数据分析与处理技术——数据可视化

商学院　徐宁

本章参考内容

《R语言实战》第6章，第11章

《R语言教程》：

28 基本R绘图 | R语言教程 (pku.edu.cn)

29 ggplot作图入门 | R语言教程 (pku.edu.cn)

30 ggplot的各种图形 | R语言教程 (pku.edu.cn)

《R数据科学》第1章

《ggplot2:Elegant Graphics for Data Analysis》：

https://ggplot2-book.org/index.html

数据可视化

1.图形设备

绘图面板

图形文件

中文字体

内部图形设备

集成在Rstudio工具种的画板被作为默认图形设备

示例：常用内部图形设备（画板）命令

```{r}
dev.new()   #打开新画板命令
dev.set(4) #设置4号画板为待绘图状态
dev.cur()   #查询当前画板数量和状态
```

图形文件格式

矢量图格式

- eps/pdf/svg/ai/cdr/dwg
- 常见编辑软件：Adobe illustrator/CorelDrwa/CAD

位图，点阵图/栅格图

- JPG/JPEG/PNG/BMP
- 常见编辑软件：photoshop，画图板，……

外部图形设备

任何工具绘图前都会生成待输出的图形设备，其中外部图形设备主要指pdf、eps、jpeg、png等图形文件

示例：打开外部pdf文件作为待输出图形设备

```
```{r}
pdf("mygraph.pdf")
plot(cars)
dev.off()
```
```

绘图结束后必须关闭pdf文件

```
> jpeg(filename = "myplot.jpeg",width = 1500,height = 1400,units = "px",res = 300)
> barplot(c(88,79,99))
> dev.off()
```

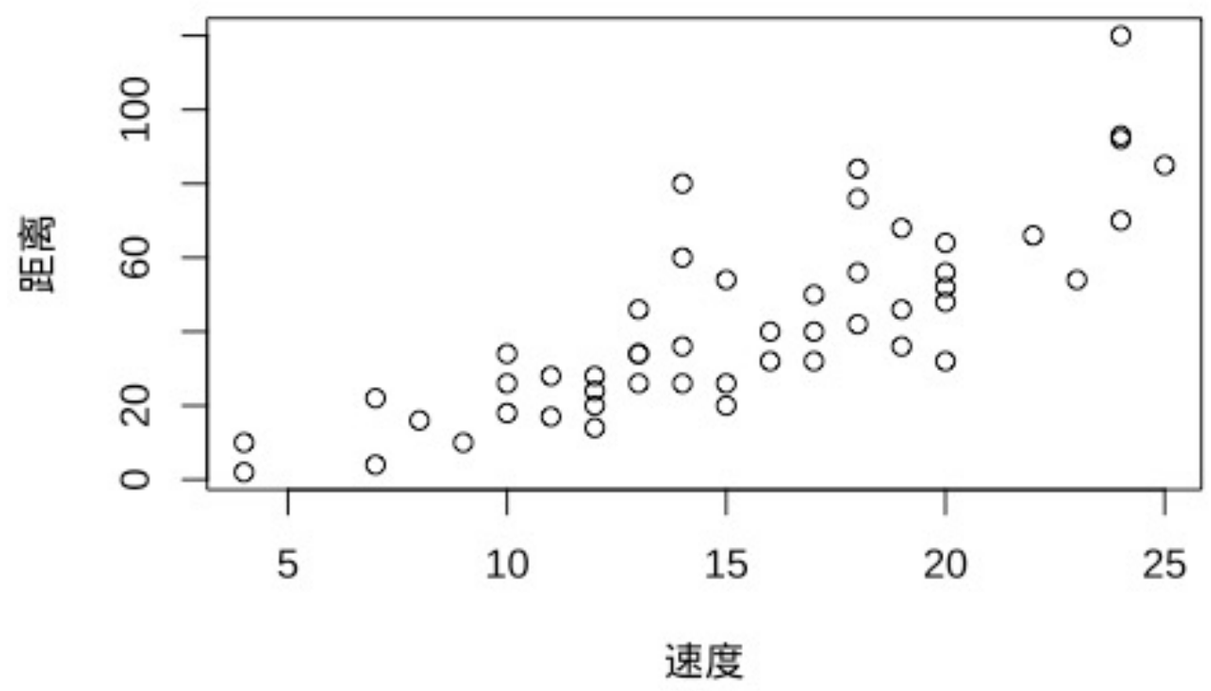
units：px(像素),cm(厘米),in(英尺)

res：单位区域像素密度，ppi

中文字体的补充

中文字体的缺失导致很多绘图系统输出文字时出现错误，需要用到补充字体工具：showtext,sysfonts,showtextdb三个工具包，其中showtext会关联其他工具包的加载。

```
> library(showtext)
> showtext_auto()
> plot(cars,xlab="速度",ylab="距离")
```



数据可视化

2.基础绘图系统

绘图原理

添加元素

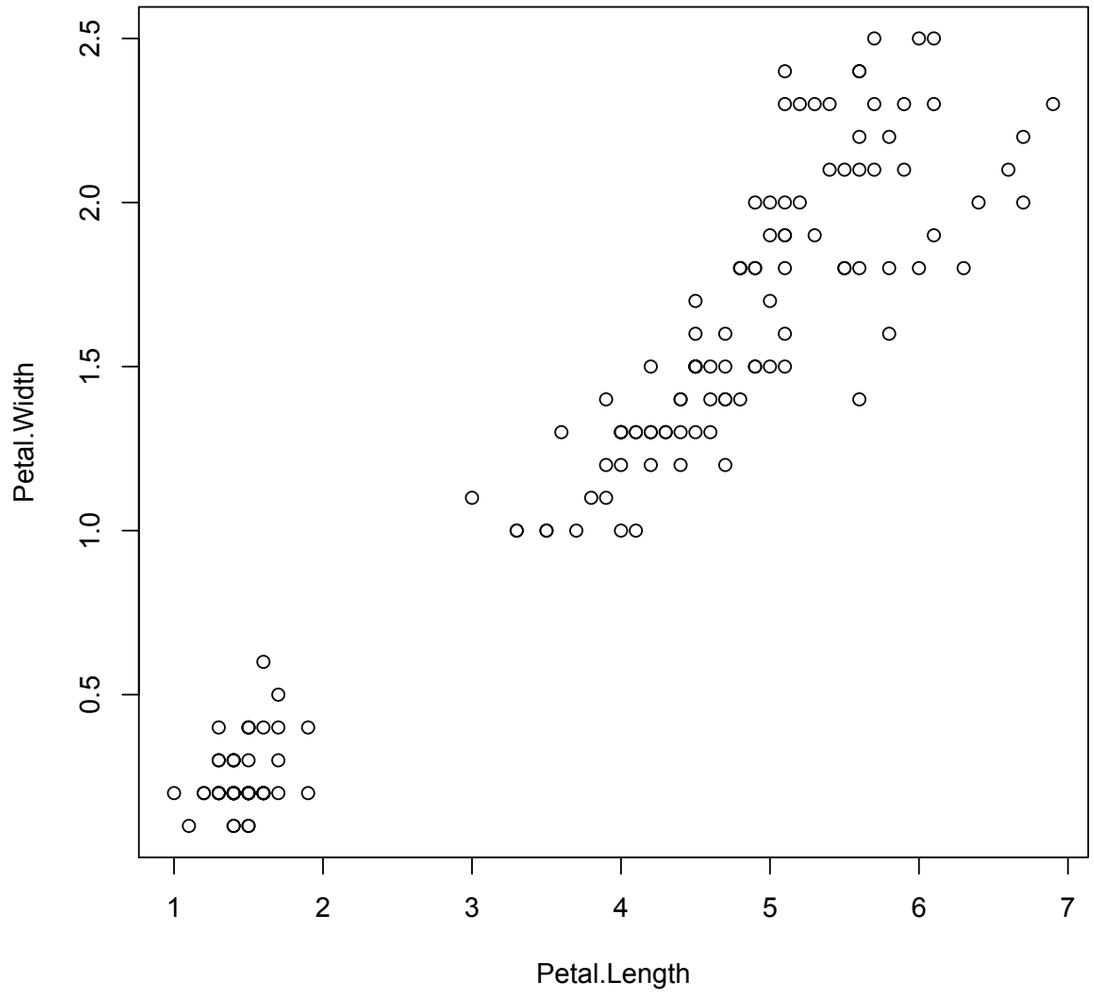
页面布局

plot( )函数画图

```{r}
plot(x=iris$Petal.Length,y=iris$Petal.Width)
```

散点图绘制函数格式：
plot(图形参数=数据)

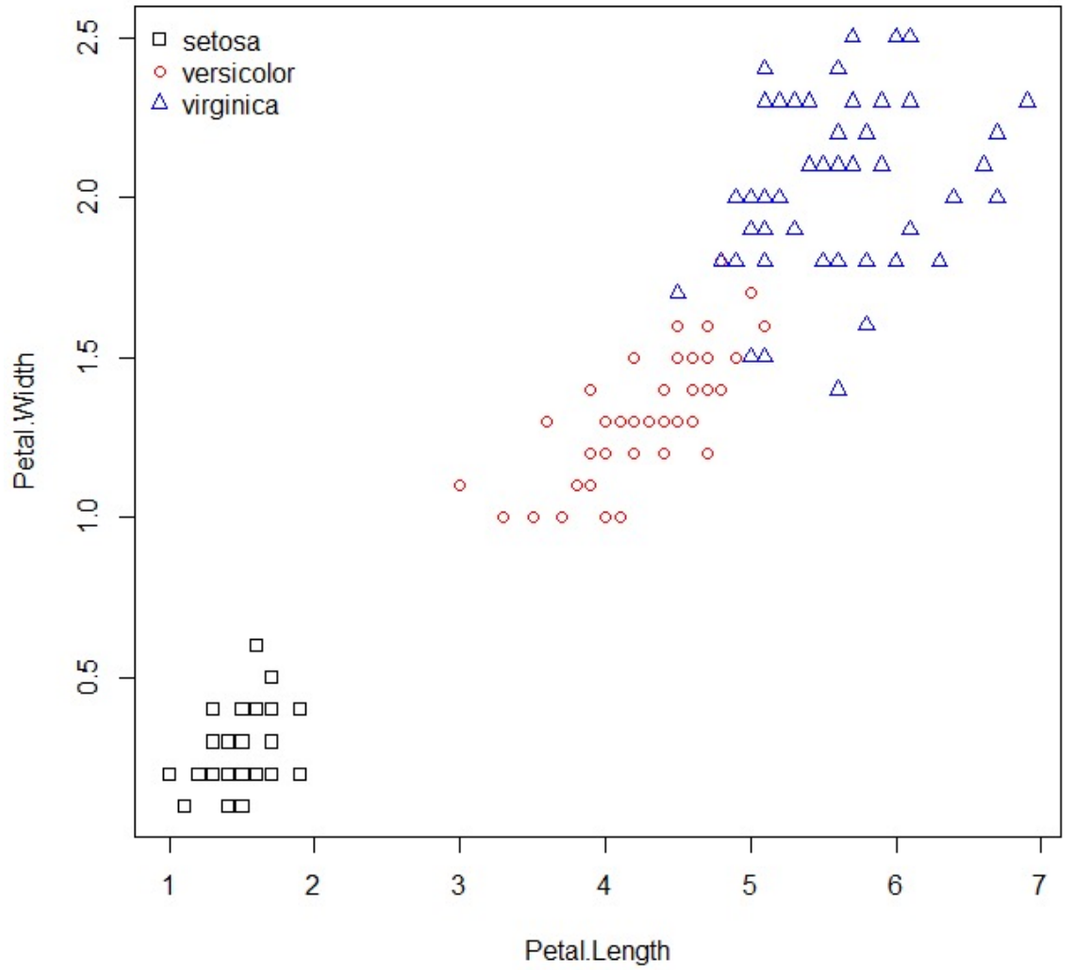
思考：如何将第三个变量画入图中？



基本图形元素

数据的第三维度画在了哪里？

	Petal.Length	Petal.Width	Species
1	1.4	0.2	setosa
2	1.4	0.2	setosa
3	1.3	0.2	setosa
4	1.5	0.2	setosa
5	1.4	0.2	setosa
6	1.7	0.4	setosa



