数据分析与处理技术

数据可视化：ggplot2

ggplot2基本原理

R当中除了基础做图包Graphics中的图形系统外，还有许多广受欢迎的图形工具包，其中最为著名的就是lattice和ggplot2。

Leland Wilkinson在《The Grammar of Graphics》构建了一套以图层化为基础的图形语法，随后Hadley Wickham在此基础上开发了ggplot2工具包。

这套图形语法把图形元素划分成图层(layer)，不同图层描述了数据的绘制特性、绘制类型、坐标系、图形特性等相关标度的映射。

Hadley Wickham的电子书《高级R语言编程指南》： http://adv-r.had.co.nz/

从plot到qplot

plot函数是基础图形系统中的通用型函数，ggplot2设置了qplot函数作为对应，方便初学者适应，但熟悉之后则不再主张使用这个方式。

qplot与基础包的plot用法几乎一样：

```
> qplot(speed,dist,data=cars,main='myggplot-1')
```

但是参数从缩写变成了全程，如color,size,shape分别对应了col,pch,cex

```
> qplot(speed,dist,data=cars,main='myggplot-1',color='red',xlab='x')
```

除此以外，增加了alpha(透明度),fill(阴影填充)

```
> qplot(speed,dist,data=cars,main='myggplot-1',alpha=I(1/10))
```

而main,xlab,ylab,xlim,ylim则完全一样。

注意ggplot的一个特点：除x y两个位置属性外，ggplot不需要在建立映射关系前手动去标度变量，而是自动将变量根据属性值类型标度和映射给图形属性。

常用图形参数的对应

x
y
color/colour
alpha
fill
linetype
shape
size
group
Order

图层layer

qplot虽模仿plot用法，但已经具备了图层功能

```
> qplot(speed,dist,data=cars)+ggtitle('ralationship of speed and dist')
+ylab('y axis:dist')+xlab('x axis:speed')
```

加号+连接了两个图层，不同图层具有不同的功能，最终叠在一起形成图形。

而每一个图层都可以带上自己的属性数据，进而达到远超基础做图的灵活性。

```
> p<-qplot(speed,dist,data=cars,main='myggplot-1')
```

另外，ggplot2产生的图形可以当做一种独特变量保存在环境中，下次调用变量p时则相当于将图输出到设备上。

```
> p+xlab('x:speed')
```

由此，变量的基本操作和保存方法则都可以适用于图形对象p

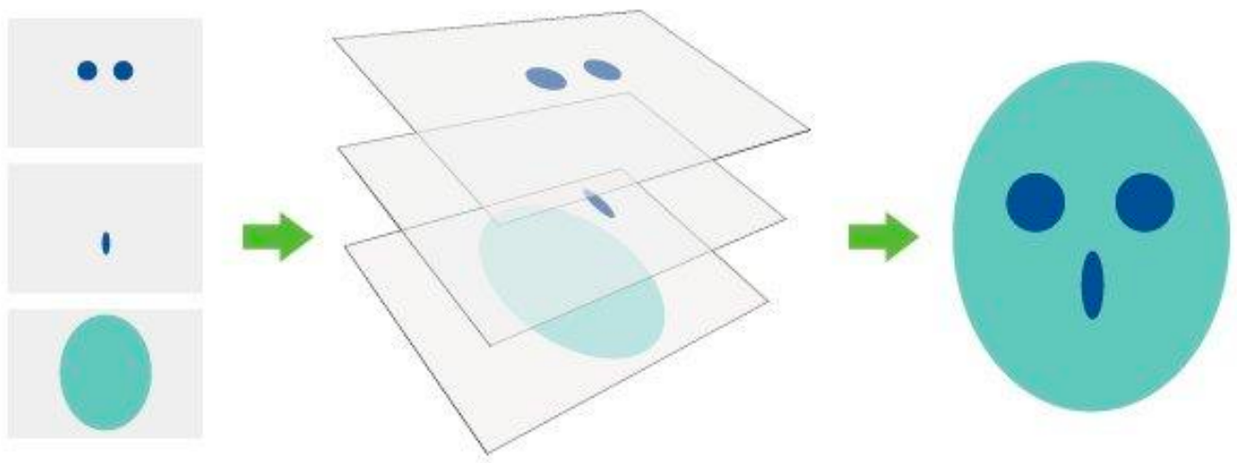
ggplot2基础

仍然以cars数据集为例，画出散点图并存入变量p中

```
> p<-ggplot(data=cars)+geom_point(aes(x=speed,y=dist),color='red')
```

ggplot( )函数必须作为第一个图层，它用来生成最底层的画板，在ggplot( )中传入的参数将是后续所有图层公用的，类似par( )的设备全局参数。

geom_point( )是叠加上的第二个图层，它用来画点，括号中传入的参数将只控制点的属性，即位置属性与speed和dist建立映射，颜色设置为’red’

