R简介

2015.04

R是什么？

什么是R

- R是一个有着统计分析及强大作图功能的软件系统。
- 它是AT&T贝尔实验室（Bell Laboratories）所创的S语言发展出的一种方言，是S语言的一种实现。
- R是在GNU协议Genneral Public Licence下发行的，所以R也称为“GNU S”。
- 它提供了一系列统计和图形显示工具（线性和非线性模型，统计检验，时间序列分析，分类，聚类，……）。

R的优势

- R是跨平台的自由开源软件，它不会收取任何费用，而且它的能力不比同类型的商业软件差。
- R是统计学界分析领域实际上的通用语言，备受学界的支持与推崇。
- R是全面的统计研究平台，内置丰富的统计分析、绘图、数据处理工具，并通CRAN(Comprehensive R Archive Network)安装可选的扩展包，增强R功能，囊括了其他软件尚不能用的、先进的统计算法。
- R和其他编程语言、数据库之间有良好的接口，它是彻底的面向对象的统计语言，简洁而高效便于使用者理解与使用。

R与其他分析软件的简单对比

	R	其他
算法丰富	✸	✖
更新快速	✸	✖
扩展丰富	✸	✖
风靡业界	✸	✖
开发者众多	✸	✖
代码开放	✸	✖
优秀的语言	✸	✖
调用与对接	✸	✖

选R而非SAS

- **算法方面：**SAS算法只有在安装新版本时，可能会新增或优化其算法，无法使用新颖的分析方法。而R的扩展包是由专业人士以开源方式不定期更新优化，使得用户可以使用流行、灵活的数据分析方法。

- **流行趋势方面：**SAS是老旧的商业统计软件，而R在业界的流行使得其被更多的人关注，其开源的运作方式，越来越多的人士为项目作出贡献，R发展得也越来越好。

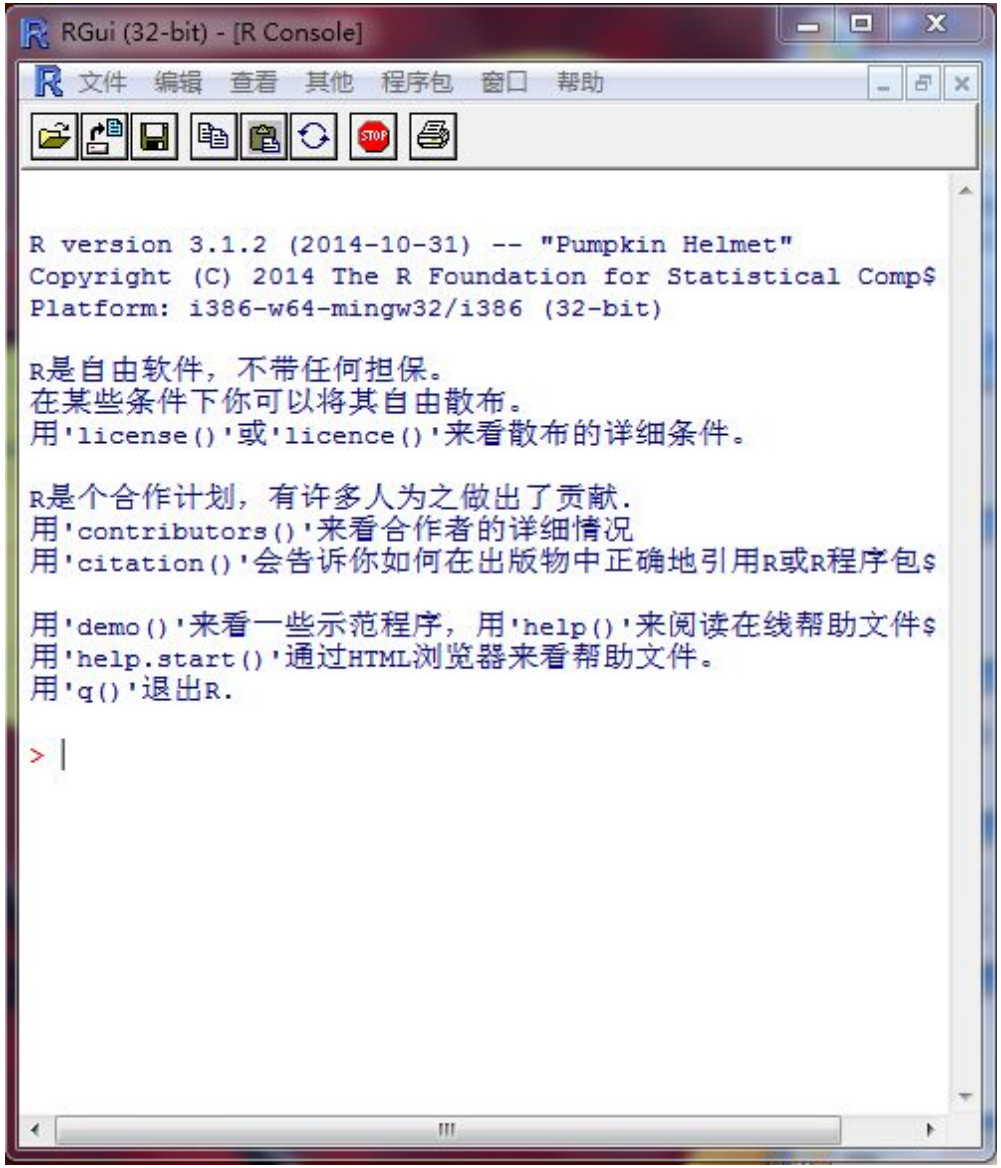
- **语言特性方面：**SAS是类命令语言，将分析结果存储在数据集内，获取困难，给后续使用带来极大麻烦。而R是优秀的脚本语言，将结果赋给某个变量，简洁高效，节省了无谓的时间。

- **系统集成方面：**SAS是商业软件，算法是其技术核心不会对外公开。而R是以开源方式运作（GPL）程序算法均是公开的，易于与其他工业级的语言（c、java等）或系统进行调用与对接，可以发挥R的算法优势。