如何用图形表示变量的变化？

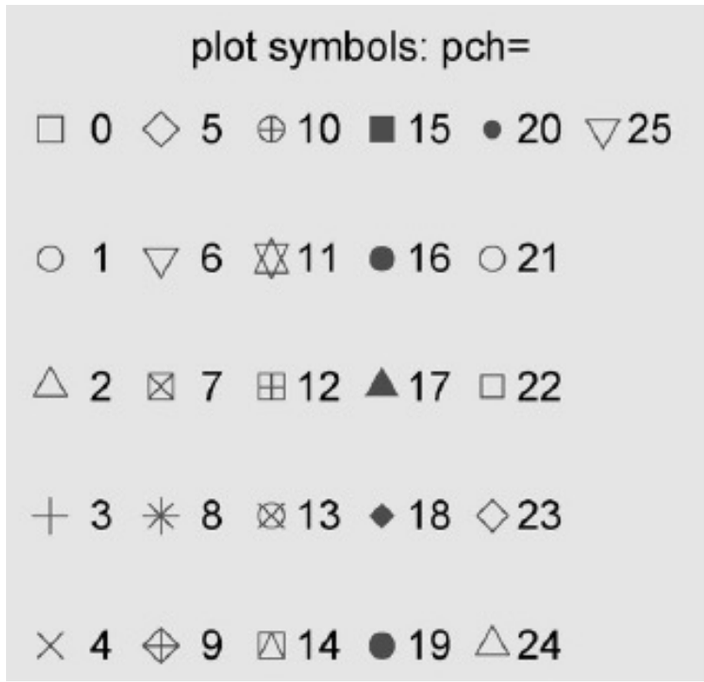
颜色、形状、线型、比例等作为元素存储在图形元素库中，数据通过与图形元素建立起映射关系进行可视化。

```
Petal.Length Petal.Width    Species
1          1.4         0.2     setosa
2          1.4         0.2     setosa
3          1.3         0.2     setosa
4          1.5         0.2     setosa
5          1.4         0.2     setosa
6          1.7         0.4     setosa
```

图形参数以整数编号形式
从计算机图形库中调取

图形参数与图形元素列表

x	横轴位置
y	纵轴位置
col	颜色
pch	点形状
fill	填充颜色
lty	线型
cex	大小



数据转为图形元素的过程

RStudio Source Editor

iris

Filter

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6			
6	5.4	3.9			
7	4.6	3.4			
8	5.0	3.4			
9	4.4	2.9			

Showing 1 to 10 of 150 entries

映射

x	y	color	shape	...
1.4	0.2	setosa		
1.4	0.2	setos		
1.3	0.2	setos		
1.5	0.2	setos		
1.4	0.2	setos		
1.7	0.4	setos		
1.4	0.3	setos		

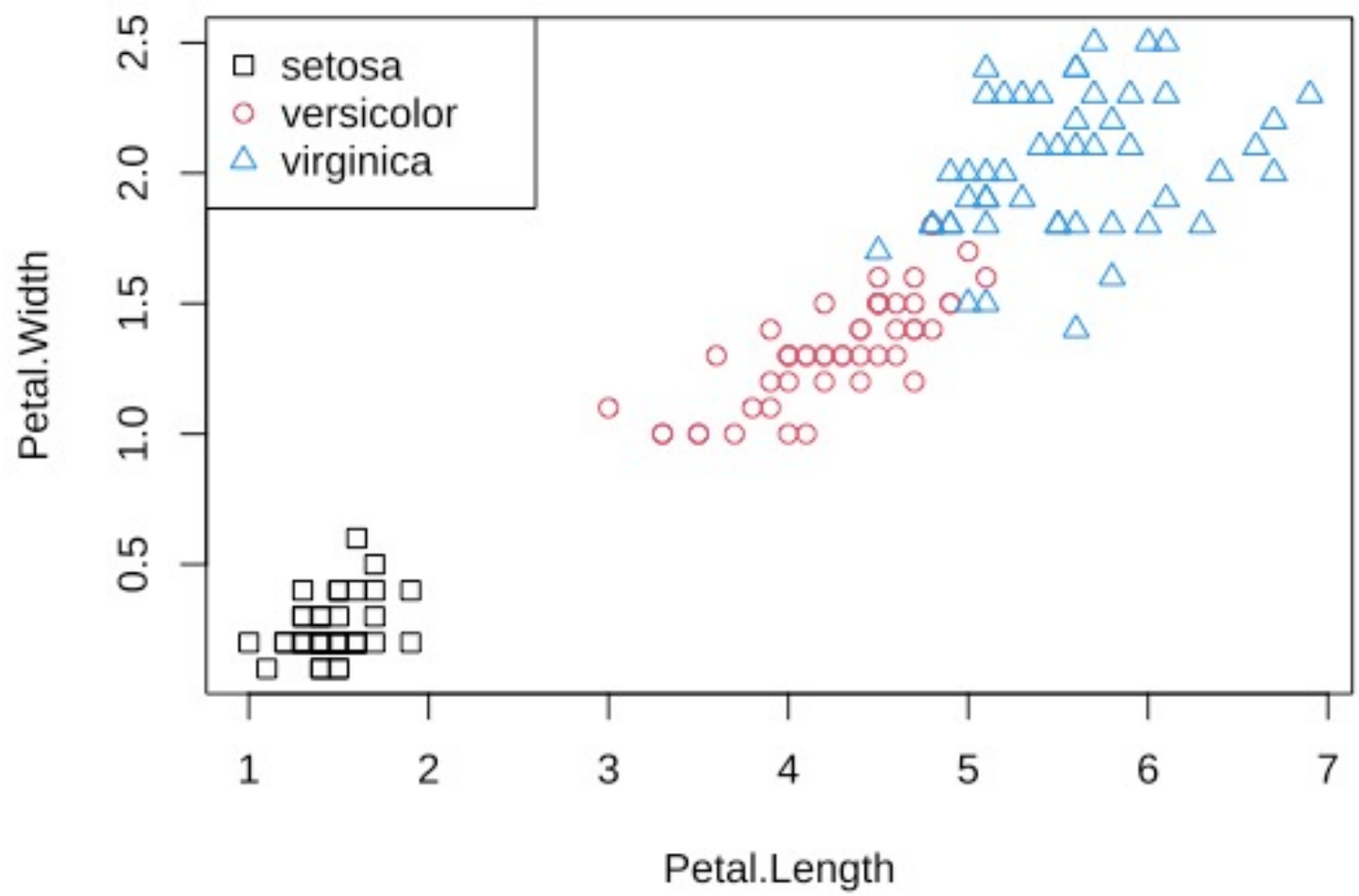
标度

x	y	color	shape	...
1.4	0.2	2		
1.4	0.2	2		
1.3	0.2	2		
1.5	0.2	2		
1.4	0.2	2		
1.7	0.4	2		
1.4	0.3	2		

分类型数据的标度

标度转化：强制数据转换为整数型

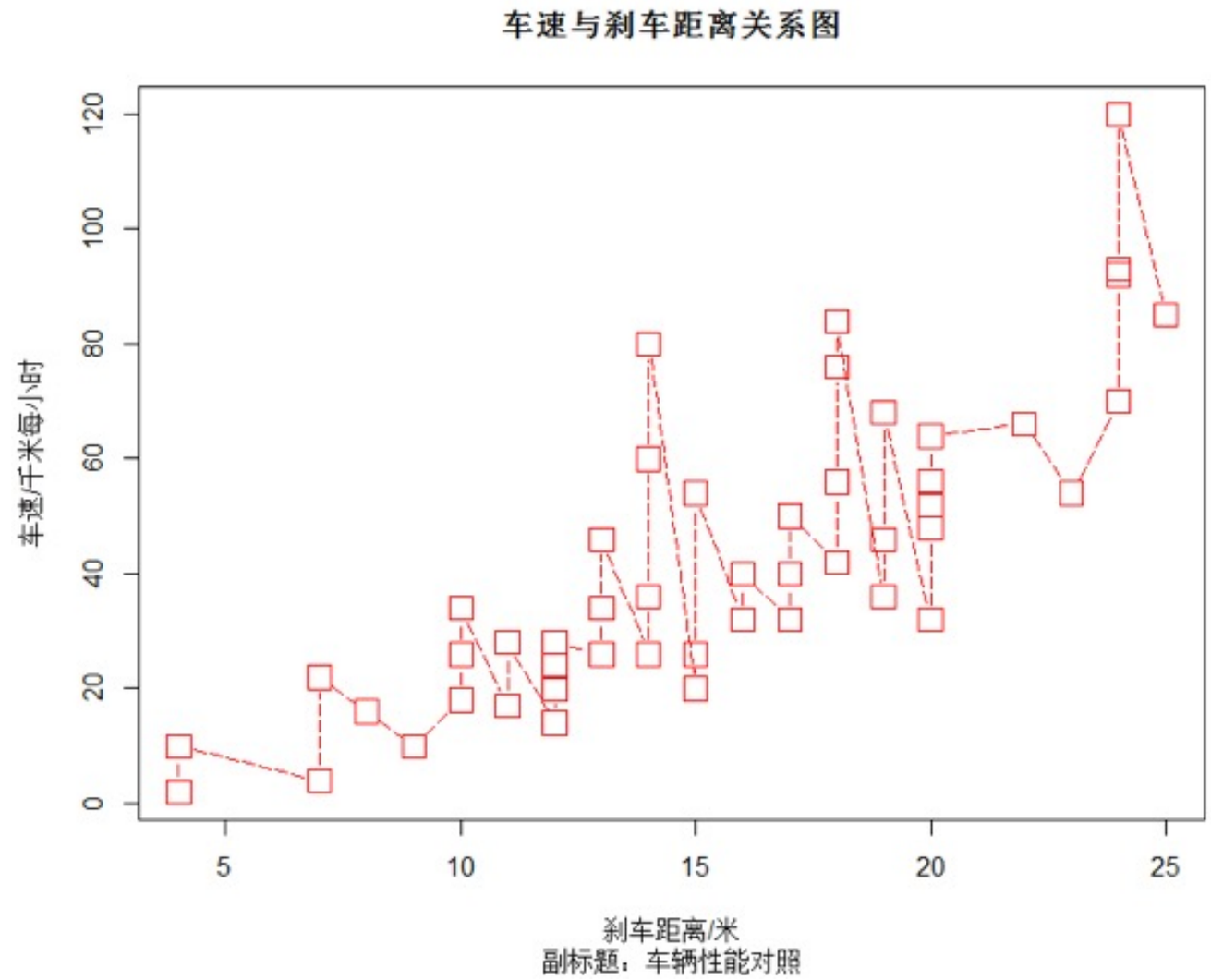
```
#转为整数
dx<-as.integer(iris[,5])

#绘制散点图
plot(iris[,c(3,4)],
       pch=c(0,1,2)[dx],
       col=c(1,2,4)[dx])

#添加图例
legend("topleft",
          legend=levels(iris$Species),
          pch=c(0,1,2),col=c(1,2,4))
```



点线图

对数据集cars的两个变量使用散点图函数绘图，调节其中的标题、图形元素比例.