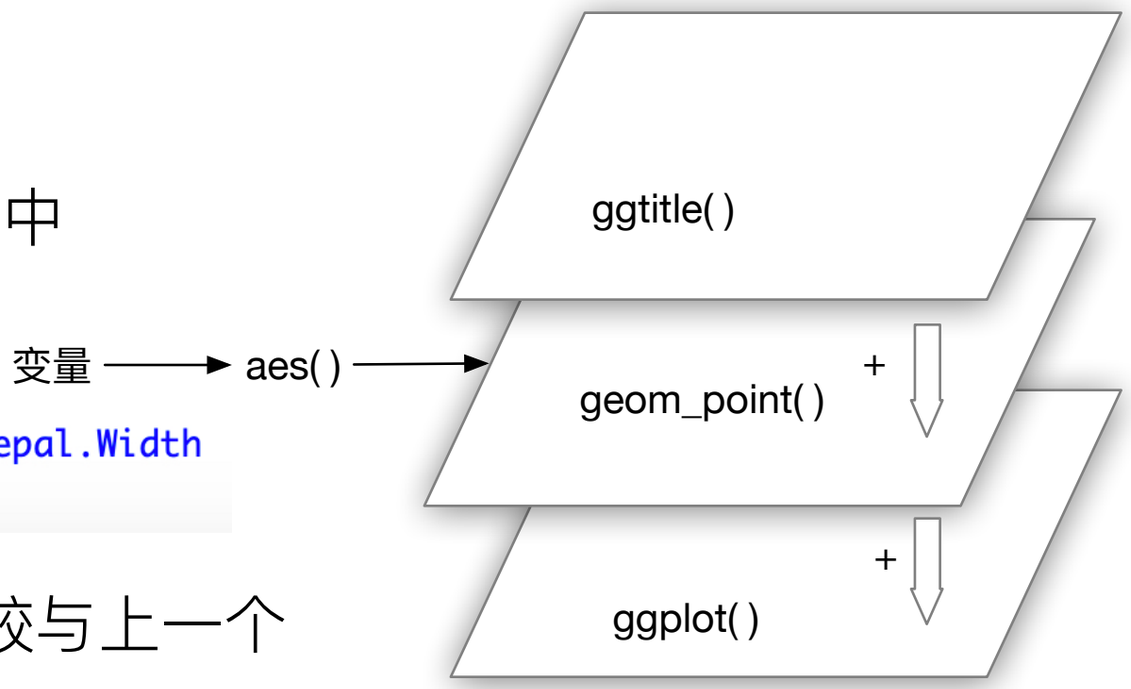


aes()函数

每一个图形对象函数都可以设定图形参数，手动标度属性值的方式与基础包一样，直接写入即可。但如果要映射变量到属性上则必须使用aes( )，在aes( )中完全不必考虑转换变量数据的标度方式，函数会自动根据变量数据类型施加标度转换。

```
> p<-ggplot(data=mtcars)
> p+geom_point(aes(x=wt,y=mpg,color=qsec))
```