R怎么用？

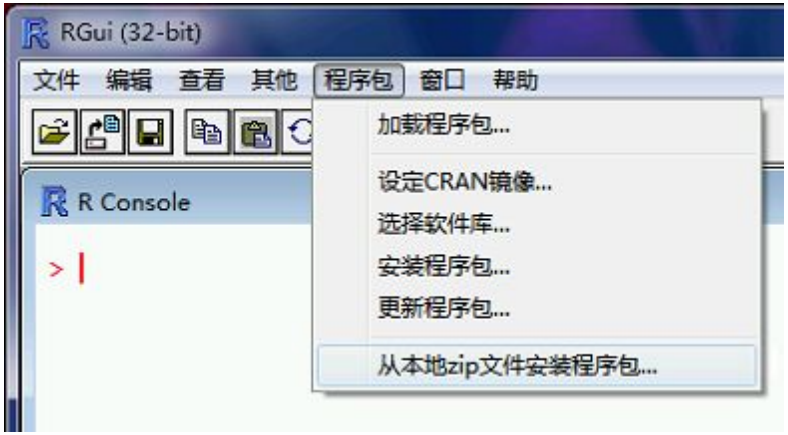
R的安装

- R的安装包可在CRAN.R/ project.org上取得。
- 其他扩展包、及相应资料也可在CRAN或镜像站点上获得。
- 在选择与操作系统对应的版本（32bit、64bit）默认安装后，启动Rgui即可。
- 其他：RStudio，R的第三方集成开发工具。



Rgui的重要功能

打开程序脚本：打开需要运行的脚本程序，它会新开一个脚本编辑窗口，可以编辑、运行、保存脚本。



从本地zip文件安装程序包：

离线情况下，以windows为例，先从CRAN上找到并下载所需windos版的包。得到包之后，选择该选项，选取对应本地包即可安装。

非离线下，设定好CRAN镜像后，使用install.packages()函数即可安装。

R概述

R概述—常用对象的类型

- 向量(vector)、数组(array)、矩阵(matrix)、因子(factor)、列表(list)、数据框(data frame)、函数(function)等。
- 其中最重要的类型为data frame，和矩阵类似的结构，列是不同的维度对象，行是观测个体的数值、分类变量，它很好地描述统计分析数据。

```
> manager <- c(1,2,3,4,5)
> Datedata <- c("10/24/08","10/28/08","10/1/08","10/12/08","5/1/09")
> country <- c("US","US","UK","UK","UK")
> gender <- c("M","F","F","M","F")
> age <- c(32,45,25,39,99)
> q1 <- c(5,3,3,3,2)
> q2 <- c(4,5,5,3,2)
> q3 <- c(5,2,5,4,1)
> q4 <- c(5,5,5,NA,2)
> q5 <- c(5,5,2,NA,1)
> leadership <- data.frame(manager,Datedata,country,gender,age,
+ q1,q2,q3,q4,q5,stringsAsFactors=FALSE)
> leadership
  manager Datedata country gender age q1 q2 q3 q4 q5
1       1 10/24/08      US      M  32  5  4  5  5  5
2       2 10/28/08      US      F  45  3  5  2  5  5
3       3  10/1/08      UK      F  25  3  5  5  5  2
4       4 10/12/08      UK      M  39  3  3  4 NA NA
5       5   5/1/09      UK      F  99  2  2  1  2  1
```

R概述—常用操作符及常量

EX:
(2^3)/2 >= 2 -> x;x

y <- c(1:10,NA,NaN,"\n a \t");y

y == NA;is.na(y);

y[!is.na(y)]

```
> (2^3)/2 >= 2 ->  x;x
[1] TRUE
>
> y <- c(1:10,NA,NaN,"\n a \t");y
 [1] "1"       "2"       "3"       "4"       "5"
 [6] "6"       "7"       "8"       "9"       "10"
[11] NA        "NaN"     "\n a \t"
>
> y == NA;is.na(y);
 [1] NA NA NA NA NA NA NA NA NA NA NA NA NA
 [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [9] FALSE FALSE  TRUE FALSE FALSE
>
> y[!is.na(y)]
 [1] "1"       "2"       "3"       "4"       "5"
 [6] "6"       "7"       "8"       "9"       "10"
[11] "NaN"     "\n a \t"
```