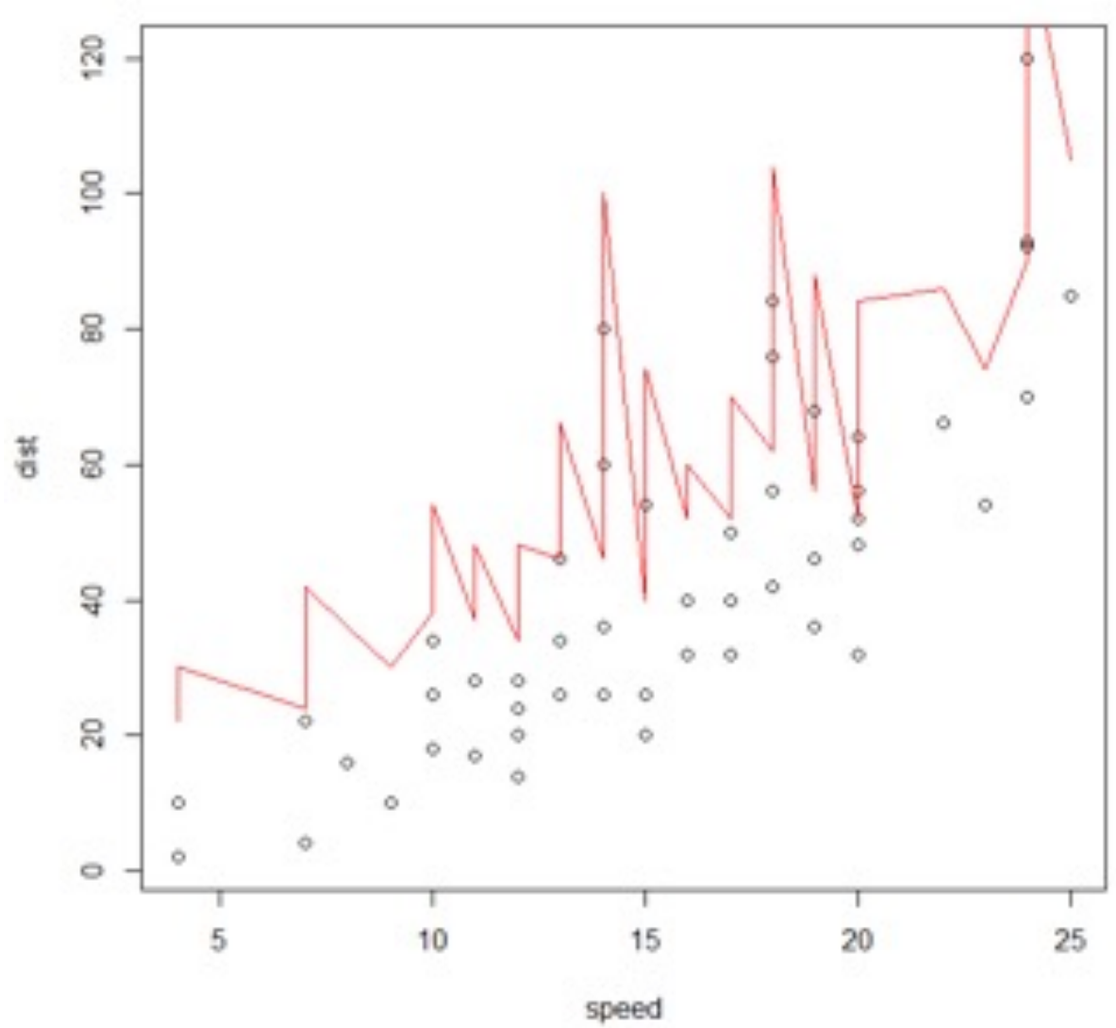
```{r}
plot(x = cars$speed, y = cars$dist,
     main = "车速与刹车距离关系图",
     xlab = "刹车距离/米",
     ylab = "车速/千米每小时",
     sub = "副标题：车辆性能对照",
     pch = 0,type = "b", lty = 5)
```



lines( )添加线

lines函数用于添加第二组数据进入plot绘制的图形中

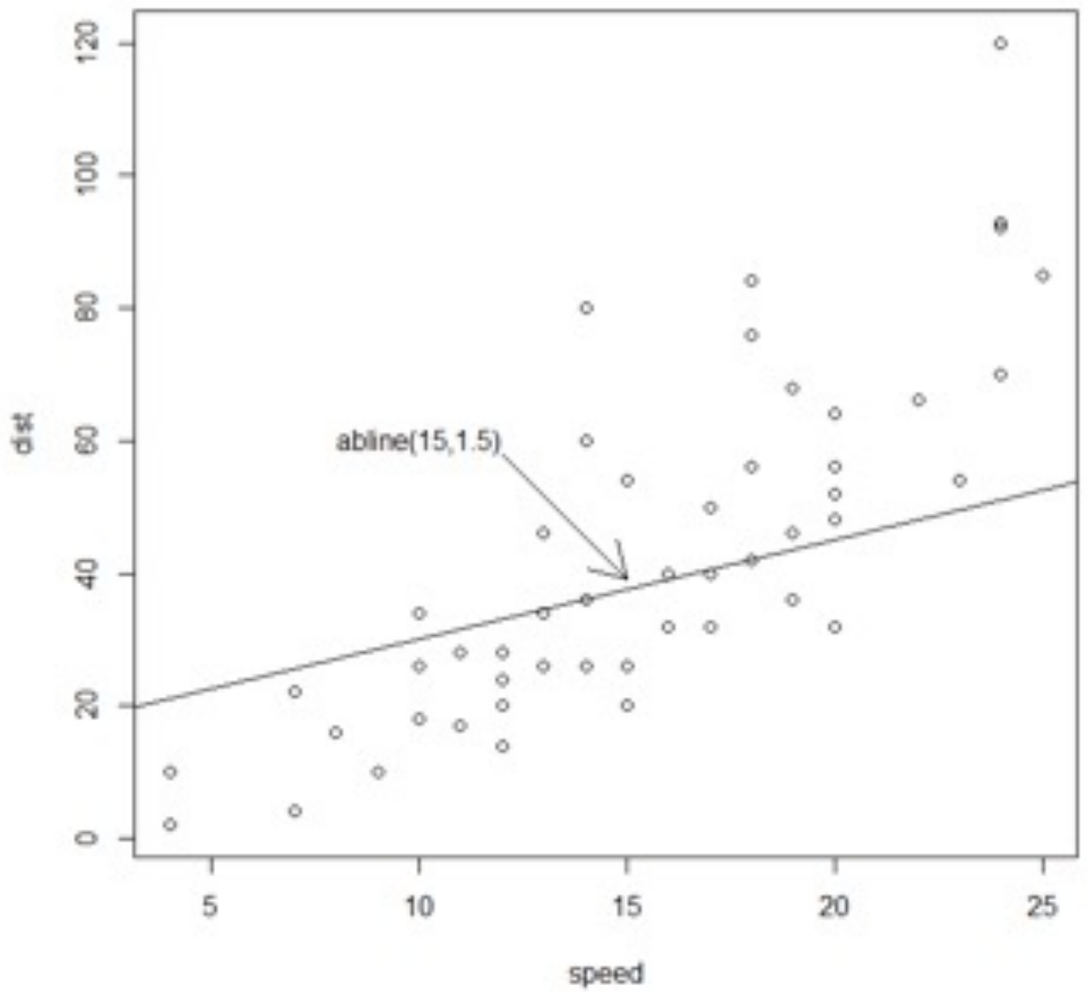
```{r}
plot(x=cars$speed,
     y=cars$dist)
lines(x=cars$speed,
      y=cars$dist+20,
      col="red")
```



abline( )辅助线

abline()即截距-斜率方式的辅助线添加函数

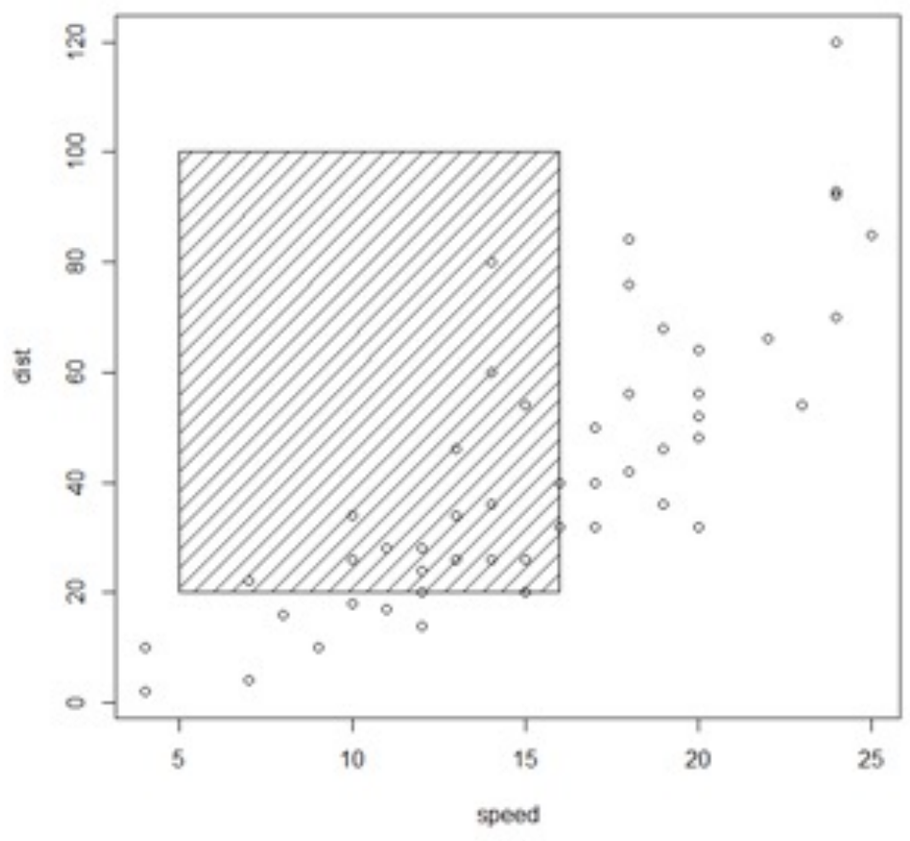
```{r}
plot(x=cars$speed,
     y=cars$dist)
abline(a=15,b=1.5)
text(10,60,
     labels = "abline(15,1.5)")
arrows(10,58,
       15,39)
```



顾名思义，a参数为截距，b为斜率

rect添加形状

```r
plot(x=cars$speed,
     y=cars$dist)
rect(5,20, 16,100,
     density = 10)
```

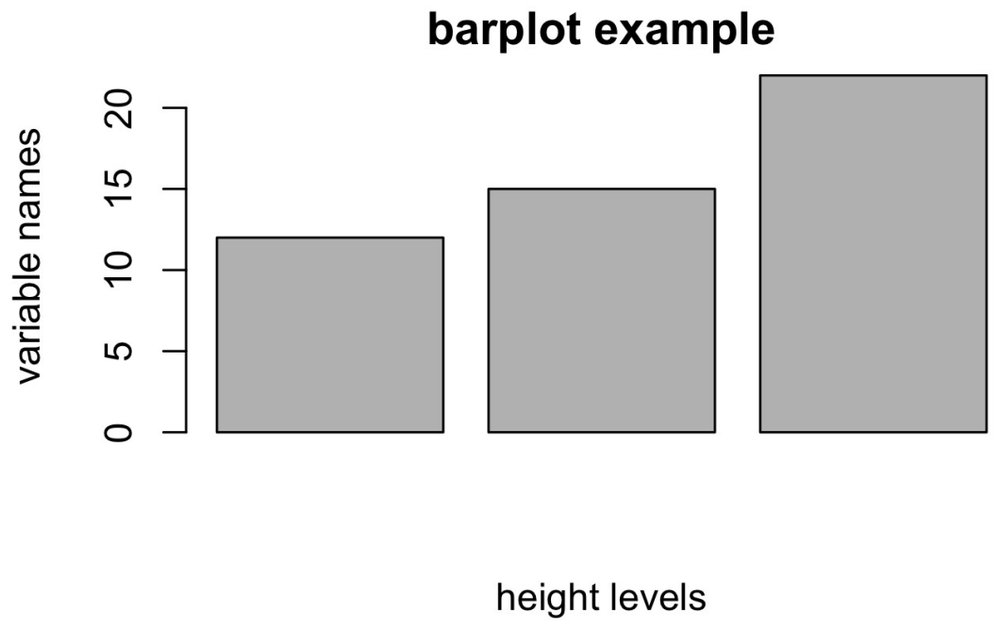


title( )	添加标题
text( )	在绘图区添加文字
mtext( )	在边界区添加文字
points( )	添加点
lines( )	添加线
abline( )	添加参考线
axis	添加坐标轴
legend	添加图例
polygon	添加多边形

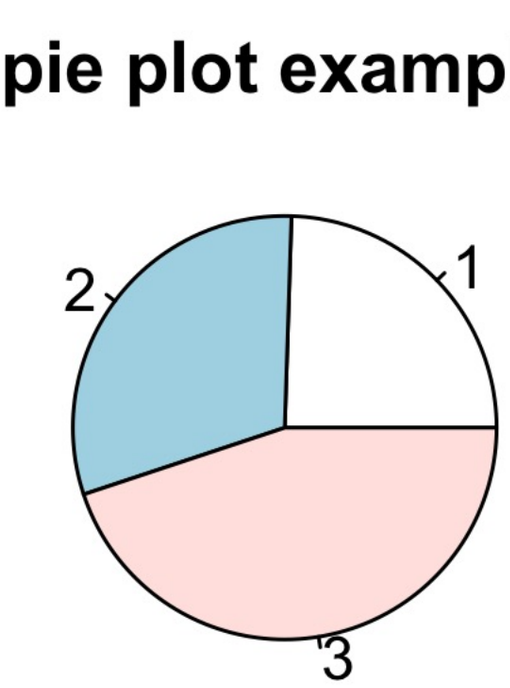
其他图型

条形图函数

```{r}
barplot(height=c(12,15,22),
        main = "barplot example",
        xlab = "height levels",
        ylab = "variable names")
```




```{r}
pie(x=c(12,15,22),
    main = "pie plot example")
```



par设置页面参数

图形界面的分割需要在设备默认参数里操作，即通过par()

