对象的5种基本类型

（1）数值（numeric:real numbers)

创建变量X，并将其赋值为1，赋值符号为左箭头加一个横杠

```
x<-1
x #查看变量x的值
[1] 1  #查看变量x的值返回的结果 [1]代表它后面接着的元素是x种的一个元素，没有方括号的1代表x中
class(x)  #用class()函数查看变量x的类型,complex代表复数，character代表字符，integer代表整

#区分大小写
x<-5
X<-5

```

属性(attribute)

名称(name)

维度(dismensions: matrix,array)

类型(class)

长度(length)

向量(vector)

向量是一种可以包含多个元素的数据结构

（1）使用vector()函数创建向量

```
vector(mode="logical",length= OL)
#它包含两个参数：第一个参数是这个向量里参数的类型，第二个参数是这个向量包含的元素个数
x<-vector("character",length=10)
#变量x是字符型，有10个元素，每个元素是空的
```

（2）直接在R中创建向量

```
x1<-1:3
# : 的意思是“到”，变量x1的内容是1，2，3
```

（3）用c()函数创建向量

```
x2<-c(1,2,3)
#如果往c()函数中传入的元素类型不同的话，R会强制转换为同一类型的元素
x3<-c(TURE,10,"a")#把三个不同类型的元素都转化为字符型元素
x4<-c("a","b","c")
#向量x4是一个字符类型向量，若想把它强制转换为数值型向量，需要用到as.numeric()函数
as.numeric(x4)
[1] NA NA NA  #虽然强制把字符向量转化为数值型向量，但是R不知道如何把a,b,c转化为数字，因此用
```

一、矩阵（matrix)

我们可以把矩阵看作：向量+维度属性，这个维度是一个整数向量，只包含两个元素行数和列数，分别用nrow和ncol表示

1.用matrix()函数创建矩阵

```
x<-matrix（nrow=3,ncol=2)
x<-matrix(1:6,nrow=3,ncol=2)#产生了一个3行2列，内容为1到6，且按列排列的矩阵
dim(x) #用矩阵维度的属性dim()查看矩阵的行列数
#返回 [1] 3 2
attributes(x) #查看矩阵x有多少属性，有哪些属性
```

2.用向量+维度属性创建矩阵

```
#创建一个向量
y<-1:6
#给向量添加维度信息
dim(y)<-c(2,3) #创建出一个2行3列的矩阵y
```

3.拼接矩阵

```
#创建一个矩阵
y2<-matrix(1:6,nrow=2,ncol=3)

#用rbind()函数按行拼接矩阵
rbind(y,y2)
#由于矩阵y和y2都是2行3列的矩阵，那么按行拼接就是4行3列的矩阵

#用cbind()函数按列拼接矩阵
cbind(y,y2)
#由于矩阵y和y2都是2行3列的矩阵，那么按列拼接就是2行6列的矩阵
```

二、数组（array)

数组与矩阵类似，但是数组的维度可以大于2

1、用array()函数创建2维数组

```
> x<-array(1:24,dim=c(4,6))
> x
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    5    9   13   17   21
[2,]    2    6   10   14   18   22
[3,]    3    7   11   15   19   23
[4,]    4    8   12   16   20   24
```

2、用array()函数创建3维数组

```
> x1<-array(1:24,dim=c(2,3,4))
> x1
, , 1

     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6

, , 2

     [,1] [,2] [,3]
[1,]    7    9   11
[2,]    8   10   12

, , 3

     [,1] [,2] [,3]
[1,]   13   15   17
[2,]   14   16   18

, , 4

     [,1] [,2] [,3]
[1,]   19   21   23
[2,]   20   22   24
```

1：24的数字的排序是：先排完第3个维度的第1个元素的2行3列的矩阵，再排第3个维度的第2个元素的2行

一、创建列表

向量、矩阵和数组，它们的共性是可以包含多个元素，并且元素和元素之间的类型必须是相同的，列表和它们最大的区别是：它可以包含不同类型的对象

1.用list()函数创建列表

```
> m<-list("a","1","2L",1+2i,FALSE)
> m
[[1]]
[1] "a"

[[2]]
[1] "1"

[[3]]
[1] "2L"

[[4]]
[1] 1+2i

[[5]]
[1] FALSE

#该列表共包含5个元素，它们分别是字符型、数值型、整数型、复数型和逻辑型这5种类型的对象
```

2.用list()函数给列表里的元素命名

```
> a<-list(a=1,b=2,c=3)
> a
$a
[1] 1

$b
[1] 2

$c
[1] 3
```

```
11
12 #该列表包含3个元素a,b,c,它们内容分别是1，2，3
```

3.用list()函数创建列表中的每一个元素包含的元素个数大于1的列表

```
> c<-list(c(1,2,3),c(4,5,6,7,8))
> c
[[1]]
[1] 1 2 3

[[2]]
[1] 4 5 6 7 8
```

4.引入矩阵的维度

（1）先用matrix()函数创建一个矩阵

```
> x<-matrix(1:6,nrow=2,ncol=3)  #1：6先按列排列再按照行
> x
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

（2）再给矩阵的行列命名

```
> dimnames(x)<-list(c("a","b"),c("c","d","e"))
> x
  c d e
a 1 3 5
b 2 4 6
```

二、构建列表的子集

1、先用list()函数构建一个列表

```
> x<-list(id=1:3,height=180,gender="male")
> x
$id
[1] 1 2 3

$height
[1] 180

$gender
[1] "male"
```

2、查看列表第一个元素的名称及内容

```
#两种方法
> x[1]
$id
[1] 1 2 3

> x["id"]
$id
[1] 1 2 3
```

3、只查看列表第一个元素的内容

```
#三种方法
> x[[1]]
[1] 1 2 3
> x[["id"]]
[1] 1 2 3
> x$id
[1] 1 2 3
```

4、查看列表中多个元素的名称及内容

```
> x[c(1,3)]
```

```
$id
[1] 1 2 3
$gender
[1] "male"

> x[c(1,2)]
$id
[1] 1 2 3
$height
[1] 180

> x[c(1,2,3)]
$id
[1] 1 2 3
$height
[1] 180
$gender
[1] "male"
```

一个很容易犯错的小知识点

```
> y<-"id"
> x[["id"]]
[1] 1 2 3
> x[[y]]
[1] 1 2 3
> x$id
[1] 1 2 3
> x$y
NULL
#使用嵌套的方括号[[]]能够引用包含了名称的变量y
#x$y返回的是空值，所以我们使用$符号只能直接引用名称“id"，不能引用包含了名称的变量y
```

5、从列表中获取嵌套的元素

(1)先创建一个列表

```
> x<-list(a=list(1,2,3),b=c("Monday","Tuesday"))
> x
```

```
$a
$a[[1]]
[1] 1

$a[[2]]
[1] 2

$a[[3]]
[1] 3


$b
[1] "Monday"  "Tuesday"
```

（2）提取列表中的列表的元素的内容

使用两个嵌套的方括号[[]]提取

```
> x[[1]][[1]]
[1] 1
> x[[1]][[2]]
[1] 2
> x[[1]][[3]]
[1] 3
```

```
> x[[c(1,1)]]
[1] 1
> x[[c(1,2)]]
[1] 2
> x[[c(1,3)]]
[1] 3
> x[[c(2,1)]]
[1] "Monday"
> x[[c(2,2)]]
[1] "Tuesday"

#c()函数中的第一个参数2表示列表x中的第二个元素，第二个参数1表示第二个元素中的第1个元素
```

采用

一、判断缺失值

缺失值有两种表示方式：NA和NaN，两者之间的关系是：NaN属于NA，NA不属于NaN。原因是：NaN一般表示数值类型的缺失值，NA表示的数据类型可以多种，比如：数值的缺失值、字符的缺失值等。

1.先用c()函数创建一个向量

```
> x<-c(1,NA,NA,2,3)
> x
[1]  1 NA NA  2  3
```

2.用is.na()和is.nan()函数判断缺失值

```
> is.na(x)
[1] FALSE  TRUE  TRUE FALSE FALSE
> is.nan(x)
[1] FALSE FALSE FALSE FALSE FALSE
```

从结果可以看出：is.na()函数判断出了每个元素是否为缺失值，缺失值返回了真，数值返回了假；is.nan()函数没有判断出缺失值，说明在向量x中不存在NaN这种类型的缺失值，同时也印证了NA不属于NaN。若我们把向量x里的NA改成NaN，结果如下：

```
> x<-c(1,NaN,NaN,2,3)
> x
[1]   1 NaN NaN   2   3
> is.na(x)
[1] FALSE  TRUE  TRUE FALSE FALSE
> is.nan(x)
[1] FALSE  TRUE  TRUE FALSE FALSE
```

二、处理缺失值

```
#1.先用c()函数创建一个包含缺失值的向量
> x<-c(1,NA,NA,2,3)
> x
[1]  1 NA NA  2  3
#2.取向量中不是缺失值的元素，！的意思是取反
> x[!is.na(x)]
[1] 1 2 3

#3.用complete.case()函数取多个向量对应位置都不是缺失值的元素
> x<-c(1,NA,NA,2,3)
> y<-c(NA,"a","b","c","d")
> z<-c(1L,2L,3L,NA,4L)
> w<-complete.cases(x,y,z)
> w
[1] FALSE FALSE FALSE FALSE  TRUE
#返回结果是逻辑向量，其中只有x和y对应位置都不是缺失值的才会返回TRUE，否则返回FALSE。我们用x[
```

```
> x[w]
[1] 3
> y[w]
[1] "d"
> z[w]
[1] 4
```

三、实例

1.加载数据集所在的包

```
> library(datasets)
```

2.用head()函数查看数据集的前6行

```
head(airquality)
  Ozone Solar.R Wind Temp Month Day
1    41     190  7.4   67     5   1
```

```
2    36     118  8.0   72     5   2
3    12     149 12.6   74     5   3
4    18     313 11.5   62     5   4
5    NA      NA 14.3   56     5   5
6    28      NA 14.9   66     5   6
```

3.用complete.cases()函数选择在变量

```
g<-complete.cases(airquality)
> g
  [1]  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE
 [11] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [21]  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE
 [31]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE
 [41]  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
 [51]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [61] FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
 [71]  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
 [81]  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [91]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE
[101]  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
[111]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE
[121]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[131]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[141]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
[151]  TRUE  TRUE  TRUE

```

4.选择该数据集不包含缺失值记录的前10行，且每个变量都要

```
> airquality[g,][1:10,]
   Ozone Solar.R Wind Temp Month Day
1     41     190  7.4   67     5   1
2     36     118  8.0   72     5   2
3     12     149 12.6   74     5   3
4     18     313 11.5   62     5   4
7     23     299  8.6   65     5   7
8     19      99 13.8   59     5   8
```