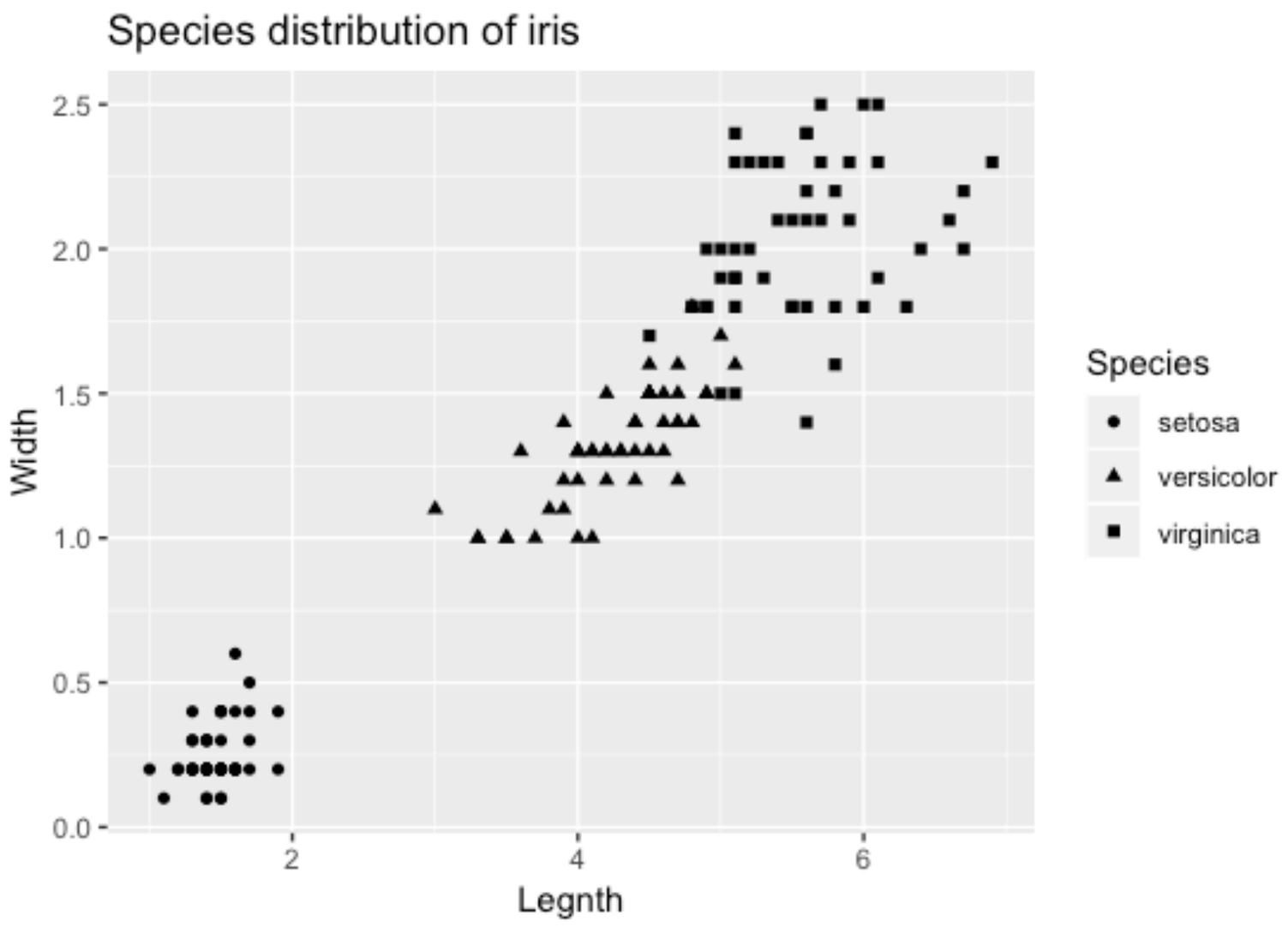
上例中，mtcars下的qsec映射到颜色属性中




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

iris数据集的Species映射到颜色属性，比较与上一个图形颜色配置的差别

```
> fig<-ggplot(data=iris,aes(Petal.Length,Petal.Width))+geom_point(aes(shape=Species))
> fig+ggtitle('Species distribution of iris')+xlab('Legnth')+ylab('Width')
```



几何对象

ggplot( )必须加上至少一个图层的几何对象才能形成一个完整的图形。

所有几何对象函数都是以geom_开头，后边接几何名称，

geom_point
geom_line
geom_abline
geom_boxplot
geom_bar
geom_polygon

geom_XXX(mapping,data,…,stat,position)

mapping, 图形属性映射

data,数据框格式的数据集

… geom的参数， 如fill填充颜色等

stat,统计变换方法

```
> ggplot(data=Titanic)+geom_bar(aes(Pclass),fill='blue')
```

```
> ggplot(mtcars,aes(mpg))+geom_histogram(aes(y=..density..),stat='bin',binwidth = 0.8)
```

统计变换

统计变换将输入的数据变换后作为新变量放入图形参数中。几何对象geom_XXX中可以选择统计方法，或使用统计变换图层

stat_XXX(mapping,data,…,geom,position)

mapping, 图形属性映射

data,数据框格式的数据集

… geom的参数，如binwidth组距，光滑曲线的bandwidth带宽等

geom,几何对象

赋予y的统计量

..count.. 观察值数目

..density.. 观察值密度

..x.. 组中心位置

```
> ggplot(mtcars,aes(x=am))+geom_bar(aes(y=stat(count/3)))
```

```
> ggplot(cars,aes(speed,dist))+geom_point()+geom_smooth(method='loess',se=F)
```