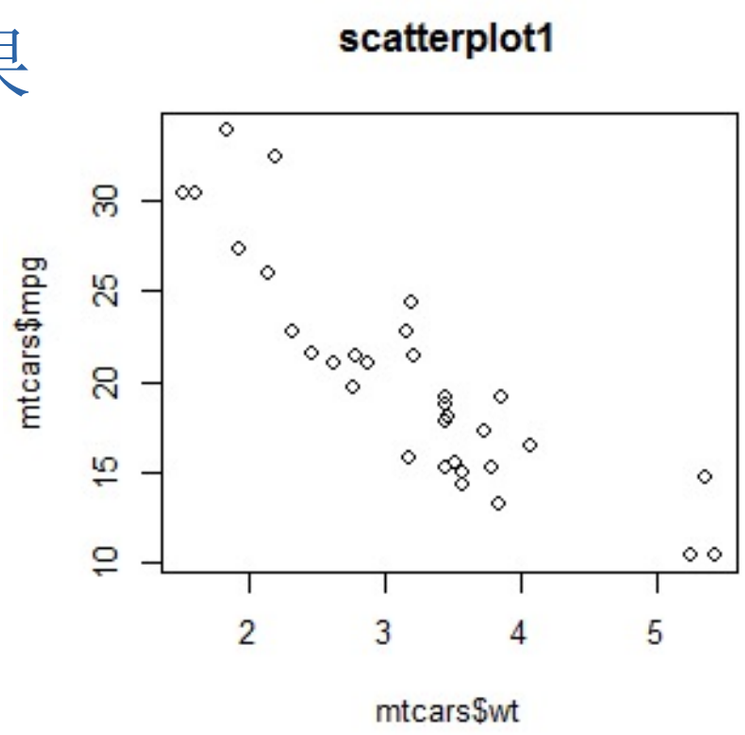
例如:par(mfrow=c(2,2))

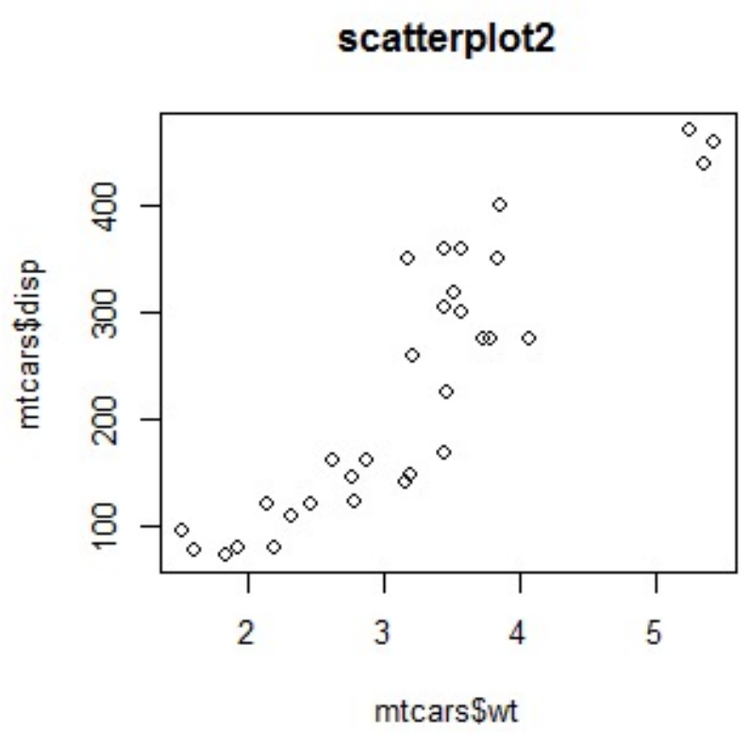
将图形窗口分割为2X2的四个子图
在随后的做图过程中，每画一个图就按右侧示意图中的编号顺序摆放。
尝试下列代码会生成什么样的图形

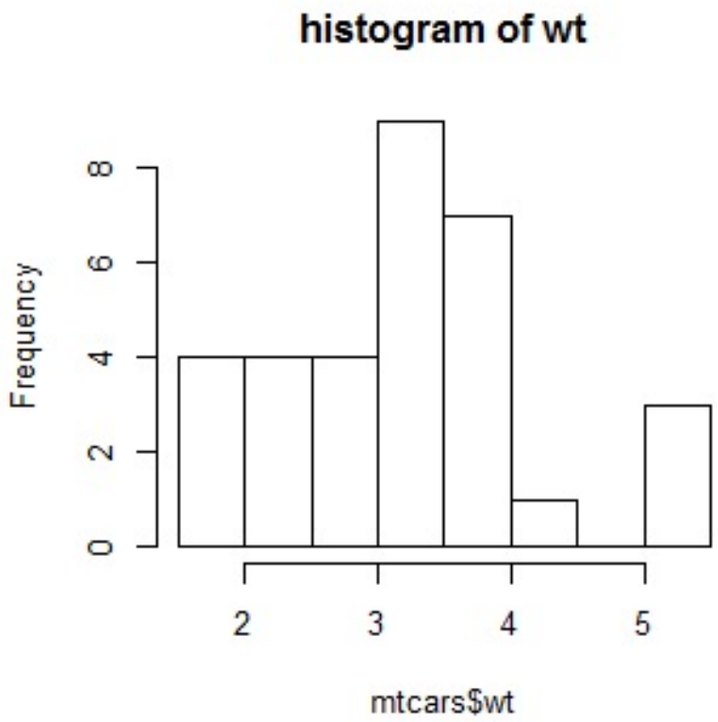
1	2
3	4

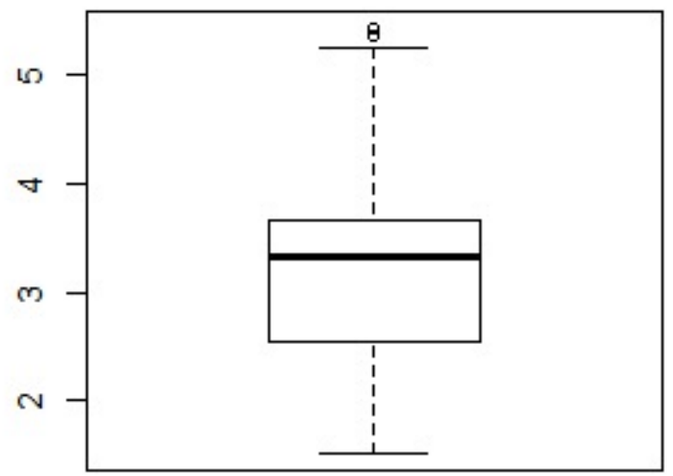
```
> par(mfrow=c(2,2))
> plot(mtcars$wt,mtcars$mpg,main="scatterplot1")
> plot(mtcars$wt,mtcars$disp,main="scatterplot2")
> hist(mtcars$wt,main="histogram of wt")
> boxplot(mtcars$wt,main="boxplot of wt")
```

图形输出结果


scatterplot1
mtcars$mpg
10
15
20
25
30
2
3
4
5
mtcars$wt


scatterplot2
mtcars$disp
100
200
300
400
2
3
4
5
mtcars$wt

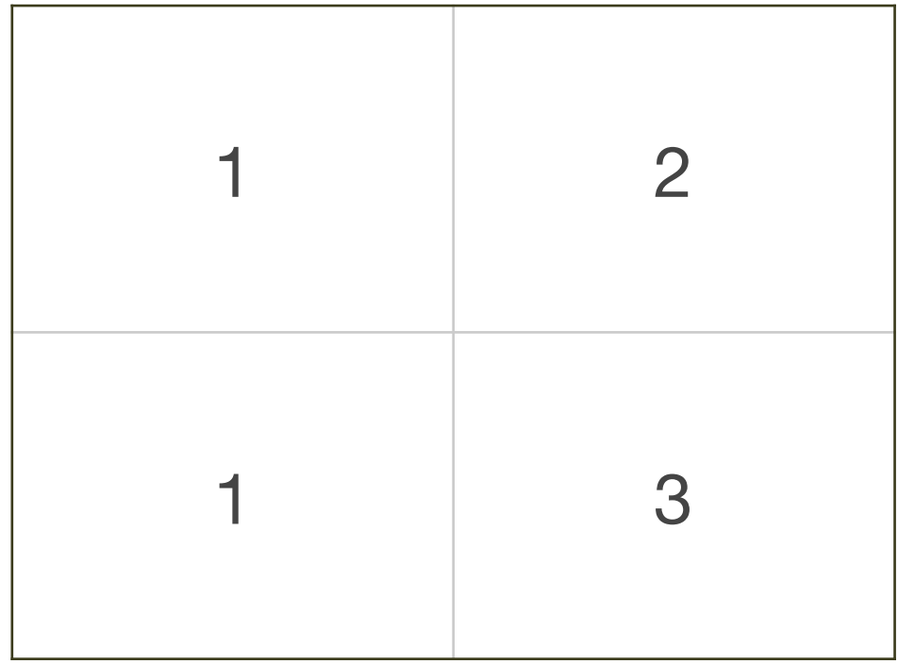

histogram of wt
Frequency
0
2
4
6
8
2
3
4
5
mtcars$wt


boxplot of wt
2
3
4
5

layout设置页面结构

layout函数将一个写好顺序编号的矩阵替代默认分割方式，进而可以更灵活的将部分分面合并，任意构造出分面

```
> layout(matrix(c(1,1,2,3),nrow=2))
> hist(mtcars$wt)
> hist(mtcars$mpg)
> hist(mtcars$disp)
```