R的简单使用（一）-- 描述分析

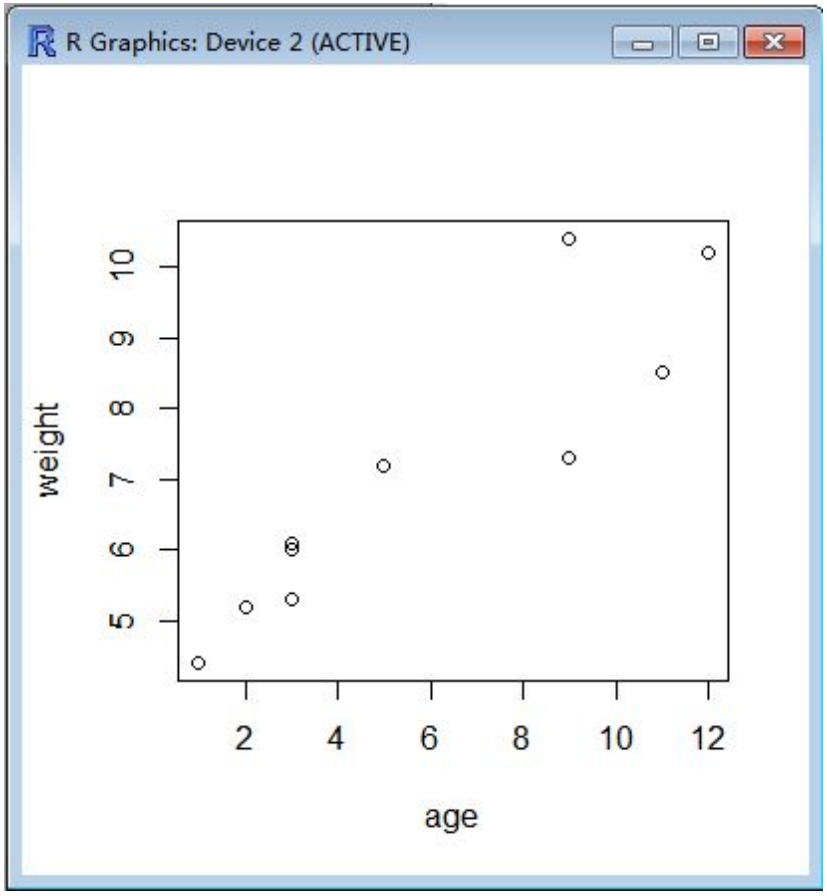
例：研究10名婴儿在出生后一年内的月龄和体重的关系。

C函数返回一个数组或是一个向量

```
> age <- c(1,3,5,2,11,9,3,9,12,3)
> age
 [1]  1  3  5  2 11  9  3  9 12  3
> age[1]
[1] 1
> age[2]
[1] 3
> age[2,1]
错误于age[2, 1] : 量度数目不对
> weight <- c(4.4,5.3,7.2,5.2,8.5,7.3,6.0,10.4,10.2,6.1)
> mean(weight)
[1] 7.06
> sd(weight)
[1] 2.077498
> cor(age,weight)
[1] 0.9075655
> plot(age,weight)
>
```

mean函数返回均值

plot函数进行绘图



可以看到，这10名婴儿的平均体重是7.06 kg，标准差为2.08 ，月龄和体重之间存在较强的线性关系（相关度 = 0.91）。

R的简单使用（二）-- 线性回归

```
> fit <- lm(weight~age)
> summary(fit)

Call:
lm(formula = weight ~ age)

Residuals:
     Min       1Q   Median       3Q      Max
-1.24986 -0.44858  0.07642  0.32107  1.85014

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.35962    0.52985   8.228 3.57e-05 ***
age          0.46558    0.07616   6.113 0.000285 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9253 on 8 degrees of freedom
Multiple R-squared:  0.8237,    Adjusted R-squared:  0.8016
F-statistic: 37.37 on 1 and 8 DF,  p-value: 0.0002853

> coefficients(fit)
(Intercept)         age
  4.3596206   0.4655827
> coefficients(fit)["age"]
      age
0.4655827
```

lm函数进行回归分析

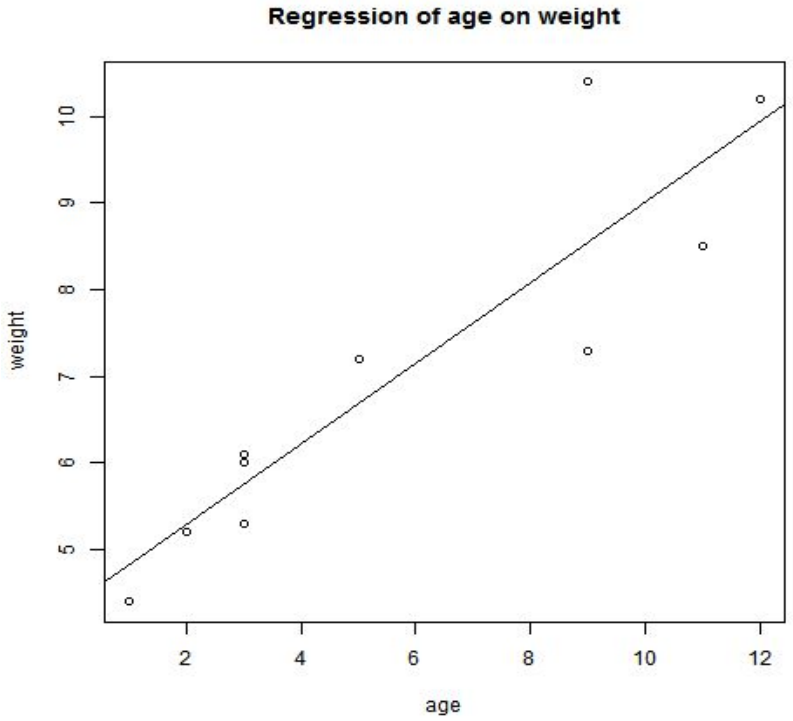
summary函数返回回归分析的详细结果

函数返回回归分析结果的系数

```
> png("lm.png")
> plot(age,weight)
> abline(fit)
> title("Regression of age on weight")
> dev.off()
null device
          1
>
```

将绘图保存为PNG图像

abline函数添加参考线

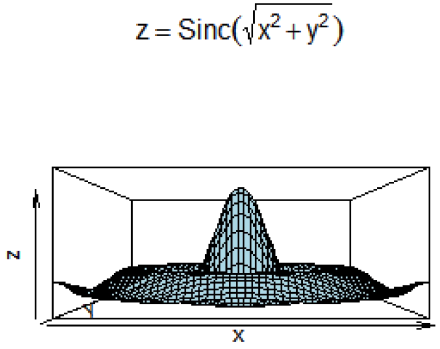


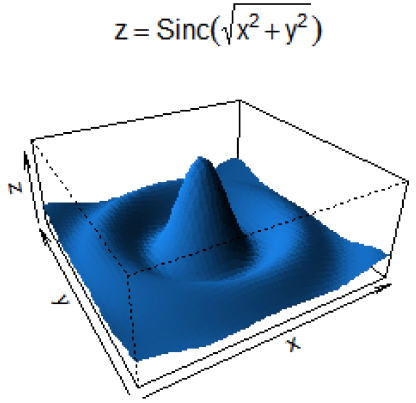
通过简单的回归得到，weight=0.46558*age+4.35962，R方为0.8237，p值很小，模型比较显著。

