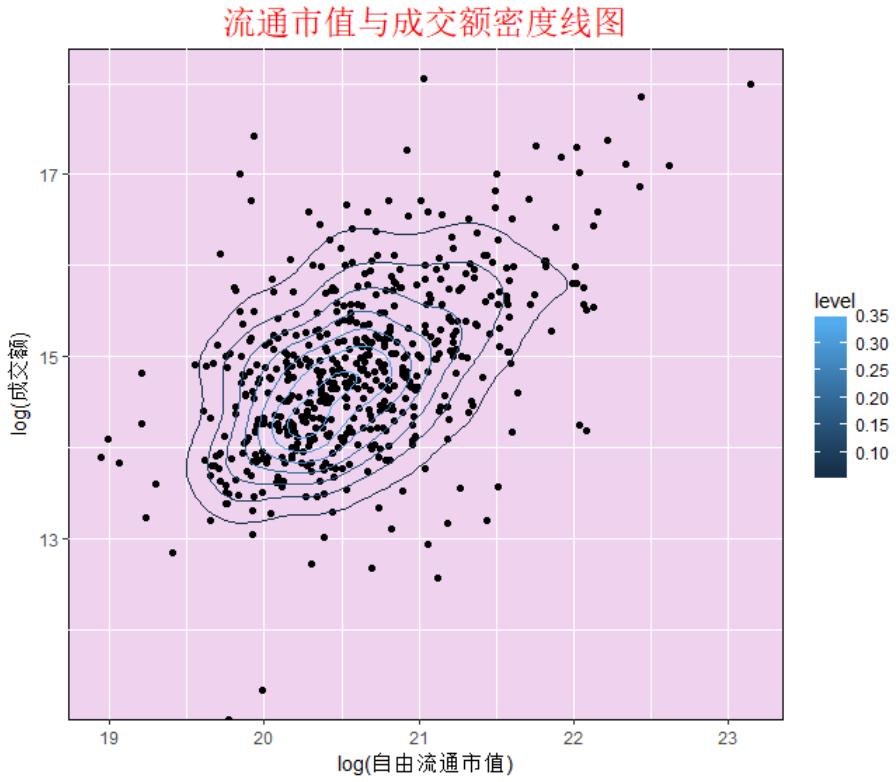

图 1-5 密度曲线图

```
##绘制瓦片密度图
p+geom_point()+stat_density2d(aes(alpha=..density..),geom="tile",contour= FALSE)
  +xlab("log(自由流通市值)") + ylab("log(成交额)")
  +ggtitle('流通市值与成交额密度线图')          #有填充颜色
  +theme(plot.title = element_text(hjust = 0.5,
            family="myFont",size=18,color="red"),
            panel.background=element_rect(fill='thistle2',color='black'))
```

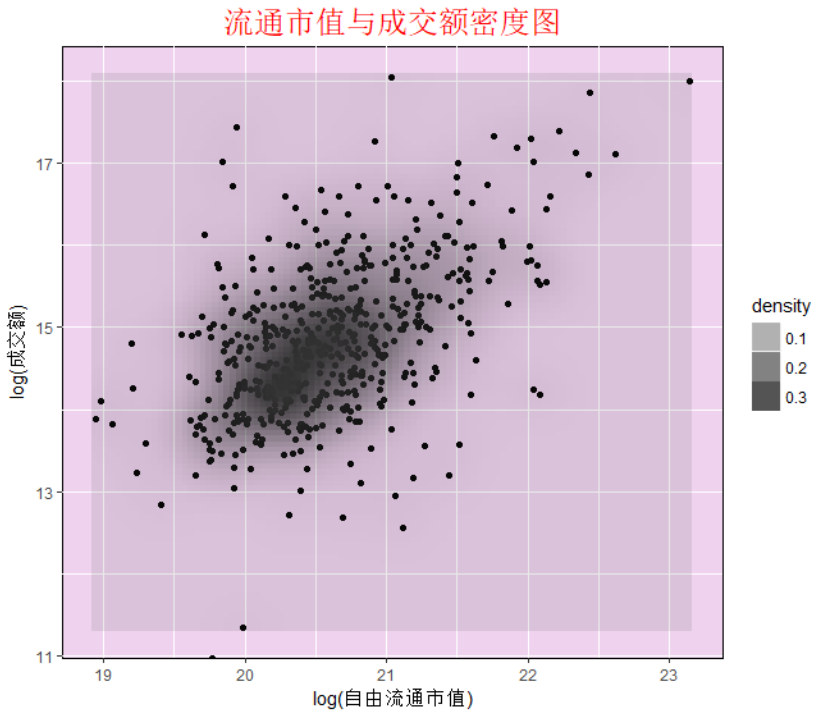

图 1-6　密度灰度图

图 1-6 显示的分别是根据核密度估计作图的密度曲线图和密度瓦片图。利用的是自由流通市值的对数值和成交量的对数值，根据样本点分布，利用核密度估计来估计某个位置的分布概率，并分别通过密度曲线和瓦片的填充在图中显现出来。

