title: "Lect02 ggplot画图" author: "郑骏明" date: "2019/11/3" output: html_document

ggplot绘图的语句基本上是下面这样的形式（以散点图为例）

```
ggplot(数据, aes(x=变量1,y=变量2,...))
+geom_poin(属性设置)
+...
```

比如iris这个数据集，总共有150个观测记录，有5个变量Sepal.Length, Sepal.Width, Petal.Length, Petal.Width, 和 Species. 前面4个变量是连续变量，后面Species是factor变量

```
class(iris$Sepal.Length)
class(iris$Species)
levels(iris$Species)
```

如果我要用Sepal.Length作为x，Sepal.Width作为y绘制图表：

```
library(ggplot2)
ggplot(iris,aes(Sepal.Length,Sepal.Width))+geom_point()
```

> 如果数一下的话就会发现肯定没有150个点，这是因为有些点互相重叠了。

Species作为类别变量，还可以映射给color：

```
ggplot(iris,aes(Sepal.Length,Sepal.Width,color=Species))+geom_point()
```

当然aes函数不一定要放在ggplot里面,下面的代码和上面的代码运行效果其实看不出什么区别

```
ggplot(iris)+geom_point(aes(Sepal.Length,Sepal.Width,color=Species))
```

比较一下下面两个代码的结果有什么不一样：（geom_smooth是生成拟合曲线的一种几何图像）

```
g1 = ggplot(iris,aes(Sepal.Length,Sepal.Width))
g1+geom_point(aes(color=Species))+geom_smooth(method='lm')
```

```
g2 = ggplot(iris,(aes(Sepal.Length,Sepal.Width,color=Species)))
g2+geom_point()+geom_smooth(method='lm')
```

从上面的散点图的例子可以看到，ggplot的绘图是分层实现的。ggplot的分层体现在：

- 首先在ggplot函数里面声明第一层次：数据。
- 然后是通过aes函数确定如何将数据的各个变量映射(mapping)到图像的各个**参数**里
- 图像的各个参数除了可以使用aes定义映射，也可以直接作为属性定义
- 映射可以通过scale系列函数进一步微调
- 之后是向这个做好了的映射上面安装图像呈现机制（`geom_*_*`）
- 再之后还可以进一步设置主题，facet等内容

ggplot提供了很多的`geom_*_*`函数，这些函数有不少是绘图的快捷函数，比如刚才我们用到的`geom_smooth(method='lm')`。话说回来，本来R就……挺适合作quickfix的，所以很多时候我们想画一个东西的时候就直接上网找教程然后照葫芦画瓢弄一个交差。r graph gallery收集了很多类型图表的教程，可以多去看一下。

但是就算只想求快，grammar of graphic还是必须要有所了解的。grammar of graphics这个页面可以作为一个入门的介绍。

接下来对课堂上提到的几个图表给一个参考解

柱状图

以mtcars数据集为例, 现在想绘制气缸数4,6和8的车辆的数目, 这个可以用geom_bar()来实现

```
ggplot(mtcars,aes(x=factor(cyl)))+geom_bar()
```

如果现在要按照气缸数将车辆数据分组,统计各组车辆车重, 以平均值作为柱形图高度, 添加errorbar的话(平均值加减标准差). 这样就需要计算出各组的平均值和标准差. 这两种数值有很多的计算方式, aggregate函数是其中一种. aggregate的第一个参数是方程,波浪号左边表示因变量,右边是自变量. 我们的因变量是重量(wt), 分组的依据(自变量)是气缸数(cyl), 所以第一个参数是`wt~cyl`. aggregate的第二个参数是mtcars, 第三个参数是用来做累计计算的函数名称, mean是求一组数据平均值的函数, sd是求一组数据标准差的函数, 而length则是返回一组数据数量的函数. 所以下面程序的前两行分别会计算各个cyl组的wt平均值与标准差.

自己运行这个程序的时候可以看一下`mtcars_mean`和`mtcars_sd`长成什么样子.

接下来使用cbind将两个dataframe进行组合, 组合以后用names()命名各列, 形成包含cyl, 平均值(avg)和标准差(sd)的dataframe. 然后利用这个dataframe去绘制柱形图.