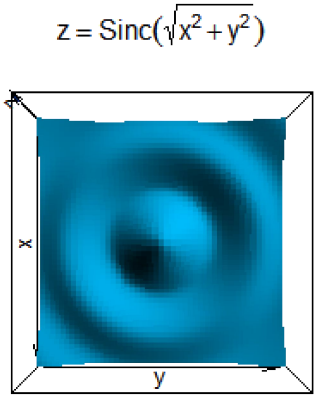
R的简单使用（三）—— 画透视图

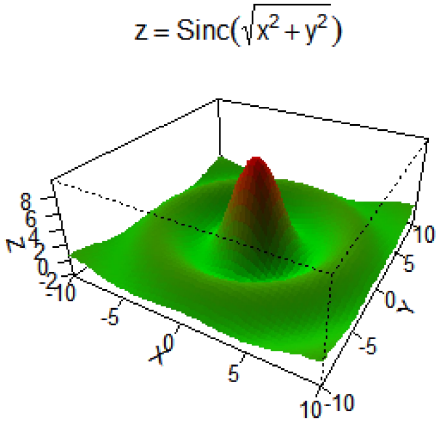
例：展现R强大而简单的绘图功能。用R画一个3D水滴。

```
x <- seq(-10, 10, length.out = 50)
y <- x

rotsinc <- function(x,y){
sinc <- function(x) {
    y <- sin(x)/x ;
    y[is.na(y)] <- 1; y}

10 * sinc( sqrt(x^2+y^2) )
}

sinc.exp <- expression(z ==
Sinc(sqrt(x^2 + y^2)))

z <- outer(x, y, rotsinc)

par(mfcol=c(2,2))

persp(x, y, z, theta = 0, phi = 0, expand
= 0.4, col = 'lightblue')

title(main = sinc.exp)
```









注：上面为绘图部分代码，persp函数：theta gives the azimuthal 方位角 direction and phi the colatitude 余纬度.

R的简单使用（四）–导入数据

例：分析前准备，导入三个CSV文件。

```
setwd("E:/R") #设置当前工作区路径

da1_file </  paste(getwd(),
             '/M3DATA/ 14/ 11.csv',sep='')
da2_file </  paste(getwd(),
             '/M3DATA/ 14/ 12.csv',sep='')
da3_file </  paste(getwd(),
             '/M3DATA/ 15/ 01.csv',sep='')
da4_file </  paste(getwd(),
             '/全国邮政.csv',sep='')
#为对导入文件路径赋值

da1 </  read.table(da1_file,header=TRUE,sep=',')
da2 </  read.table(da2_file,header=TRUE,sep=',')
da3 </  read.table(da3_file,header=TRUE,sep=',')
da4 </  read.table(da4_file,header=TRUE,sep=',')
#使用read.table函数导入以逗号分隔的csv数据

ALL_DATA </  rbind(da1,da2,da3)
#纵向合并多个数据框
```

需要RODBC包、xlsx包、xml包、Rcurl包、foreign包

R的简单使用（五） –广义线性模型拟合

```
all_dat1 </ all_dat1[which
(all_dat1% blk != 'F'),]
```

在分析样本中剔除F code

```
fit.reduced2 </ glm(paid ~ hwife + tot + year
+ mo + tmp_mnt_n + d2 + d3 + d4 + d5 + d7 +
d8 + d12 + d14 + d15 + d18 + d21 + d22
+ act_4 + act_9 + csc3 + blk_Q + blk_R + blk_C
+ f1 + f2 + f4 + f5 + f9 + f15
,data=all_dat1,family=binomial(link = 'logit'))
```

使用glm函数进行拟合，选取binomial分布族（二项式），logit函数。

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.8498762  0.0363586  50.879  < 2e-16 ***
hwife       -0.1419516  0.0533418  -2.661 0.007787 **
tot         -0.0082002  0.0004007 -20.466  < 2e-16 ***
year        -0.0229995  0.0010239 -22.462  < 2e-16 ***
mo           0.0071511  0.0002910  24.571  < 2e-16 ***
tmp_mnt_n   -0.0158936  0.0020484  -7.759 8.55e-15 ***
d2          -1.0243843  0.0947146 -10.815  < 2e-16 ***
d3           0.8799965  0.0438020  20.090  < 2e-16 ***
d4           0.5889544  0.1501067   3.924 8.72e-05 ***
d5          -1.4326257  0.0617949 -23.184  < 2e-16 ***
d7          -1.5809470  0.0737020 -21.451  < 2e-16 ***
d8           2.6823146  0.7228624   3.711 0.000207 ***
d12         -1.1260582  0.1183903  -9.511  < 2e-16 ***
d14          1.8856724  0.3348744   5.631 1.79e-08 ***
d15         -5.0695732  1.0082313  -5.028 4.95e-07 ***
d18          0.5016460  0.0258154  19.432  < 2e-16 ***
d21          0.5253948  0.0252341  20.821  < 2e-16 ***
d22          0.2604518  0.0231658  11.243  < 2e-16 ***
act_4       -0.5222931  0.0162385 -32.164  < 2e-16 ***
act_9        0.2380700  0.0819289   2.906 0.003663 **
csc3        -2.8439165  0.7155650  -3.974 7.06e-05 ***
blk_Q       -0.1358147  0.0533882  -2.544 0.010962 *
blk_R       -1.0019431  0.0318877 -31.421  < 2e-16 ***
blk_C       -1.0220694  0.0209632 -48.755  < 2e-16 ***
f1          -0.4552348  0.0168339 -27.043  < 2e-16 ***
f2           0.3710206  0.1084140   3.422 0.000621 ***
f4          -1.1753012  0.0577498 -20.352  < 2e-16 ***
f5          -0.5269201  0.0389995 -13.511  < 2e-16 ***
f9          -0.5444500  0.0771402  -7.058 1.69e-12 ***
f15         -0.4946641  0.0323834 -15.275  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