➢ 热图

```
##热图的数据的提取
a=0
yizhi<-vector(mode="numeric",length=0)
while (i <= 5322012){
  i=a*1132+1
  yizhi<-rbind (yizhi,data[i,])
  a=a+1
}      #提取出第一只股票的 2003 年-2015 年之间的每天的交易数据，构成时间序列矩阵
month<-month(yizhi$datetime)        #提取出月份
month<-month[-4703]
jidu<-vector(mode="numeric",length=0)       #构造季度向量
for(i in 1:length(month))
{
  if( month[i]>=1 &&month[i]<=3 ){
    jidu[i]="1"
  }
```

```
  if( month[i]>=4 &&month[i]<=6 ){
    jidu[i]="2"
  }
  if( month[i]>=7 &&month[i]<=9 ){
    jidu[i]="3"
  }
  if( month[i]>=10 &&month[i]<=12 ){
    jidu[i]="4"
  }
}                                          #构造季度向量，也是一个分类变量
yizhi<-yizhi[-4703,]
yizhi=data.frame(yizhi,jidu)               #合并季度向量到股票时间序列数据
year<-year(yizhi$datetime)                 #提取年度数据
year<-year[-4703]
yizhi=data.frame(yizhi,year)               #合并到股票时间序列数据
##绘制时间序列热力图
b=1
label<-vector(mode="numeric",length=0)
for(i in 1:dim(yizhi)[1])
{
  if( yizhi$close[i]>80){
    label[b]<-i
    b=b+1
  }
}
yizhi<-yizhi[-label,]                      #剔除收盘价中大于 80 的异常值
write.csv(yizhi,"D:/大数据作业/可视化/第二次作业/yizhi.csv",row.names = TRUE)
#输出构造好的时间序列矩阵
p<-ggplot(yizhi,aes(x=year,y=jidu,fill = close))
p+ geom_tile()+ scale_fill_gradient(low = "yellow",high = "red")
 #绘制热力图，并设定色域为常见的黄到红
```

绘制出的时间序列热力图如图 1-7 时间序列热力图，利用的是编号为 000001.SZ 的股票从 2003 年到 2015 年每天的交易记录中的收盘价，输出的热图有 4×13 个小方格构成，其中第 i 行 j 列的小方格展示的是第 j 年第 i 季度的收盘价信息，我们指定红色表示更高的收盘价格，黄色表示更低的收盘价格。从图中我们读到的信息有，2009 年 3 月份这只股票的收盘价创造了新高，为比较重的红色，而其他时间的收盘价格都想差不多。

图 1-7　时间序列热力图

```
##绘制样本之间的欧式距离的热力图
mydata3<-mydata1[-c(1)]
rili<-subset(mydata3,select = c(open,high,volume,chg,mkt_cap))
#提取五个变量作为绘制热力图数据
rili[,1:5]<-scale(rili[,1:5])                    #标准化处理
rili<-na.exclude(rili)                           #删除有缺失值的行
distance_rili<-dist(rili,method="euclidean")     #计算欧式距离
distance_rili<-as.matrix(distance_rili)          #数据类型转化为矩阵
distance_rili<- distance_rili[1:100,]
#由于样本量过大截取前 100 只股票作为热力图数据
heatmap(distance_rili,main="Heatmap")
##热图中区块的颜色深浅表示支股票的距离远近，邻近的点对应的方格的颜色更深，而远处的点对应的方格颜色浅
```

绘制出的 1-8 样本间的聚类热力图所示，利用 100 支股票的开盘价，收盘价，最高价，最低价，成交额五个变量进行利用欧式距离的聚类分析，热图中区块的颜色深浅表示两个股票的距离远近，邻近的点对应的方格的颜色更深，而远处的点对应的方格颜色浅。将股票种类间的类别进行了划分。