```
9      8      19 20.1   61     5   9
12    16     256  9.7   69     5  12
13    11     290  9.2   66     5  13
14    14     274 10.9   68     5  14
```

四、识别缺失值的模式

```
install.packages("mice")
library(mice)
```

```
#用md.pattern()函数查看数据缺失值的分布
md.pattern(data)
```

五、处理缺失值

1.行删除法

用na.omit()函数删除不完整观测

```
library(datasets)
> data<-head(airquality)
> data
  Ozone Solar.R Wind Temp Month Day
1    41     190  7.4   67     5   1
2    36     118  8.0   72     5   2
3    12     149 12.6   74     5   3
4    18     313 11.5   62     5   4
5    NA      NA 14.3   56     5   5
6    28      NA 14.9   66     5   6
> newdata<-na.omit(data)  #用na.omit()函数删除不完整观测
> newdata
  Ozone Solar.R Wind Temp Month Day
1    41     190  7.4   67     5   1
2    36     118  8.0   72     5   2
3    12     149 12.6   74     5   3
4    18     313 11.5   62     5   4
```

2.多重插补法

多重插补法（MI）是一种基于重复模拟的处理缺失值的方法，多重插补是从一个包含缺失值的数据集中生成一组完整的数据集。每个模拟数据集中，缺失数据将使用蒙特卡洛方法来填补。

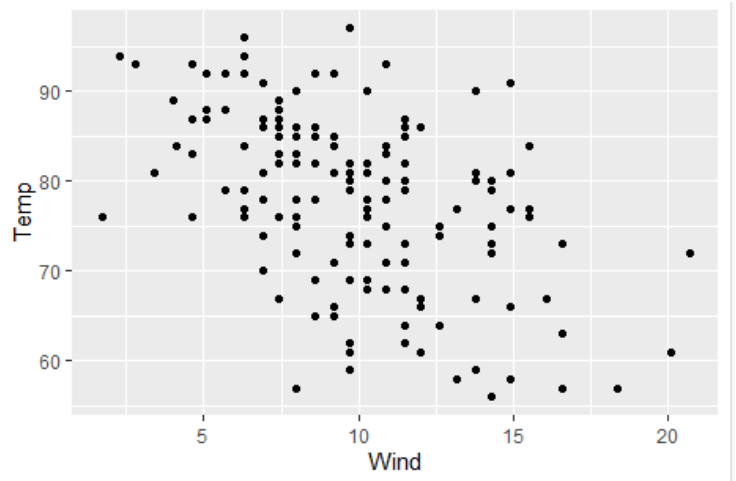
```
library(datasets)
data<-head(airquality)
data

data1<-mice(data,m=6)
#用mice()函数从包含缺失数据的数据框开始，进行6重插补，即生成6个完整数据集
#mice()函数的第一个参数data为数据，第二个参数m为要返回的完整数据集的个数

fit<-with(data1,lm(Solar.R~Wind+Temp+Month+Day+Ozone))
fit
#用with()函数依次对6个完整数据集分别应用lm()模型
#结果分别返回6个完整数据集的回归结果

jieguo<-pool(fit)
jieguo
summary(jieguo)
#用pool()函数汇总回归结果

result<-complete(data1,action=3)
result
#选择第三个插补数据集作为结果
```

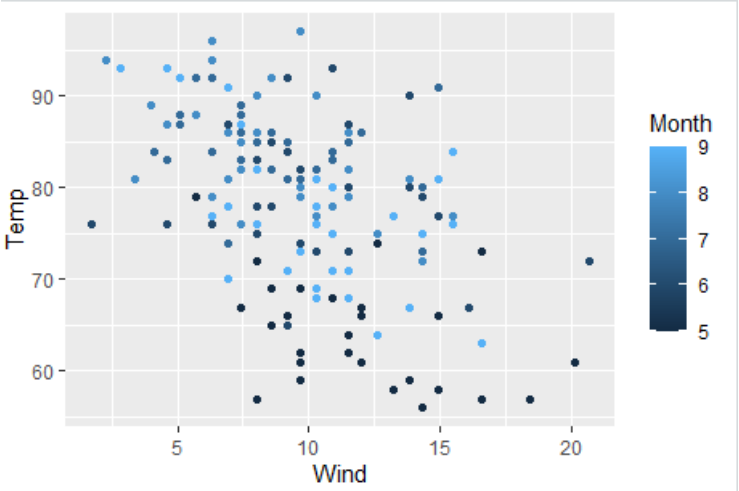
```
a<-read.csv()
#更改工作空间目录，直接用文件名读入数据
#（1）用read.csv()读入数据
#（2）用setwd()函数重新设定工作空间
> setwd("D:\\")
# (3) 用getwd()函数查看目前工作空间所在位置
> getwd()
# (4) 再次用read.csv()函数读入数据
> a<-read.csv()



#安装readxl程序包
install.packages("readxl")
library(readxl)
#用read_excel()函数读入数据
data=read_excel()

#表格状数据的读入，用read.table()函数
zhushi<-read.table("竹石.txt")
zhushi<-read.table("竹石.txt",header=T) #数据中第一行的数据作为列数据的变量名

#读取没有任何规则的数据需要readlines()函数
shuju1=readlines("数据小说数据.txt")
```

层（layer)

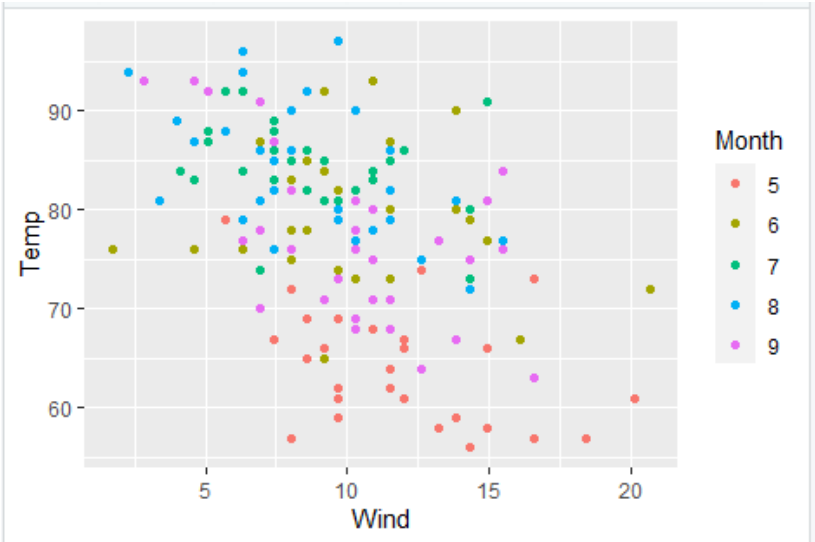
Data	感兴趣的变量（data frame）
Aesthetics(美学层)	x-axis(x轴）/y-axis(y轴）/color(颜色）/fill(填充的颜色）/size(大小）/labels(标签）/alpha(透明度）/shape(形状）/linearwidth(线宽）/lineartype(线的类型)
Geometries(几何客体层)	point(散点图）/line(线图）/histogram(柱状图）/bar(条形图）/boxplot(箱图)
Facets（划分绘图面板）	columns(行）,rows(行)
Statistics（统计层）（目的：添加统计信息）	binning/smoothing/descriptive/inferential
coordinates(坐标系)	cartesian/fixed/polar/limits
Themes(主题)	non-data ink(和数据无关的风格设计）

```
install.packages("ggplot2")
library(ggplot2)
#绘制风速~温度的散点图
qplot(Wind,Temp,data=airquality)
#按月份显示不同的颜色
qplot(Wind,Temp,data=airquality,color=Month)
```

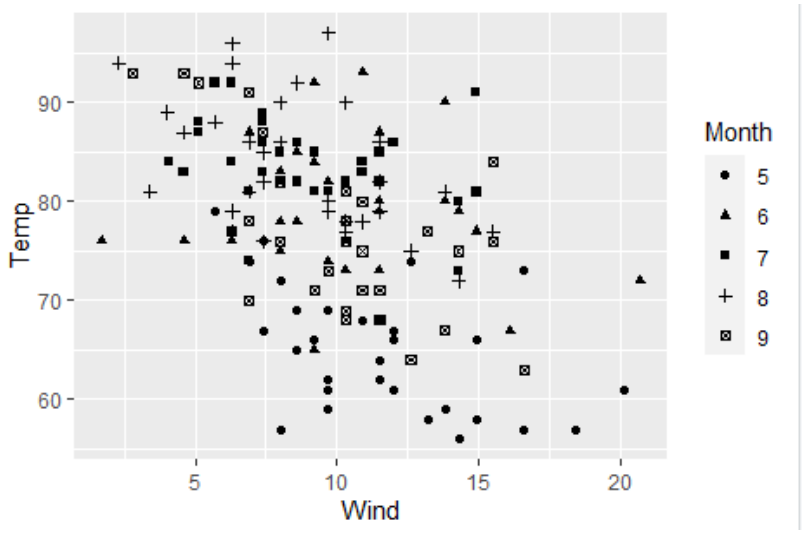




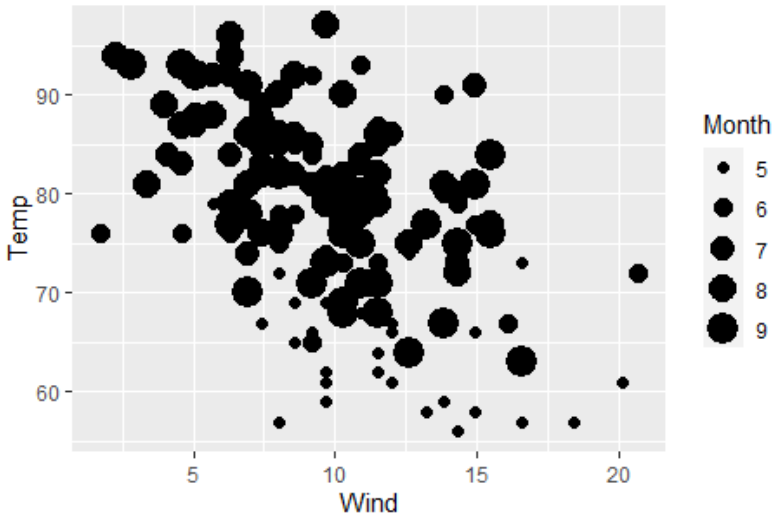