glm回归结果的系数

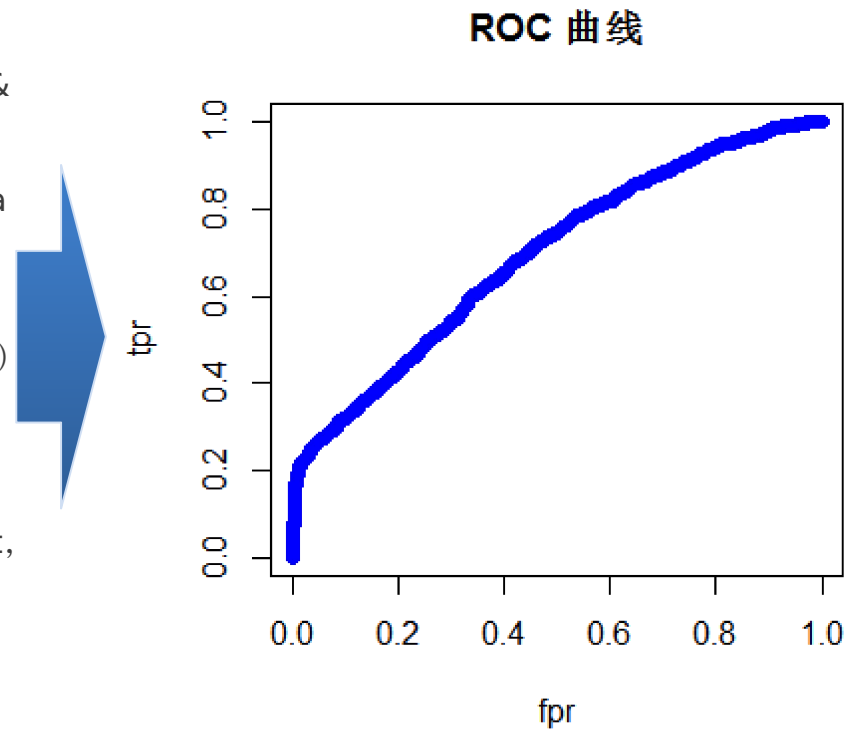
glm即Generalized Linear Models，预测时使用predict()函数。

R的简单使用（六） --ROC与AUC值(1)

```
getROC <- function(fit,obj,reduce){

pre <- predict(fit,type='response')
#将预测概率prob和实际结果paid放在一个数据框中
len <- length(obj)
dis <- len / length(pre)
data <- data.frame(prob=pre,obs=obj[c(dis+1): len])

#按预测值从低到高排序
data <- data[order(data$prob),]
n <- nrow(data)
data <- data[seq(1,n,reduce),]
tpr <- fpr <- rep(0,nrow(data))

#根据不同的临界值threshold计算TPR 真正率 灵敏度 FPR
for ( i in seq(1,nrow(data)) ){
  threshold <- data$prob[i]
  tp <- sum(data$prob | threshold & data$obs == 1)
  fp <- sum(data$prob | threshold & data$obs == 0)
  tn <- sum(data$prob < threshold & data$obs == 0)
  fn <- sum(data$prob < threshold & data$obs == 1)
  tpr[i] <- tp/(tp+fn) #真正率
  fpr[i] <- fp/(tn+fp) #假正率
}

plot(fpr,tpr,main='ROC 曲线',type='p',
             xlim=c(0,1),ylim=c(0,1),col='blue')
```

R的简单使用（六）--ROC与AUC值(2)

```
###/  /  /   AUC /  /  /  ###
roc_dat </   data.frame(TPR=tpr,FPR=fpr)
roc_dat </   roc_dat[which(roc_dat% TPR !=0 &
roc_dat% FPR !=0 ),]
roc.fit </   lm(TPR ~
I(FPR^4)+I(FPR^3)+I(FPR^2)+FPR,data=roc_da
t)

roc_line </   coefficients(roc.fit)
auc </    roc_line[1] + (1/5) * roc_line[2] + (1/4)
* roc_line[3] + (1/3) * roc_line[4] + (1/2) *
roc_line[5]

# 删除无用变量，释放资源
rm(pre,len,dis,data,n,tp,fp,tn,fn,roc_dat,roc.fit,
roc_line)
gc()

#返回的结果
roc.out </   list(tpr=tpr,fpr=fpr,auc=auc)
}

#调用函数
getROC(fit.reduced2,all_dat1% paid,20)
```