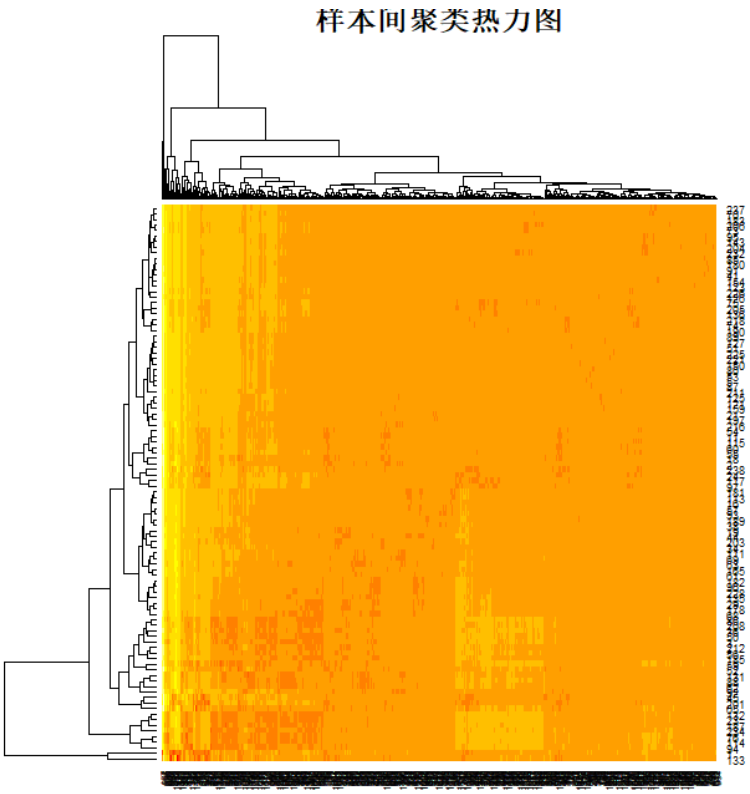

图 1-8　样本间聚类热力图

➢ 平行坐标图

```
pingxing<-subset(mydata1,select = c(open,high,volume,chg,mkt_cap,shizhi))
#提取几个变量构造平行图数据矩阵
pingxing[,1:5]<-scale(pingxing[,1:5])
#标准化处理
parallelplot(~pingxing[,1:5],pingxing,group=shizhi,horizontal.axis = FALSE)
#以市值为分组绘制平行图
```

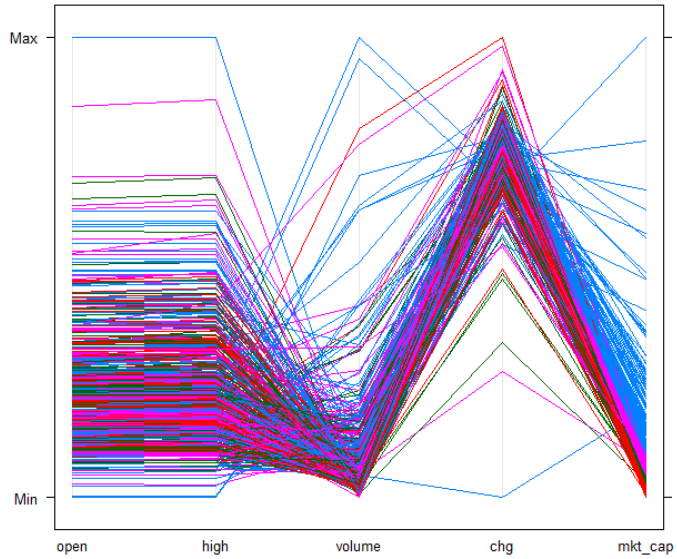

图 1-9 分组绘制平行图

```
parallelplot(~pingxing[,1:5]|shizhi, pingxing,horizontal.axis = FALSE, scales = list(x = list(rot = 90)))
#分面化处理，生成基于不同组别内的平行坐标图，反应某组内观测点的差异
```

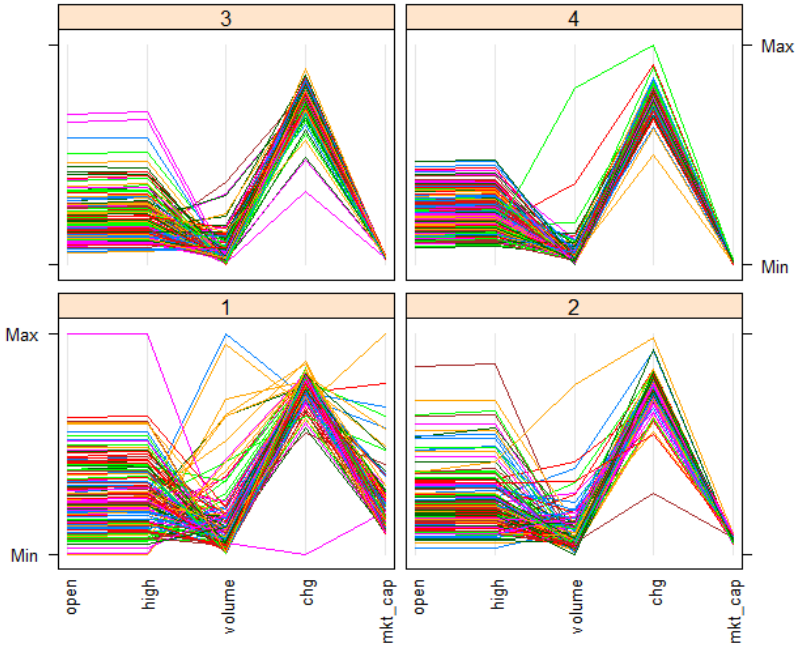

图 1-10　分面平行图

首先提取了开盘价，最高价，成交量，涨跌，总市值以及市值分类作为平行图数据矩阵，图 1-9 为以不同颜色基于不同市值组别的平行图，平行坐标轴的绘制原理是在横轴上选择几个等距的点表示不同的变量，变量的取值是经过标准化处理的值被映射到纵坐标上。图 1-10 用 lattice 包做分面化处理，生成基于不同组别 内的平行坐标图，反应某组内观测点的差异。

➢ 雷达图

```
stars(pingxing[c(1:10), -6],scale = TRUE,main="星图")
#绘制前 10 个股票标准化后观测的星图
```