```
#由于元数据集中Month变量是一个连续变量，因此颜色为渐变色。如果想要不同月份显示不同颜色（非连续的渐变色），需要
airquality$Month<-factor(airquality$Month)
qplot(Wind,Temp,data=airquality,color=Month)
```



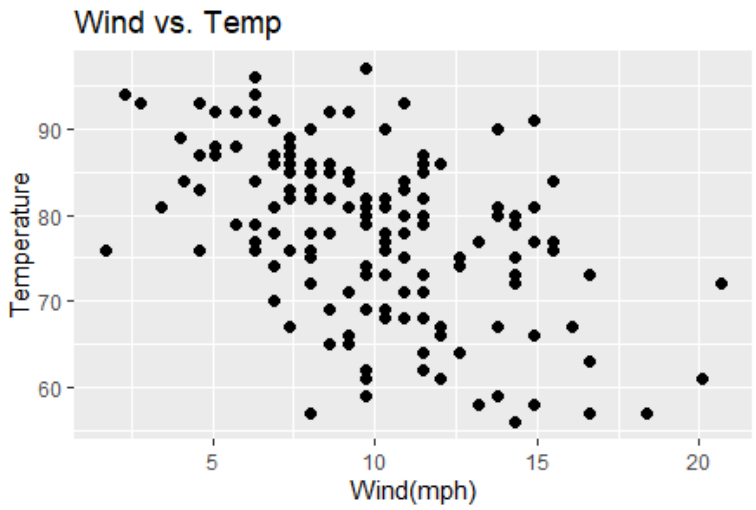

```
#不同月份显示不同形状
qplot(Wind,Temp,data=airquality,shape=Month)
```



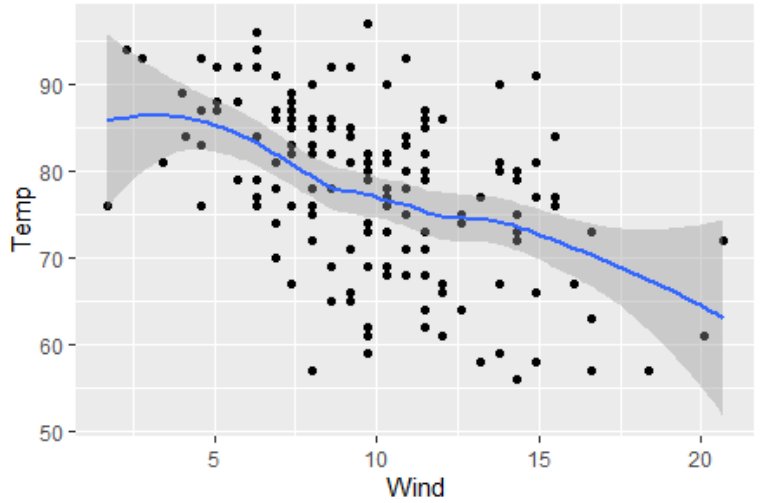

```
#不同月份显示不同的大小
qplot(Wind,Temp,data=airquality,size=Month)
```



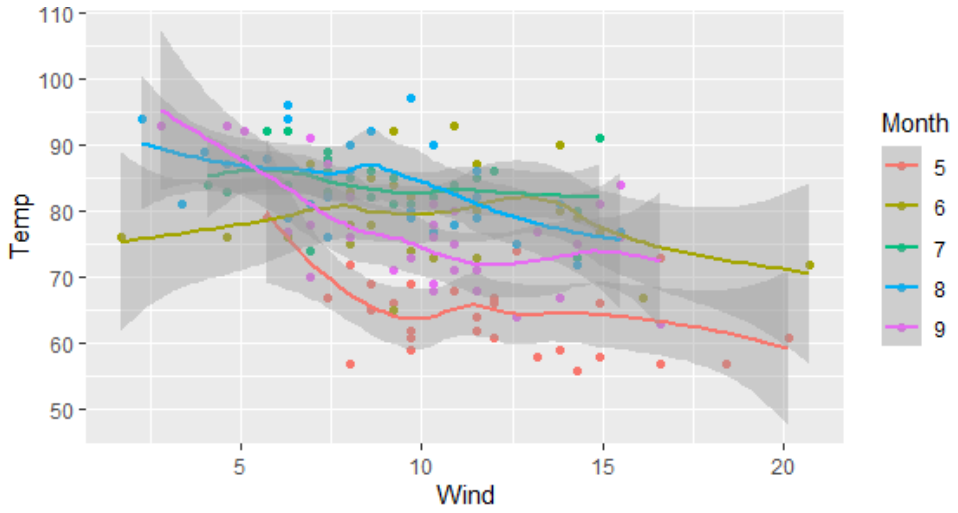

```
#设置大小为2，x,y轴标签分别为“Wind(mph)","emperature",标题为”Wind vs. Temp"
qplot(Wind,Temp,data=airquality,size=I(2),main="Wind vs. Temp",xlab="Wind(mph)",ylab="Temperature")
```



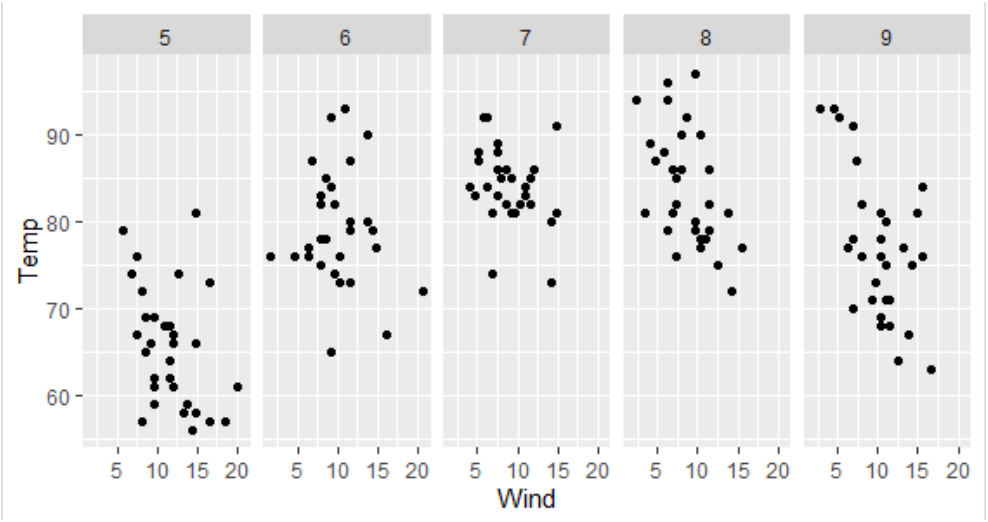
```r
#生成带平滑回归线的散点图
qplot(Wind,Temp,data=airquality,geom = c("point","smooth"))
#代码中的smooth会给一条根据point(点）拟合出来的回归线，它可以算作统计信息，
#可以看出，图中多了一条按照默认方法拟合出的平滑的蓝色曲线(回归线）和灰色的条块（置信区间）
```



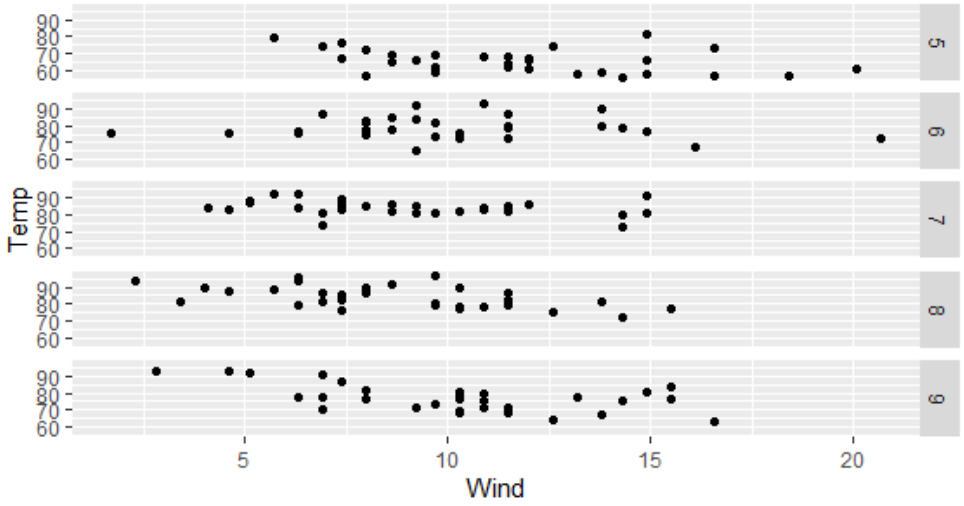

```r
#按月份生成不同颜色的平滑回归线
qplot(Wind,Temp,data=airquality,geom = c("point","smooth"),color=Month)
```




```r
#按月份生成从左至右的多个散点图
qplot(Wind,Temp,data=airquality,facets = .~Month)
```

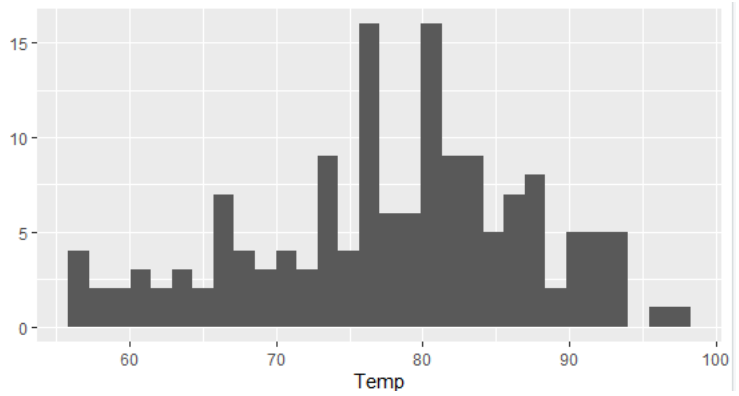


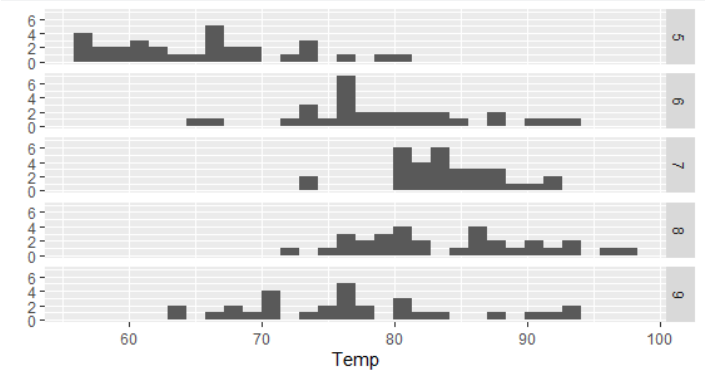