标度

图形属性的外观由标度图层控制，在不指定标度方式情况下ggplot2会自动按照主题模版进行标度并生成图例。标度会与图例自动匹配。

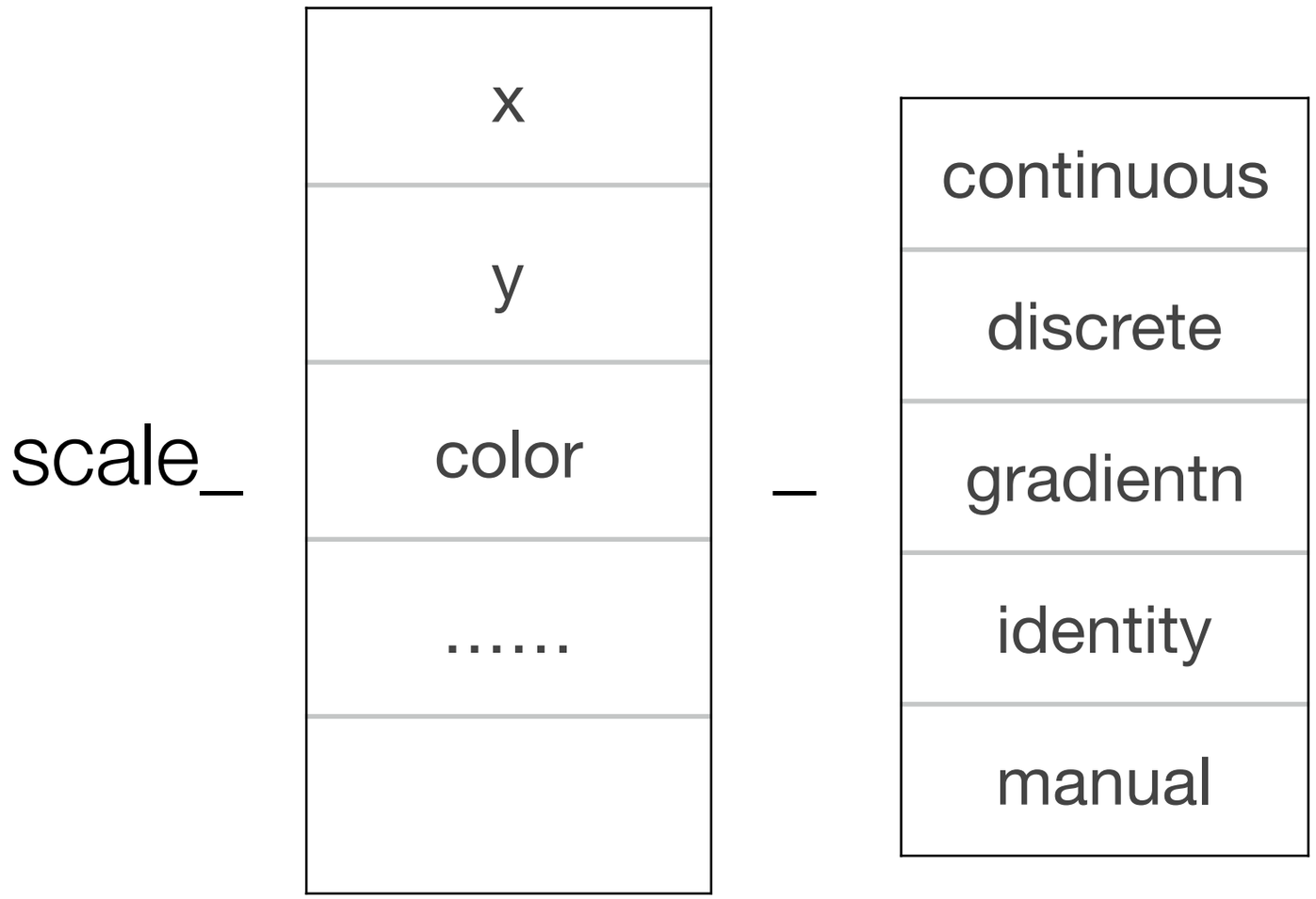
标度图层函数具有统一命名规则：scale_图形元素_标度类型
图形元素包括：color, size, shape, alpha等
标度类型包括：manual 手动修改标度，discrete 离散型，continuous连续性和gradientn 颜渐变型等

```
> p<- ggplot(data=mtcars)+geom_point(aes(wt,mpg,shape=as.factor(cyl)),color='blue')

> p+scale_shape_manual('factor number',values =c(3,5,2))
```

我们可以通过准确指定scale层的shape元素discrete型标度方式修改图例的名称

```
> p+scale_shape_discrete('factor')
```



