GGpplot 应用一

上海财经大学　统计与管理学院

一、图形与语法

一张统计图形是从**数据**到**几何对象**的图形属性的一个映射。此外，图形中还可能包含数据的**统计变换**，最后绘制在某个特定的**坐标系**中，而**分面**则可以用来生成数据不同子集的图形。
一张统计图形就是由上述这些独立的图形部件所组成的。

- 最基础的部分是你想要可视化的**数据（data）**以及一系列将数据中的变量对应到图形属性的**映射（mapping）**；
- **几何对象（geom）**：在图中实际看到的图形元素，如点、线、多边形等；
- **统计变换（stats）**：对数据进行的某种汇总，如分组计数、线性回归等。可选，但很有用。
- **标度（scale）**：将数据的取值映射到图形空间，如用颜色、大小或形状来表示不同的取值，使读者可以从图形中读取原始数据。展现方式为绘制**图例**和**坐标轴**。
- **坐标系（coord）**：描述数据是如何映射到图形所在的平面，同时提供看图所需的坐标轴和网络线。通常使用笛卡尔坐标系，但也可以变换为其他类型，如极坐标和地图投影。
- **分面（facet）**：描述如何将数据分解为各个子集，以及如何对子集作图并联合进行展示。分面也称为条件作图或网格作图。

二、qplot（quick plot）相关作图

主要学习：

- qplot()的基本用法
- 数据和映射
- 图形属性（如颜色、大小和形状）
- 几何对象(如点、线、条形等））
- 分面（条件作图）
- 外观

数据集

ggplot2包中的diamonds数据集包含约54000颗钻石的信息，数据涵盖：

- 钻石质量的四个C：
  - 克拉重量（carat）
  - 切工（cut）
  - 颜色（color）
  - 净度（clarity）
- 五个物理指标
  - 深度（depth）
  - 钻面宽度（table）
  - 其他测量尺度（x, y, z）

数据集

ggplot2包中的diamonds数据集包含约54000颗钻石的信息，数据涵盖：

- 钻石质量的四个C：
  - 克拉重量（carat）
  - 切工（cut）
  - 颜色（color）
  - 净度（clarity）
- 五个物理指标
  - 深度（depth）
  - 钻面宽度（table）
  - 其他测量尺度（x, y, z）

数据集

```
library(gridExtra)  ##支持ggplot2多图并列
library(ggplot2)
data("diamonds")
head(diamonds)

set.seed(123456)
dsmall <- diamonds[sample(nrow(diamonds),100),]
```

1st Plot

```
attach(diamonds)
p1 <- qplot(carat, price)  #价格和重量之间的关系
p2 <- qplot(log(carat), log(price))  #变量变换
p3 <- qplot(carat,x*y*z)  #变量组合,体积和重量之间关系
grid.arrange(p1,p2,p3, ncol=3)
detach(diamonds)
```

1st Plot

```
attach(dsmall)
p4 <- qplot(carat, price, color=color) #将color变量映射到点的颜色
p5 <- qplot(carat, price, shape=cut)  # 将cut变量映射到点的形状
grid.arrange(p4,p5,ncol=2)
detach(dsmall)
```

图形属性

不同类型的变量适合不同的图形属性：

- 颜色和形状适合于分类变量；
- 大小适合于连续变量；
- 数据量很大，不同组的数据之间很难区分，适合于分面。

几何对象

适用于考察二维变量关系的几何对象：

- geom="point"，绘制散点图。
- geom="smooth"，拟合一条平滑曲线，并将曲线和标准误展示在图中。
- geom="boxplot"，绘制箱形胡须图，用以概括一系列点的分布情况。
- geom="path"和geom="line"可以在数据点之间绘制连线。
  - 路径图可以是任意的方向；
  - 线条图只能创建从左到右的连线。

几何对象

适用于考察一维变量分布的几何对象：

对于连续变量：

- geom="histogram"， 绘制直方图；
- geom="freqpoly"，绘制频率多边形；
- geom="density"，绘制密度曲线。
- qplot()默认直方图。

对于离散变量：

- geom="bar"，绘制条形图。

平滑曲线

```
p5 <- qplot(carat, price, data=dsmall, geom=c("point","smooth"))

#利用c()函数将多个几何对象组成一个向量传递给geom。几何对象会按指定的顺序进行堆叠。

#如果不想绘制标准误，可以使用se=FALSE

p6 <- qplot(carat, price, data=dsmall, geom=c("point", "smooth"), se=FALSE)
grid.arrange(p5, p6, ncol=2)
```

平滑曲线

利用method参数可以选择许多不同的平滑器：

- method="loess"，当n<1000时是默认选项，使用的是局部回归的方法。曲线的平滑程度由span参数控制，取值范围为0（很不平滑）至1（很平滑）。

```
qplot(carat, price, data=dsmall, geom=c("point","smooth"),
span=0.2)
```

- method="lm"，默认拟合一条直接。通过指定formula = y ~ poly(x,2)可以拟合二次多项式。通过加载splines包可以使用样条模型：formula = y ~ ns(x,2)。第二个参数是自由度：自由度越大，曲线波动越大。

```
p9 <- qplot(carat, price, data=dsmall, geom=c(“point","smooth"), method="lm")
library(splines)
p10 <- qplot(carat, price, data=dsmall, geom=c("point","smooth"),
method="lm", formula=y~ns(x,5))
grid.arrange(p9,p10, ncol=2)
```