```r
#按月份生成从上至下的多个散点图
qplot(Wind,Temp,data=airquality,facets = Month~.)
```

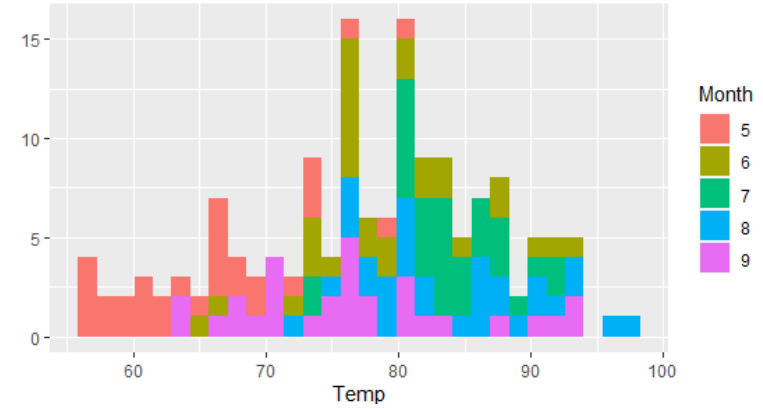


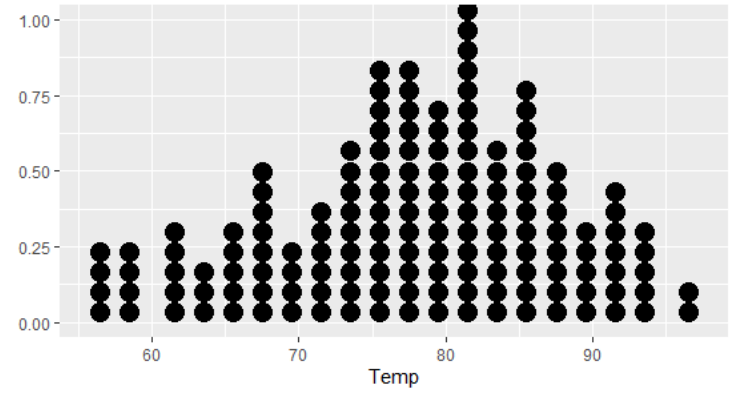

```r
#可以看出，在定义参数facets时，若为~.变量名，则按变量不同从左到右生成不同的图
#若为变量名~.,则从上至下生成不同的图
```

```r
#若在qplot函数中只输入了一个数据参数，则默认生成柱状图，如：
qplot(Temp,data=airquality)
qplot(Temp,data=airquality,facets = Month~.)

#在qplot函数中定义参数fill,可以得到一个累加的柱状图，如：
qplot(Temp,data=airquality,fill=Month)

#生成温度的密度点图
qplot(Temp,data=airquality,geom="dotplot")

```

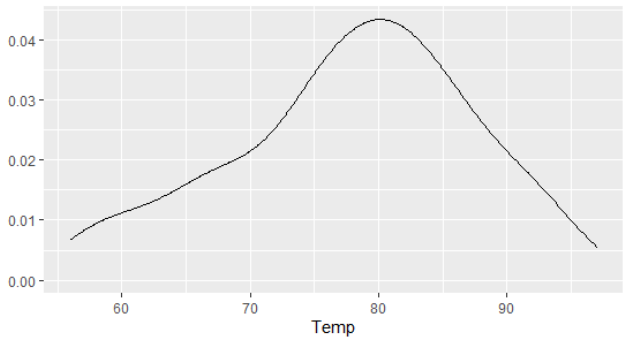


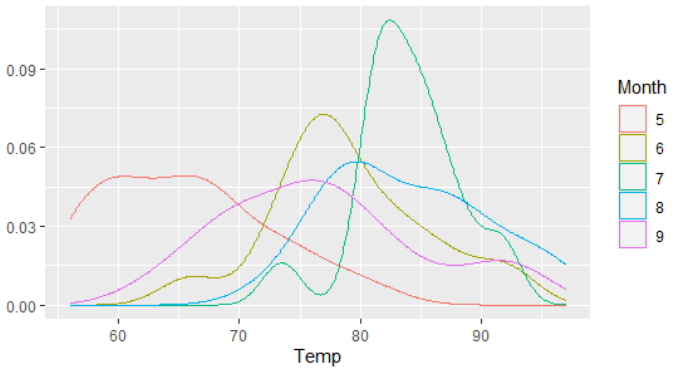







```
#生成温度的密度曲线
qplot(Temp,data=airquality,geom = "density")
#按月份生成不同颜色的温度的密度曲线
qplot(Temp,data=airquality,geom = "density",color=Month)
```


0.04
0.03
0.02
0.01
0.00
60
70
80
90
Temp


0.09
0.06
0.03
0.00
60
70
80
90
Temp
Month
5
6
7
8
9

Lattice绘图系统的绘图函数

lattice包

xyplot/bwplot/histogram/stripplot/dotplot/splom/leveplot/contourplot

- xyplot函数的格式：xyplot(y~x|f*g,data),其中，y代表因变量，x代表自变量，f*g指分类变量，data指数据集，第一个参数指一个公式，其 | 的左半部分是必须存在的，右半部分不是必须存在的，若只存在左半部分，说明我们不考虑交互作用，只考虑y和x这两个变量之间的关系；若左半部分和右半部分都有，则考虑交互作用，这样就提供给我们看了x和y这两者的关系在分类变量的不同水平下进行的变化。
- panel函数，用于控制每个面板的绘图

grid包

- 实现了独立于base的绘图系统
- lattice包是基于grid创建的，很少直接从grid包调用函数

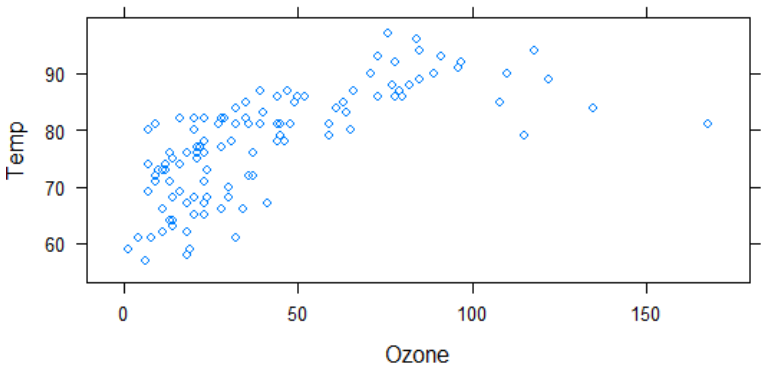
Lattice与Base的重要区别

Base绘图函数直接在图形设备上绘图

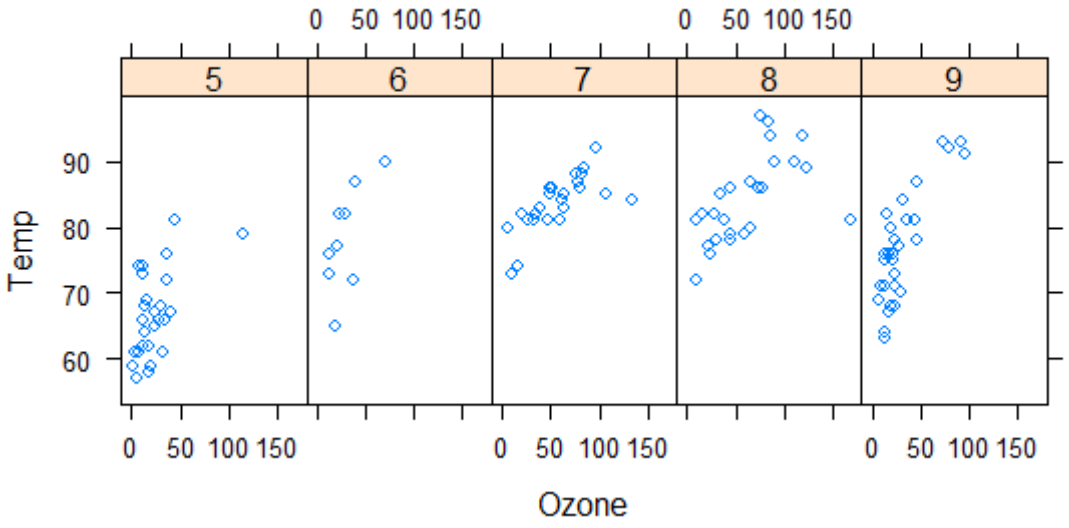
Lattice绘图函数返回trellis类对象

- 打印函数真正执行了在设备上绘图
- 命令执行时，trellis类对象会被自动打印，所以看起来就像是lattice函数直接完成了绘图