3.ggplot2绘图系统

ggplot2绘图原理

修改图形元素

主题元素

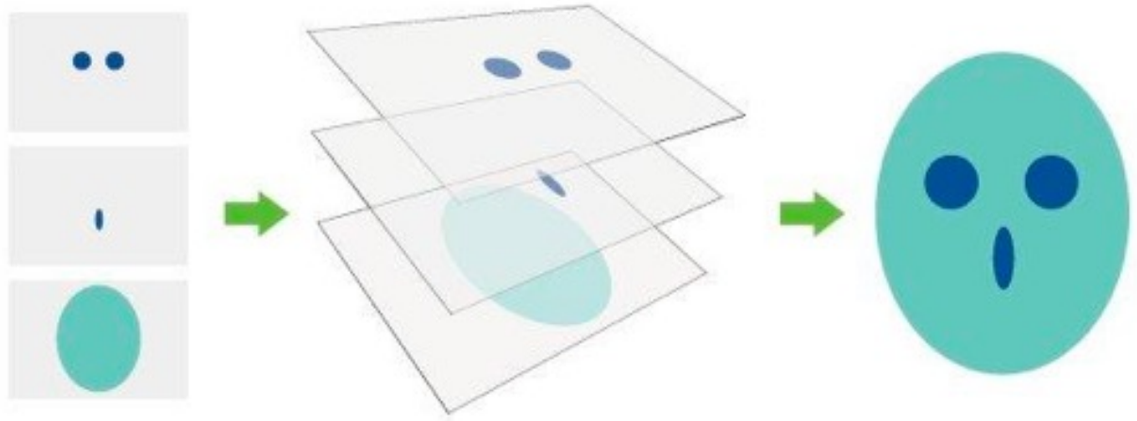
图形语法

《The Grammar of Graphics》建立了一套图形语法

ggplot2是R语言中图形语法的一套实现工具

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

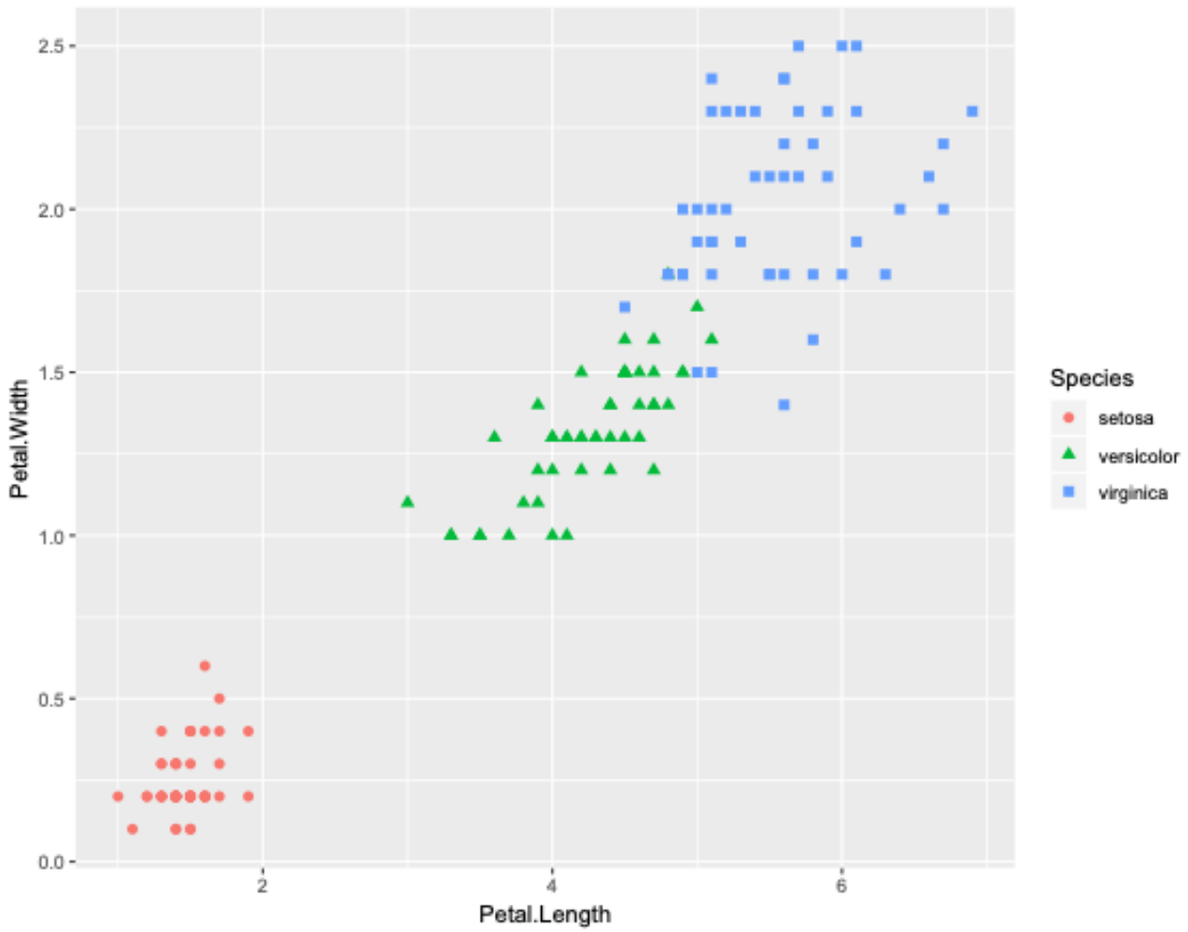
ggplot以图层方式描述图形，数据集默认只接受data.frame格式



iris散点图

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

```
> head(iris[,3:5])
  Petal.Length Petal.Width Species
1          1.4         0.2  setosa
2          1.4         0.2  setosa
3          1.3         0.2  setosa
4          1.5         0.2  setosa
5          1.4         0.2  setosa
6          1.7         0.4  setosa
```

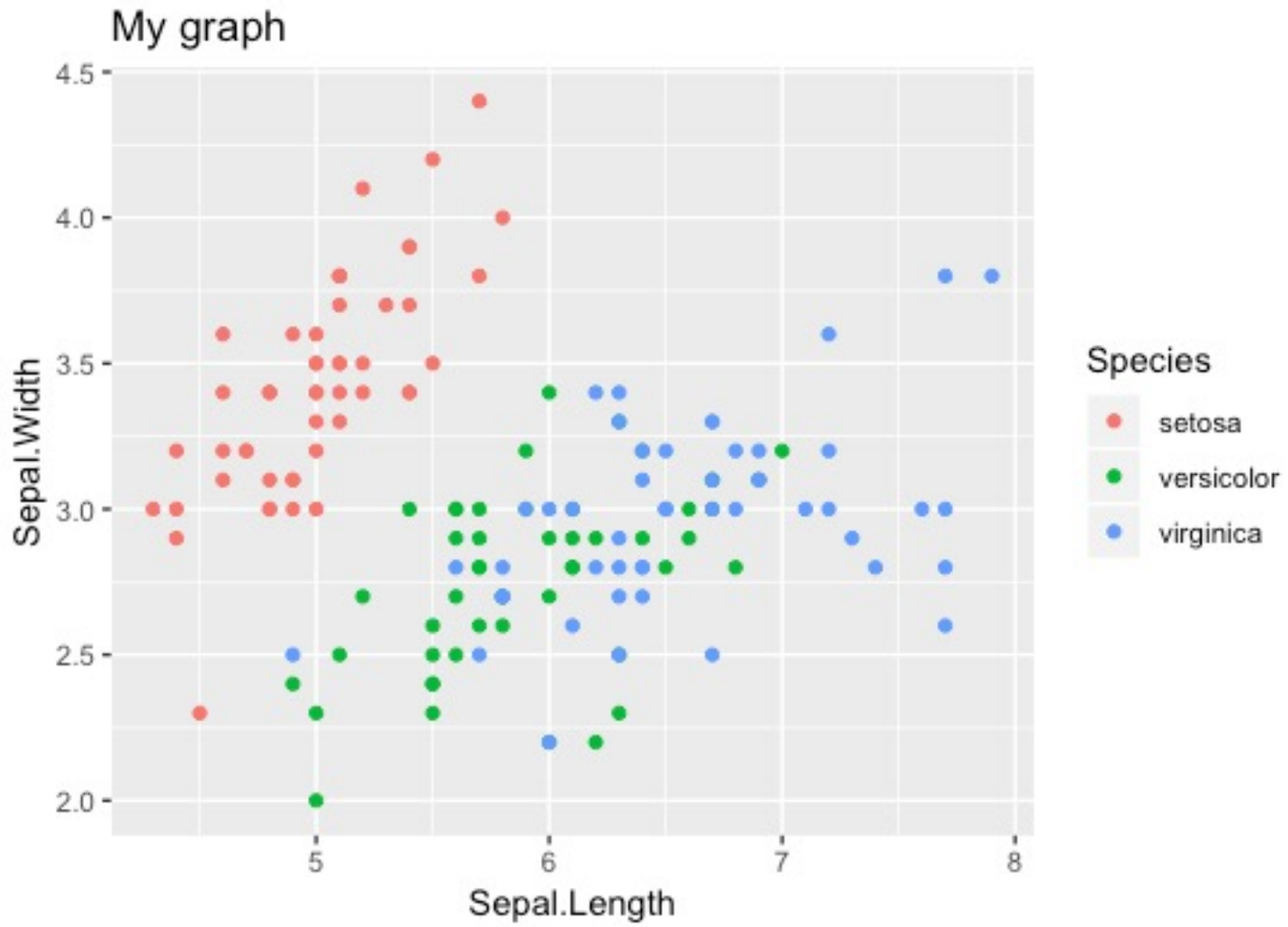


ggplot2图像三要素：

1. 数据
2. 图形元素映射
3. 几何对象 geom
4. （统计变换 stat）

代码分析

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

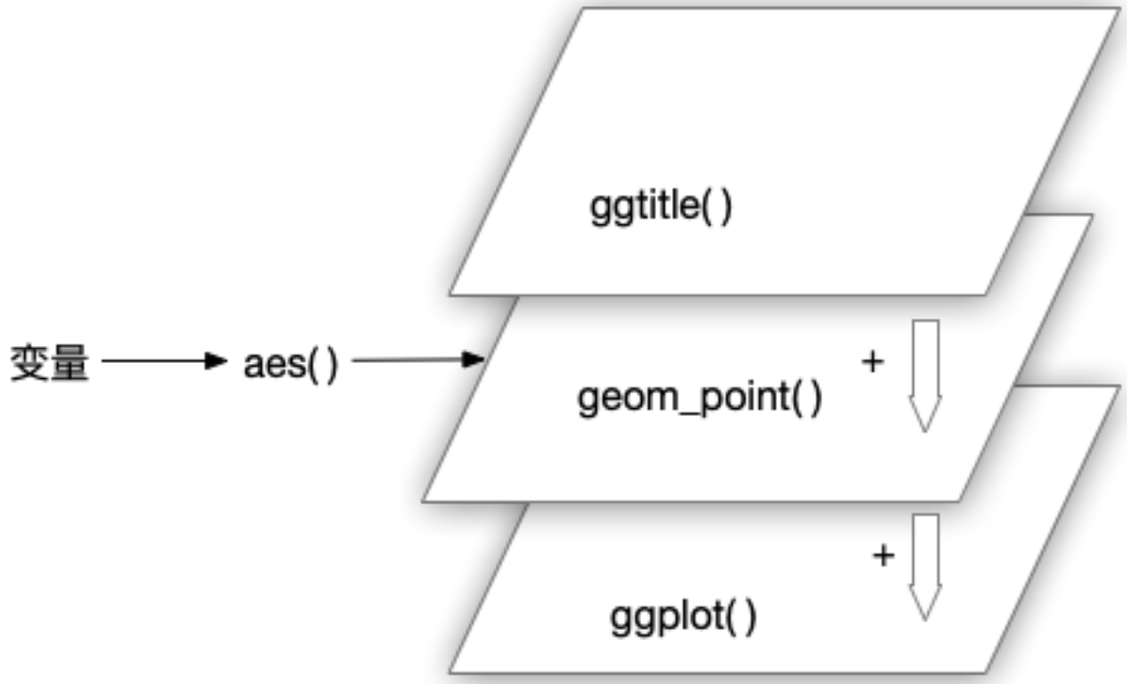


绘图语法原理

ggtitle()

geom_point()+

ggplot()+



最底层ggplot()对所有图层是透明的，其他图层之间相互独立，优先使用本层函数内数据，以底层数据作为默认缺省数据。

aes()：实现数据与图形元素之间的映射，并配以完整图例

常用几何对象与图形元素

x	横轴位置
y	纵轴位置
color/colour	颜色
alpha	透明度
fill	填充颜色
linetype	线型
shape	点形状
size	大小

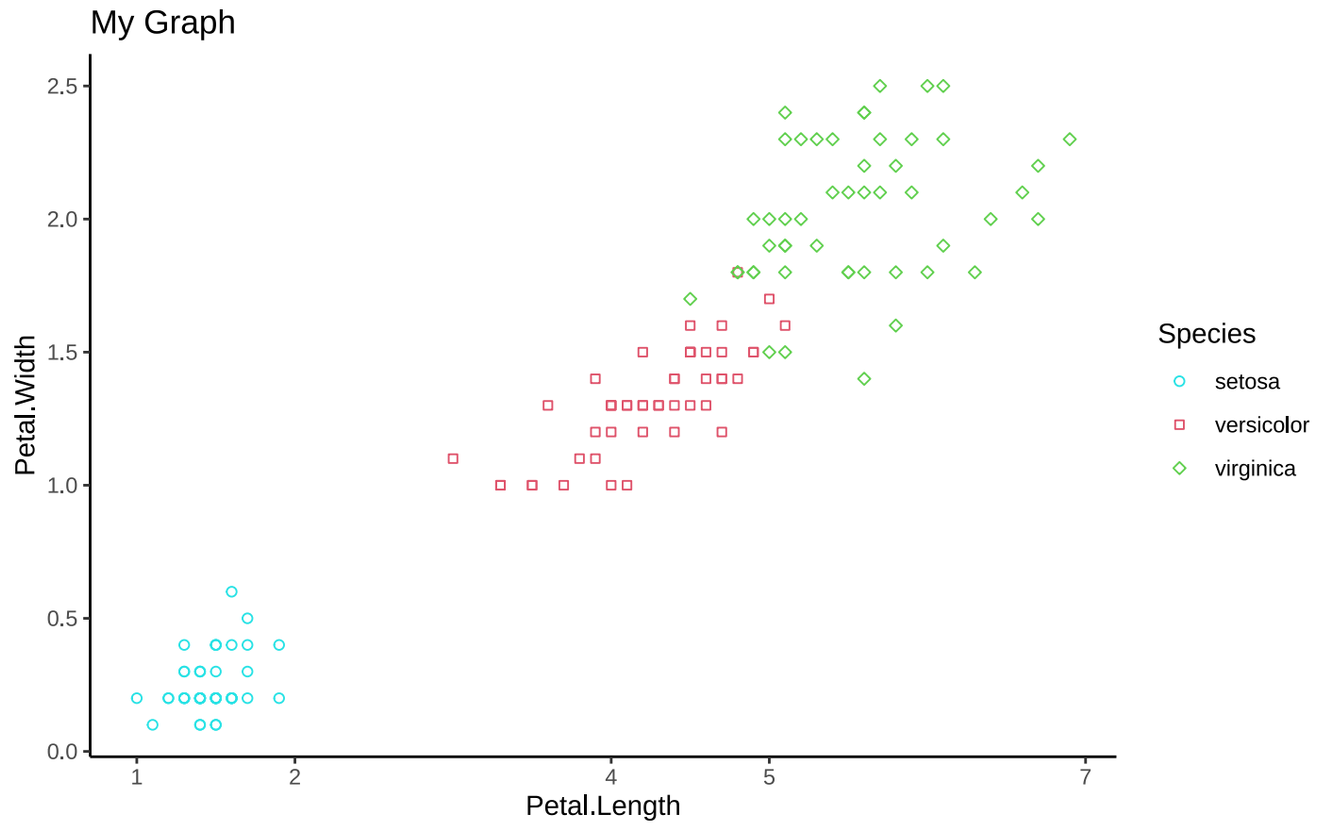
geom_point( )	点
geom_jitter( )	扰动点图
geom_line( )	线
geom_abline( )	辅助线
geom_boxplot( )	箱图
geom_violin( )	琴线图
geom_histgram( )	直方图
geom_freqpoly( )	频率图
geom_bar( )	大小