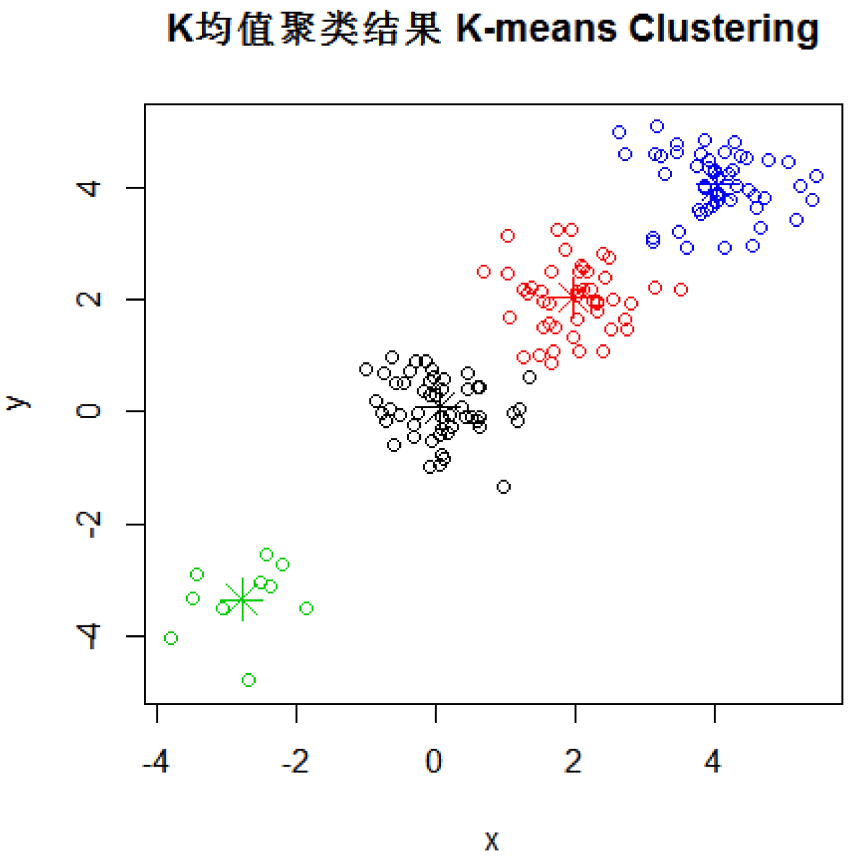

```
$auc
(Intercept)
  0.6898663
```

R进阶

R进阶–K均值聚类

K均值聚类(**K- Means Clustering**)适用于大样本的样品聚类，使用逐步聚类法进行聚类。

```
x <- rbind(
matrix(rnorm(100, sd = 0.6), ncol = 2),
matrix(rnorm(100, mean = 2, sd = 0.6),
ncol = 2),
matrix(rnorm(100, mean = 4, sd = 0.6),
ncol = 2),
matrix(rnorm(20, mean = -3, sd = 0.6),
ncol = 2)
)
colnames(x) <- c("x", "y")
cl <- kmeans(x, 4)

cl

plot(x,col = cl$cluster,main="K均值聚类
结果 K-means Clustering")
points(cl$centers, col = 1:4, pch = 8,
cex = 2)
```



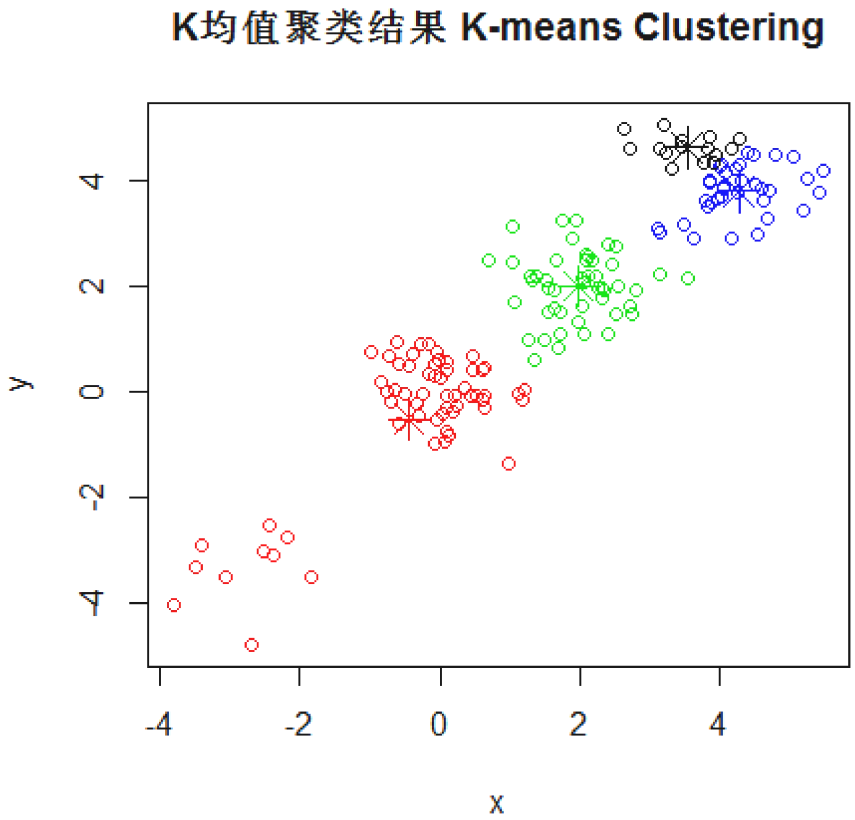
R进阶–K均值聚类判别优劣

相关指标：

Totss
The total sum of squares.

Betweenss
The between/ cluster sum of squares.

withinss
Vector of within/ cluster sum of squares, one component per cluster.
群内距：样本到其中心点的距离平方和。




```
Within cluster sum of squares by cluster:
[1] 30.027774 31.999207  7.615486 36.189445
 (between_SS / total_SS =  92.5 %)
> 1-(cl$tot.withinss/cl$totss)
[1] 0.924721
> cl$size
[1] 51 48 10 51
```

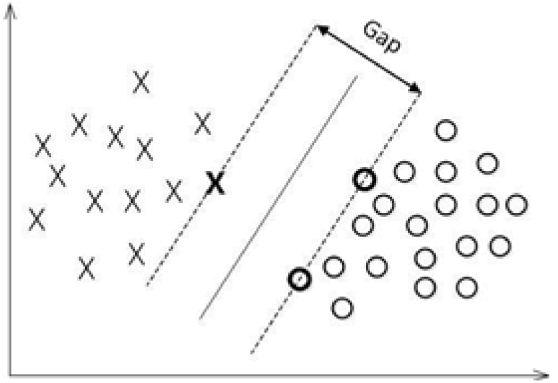
```
Within cluster sum of squares by cluster:
[1]   4.153859 198.182116  34.352290  18.899446
 (between_SS / total_SS =  81.8 %)
> cl$withinss
[1]   4.153859 198.182116  34.352290  18.899446
> 1-(cl$tot.withinss/cl$totss) #1-(sum(cl$withinss)/cl$totss)
[1] 0.8181987
```