直方图和密度曲线图

```
p13 <- qplot(carat, data=diamonds, geom="histogram")  #直方图
p14 <- qplot(carat, data=diamonds, geom="density")  #密度曲线图
grid.arrange(p13, p14, ncol=2)
```

对于密度曲线，adjust参数控制曲线的平滑程度，取值越大，曲线越平滑。

对于直方图， binwidth参数，通过设定组距来调节平滑度。 组距越小，越体现细节。

```
p15 <- qplot(carat, data=diamonds, geom="density", color=color)
p16 <- qplot(carat, data=diamonds, geom="histogram", fill=color)
#等式后面的color是数据集中的变量color钻石颜色，按color分组。
grid.arrange(p15,p16,ncol=2)
```

分面

分面将数据分割成若干子集，然后创建一个图形矩阵，将每一个子集绘制到图形矩阵的窗格中。所有的子图采用相同的图形类型，并进行一定的设计，使得它们之间方便比较。

qplot()通过*row_var ~ col_var*表达式进行指定。可以指定任意数量的行变量和列变量，但注意当变量数超过**两个**时，生成的图形可能非常多，以至于不适合在屏幕上显示。

如果只想指定一行或一列，可以使用.作为占位符，例如row_var ~ . 创建一个单列多行的图形矩阵。

```
p21 <- qplot(carat, data=diamonds, facets = color ~ ., geom = "histogram",
binwidth=0.1, xlim = c(0,3))
p22 <- qplot(carat, ..density.., data=diamonds, facets = color ~ ., geom =
"histogram", binwidth=0.1, xlim = c(0,3))
# ..density..告诉ggplot2将密度而不是频数映射到y轴。
grid.arrange(p21, p22, ncol=2)
```

#左图展示的是频数，右图展示的是频率。在比较不同组分布时，频率图不受该组样本量大小的影响。图形显示，高质量的钻石（颜色D)在小尺寸分布上是偏斜的，但随着质量下降，重量分布越来越平坦。

其他选项

其他一些控制图形的外观的参数：

- **xlim, ylim:** 设置x轴和y轴的显示区间。取值是一个长度为2的向量。例如xlime=c(0,20)；
- **log：** 一个字符型向量，说明哪个坐标轴应该取对数。例如：log="x"，表示对x轴取对数；log="xy"，表示对x轴和y轴都取对数。
- **main：** 图形的主标题，可以是字符串，也可以是一个表达式。例如main = expression(beta[1]==1)。更多数学表达式参见?plotmath命令。
- **xlab, ylab：** 设置x轴和y轴的标签文字。可以是字符串，也可以是表达式。

```
p23 <- qplot(carat, price, data=dsmall,xlab ="Weight(carats)" , ylab="Price($)",
    main="Price-weight relationship")

p24 <- qplot(carat, price/carat, data=dsmall,xlab = "Weight(carats)", ylab=
    expression(frac(price,carat)),main="Price-weight relationship",xlim=c(.2,1))
p25 <- qplot(carat, price, data=dsmall, log="xy")

grid.arrange(p23,p24,p25,ncol=3)
```

ggplot

Data	感兴趣的变量 (data frame)
Aesthetics	x-axis / y-axis / color / fill / size / labels / alpha / shape / linear width / linear type
Geometries	point / line / histogram / bar / boxplot
Facets	columns / rows
Statistics	binning / smoothing / descriptive / inferential
Coordinates	cartesian / fixed / polar / limits
Themes	non-data ink

第一层是数据层

第二层是美学层

第三层是几何层，是最基本的层

第四层是面，绘图面板划分成多少行列，对应一个分类变量

第五层是统计层

第六层是坐标系，主要应用的都是笛卡尔坐标系

第七层是主题，与数据本身无关

Ggplot:airquality数据集绘图

```
# 加载ggplot2包，使用airquality数据集绘图
library(ggplot2)
ggplot(data = airquality,aes(x=Wind,y=Temp)) +
  geom_point(color='green')


# 风速与温度，不同月份对应不同颜色
ggplot(data = airquality,aes(x=Wind,y=Temp)) +
  geom_point(aes(color=factor(Month)))
```

Ggplot:airquality数据集绘图

```
ggplot(data=airquality,aes(x=Wind,y=Temp)) +
  geom_point() +
  geom_smooth()

# 使用stat也可以实现
ggplot(data=airquality,aes(x=Wind,y=Temp)) +
  geom_point() +
  stat_smooth()
```

Ggplot:airquality数据集绘图

```
# 指定回归方法为线性回归，关闭置信区间显示
ggplot(data=airquality,aes(x=Wind,y=Temp)) +
  geom_point() +
  stat_smooth(method='lm',se=FALSE)


# 每个月对应一条不同颜色的回归线
ggplot(data=airquality,aes(x=Wind,y=Temp)) +
  stat_smooth(method='lm',se=FALSE,aes(color=factor(Month)))
```

Ggplot:airquality数据集绘图

```
# 对总体绘制回归线，再按照月份绘制回归线
ggplot(data=airquality,aes(x=Wind,y=Temp)) +
  geom_point(alpha=0.7,size=0.5) +
  stat_smooth(method='lm',se=FALSE,aes(group=1),color='yellow') +
  stat_smooth(method='lm',se=FALSE,aes(color=factor(Month)))


# 按照月份分成五个面板
ggplot(data=airquality,aes(x=Wind,y=Temp)) +
  geom_point(alpha=0.7,size=0.5) +
  stat_smooth(method='lm',se=FALSE,aes(color=factor(Month))) +
  facet_grid(.~Month)
```

Thank You !