scale修改图形元素

scale函数用于修改映射中的图形元素显示方式

```
> p= p+
      scale_shape_manual(values = c(21,22,23) )+
      scale_color_manual(values = c(5,2,11))+
      scale_x_continuous(breaks = c(1,2,4,5,7))
> p
```

仅作用于参与展示数据的图形元素



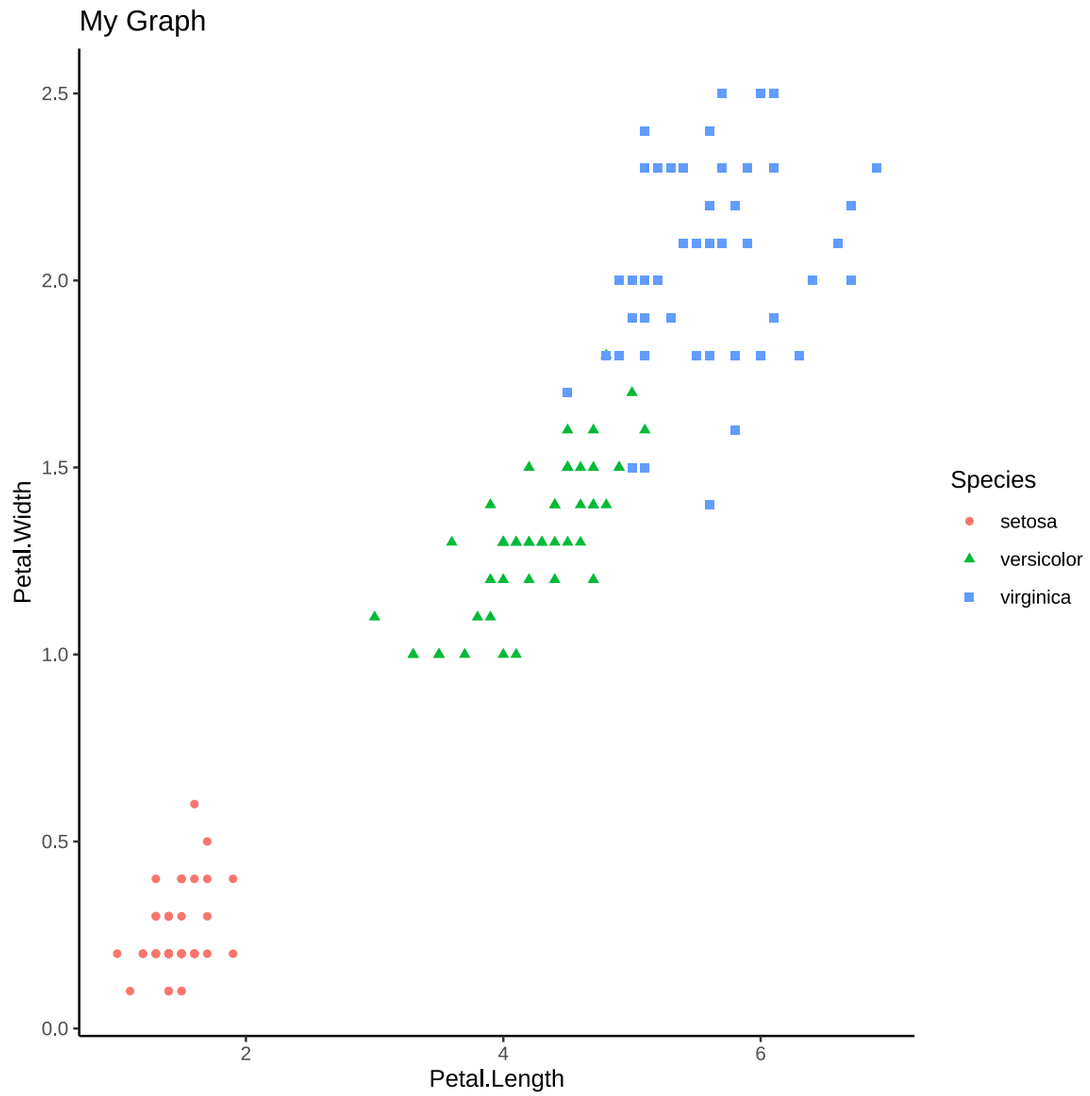
theme主题元素调整

theme_XXX函数能够为图形提供一整套的主题配色方案，与叠加几何对象相同的方式使用：

```
> p= p+theme_classic()
> p
```

其他拓展主题包提供了非常多且专业的主题，例如ggthemes工具包。

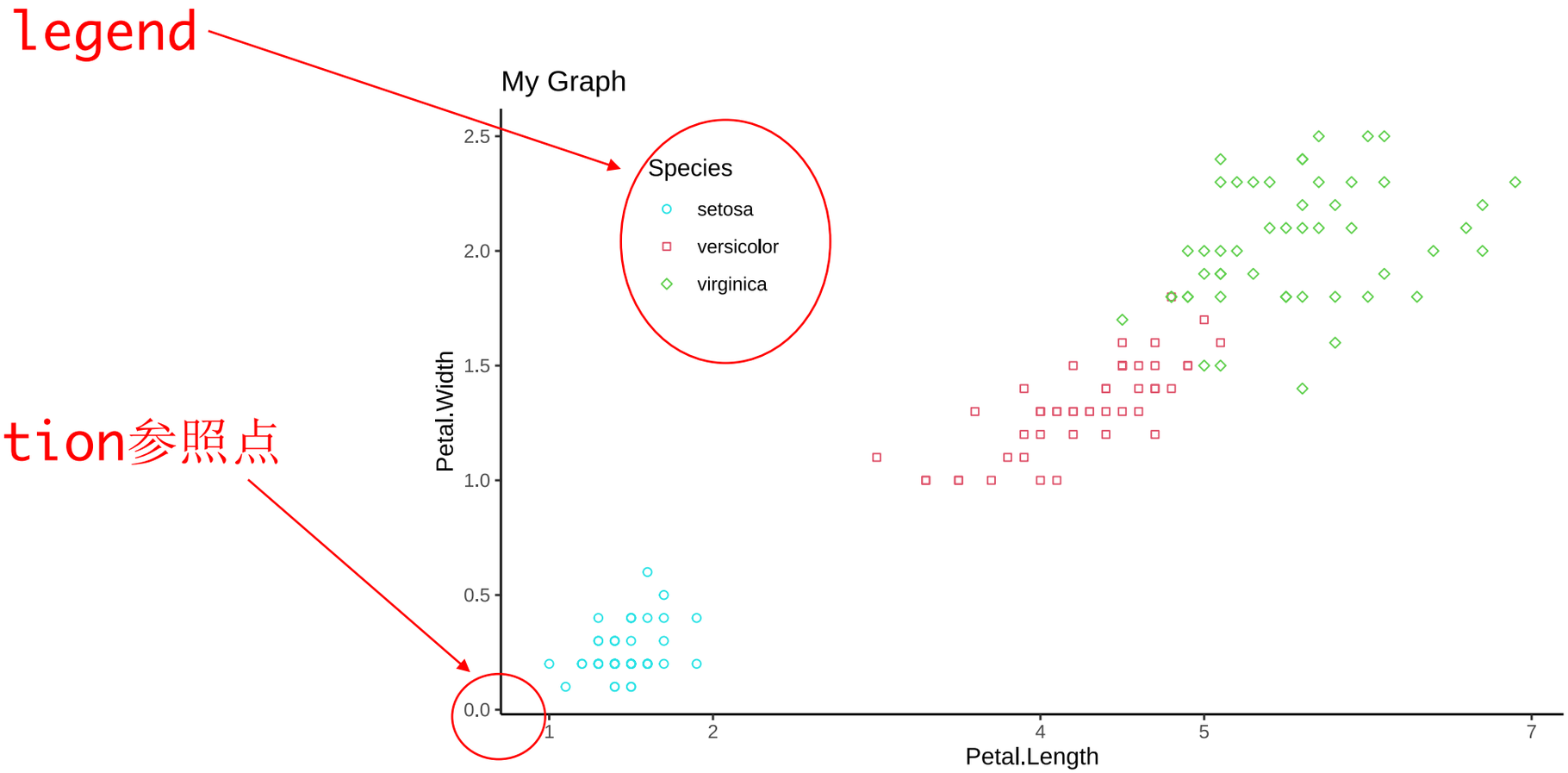
使用theme类函数会覆盖前边的主题修改。



theme主题元素调整

对主题元素中的图例调整：

```
> p=p+theme(legend.position = c(0.2,0.8))
> p
```



数据可视化

4.绘图技巧

条形图展示数据

直方图展示数据

其他图形类型

图形保存

单变量条形图

geom_bar条形图对象需要考虑统计方法，默认单变量情况下对相同的x值进行计数堆积

尝试以下例子

```
ggplot(mpg,aes(class))+geom_bar()
```

同样，条形图中fill参数可以接受分类变量

```
ggplot(mpg,aes(class))+geom_bar(aes(fill=factor(year)))
```

条形图位置调整

position参数控制图形元素以什么逻辑排放，在条形图中常用dodge和stack控制展示方式。如下

```
ggplot(mpg,aes(class))+
  geom_bar(aes(fill=factor(year)),position = "dodge")
```

```
ggplot(mpg,aes(class))+
  geom_bar(aes(fill=factor(year)),position = "stack")
```

position参数含义：
“dodge” 并排避免重复
“fill” 堆叠图形并标准化为1
“identity” 不做调整
“jitter” 添加扰动
“stack” 堆叠起来

坐标系变换

coord_XXX坐标系对象将调整绘图所在的坐标体系，其中极坐标系coord_polar()函数是常用的坐标转换方式。

```
ggplot(mpg,aes(class))+
  geom_bar(aes(fill=factor(year)),position = "dodge")+
  coord_polar()
```

不难发现，饼图与条形图其实是相同图形在不同坐标系下的表现方式

单变量直方图

geom_histogram是直方图几何对象

```
ggplot(mpg,aes(hwy))+geom_histogram(binwidth = 1.5)
```