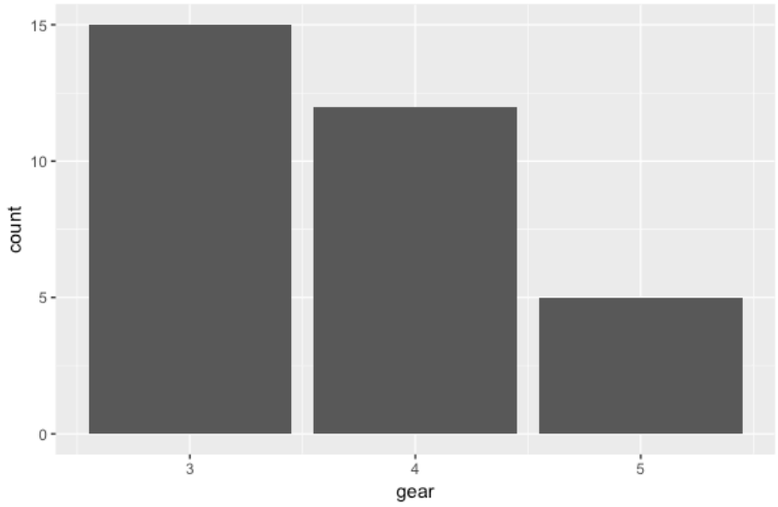

```
> ggplot(cars,aes(speed,dist))+geom_point()+scale_x_continuous(limits = c(-5,40))
```

坐标系

常用的做图是在笛卡尔坐标系下，但有时会用到极坐标，例如ggplot2中是没有pie饼图的，因为geom_bar条形图在极坐标下就是饼图。

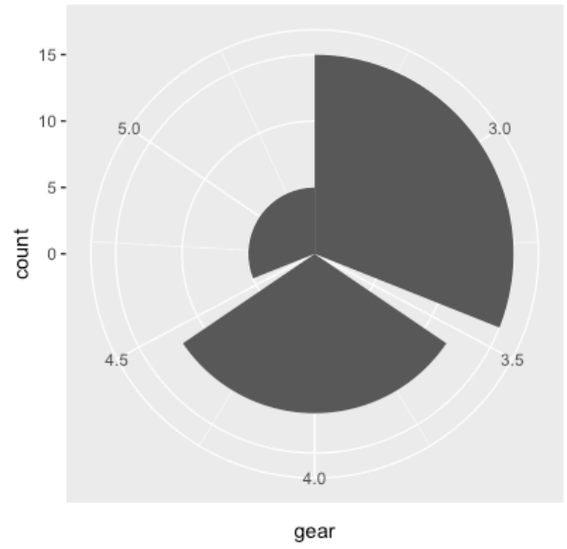
默认为笛卡尔坐标系，添加坐标系图层coord_polar( )可以指定使用极坐标

```
> ggplot(data=mtcars)+geom_bar(aes(gear))
> ggplot(data=mtcars)+geom_bar(aes(gear))+coord_polar()
```



另外常用的坐标系函数coord_flip( )横纵坐标互换

```
> ggplot(data=mtcars)+geom_bar(aes(gear))+coord_flip()
```