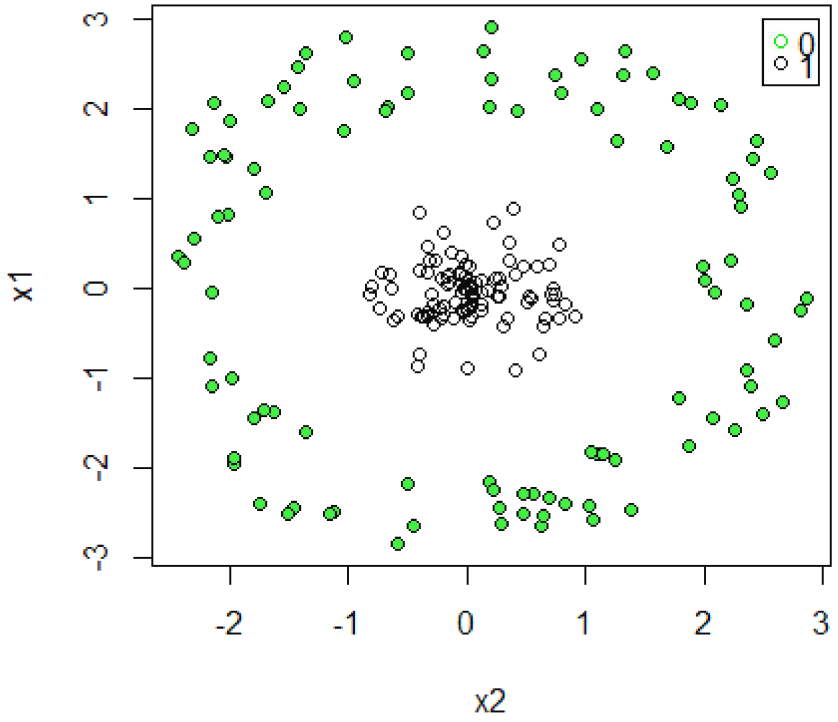
R进阶–非线性支持向量机 Non–Liner SVM(1)

支持向量机(SVM):一种线性和非线性数据的分类方法，它使用非线性映射将原始数据映射到高维空间，在该空间内搜索最佳分离超平面。

例：1.生成线性不可分的样本。

```
n = 100

r = runif(n);a = 2*pi*runif(n)
a1 = r*sin(a);a2 = r*cos(a)

r = 2+runif(n);a = 2*pi*runif(n)
b1 = r*sin(a);b2 = r*cos(a)

x =
rbind(matrix(cbind(a1,a2),,2),
            matrix(cbind(b1,b2),,2))
y <- matrix(c(rep(1,n),rep(0,n)))

ro_dat <-
as.data.frame(cbind(x,y))
```



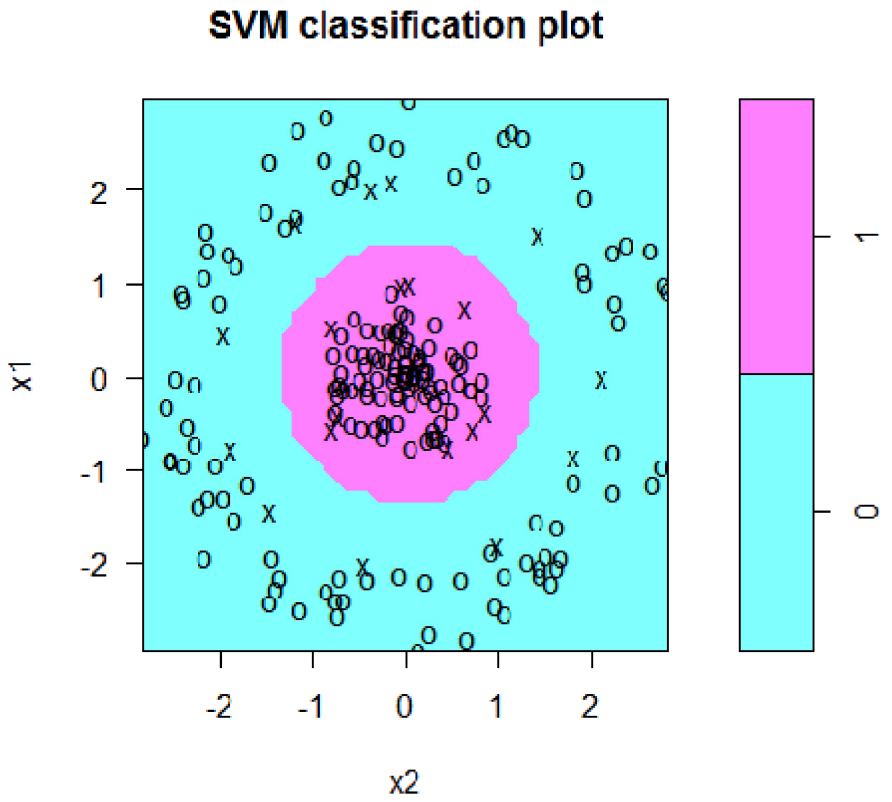


R进阶–非线性支持向量机 Non–Liner SVM(2)

例：运用SVM对样本进行分类。

```
library(e1071)

radial.svm.fit </  svm(y ~ .,
            data=ro_dat,C=100,
            type='C/  classification',
            kernel ='radial',
            probability=TRUE);
summary(radial.svm.fit);

plot(radial.svm.fit,data=ro_dat)
```

```
Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.5

Number of Support Vectors:  17

 ( 7 10 )


Number of Classes:  2

Levels:
 0 1
```