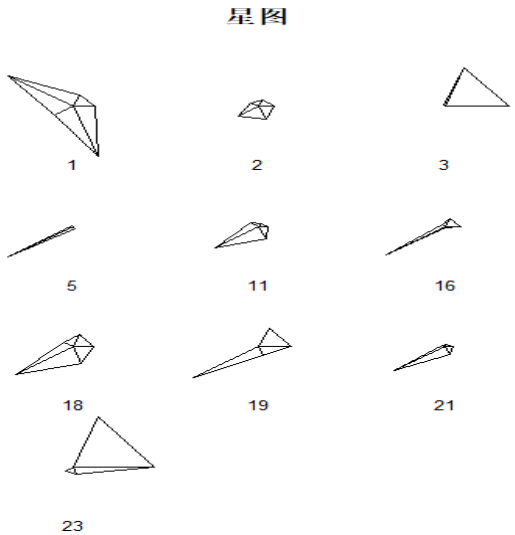

图 1-11　星图

利用数据集前五个变量和和前 10 个数据进行性图的绘制，星图向外延展的大小则是该样本对应指标上的表现相对好坏。

```
leida<-subset(mydata1,select= c(trade_code,open,high,volume,chg,mkt_cap,shizhi))
 #构造雷达图数据
stars(leida[1:4,2:6],locations=c(0,0),col.lines = 2:7,radius=FALSE,scale = TRUE,
key.loc=c(0,0),lwd=1.5)
#绘制前 4 个股票样本数据的标准化雷达图
legend(0.5,0.8,cex=0.5,legend=leida$trade_code[1:4],col=c(1:4),lty=1)
 #添加图例
leida1<-leida[1:4,-7]
for(i in 2:dim(leida1)[2])
{
  leida1[,i]<-(leida1[,i]-min(leida1[,i]))/(max(leida1[,i])-min(leida1[,i]))
}
rownames(leida1) <- LETTERS[1:4]          #整合雷达矩阵,并将数据进行归一化
ggradar(leida1)          #另一种雷达图利用基于 ggplot 插件 ggradar 进行绘制
```

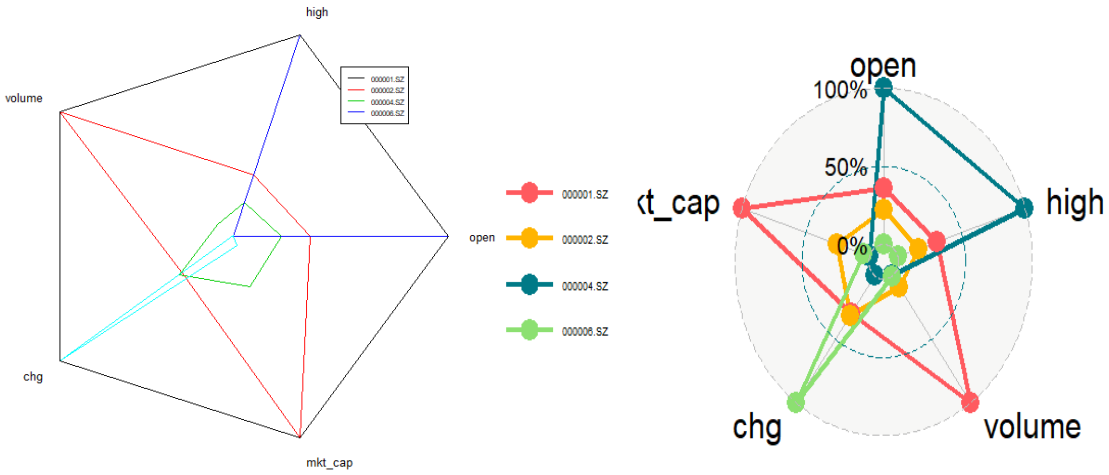

图 1-12 前 4 支股票的雷达图对比

利用前四支股票的五个变量进行雷达图的绘制，分别利用 ggplot 和原始作图的雷达图进行对比，图 1-12 左图为原始雷达图，相对来讲 ggplot 的插件 ggradar 作图更加美观。对比度更高。

➢ 交互图

```
##交互可视化
library("plotly")
p<-plot_ly(mydata1,x=close,color=shizhi,type="box")  #交互箱线图
layout(p,
       title="收盘价与市值交互式箱线图",
       xaxis=list(title="市值水平",showgrid=F),
```

```
        yaxis=list(title="频数"),
        margin=list(l=50,r=50,b=50,t=50,pad=4))
#采用 layout 函数来单独定义，并且用 xaxis、yaxis、title 对横纵坐标轴以及作图标题进行编辑。利用 margin 对作图细节进行编辑。
```

图 1-13　交互式收盘价与市值箱线图

利用截面数据集 mydata1 进行绘图，绘制以市值类型为分类标准的交互式箱线图，反应每个类型的市值的分布状况如图 1-13 所示。链接如下：

```
##交互式饼图
summary（mydata$shizhi） #利用 summary 统计市值在各个档次的统计值
piedata=data.frame(value=c(138,137,138,136),group=c("1","2","3","4"))
plot_ly(piedata,values=~value,labels=~group,type="pie") #交互式饼图
```