```
install.packages("lattice")
library(lattice)
#用xyplot()函数查看温度和臭氧含量的关系
xyplot(Temp~Ozone,data=airquality)
```



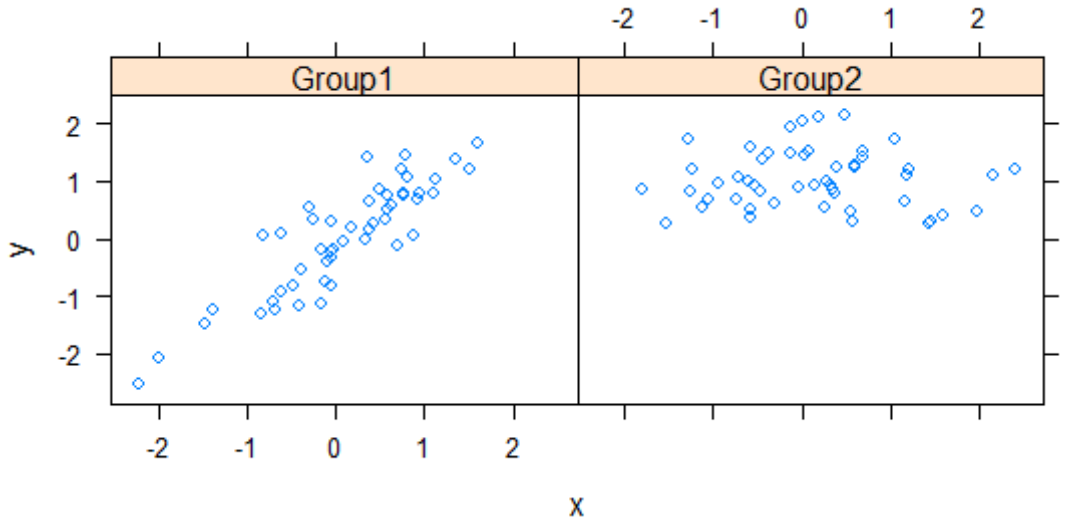
```r
#将月份转化为分类变量
airquality$Month<-factor(airquality$Month)
airquality$Month
#查看不同月份下温度和臭氧含量的关系
xyplot(Temp~Ozone|Month,data=airquality,layout=c(5,1))
#结果显示：将月份转化为分类变量时有5个水平，分别为5、6、7、8、9月份，所以，我们在查看不同月份
#参数设置成了一行5列，其中，第一列指5月份温度和臭氧含量之间的散点图，第2列指6月份温度和臭氧含
#从总体上来看，温度和臭氧含量之间的关系随着月份的不同时有所变化的，这也意味着温度和臭氧含量之
#这也展示了lattice系统适合呈现交互作用的优势
```



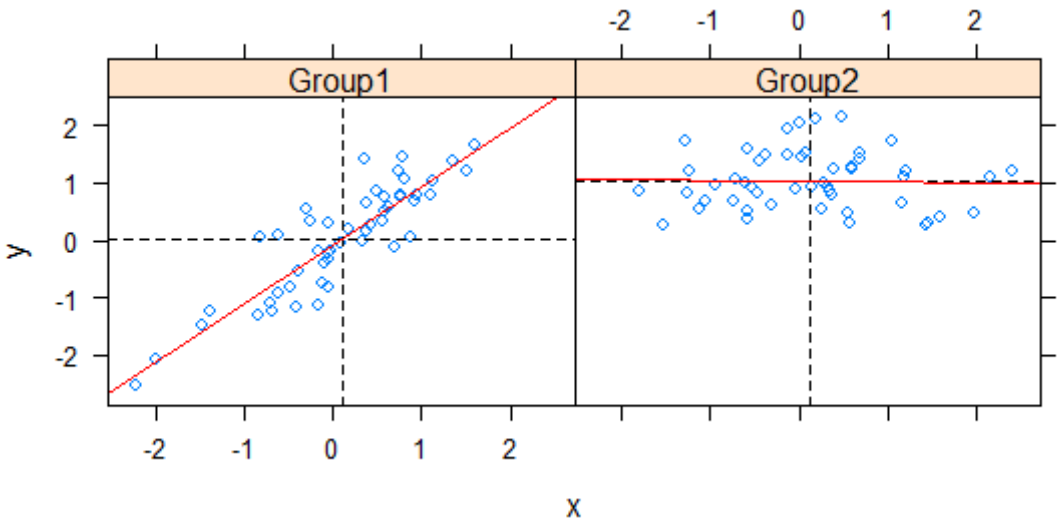
Panel函数应用

```r
#1.用set.seed()函数设置种子数，其里面的数字可以输任何数，这里我们输入1
set.seed()
#设置种子点的意义是让我们每次产生的随机数是一样的

#2.在标准正态分布中产生100个随机数
x<-rnorm(100)
#3.重新创建一个变量f,该变量中只包含0和1这两个数，且每个数出现的次数均为50次
f<-rep(0:1,each=50)
#4.用x和f进行计算，将结果赋值给y
y<-x+f-f*x+rnorm(100,sd=0.5)
#目的是让x和y之间的关系与f有交互作用，我们为了让画出的图的点不在一条直线上，则加入一个随机数
#中产生100个随机数，设置正态分布的均值为0，标准差为0.5，否则正态分布的默认标准差为1


#5.把f变量转化为分类变量
```

```
f<-factor(f,labels = c("Group1","Group2"))#由于我们不知道变量f中的0和1代表什么，所以，我
#6.查看在f的不同水平下x和y之间的关系
xyplot(y~x|f,layout=c(2,1))
```




```
#7.自己设置面板风格
xyplot(y~x|f,panel = function(x,y){panel.xyplot(x,y)
    +panel.abline(v=mean(x),h=mean(y),lty=2)
    +panel.lmline(x,y,col="red")})

#上述代码中，第二个参数指自己定义的函数，函数体用花括号括起来，panel.abline()函数中的第一个
#panel.abline()函数中的第二个参数指在y的均值处画一条水平的直线，第三个参数的类型指线的类型是
#panel.lmline()函数的目的是在每个面板中添加一条水平的虚线和一条垂直的虚线，它们分别对应x的均
#这条红色的线是对数据进行拟合得到的回归线，在Group2水平下的结果和Group1类似
```



1.任意步长的等差数列构成的向量，用函数seq()

2.使用重复函数 rep()

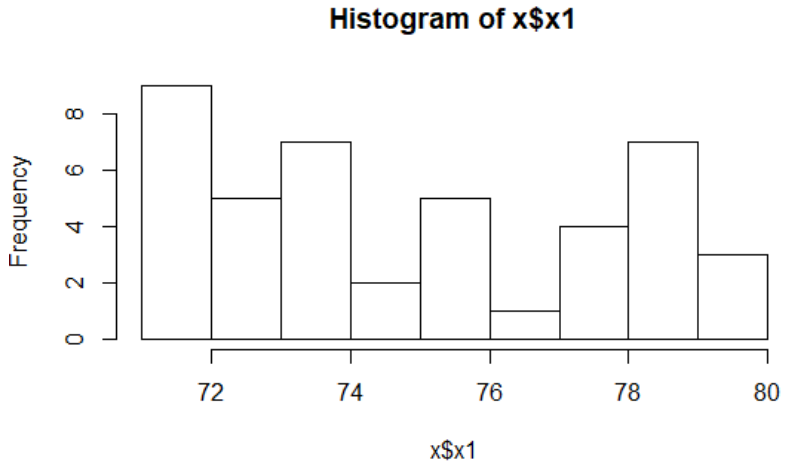
```
rep(x,time=,each=,length.out=)
#x:数量、向量或是数据对象，是要复制的对象
#times:表示x被复制的次数
#each:表示每个元素重复的次数
#length.out:表示截取前多少个元素
```

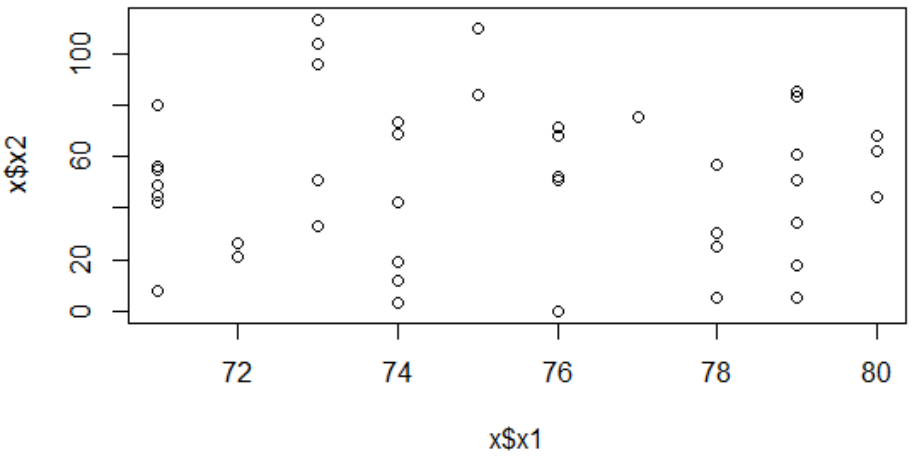
```
> #求余
> 9%%4
[1] 1
> #求商
> 9%/%4
[1] 2
> #乘方
> 2^3
[1] 8
> #开方
> sqrt(9)
[1] 3
```

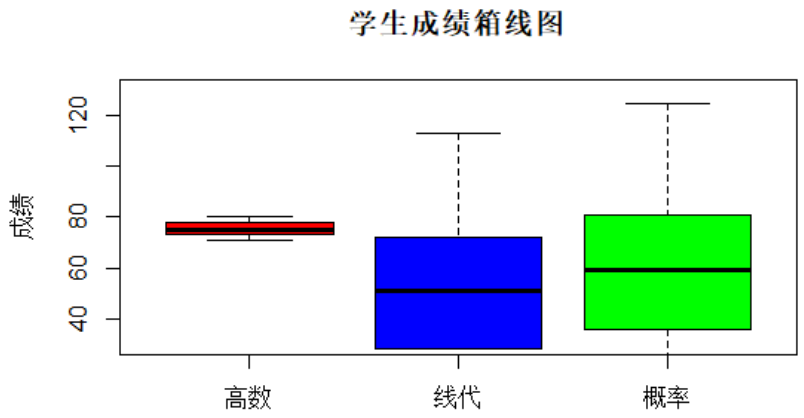
例1：用一个班43位同学高数、线代、概率论与数理统计的成绩为例，在R语言中演示基本的统计分析

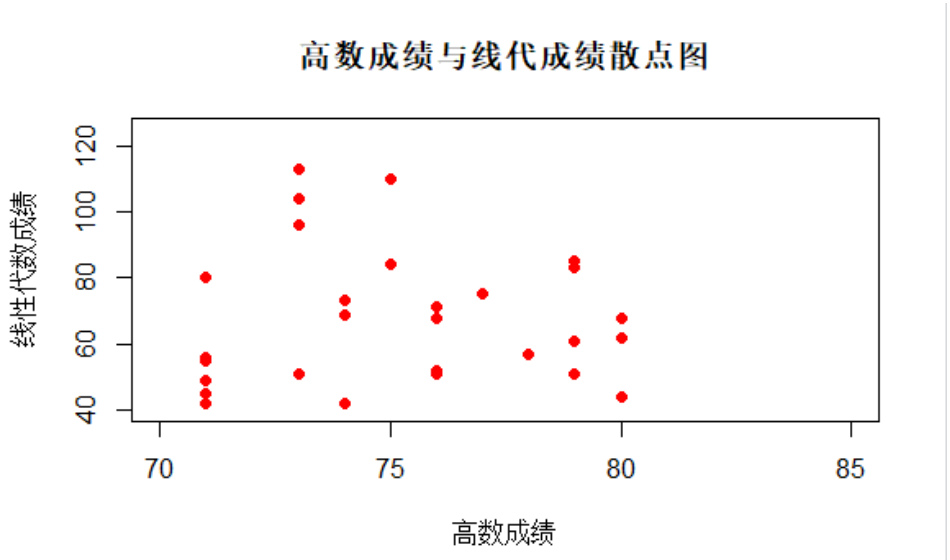
```
#1.生成43个学号
num<-seq(1010913418,1010913460)
num

#2.依次生成高数成绩x1,线代成绩x2,概率论与数理统计成绩x3为:
x1<-round(runif(43,min=70,max=80))
x2<-round(rnorm(43,mean=60,sd=30))
x3<-round(rnorm(43,mean=55,sd=29))
#round(x)表示对x取整数,runif(n,min=,max=)表示生成n个最小值为，最大值为 的服从均匀分布的数
#x2=round(rnorm(43,mean=60,sd=30))#表示生成43个均值为60，方差为30的服从正态分布的数
```