直方图对象存在一个常用参数binwidth，其中bin指直方图中的条形块，通过设置条形块宽度调节密集程度

geom_freqpoly是频数线图，与直方图相似

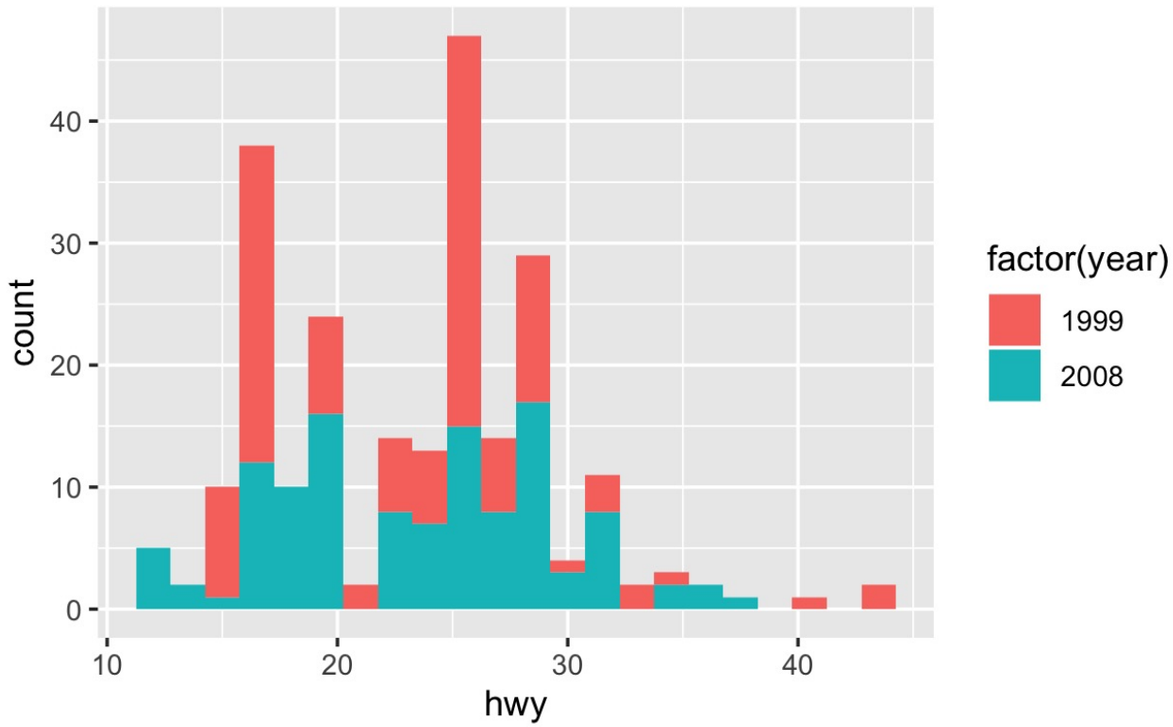
```
ggplot(mpg,aes(hwy))+geom_freqpoly(binwidth=1.2)
```

binwidth参数同样控制了区间划分的密集程度

直方图中的分类

fill参数控制填充色彩，仅接受分类型数据

```
ggplot(mpg,aes(hwy))+geom_histogram(aes(fill=factor(year)))
```



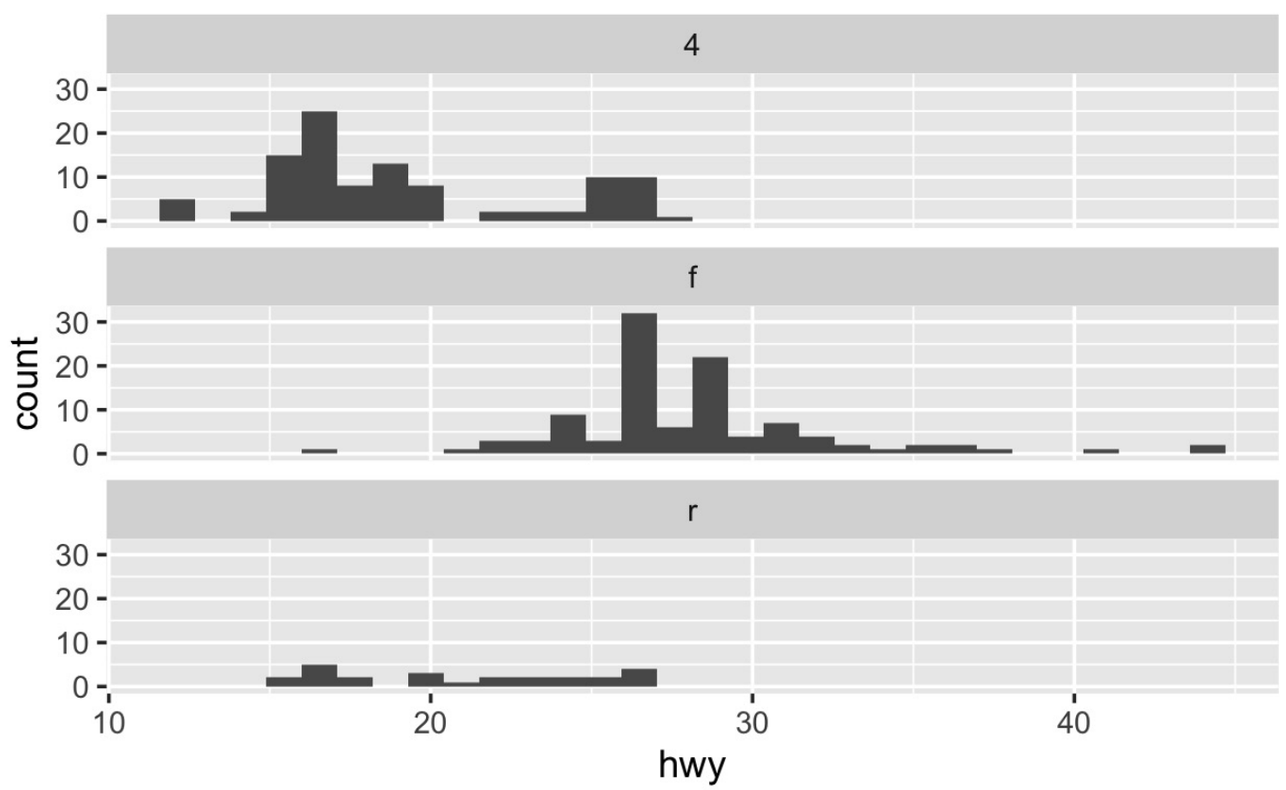
分面展示分组

facet分面将数据对象按某变量属性进行分类，是分组统计的方式之一。注意：分面并非页面分割。

控制分面的函数有：
facet_wrap()
facet_grid()

```
ggplot(mpg,aes(hwy))+
  geom_histogram()+
  facet_wrap(~drv,ncol=1)
```

上例中以公式表达drv为控制分面变量，ncol设置的展示为1列



两连续型数据

案例数据集：mpg数据集存放在ggplot2工具包中，记录了美国1999年和2008年部分汽车的制造厂商、型号、类型、驱动系统和油耗量信息，数据来源于美国环境保护署公开信息。

绘制简单散点图

```
ggplot(mpg,aes(displ,hwy))+geom_point()
```

图形对象之间靠+进行叠加

```
ggplot(mpg,aes(displ,hwy))+geom_point()+geom_smooth(span=0.2)
```

geom_smooth添加了趋势线，其中span控制趋势线的适应程度

添加标题信息

同样原理，图标题和坐标轴标题也是以图层对象方式出现

```
ggplot(mpg,aes(displ,hwy))+
  geom_point(alpha=1/2)+
  xlab("city driving(mpg)")+
  ylab("highway driving(mpg)")+
  ggtitle("mpg dataset")
```