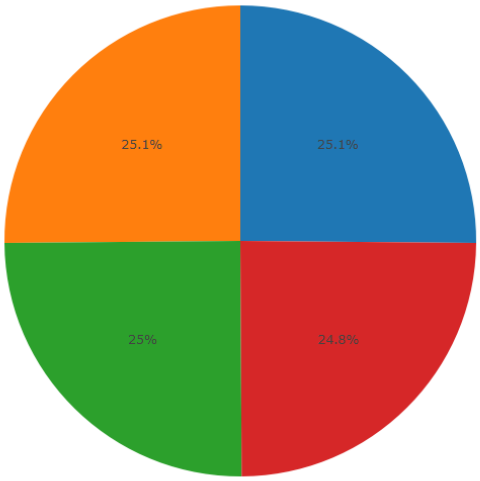

图 1-14 股票市值分布交互式饼图

```
#交互式密度图
p<+ggplot(mydata1,aes(x=close,fill=shizhi))+geom_density(alpha=.35)
  +facet_grid(shizhi~.)    #ggplot 绘制密度统计图并以市值类型作为分面变量
library("devtools")        #载入 devtools 下载工具包
devtools::install_github('hadley/ggplot2')    #该交互功能需在GitHub上下载ggplot
library("ggplot2")
p<-plotly(p)                    #绘制交互式分面密度图
layout(p,
       title="收盘价与市值交互式密度统计图",
       xaxis=list(title="收盘价",showgrid=F),
       yaxis=list(title="市值数平"),
       margin=list(l=50,r=50,b=50,t=50,pad=4))
piedata=data.frame(value=c(138,137,138,136),group=c("1","2","3","4"))
#采用 layout  函数来单独定义，并且用 xaxis、yaxis、title 对横纵坐标轴以及作图标题进行编辑。利用 margin 对作图细节进行编辑。
```

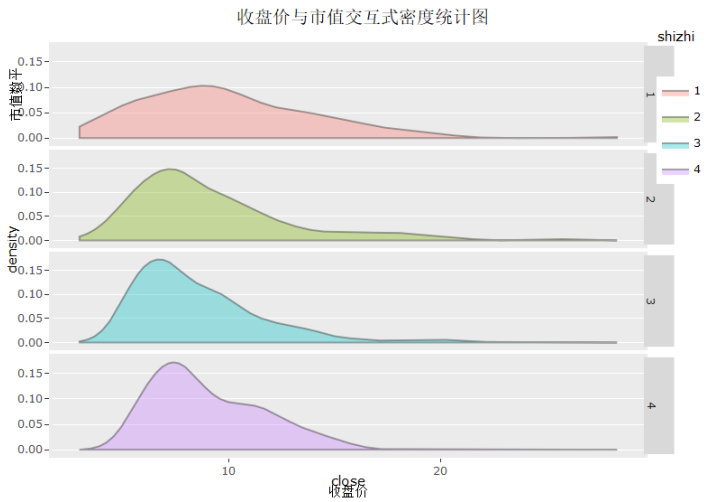

图 1-15　交互式分布密度统计图

利用截面数据集 mydata1 进行绘图，绘制以市值类型为分类标准的收盘价统计分布密度交互式饼图，反应每个类型的市值的分布状况如图 1-13 所示。

➢ 密度曲线直方图

```
ggplot(mydata1,aes(x=close,y=..density..))+geom_histogram(fill="tomato1")
        +geom_line(stat="density",adjust=.25,colour="black")
        +xlab("收盘价") + ylab("频率") +ggtitle('收盘价频率直方图')
        +theme(plot.title = element_text(hjust = 0.5,
                family="myFont",size=18,color="red"),
          panel.background=element_rect(fill='lightsteelblue1',color='black'))
```

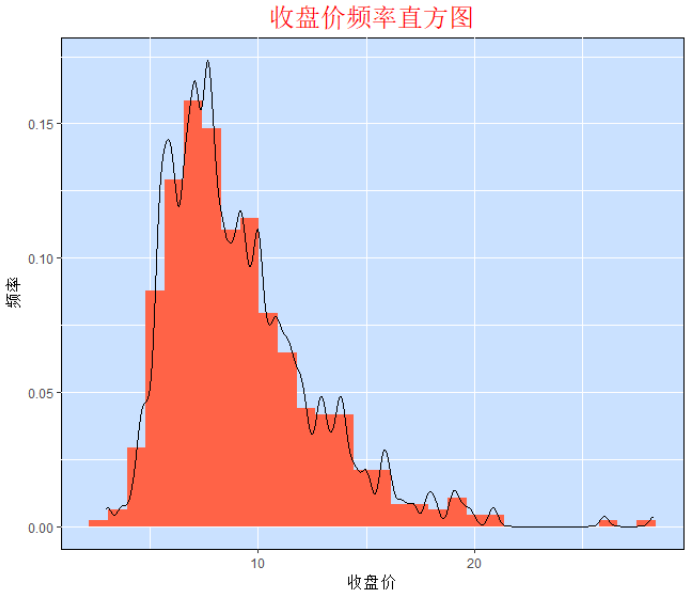

图 1-16 收盘价频率直方图

利用 2003 年 1 月 2 日的各只股票收盘价的数据通过 ggplot 绘图包绘制频率直方图，并在直方图上绘制出分布密度曲线，通过调节频率直方图的参数最终达到效果。