```r

#3.将学号连同三科成绩合并成数据框x中
x<-data.frame(num,x1,x2,x3)
x

#4.计算各科的均值
colMeans(x[,2:4])
apply(x,2,mean)#matrix 1 indicates rows, 2 indicates columns, c(1, 2) indicates rows
apply(x[,2:4],2,mean)

#5.计算各科的最大、最小值
apply(x[,2:4],2,max)
apply(x[,2:4],2,min)

#6.求出每人的最高分
apply(x[,2:4],1,sum)

#7.求出最高分的同学程序为:
which.max(apply(x[,2:4],1,sum)) #which.max表示找到最高分同学的位置
x$num[which.max(apply(x[,2:4],1,sum))] #x$num表示找到最高分同学所对应的学号

#8.直方图
hist(x$x1)
#散点图
plot(x$x1,x$x2)
boxplot(x[2:4],main="学生成绩箱线图",ylab="成绩",names=c("高数","线代","概率"),col=c("re
plot(x$x1,x$x2,main="高数成绩与线代成绩散点图",xlab="高数成绩",ylab="线性代数成绩",xlim=c
```


Histogram of x$x1
Frequency
0
2
4
6
8
72
74
76
78
80
x$x1


x$x2
0
20
60
100
72
74
76
78
80
x$x1






```
#常见的绘图参数解释：
#font=字体;lty=线类型；lwd=线宽度；pch=点的类型；xlab=横坐标；ylab=纵坐标；xlim=横坐标范围

#coplot(x~y|z) 在z的每个值或每个区间上做x与y的散点图
#pairs(x) 当x为矩阵，做x各列之间的散点图
#hist(x,freq)直方图，参数freq默认为TRUE，根据频数作图；若为FALSE，则根据构成比（总和为1）作
#barplot(table(x,y),beside=FALSE) 对定性变量x,y做条图，默认堆积条图，使用beside=T则为并列
#qqnorm(x) QQ图，(正态分位数-分位数图）
#pie(table(x))#对定性变量x做饼图
#lines(x),lines(x,y) 添加折线原图上修改
#abline(lm(y~x)) 添加y对x的回归直线
#abline(a,b) a为截距，b为斜率
#abline(v=) 添加垂直线
#abline(h=) 添加水平线
```

```
#使用sort(colors()可以查看所有已知的颜色
```

假设检验（hypothesis test) 亦称显著性检验（significant test),是统计推断的另一个重要内容，其目的是比较总体参数之间有无差别。假设检验的实质是判断观察到的“差别”是由抽样误差引起还是总体上的不同，目的是评价两种不同处理引起效应的不同的证据有多强，这种证据的强度用概率P来度量和表示。除t分布外，针对不同的资料还有其他各种检验统计量及分布，如F分布、$\chi 2$分布等，应用这些分布对不同类型的数据进行假设检验的步骤相同，其差别仅仅是需要计算的检验统计量不同。

其基本思想是小概率反证法思想

1.W检验(Shapiro-Wilk)

检验数据是否符合正态分布

R函数：shapiro.test()

结果含义：当p值小于某个显著性水平α(比如0.05）时，则认为样本不是来自正态分布的总体，否则承认样本来自正态分布的总体

2.K检验(经验分布的Kolmogorov-Smirnov检验)

R函数：ks.test(),如果P值很小，说明拒绝原假设表明数据不符合F(n,m)分布。

3.相关性检验：

R函数：cor.test()

```
cor.test(x,y,alternative=c("two.sided","less","greater"),method=c("pearson","kendall"
```

结果含义：如果p值很小，则拒绝原假设，认为x,y是相关的。否则认为是不相关的。

4.T检验

用于正态总体均值假设检验，单样本、双样本都行

```
t.test()
t.test(x,y=NULL,alternative=c("two.sided","less","greater"),mu=0,paired=FALSE,var.equ
```

结果含义：P值小于显著性水平时拒绝原假设，否则，接受原假设。具体的假设要看所选择的是双边假设还是单边假设（又分小于和大于）

5.Fisher精确的独立检验

原假设：X，Y相关

```
fisher.test(x,y=NULL,workspace=200000,hybrid=FALSE,control=list(),or=1,alternative="t
```

6.Pearson χ2拟合优度检验

R语言中调用chisq.test(X)

原假设H0：X符合F分布

p-值小于某个显著性水平，则表示拒绝原假设，否则接受原假设

正态分布参数检验

例1.某种原件的寿命X(以小时计）服从正态分布N（μ，σ ）。其测得16只原件的寿命如下：

159 280 101 212 224 379 179 264

222 362 168 250 149 260 485 170

问是否有理由认为元件的平均寿命大于255小时?

解：按题意，需检验H0：μ≤255, H1 > 255

此问题属于单边检验问题，可以使用R语言t.test(x,y=Null,#只提供x为单个正态总体均值检验，否则为两个总体均值检验

alternative=c("two.side","less","greater"),#双边检验 单边检验

mu=0,#原假设：mu=0,均值为某个具体数字

paired=FALSE,

var.equal=FALSE,# 方差齐性选项

conf.level=.95)#置信水平95%

```
> X<-c(159,280,101,212,224,379,179,264,222,362,168,250,149,260)
> t.test(X,alternative = "greater",mu=225)

    One Sample t-test

data:  X
t = 0.20141, df = 13, p-value = 0.4217
```

```
alternative hypothesis: true mean is greater than 225
95 percent confidence interval:
 192.1588      Inf
sample estimates:
mean of x
 229.2143
```

可见P值为0.257>0.05,不能拒绝原假设，接受H0，即平均寿命不大于225小时

例2.在平炉上进行的一项实验以确定改变操作方法的建议是否会增加刚的得率，实验时在同一个平炉上进行的，每炼一炉刚时除操作方法外，其他条件都尽可能做到相同，先用标准方法炼一炉，然后用新办法炼一炉，以后交替进行，各炼了10炉，其得率分别为标准方法：78.1 72.4 76.2 74.3 77.4 78.4 76.0 75.5 76.7 77.3

新方法：79.1 81 77.3 79.1 80 79.1 79.1 77.3 80.2 82.1

设这两个样本相互独立，且分别来自正态总体N（μ1，σ 1）和N（μ2，σ 2），其中μ1，μ2和σ 2未知。问新的操作能否提高得率？（取α=0.05）

解：因为数据是成对出现的，所以采用成对数据t检验比上述的双样本均值检验更准确。所谓成对t检验就是Zi=Xi-Yi,再对Z进行单样本均值检验

```
> X<-c(78.1,72.4,76.2,74.3,77.4,78.4,76.0,75.5,76.7,77.3)
> Y<-c(79.1,81.0,77.3,79.1,80.0,79.1,77.3,80.2,82.1)
> t.test(X-Y,alternative = "less")

#结果如下：
    One Sample t-test

data:  X - Y
t = -3.9114, df = 9, p-value = 0.001779
alternative hypothesis: true mean is less than 0
95 percent confidence interval:
      -Inf -1.700279
sample estimates:
mean of x
     -3.2
```

可见P值<0.05,接受备择假设，即新操作能够提高得率。并且P值更小可见比双样本均值检验更准确。

例3.对例2进行方差检验，方差是否相同

```
> var.test(X,Y,ratio = 1,alternative = c("two.sided","less","greater"),conf.level = 0

#结果如下：
    F test to compare two variances

data:  X and Y
F = 1.3365, num df = 9, denom df = 8, p-value = 0.6933
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.3067324 5.4822833
sample estimates:
ratio of variances
          1.336505
```

可见P值为0.559>0.05,接受原假设，认为两者方差相同

例4.皮尔森拟合优度检验

某消费者协会为了确定市场上消费者对5种品牌啤酒的喜好情况，随机抽取了1000名啤酒爱好者作为样品进行试验：每个人得到5种品牌的啤酒各一瓶，但未表明牌子。这5种啤酒分别按着A、B、C、D、E字母的5张纸片随即的顺序送给每一个人。下表是根据样本资料整理的各种品牌啤酒爱好者的频数分布。试根据这些数据判断消费者对这5种品牌啤酒的爱好有无明显差异?

最喜欢的牌子	A	B	C	D	E
人数X	210	312	170	85	223

解：如果消费者对5种品牌的啤酒无显著差异，那么，就可以认为喜好这5种品牌啤酒的人呈均匀分布，即5种品牌啤酒爱好者人数各占20%。据此假设

H0：喜好5种啤酒的人数分布均匀

可以使用Pearson χ2拟合优度检验，R语言中调用chisq.test(X)

```
> X<-c(210,312,170,85,223)
> chisq.test(X)

    Chi-squared test for given probabilities
```

```
5
6 data:  X
7 X-squared = 136.49, df = 4, p-value < 2.2e-16
```

例5.正态W检验

已知15名学生体重如下，问是否服从正态分布 75 64 47.4 66.9 62.2 62.2 58.7 63.5 66.6 64 57 69 50 72

```
1 > w<-c(75,64,47.4,66.9,62.2,62.2,58.7,63.5,66.6,64,57,69,50,72)
2 > shapiro.test(w)
3
4     Shapiro-Wilk normality test
5
6 data:  w
7 W = 0.95448, p-value = 0.6321
```

因为P值>0.05,接受原假设，所以认为来自正态分布总体

当我们在R中读入了一个txt\csv等格式的数据时，即自成了一个数据框

数据框就是一个表格，它的每一列可以存放不同类型的数据，因此，它最接近于实际数据的形态

一、创建数据框

```
> birthday<-c(1997,1998,1971,1999,2000,2001,2003)
> gender<-c("男","男","女","男","女","女","男")
> people<-data.frame(name,birthday,gender);people
    name birthday gender
1   张三     1997     男
2   李四     1998     男
3 王大妈     1971     女
4   小二     1999     男
5   小五     2000     女
6   小六     2001     女
7   小七     2003     男
```

二、筛选、引用

```
> people[3,1]
[1] 王大妈
Levels: 李四 王大妈 小二 小六 小七 小五 张三
> people[3,1]#查看第3行第1列的人物信息
[1] 王大妈
Levels: 李四 王大妈 小二 小六 小七 小五 张三
> people[3,] #查看第3行的人物信息
    name birthday gender
3 王大妈     1971     女
> people$birthday
[1] 1997 1998 1971 1999 2000 2001 2003
> people[,2]#查看第2列人物的出生年份
[1] 1997 1998 1971 1999 2000 2001 2003
> new_shuju<-people[people$birthday==2000,];new_shuju
  name birthday gender
5 小五     2000     女
> #选择出生年份为2000年的数据