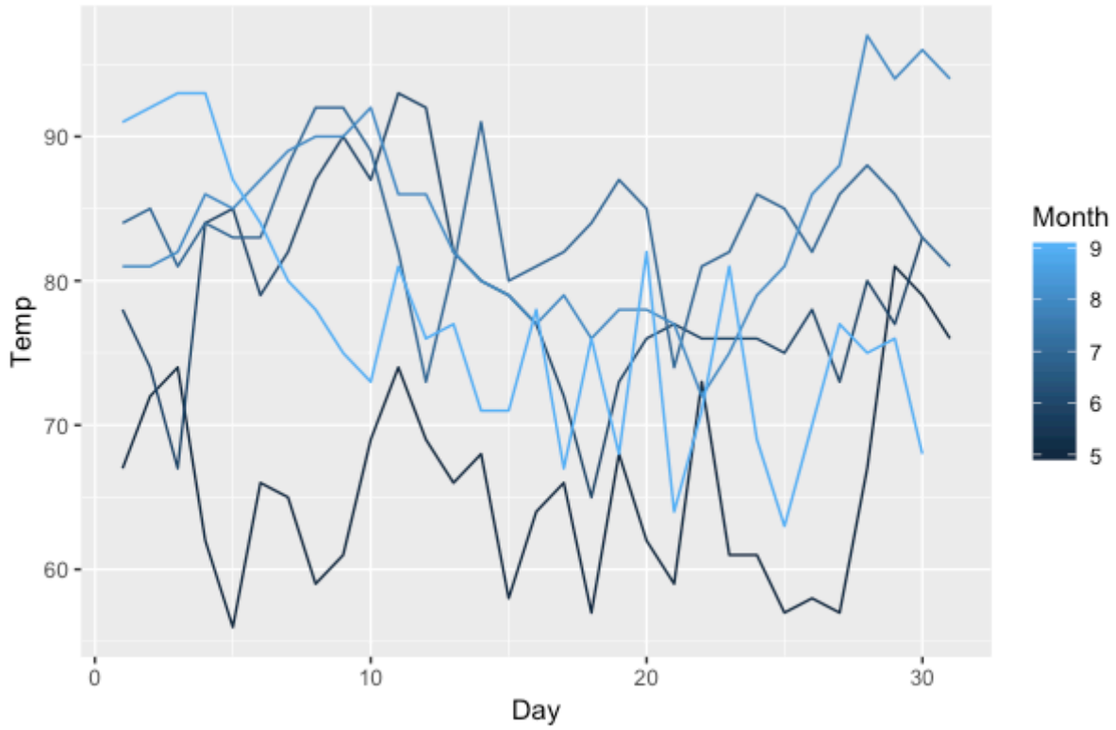


分组控制

分组画图是ggplot2中重要的做图方式，某一个变量被作为分组的标签变量，如下图数据集airquality中，记录了5到9月份逐天湿度数据

```
> ggplot(data=airquality)+geom_line(aes(x=Day,y=Temp,group=Month,color=Month))
```

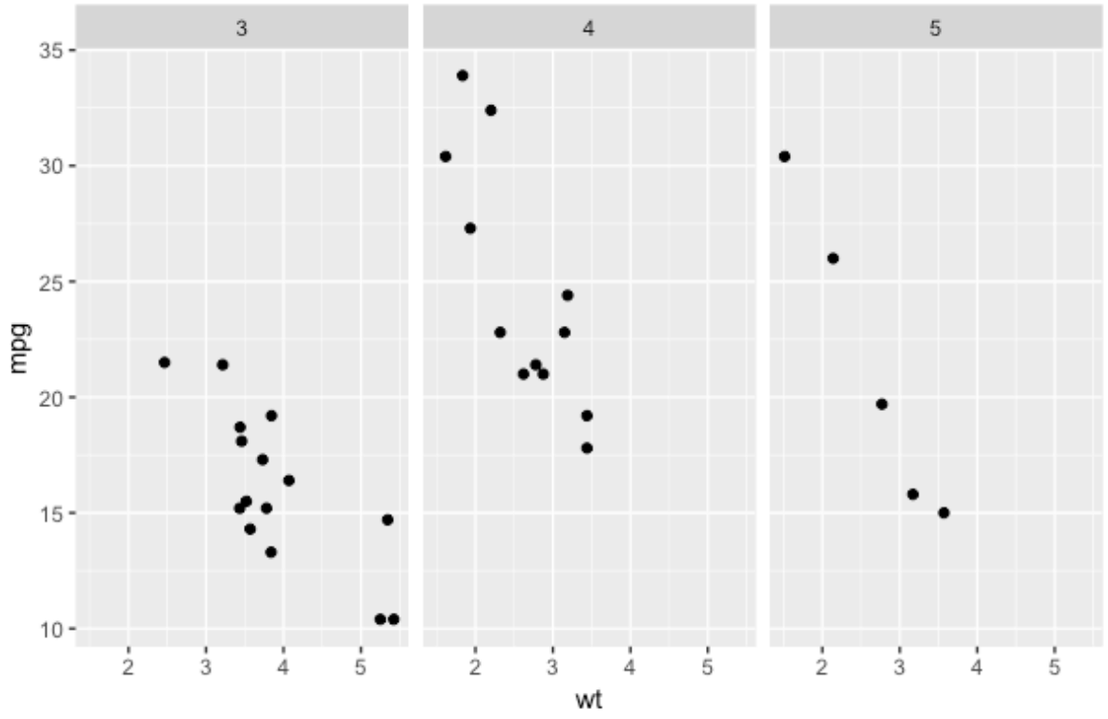
x轴映射为Day，y轴则映射湿度数据Temp，而月份Month则成为分组标签



分面控制

ggplot2中的分面并非画布的布局控制，而是根据某个标志变量分组做图，类似于分组统计。

```
> p<-ggplot(data=mtcars)+geom_point(aes(x=wt,y=mpg))
> p+facet_grid(.~gear)
```