➢ 箱线图、小提琴图

```
ggplot(mydata1,aes(x=shizhi,y=close,fill=shizhi))
    +geom_boxplot(outlier.size=1.5, outlier.shape=21,notch=TRUE,alpha=.35)
    +stat_summary (fun.y="mean",geom="point",shape=23,size=3,fill="white")
    +xlab("市值水平") + ylab("成交价格")
    +ggtitle('带异常值收盘价箱线图')
    +theme(plot.title = element_text(hjust = 0.5,
          family="myFont",size=18,color="red"),
          panel.background=element_rect(fill='aliceblue',color='black'))
```

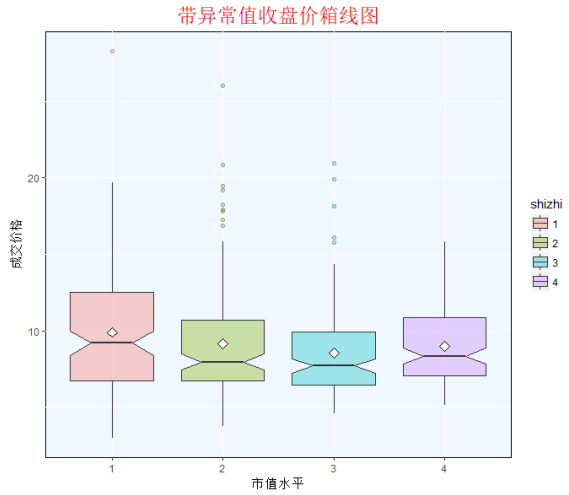

图 1-17 带异常值的箱线图

```
ggplot(mydata1,aes(x=shizhi,y=close,fill=shizhi))
    +geom_boxplot(outlier.shape=NA,notch=TRUE,alpha=.35)
    +stat_summary (fun.y="mean",geom="point",shape=23,size=3,fill="white")
    +xlab("市值水平") + ylab("成交价格")
    +ggtitle('带异常值收盘价箱线图')
    +theme(plot.title = element_text(hjust = 0.5,
            family="myFont",size=18,color="red"),
            panel.background=element_rect(fill='aliceblue',color='black'))
```

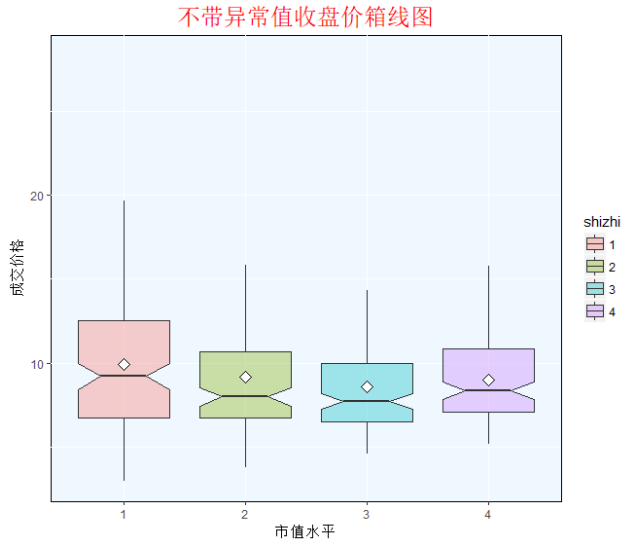

图 1-18 不带异常值的箱线图

```
ggplot(mydata1,aes(x=shizhi,y=close))+
        geom_violin()+
        geom_boxplot(width=.1,fill="black",outlier.colour=“red”)+
        stat_summary(fun.y=median,geom="point",shape=23,size=3,  fill="white")
```

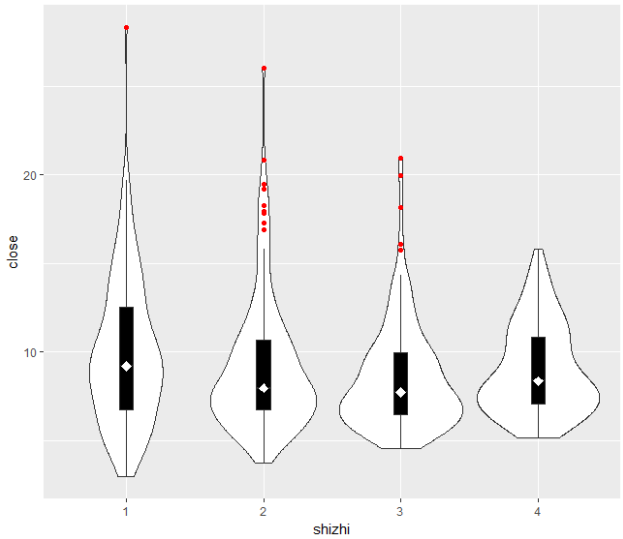

图 1-19 包含异常值点小提琴图

```
ggplot(mydata1,aes(x=shizhi,y=close))+
  geom_violin()+
  geom_boxplot(width=.1,fill="black",outlier.colour=NA)+
  stat_summary(fun.y=median,geom="point",shape=23,size=3,  fill="white")
```