> new_shuju<-people[people$gender=="男"&people$birthday>1998,]
```

```
> new_shuju
  name birthday gender
4 小二     1999      男
7 小七     2003      男
```

三、增列、合并

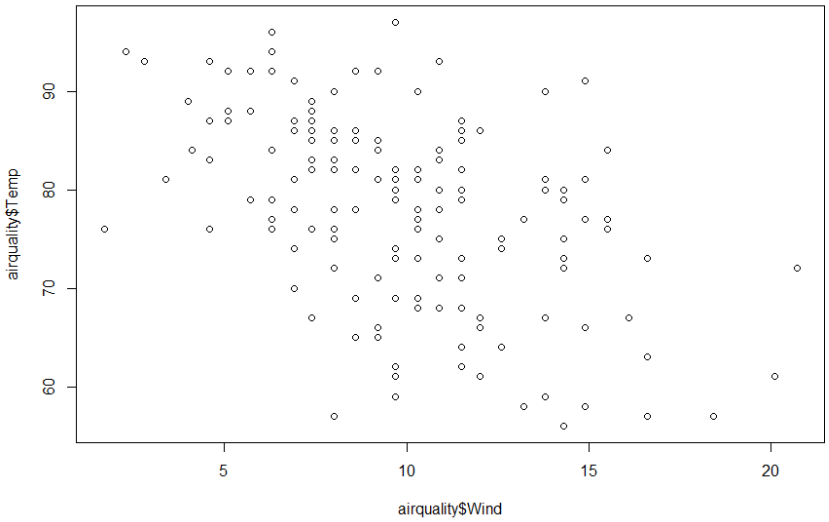
```
#1.增列
> nation<-c("汉族","汉族","回族","藏族","汉族","汉族","苗族")
> #给数据框添加一列数据：民族
> people$nat<-nation
> people$nation<-nation
> #给数据框的增列命名了2次，因此要删除其中的一列
> people[,-5]
    name birthday gender  nat
1   张三     1997      男 汉族
2   李四     1998      男 汉族
3 王大妈     1971      女 回族
4   小二     1999      男 藏族
5   小五     2000      女 汉族
6   小六     2001      女 汉族
7   小七     2003      男 苗族
```

```
#2.合并
> x1<-people[1:3,];x1
    name birthday gender  nat nation
1   张三     1997      男 汉族   汉族
2   李四     1998      男 汉族   汉族
3 王大妈     1971      女 回族   回族
> x2<-people[4:7,];x2
  name birthday gender  nat nation
4 小二     1999      男 藏族   藏族
5 小五     2000      女 汉族   汉族
6 小六     2001      女 汉族   汉族
7 小七     2003      男 苗族   苗族
> people.nat<-merge(x1,x2,by="nat");people.nat
   nat name.x birthday.x gender.x nation.x name.y birthday.y gender.y nation.y
1 汉族   张三       1997       男     汉族   小五       2000       女     汉族
```

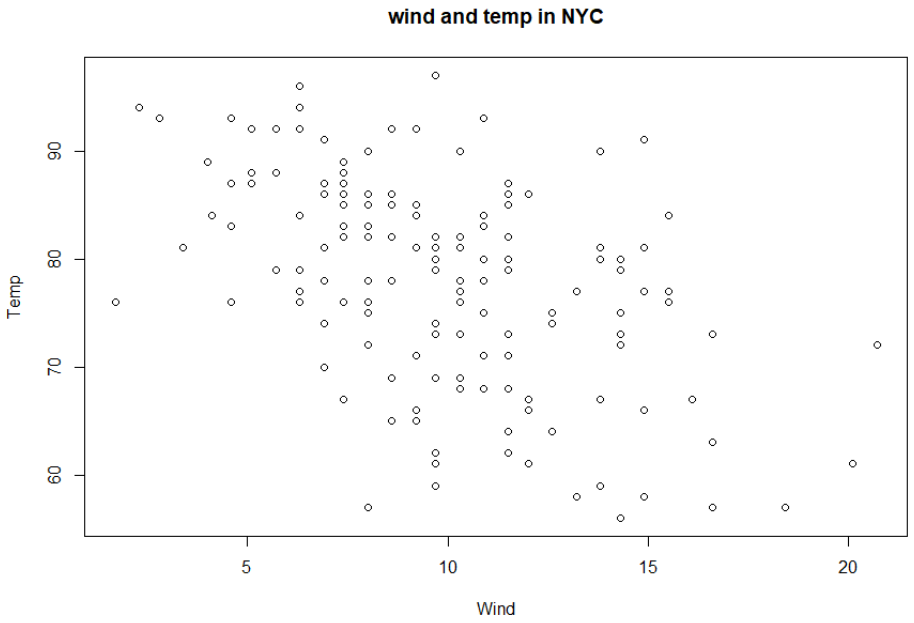
```
2 汉族    张三        1997        男      汉族    小六        2001        女      汉族
3 汉族    李四        1998        男      汉族    小五        2000        女      汉族
4 汉族    李四        1998        男      汉族    小六        2001        女      汉族
```

```
#显示匹配不到的样本，只需要在merge()函数中多传入一个参数all.x=T即可
> #all.x=T,将前一个数据框people中nat列所有的值做合并，匹配不到赋值NA
> x3<-x2[1:3,];x3
  name birthday gender  nat nation
4 小二      1999      男 藏族    藏族
5 小五      2000      女 汉族    汉族
6 小六      2001      女 汉族    汉族

> people.nat<-merge(x1,x3,by="nat",all.x = T);people.nat
   nat name.x birthday.x gender.x nation.x name.y birthday.y gender.y nation.y
1 汉族    张三        1997        男      汉族    小五        2000        女      汉族
2 汉族    张三        1997        男      汉族    小六        2001        女      汉族
3 汉族    李四        1998        男      汉族    小五        2000        女      汉族
4 汉族    李四        1998        男      汉族    小六        2001        女      汉族
5 回族  王大妈        1971        女      回族    <NA>          NA      <NA>      <NA>
```

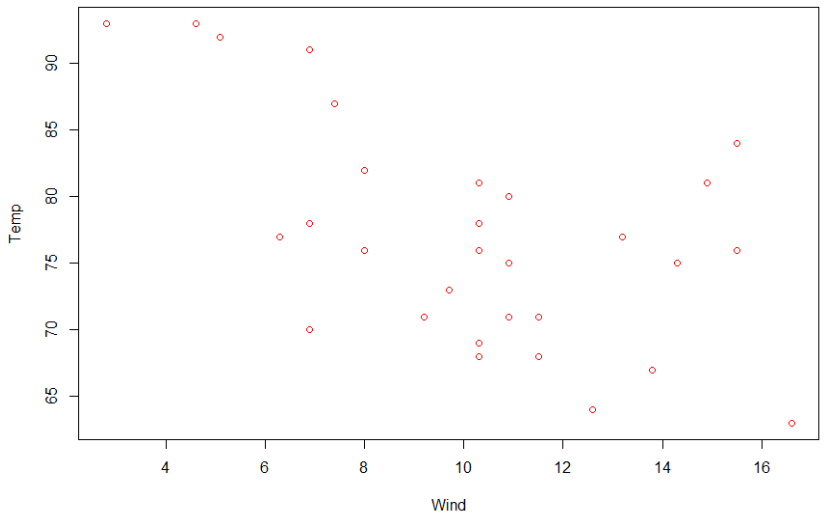
```r
#用plot()函数画空气质量这个数据集里风俗和温度的散点图
plot(airquality$Wind,airquality$Temp)
```

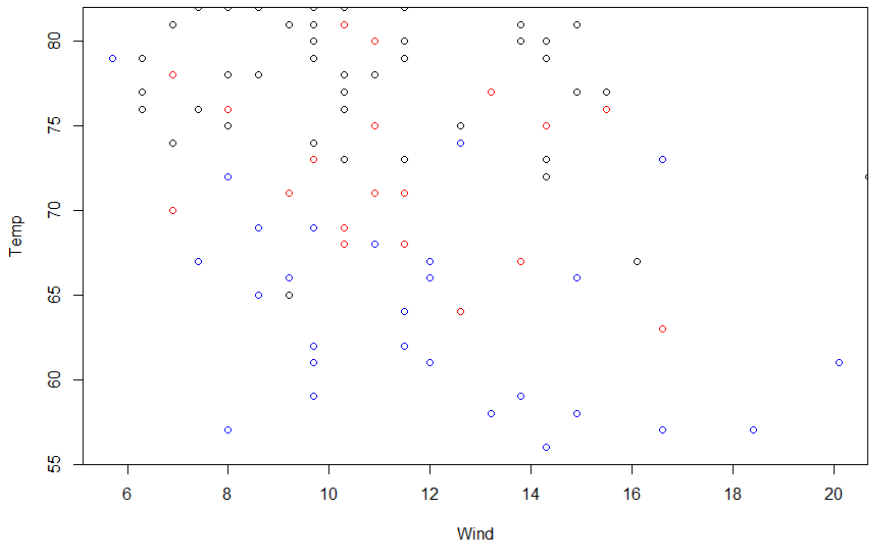



```r
#用with()函数画空气质量这个数据集里风速和温度的散点图
with(airquality,plot(Wind,Temp))
#with()函数的第一个参数为数据集，第二个参数指绘图函数

#给散点图添加标题
#法1
with(airquality,plot(Wind,Temp,main="wind and temp in NYC"))
#法2
with(airquality,plot(Wind,Temp))
title(main="wind and temp in NYC")
```

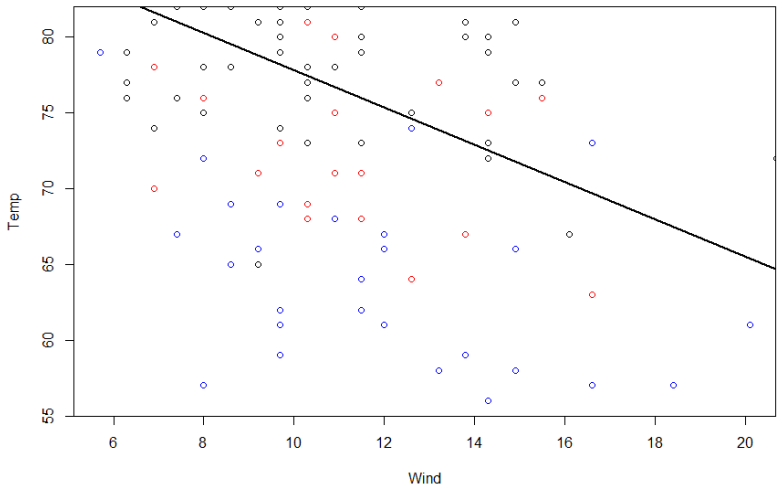