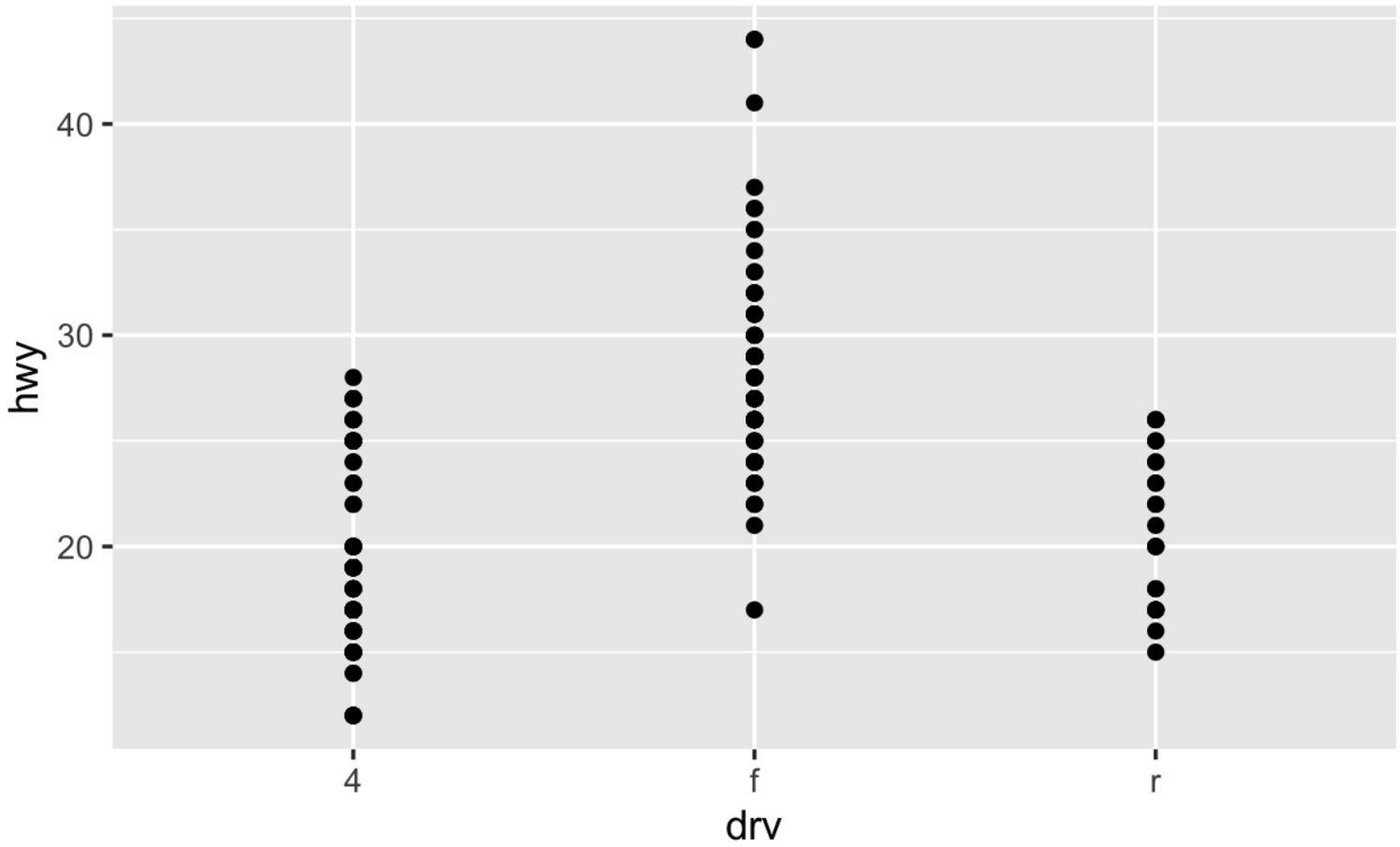
其中，手动标度了图形元素alpha（透明度）为1/2,即一半的透明度

另外,xlab(“NULL”)将设定无坐标轴标题

分类连续型分布图

当其中一个数据为离散型，散点图中的点将密集分布

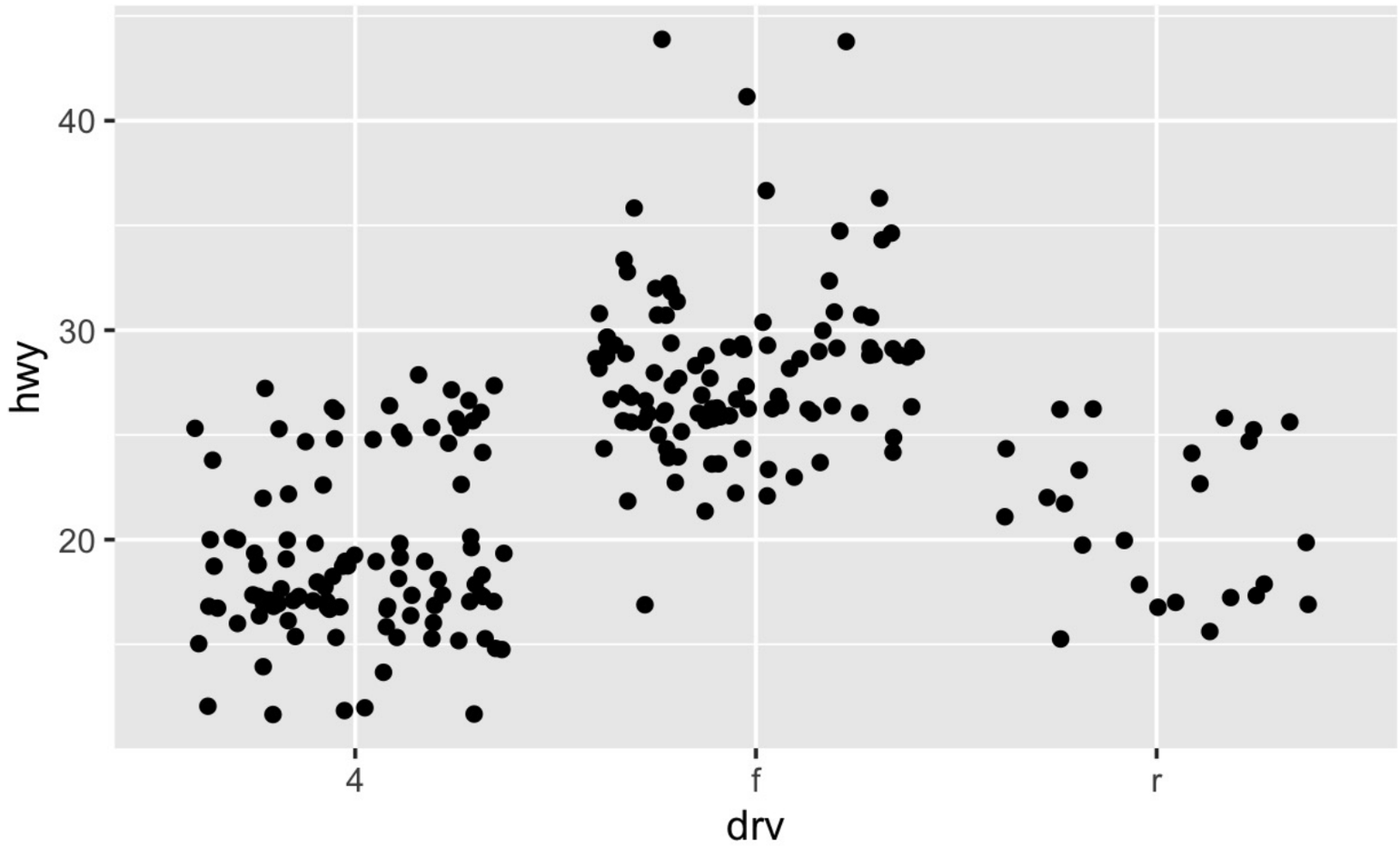
```
ggplot(mpg,aes(drv,hwy))+geom_point()
```



随机扰动散点图

geom_jitter对象在散点图中增加了防止叠加覆盖，为重叠点增加了随机扰动，常用于存在重叠点的图形

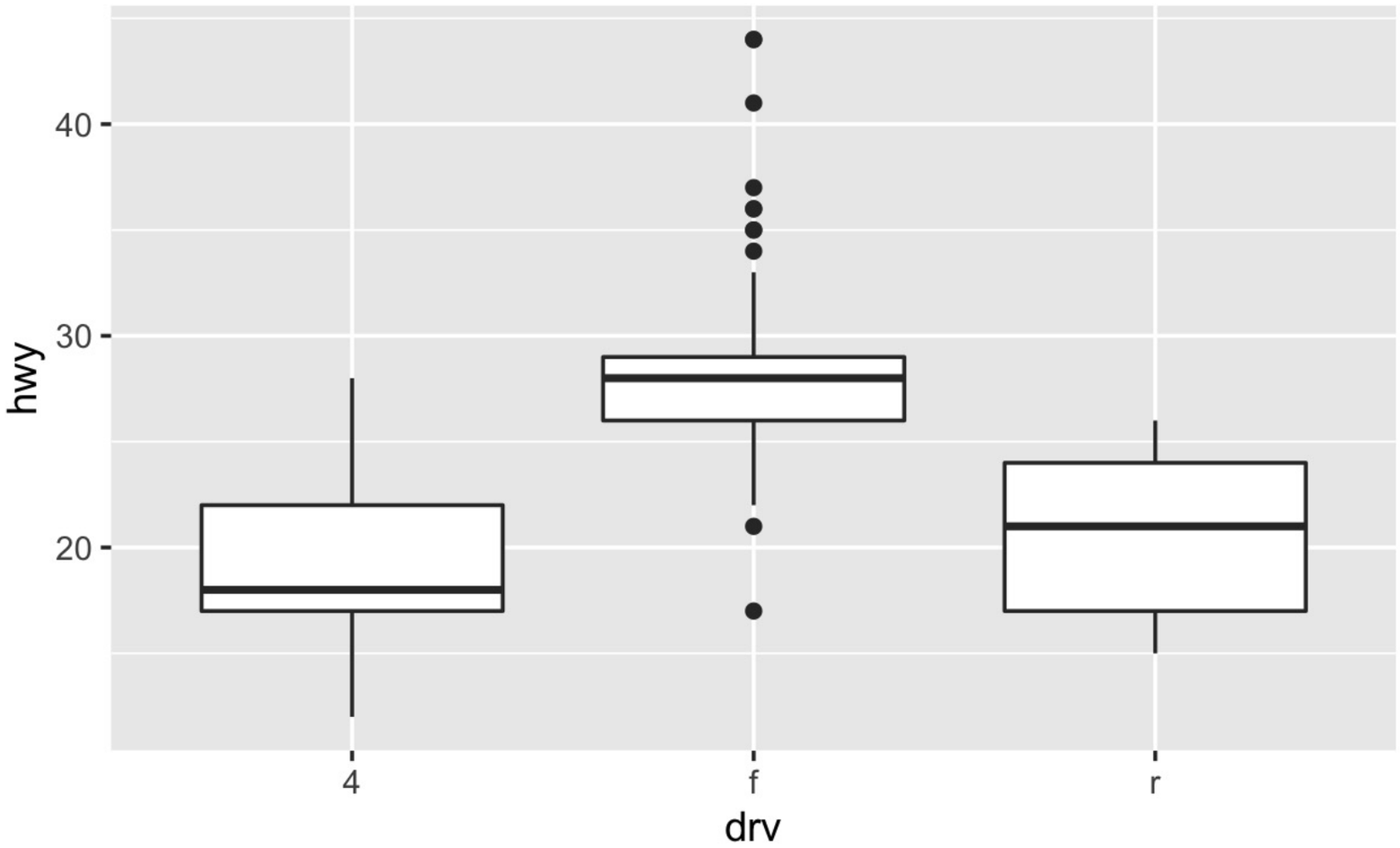
```
ggplot(mpg,aes(drv,hwy))+geom_jitter()
```



分组箱图

geom_boxplot绘制箱图对象，放在x位置的参数为分组控制参数，而y位置的参数为分析目标变量

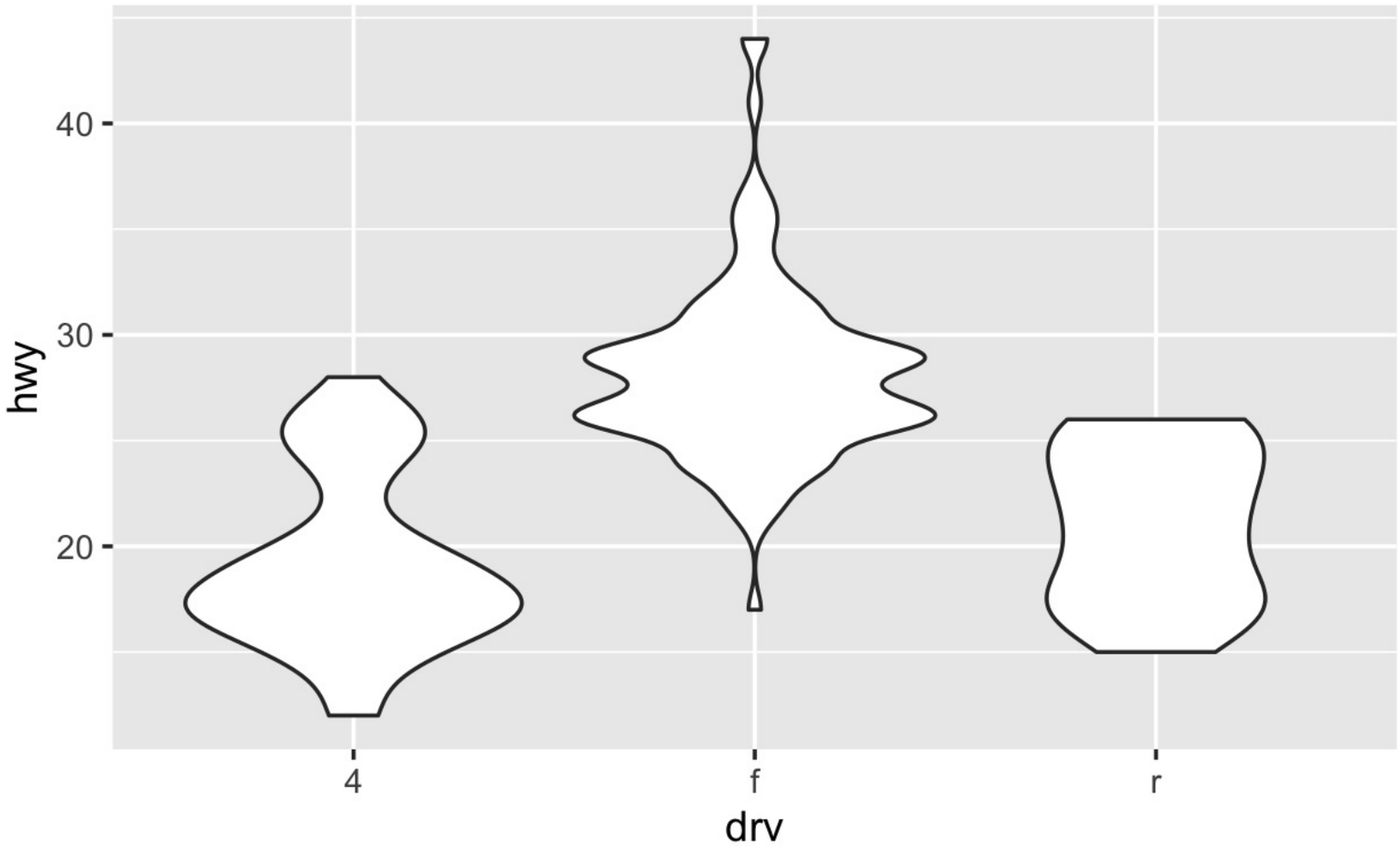
```
ggplot(mpg,aes(drv,hwy))+geom_boxplot()
```



小提琴图

小提琴图和箱图类似，增加了数据的分布密集情况

```
ggplot(mpg,aes(drv,hwy))+geom_violin()
```



页面布局

ggplot2本身做页面布局并不方便，有许多对接的工具包为其做了拓展功能。

使用gridExtra工具包，将ggplot2图形对象以变量方式放入grid.arrange函数中，类似创建矩阵的方式控制页面布局

```
library(gridExtra)
p1 <- ggplot(mpg,aes(hwy))+geom_freqpoly(binwidth=1.2)
p2 <- ggplot(mpg,aes(displ,hwy))+geom_point()+geom_smooth(span=0.2)
grid.arrange(p1,p2,ncol=2)
```

衍生GUI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
```

导出和保存图形

ggsave函数能够根据文件后缀名生成相应图形文件保存ggplot图形对象

```
p1 <- ggplot(mpg,aes(hwy))+geom_freqpoly(binwidth=1.2)
ggsave("p1.eps",p1,dpi = 300,width = 5,height = 5)
```

图形变量则可以通过saveRDS以数据文件形式存放，对应的读取函数为readRDS

```
saveRDS(p1,"p1.rds")
p <- readRDS("p1.rds")
```

练习

- ggplot2中的economics数据集，绘制unemploy变量的时间序列趋势图，使用geom_line对象生成连续曲线。
- 对persons数据集进行分析，分析班级、Math、English、Computer等变量
- 对mtcars进行绘图，将尽可能多的变量以合理方式展示
- 其他练习数据：
  - Titanic数据集
  - ggplot2中的diamonds数据集