公式左侧需用点(.)表示除标志变量外其他任意变量，而公式右侧则控制按列分面

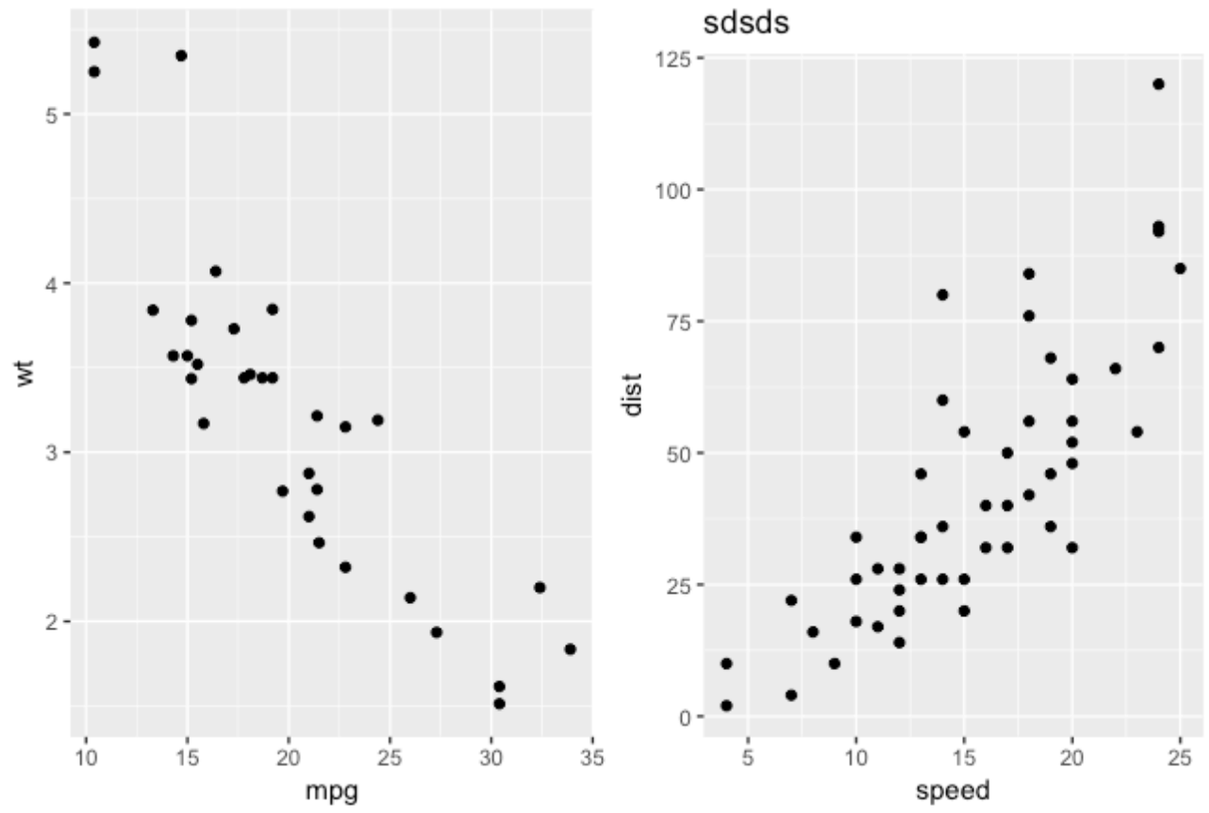
```
> p+facet_wrap(~carb)
```

facet_wrap函数同样用来分面，不同在于facet_wrap的公式不需要点(.)，并且会自动调整布局，对于分组标志变量属性值特别多时比较方便。

页面布局

分面(facet)需要依赖分类变量对画图区进行分割，必须在同一个数据集内实现。
ggplot2是以grid包为基础控制画图区域，分面布局可以直接使用grid中的工具，最方便的办法是使用gridExtra包。

```
> library(gridExtra)
> p1<-ggplot(data=mtcars)+geom_point(aes(mpg,wt))
> p2<-qplot(speed,dist,data=cars)+ggtitle('sdsds')
> grid.arrange(p1,p2,ncol=2)
```