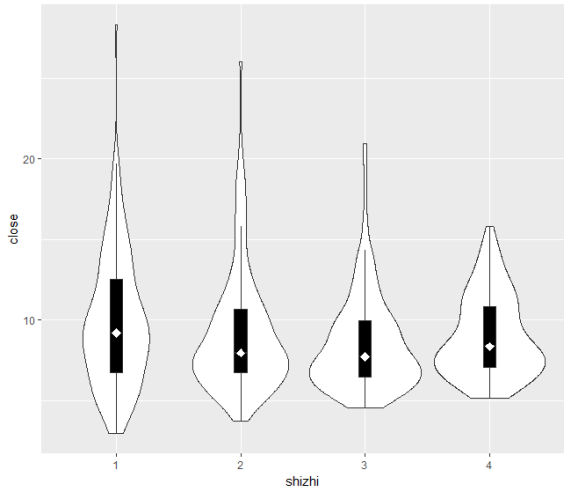

图 1-21 不包含异常值点的小提琴图

二、概率密度函数和累积分布函数

➢ T 分布

```
set.seed(1)                          #设置随机数种子
x <- seq(-5,5,length.out=100)   #生成-5 到 5 的均匀数列
y<-dt(x,1,0)                         #生成自由度为 1，非中心化参数为 0
plot(x,y,col="red",xlim=c(-5,5),ylim=c(0,0.5),type='l',xaxs="i",
yaxs="i",ylab='density',xlab='',main="t 分布概率密度函数")
lines(x,dt(x,5,0),col="green")
lines(x,dt(x,5,2),col="blue")
lines(x,dt(x,50,4),col="orange")       #画出不同自由度和非中心化参数 t 分布
legend("topleft",legend=paste("df=",c(1,5,5,50),"  ncp=",  c(0,0,2,4)),  lwd=1,
col=c("red",  "green","blue","orange")
```

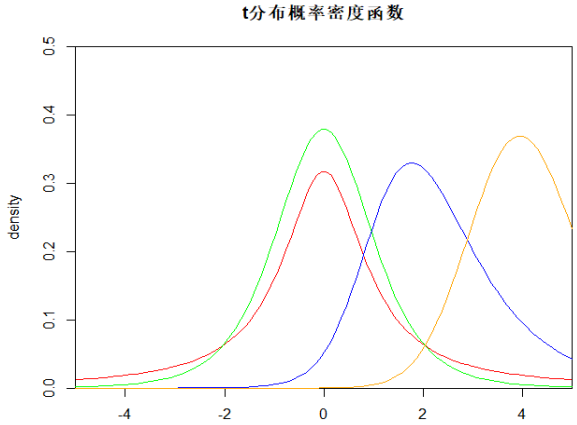

图 2-1 t 分布概率密度函数

```
set.seed(1)                    #设置随机种子
x<-seq(-5,5,length.out=1000)
y<-pt(x,1,0)                   #生成自由度为 1，非中心化参数为 0 的 t 分布累积分布概率
plot(x,y,col="red",xlim=c(-5,5),ylim=c(0,0.5),type='l',
     axs="i", yaxs="i",ylab='density',xlab='',
      main="t 累积分布函数图")
```