```r
#按月份来画点，不同月份对应的数值显示不同的颜色
#一、画空气质量数据集中风速和温度9月份的散点图
#1.筛选空气质量数据集的子集，子集是9月份的数据
x<-subset(airquality,Month==9)

#2.画图
with(x,points(Wind,Temp,col="red"))
#画空气质量数据集中风速和温度5月份的散点图
with(subset(airquality,Month==5),points(Wind,Temp,col="blue"))
#画空气质量数据集中风速和温度6、7、8月份的散点图
with(subset(airquality,Month%in% c(6,7,8)),points(Wind,Temp,col="black"))
```

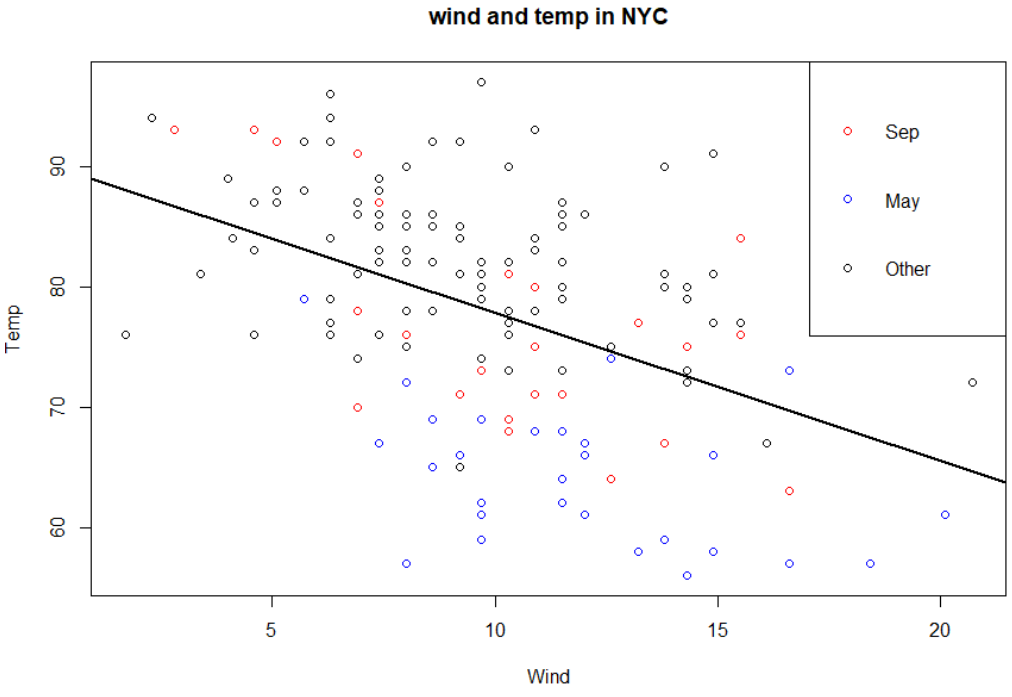





```
#在图形中添加回归线
#1.先用lm()函数拟合一个线性模型
fit<-lm(Temp~Wind,airquality)
#2.给图形添加回归线，并设置线宽
abline(fit,lwd=2)
#上述lm()函数拟合的线性模型中，Temp(风速)指因变量，Wind(温度)指自变量，它们都来自数据集airq
```



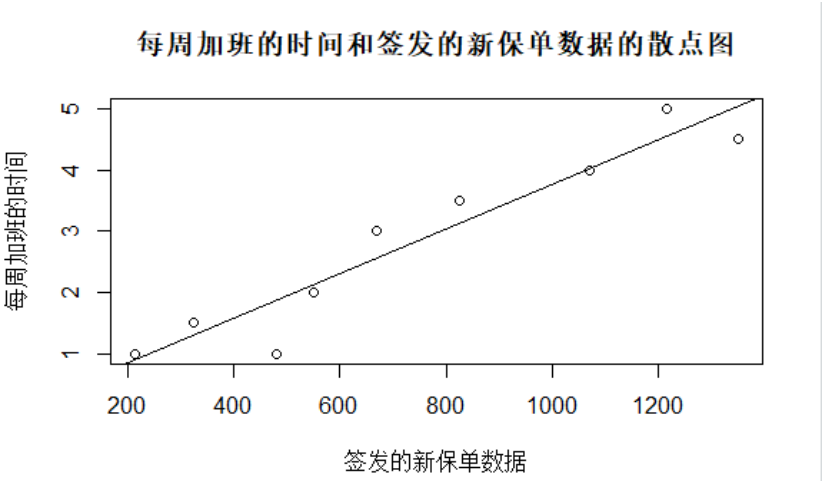

```r
#五、用legend()函数给图形添加图例
legend("topright",pch=1,col=c("red","blue","black"),legend=c("Sep","May","Other"))
#legend()函数的第一个参数指图例说明的位置在右上方，第二个参数指图例保持跟散点图一样的蓝色、黑
#第三个参数指颜色，对应于我们画图的顺序，第四个参数指颜色赋予的含义，Others指6/7/8月份
legend("topleft",pch=1,col=c("red","blue","black"),legend=c("Sep","May","Other"))
```



[ggplot绘图系统之qplot]

回归分析（Regression Analysis）是用来确定2个或2个以上变量间关系的一种统计分析方法。如果回归分析中，只包括了一个自变量X和一个因变量Y时，且它们的关系是线性的，那么这种回归分析称为一元线性回归分析。

```
x<-c(825,215,1070,550,480,1350,325,670,1215)
y<-c(3.5,1,4,2,1,4.5,1.5,3,5)
library(ggplot2)
plot(x,y,main="每周加班的时间和签发的新保单数据的散点图",xlab="签发的新保单数据",ylab="每周
abline(lm(y~x))

#每周加班的时间和签发的新保单数据呈线性关系，因而可以考虑一元线性回归模型
```




```
#2.求回归方程，并对相应的方程做检验。程序中lm()函数表示作线性模型，summary()函数提取模型的计
a<-lm(y~x)
> summary(a)

Call:
lm(formula = y ~ x)

Residuals:
     Min       1Q   Median       3Q      Max
-0.87004 -0.12503  0.09527  0.37323  0.45258

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept) 0.1215500  0.3588377   0.339    0.745
x           0.0036427  0.0004303   8.465 6.34e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.485 on 7 degrees of freedom
Multiple R-squared:  0.911, Adjusted R-squared:  0.8983
F-statistic: 71.65 on 1 and 7 DF,  p-value: 6.344e-05
```

残差的标准差（Residual standard error）为0.485，其自由度为 7 (计算方式：n-2);

相关系数的平方（Multiple R-squared）即判别系数为 0.911,

Adjusted R-squared（调整后的R方）：0.8983，说明整个模型的拟合度较高，两者存在较强的线性关系

F统计量(F-statistic）的自由度(1,7),回归方程为：Intercept的意思是截距

$$\hat{y} = 0.12155 + 0.0036427x$$

```
#3.参数的区间估计
> confint(a)
                    2.5 %      97.5 %
(Intercept) -0.726966290 0.970066254
x            0.002625117 0.004660271
```

常数项的置信区间为[-0.726966290,0.970066254],系数的置信区间为 [0.002625117,0.004660271]

```
#4.预测
> new<-data.frame(x=600)
> new
    x
1 600

> new2<-predict(a,new,interval = "confidence",level = 0.95)
> new2
       fit      lwr      upr
1 2.307166 1.897614 2.716719
```

平均值的置信区间为[1.897614,2.716719]

```
#5.残差分析
> e<-resid(a,digits=5) #计算回归a的残差，并保留5位小数
> e
          1           2           3           4           5           6
 0.37322742  0.09527080 -0.01923262 -0.12503171 -0.87004313 -0.53918695
          7           8           9
 0.19457445  0.43784500  0.45257674
```

常用包

```r
library(deplyr)
library(psych) # 本例中用到describe函数(描述统计)
library(lm.beta) # 用于在回归分析中输出标准化回归系数β
library(ggplot2) # 绘图必备
library(gridExtra) # 多图组合
library(PerformanceAnalytics) # 本例中用到chart.Correlation函数

#描述统计
data=get(data("anscombe")) # 统计学经典四组数据，具有相同的统计量(M, SD, r)，但本质不同
describe(data)
chart.Correlation(data) # 绘制变量分布及相关关系图

# 线性回归：以下4种写法在作用上等价(建模、赋值、报告结果) ------
# y1 = b0 + b1*x1
model.1=lm(y1 ~ x1, data)
summary(lm.beta(model.1))
# y2 = b0 + b1*x2
model.2=lm(y2 ~ x2, data)
model.2 %>% lm.beta() %>% summary()
# y3 = b0 + b1*x3
summary(lm.beta( (model.3=lm(y3 ~ x3, data)) ))
# y4 = b0 + b1*x4
(model.4=lm(y4 ~ x4, data)) %>% lm.beta() %>% summary()

# 散点图 ------
p1=ggplot(data=data, aes(x=x1, y=y1)) + geom_point() + geom_smooth(method="lm") + lab
p2=ggplot(data=data, aes(x=x2, y=y2)) + geom_point() + geom_smooth(method="lm") + lab
p3=ggplot(data=data, aes(x=x3, y=y3)) + geom_point() + geom_smooth(method="lm") + lab
p4=ggplot(data=data, aes(x=x4, y=y4)) + geom_point() + geom_smooth(method="lm") + lab
grid=grid.arrange(grobs=list(p1, p2, p3, p4), ncol=2, nrow=2)
ggsave("Plot.pdf", grid, width=8, height=6) # PDF保存的是矢量图，任意放缩不会模糊，推荐使
ggsave("Plot.png", grid, width=8, height=6, dpi=300) # 科研绘图一般要求分辨率至少达到 30
```

factoextra

factoextra是一个R软件包，可以轻松提取和可视化探索性多变量数据分析的输出，其中包括： 主成分分析（PCA），用于通过在不丢失重要信息的情况下减少数据的维度来总结连续（即定量）多变量数据中包含的信息。 对应分析（CA），它是适用于分析由两个定性变量（或分类数据）形成的大型列联表的主成分分析的扩展。 多重对应分析（MCA），它是将CA改编为包含两个以上分类变量的数据表格。 多因素分析（MFA）专用于数据集，其中变量按组（定性和/或定量变量）组织。 分层多因素分析（HMFA）：在数据组织为分层结构的情况下，MFA的扩展。