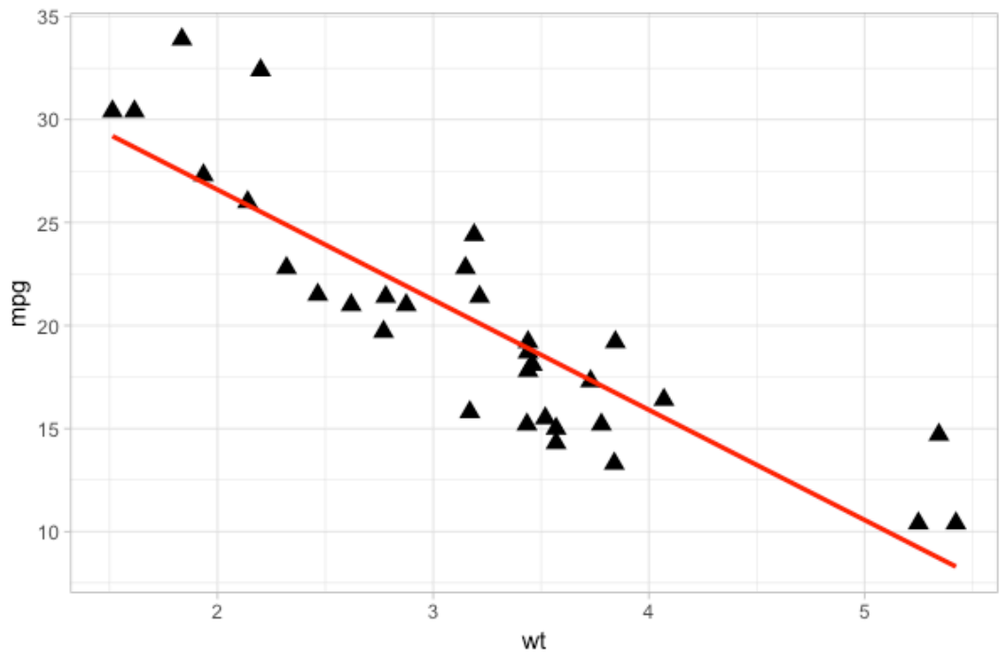
主题修改

ggplot2系统中可以自由调整做图风格，同时也准备了一些成套的主题，例如做好的一个图形p

```
> p<-ggplot(data=mtcars,aes(wt,mpg))
> p<-p+geom_point(shape=17,size=3)
> p<-p+geom_smooth(method='lm',se=F,color='red')
```

使用theme_set()函数设置一套主题，即命名为：theme_主题名( ) 的函数

```
> theme_set(theme_light())
```



除ggplot2自带的几个经典主题外，其他人也可以编写工具包为其配备主题，常用的有ggthemes包，提供了许多科研、新闻等经典主题包,使用格式与ggplot2的主题相同，ggthemes的开发者主页如下：

https://github.com/jrnold/ggthemes

插件式辅助工具

ggplot2衍生出非常多辅助工具包，并且还在快速扩充中。

工具包esquisse：针对某数据集以GUI界面辅助自动生成ggplot2代码

```
> library(esquisse)
> esquisser(mtcars)
Loading required package: shiny

Listening on http://127.0.0.1:3142
```

工具包ggThemeAssist：专门用于调整主题的工具包

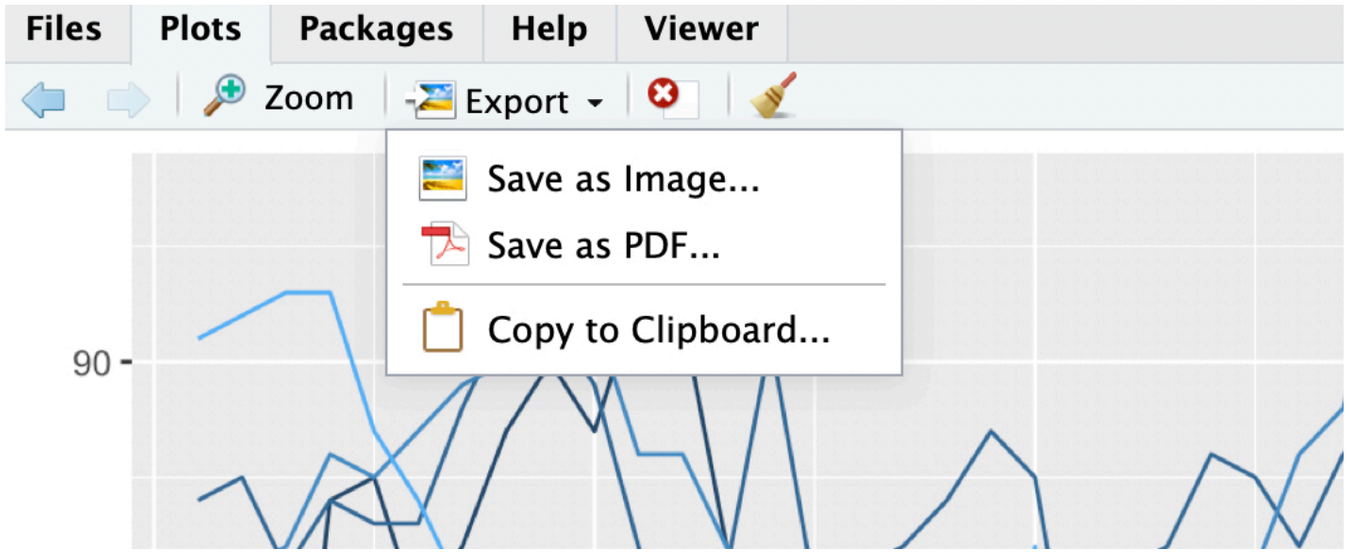
```
> library(ggThemeAssist)
> ggThemeAssistGadget(p)

Listening on http://127.0.0.1:3142
```

虽然有很多便捷工具辅助生成ggplot2代码，但最有价值的依然是它的绘图语法

保存图形

Rstudio中可以直接导出ggplot2的图片，如右图。由于图形可以存在环境的变量中，另一种方式可以用ggsave函数保存变量。




```
> ggsave('ggp.pdf',p,width=3.15,height=3.15)
```

ggplot2可以将图片保存为大多数图形格式，如pdf,png,jpeg等常见格式。