在geom_col()的基础上, geom_errorbar还需要ymin和ymax两个参数,如前所述,分别是柱形高度(平均值)加减标准差. width=0.1是让errorbar不要太宽,只要占柱形的10%. labs是给图表添加横轴与纵轴标题

```
mtcars_mean = aggregate(wt~cyl, mtcars, mean)
mtcars_sd = aggregate(wt~cyl, mtcars, sd)
mtcars_smmry = cbind(mtcars_mean,sd=mtcars_sd$wt)
names(mtcars_smmry)=c('cyl','avg','sd')
ggplot(mtcars_smmry,aes(factor(cyl),avg))+geom_col()+geom_errorbar(
  aes(ymin=avg-sd,ymax=avg+sd),
```

```
  width=0.1
)+labs(x='number of cylinder',y='weight')
```

思考题: 前面使用ggplot()+geom_bar()绘制了cyl各个组车辆的数量, 如果要用geom_col()来绘制,如何提供各组车辆的数量这个数据呢? 可以考虑用length, 请写出程序.

饼状图

其实饼状图就是柱状图做了极坐标变换, 以cyl各组车数为例:

```
ggplot(mtcars,aes(x=1,fill=factor(cyl)))+geom_bar()+coord_polar(theta='y')+theme_v
oid()
```

上面的命令中, theme_void()是去除各种标注的一个主题, coord_polar(theta='y')是作极坐标变换的一个函数, 如果将这两个命令去掉我们看看下面的命令会产生什么图形:

```
ggplot(mtcars,aes(x=1,fill=factor(cyl)))+geom_bar()
```

这个长得像蛋糕横截面的图形做了极坐标转换以后就会成为饼图了.

箱式图

箱式图的绘制相对比较简单，参考r graph gallery: boxplot的内容

```
g = ggplot(mtcars,aes(x=factor(cyl),y=wt,fill=factor(cyl)))
g+geom_boxplot()
```

与geom_col()相比，这边不需要计算wt的平均值和标准差，geom_boxplot会自动计算中值，1/4分位，3/4分位，`1/4分位-IQR*1.5`,`3/4分位+IQR*1.5`这些值，然后绘制箱式图。

有时候在箱式图上额外叠加散点图可以更好的展示数据的分布规律。可以使用geom_jitter()，在原来的boxplot上再叠加一层

首先介绍两个函数:

1.`rep(val,repeat times)`。比如我需要一个向量，由10个"a"组成，就可以用`rep('a',10)` 2. `rnorm(count,mean,sd)`根据正态分布生成一系列的数据，比如说要生成平均值为10，标准差为2的100个数据，可以使用`rnorm(100,10,2)`

```
df = data.frame(grp = c(rep("A",500),rep("B",250),rep("B",250),rep("C",100)),
val = c(rnorm(500,10,5),rnorm(250,13,1),rnorm(250,19,1),rnorm(100,23,6))
)
g = ggplot(df,aes(grp,val,fill=grp))
g+geom_boxplot()+geom_jitter(color='black',size=0.4,alpha=0.6)
```

可以看到，对于一些分布较为特殊的数据（B组），boxplot还不足以显示数据的分布规律

折线图

对于时序数据，用折线图来绘图是很常规的操作。

```
ss_df = data.frame(ss=as.matrix(sunspot.year),date=time(sunspot.year))
ggplot(ss_df,aes(date,ss))+geom_line()
```

在lect02文件夹中有一个gdp_hist.csv。这个是利用python的tushare包下载的我国过去各年的gdp数据。现在尝试读取这个数据，然后以年份为x轴，年总gdp为y轴绘制折线图

```
df = read.csv('gdp_hist.csv')
g = ggplot(df,aes(year,gdp))
g+geom_line()
```

接下来还可以用geom_smooth加入拟合曲线

```
g+geom_line()+geom_smooth()
```

如果我想单独绘制2000年到现在的gdp数据，可以怎么写呢？

facet

坐标轴变换，比如log10变换

scale_x_log10