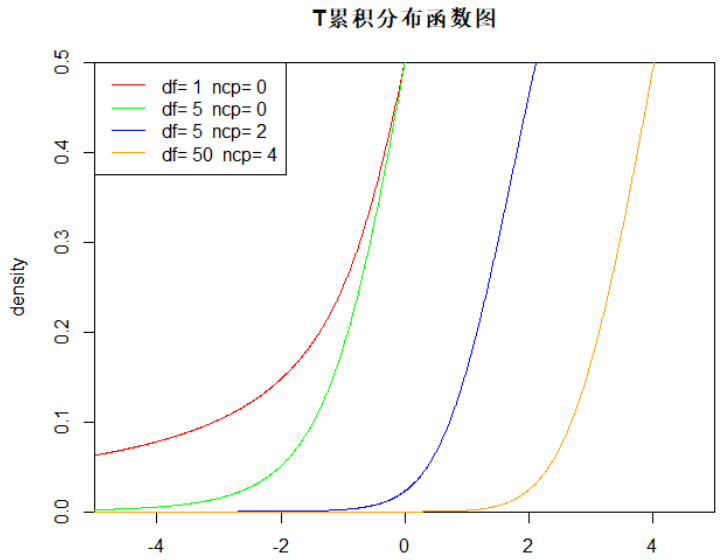

图 2-2 t 分布累积分布图

➢ 卡方分布

利用 ggplot2 来做卡方分布的分布密度函数以及累积分布函数图。

```
#自由度为 1 的卡方分布密度函数
library("reshape2")
x<-seq(0,10,length.out=1000)
自由度为 1<-dchisq(x,1)              #生成自由度为 1 的卡方分布密度
自由度为 2<-dchisq(x,2)              #生成自由度为 2 的卡方分布密度
自由度为 3<-dchisq(x,3)              #生成自由度为 3 的卡方分布密度
自由度为 4<-dchisq(x,4)              #生成自由度为 4 的卡方分布密度
dat<- data.frame(x,自由度为 1,自由度为 2,自由度为 3,自由度为 4)          #构造数据框
test <- melt(dat,id.vars="x")
ggplot(test,aes(x=x,y=value))+geom_line(aes(color=variable))
     +ggtitle('卡方分布密度图')
     +theme(plot.title = element_text(hjust = 0.5, family="myFont",
       size=18,color="black"),panel.background=element_rect(fill='aliceblue',color='black'))
```

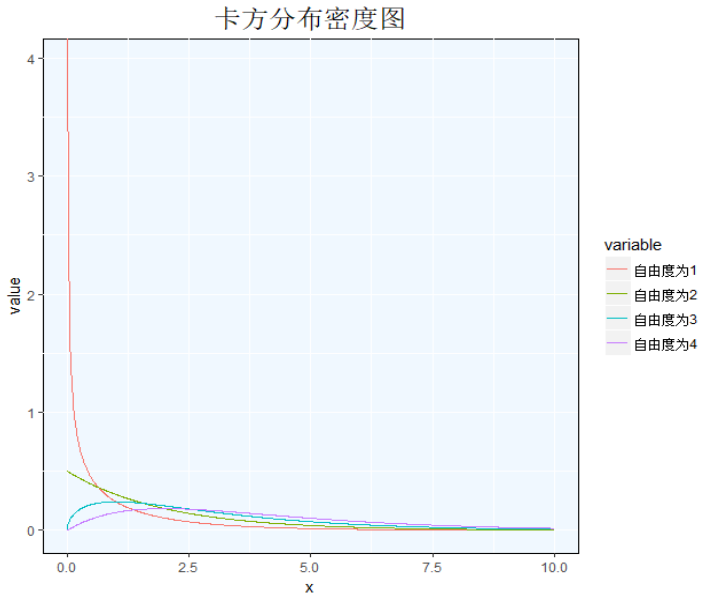

图 2-3 卡方分布密度函数

```
#卡方累积分布函数
x<-seq(0,10,length.out=1000)
自由度为 1<-pchisq(x,1)              #生成自由度为 1 的卡方分布密度
自由度为 2<-pchisq(x,2)              #生成自由度为 2 的卡方分布密度
自由度为 3<-pchisq(x,3)              #生成自由度为 3 的卡方分布密度
自由度为 4<-pchisq(x,4)              #生成自由度为 4 的卡方分布密度
dat<- data.frame(x,自由度为 1,自由度为 2,自由度为 3,自由度为 4)          #构造数据框
test <- melt(dat,id.vars="x")
ggplot(test,aes(x=x,y=value))+geom_line(aes(color=variable))
+ggtitle('卡方累积函数图') +theme(plot.title = element_text(hjust = 0.5, family="myFont",
   size=18,color="black"),panel.background=element_rect(fill='aliceblue',color='black'))
```

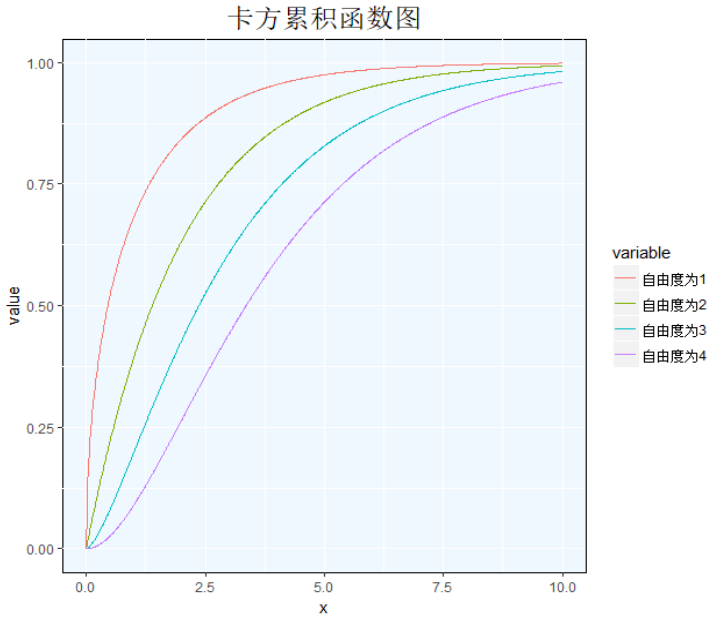

图 2-4 卡方分布累积函数图