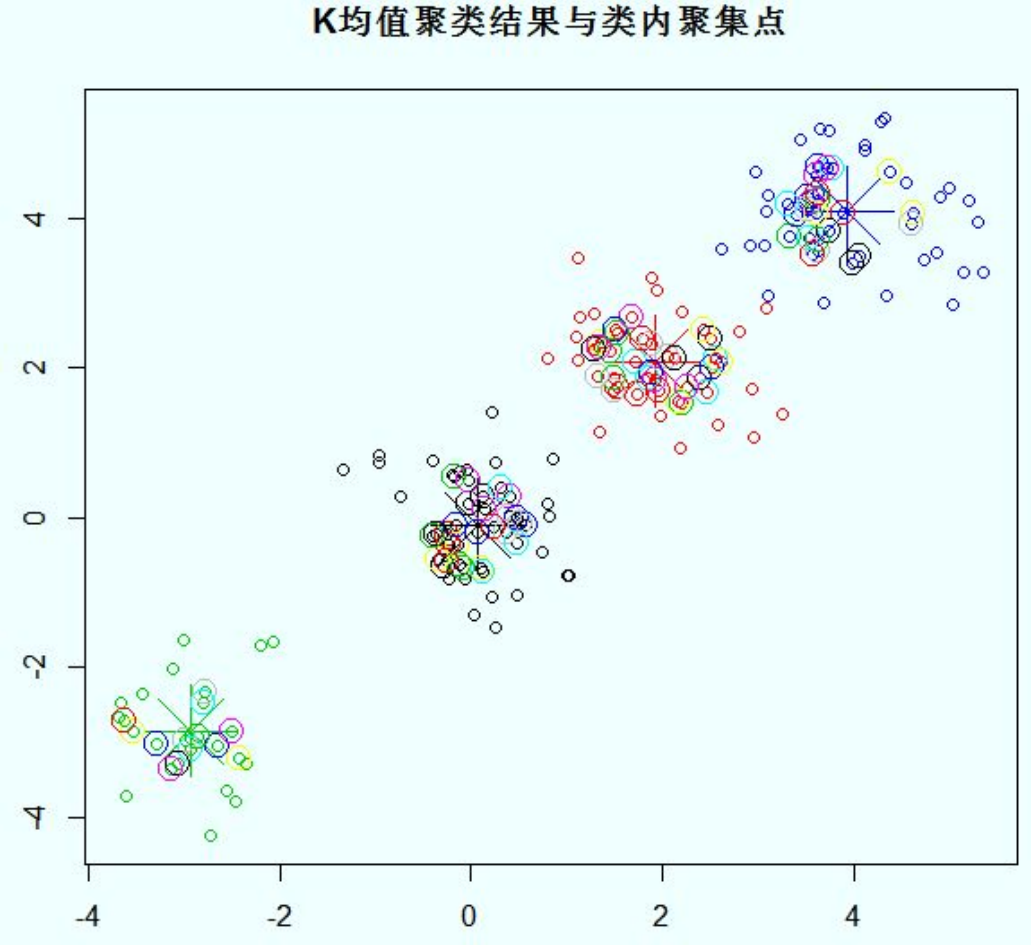
基本判断（准确率）：mean(with(ro_dat,ifelse(y == predict(radial.svm.fi

R进阶——种快速SVM的方法：SVM与K均值聚类配合使用

SVM模型的复杂度取决于支持向量的数据点的个数。

例：画出聚类结果内部的点。

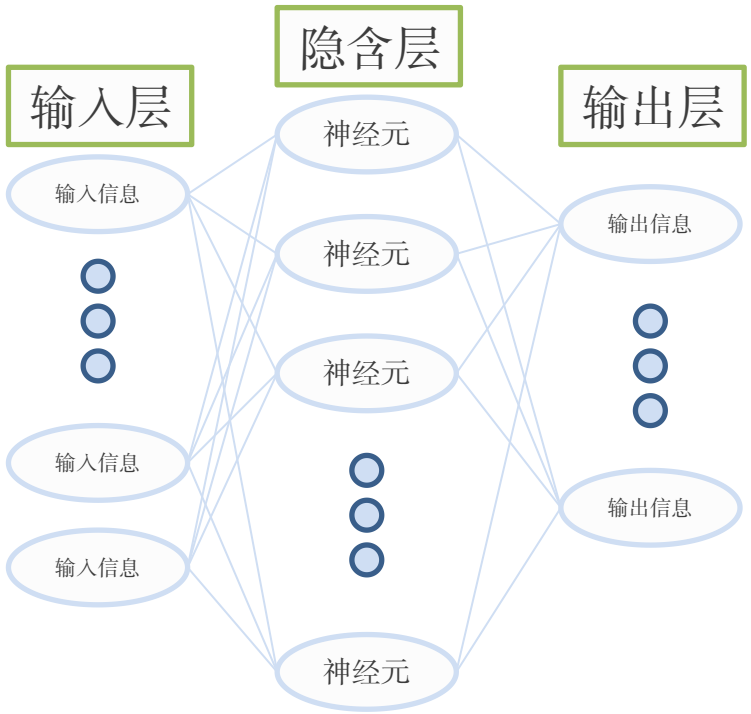
```
resid.m <-  m -  fitted(cl)

SumOfSquare <-
function(x){x[1]^2+x[2]^2}
#计算残差平方

inpoint <-
m[which(apply(resid.m,1,Sum
OfSquare) < 0.5),]
#残差小，被包裹在类里
```



R进阶–单层神经网络 Neural Networks（1）

- **神经元细胞**:主要由细胞体、轴突、树突组成。
- **细胞体**:神经元活动的能量供应和进行生化过程的中心。
- **轴突**:神经元兴奋信息的传输出口。
- **树突**:其他神经元传入信息的入口。

神经网络是由大量的、简单的神经元（处理单元）广泛地互相连接而形成的复杂的网络系统，反应了人脑的基本功能特征。特别适合于处理多因素、多条件、不精确和模糊的信息处理问题。



R进阶–单层神经网络 Neural Networks（2）

例：利用单层神经网络预测鸢尾花的品种。

```
library(nnet)

ir <- rbind(iris3[,,1],iris3[,,2],
iris3[,,3])
targets <- class.ind( c(rep("s", 50), rep("c",
50), rep("v", 50)) )

samp <- c(sample(1:50,25),
sample(51:100,25), sample(101:150,25))

ir1 <- nnet(ir[samp,], targets[samp,], size =
2, rang = 0.1,decay = 5e- 4, maxit = 200)
```

Size:隐层神经元数量，rang:初始化的权重范围，
decay:衰减度，maxit:最大迭代次数

```
test.cl <- function(true, pred) {
    true <- max.col(true)
    cres <- max.col(pred)
    table(true, cres)
}
test.cl(targets[- samp,], predict(ir1, ir[-
samp,]))
```

萼片 花瓣 品种

```
> head(iris,5)
  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          1.4         0.2  setosa
```

```
> test.cl(targets[-samp,], predict(ir1, ir[-samp,]))
     cres
true  1  2  3
   1 25  0  0
   2  0 25  0
   3  3  0 22
>
```

鸢（yuan）尾， 一种鸢尾属植物，有狭窄的剑状叶和显著的各色花朵。

扩展R

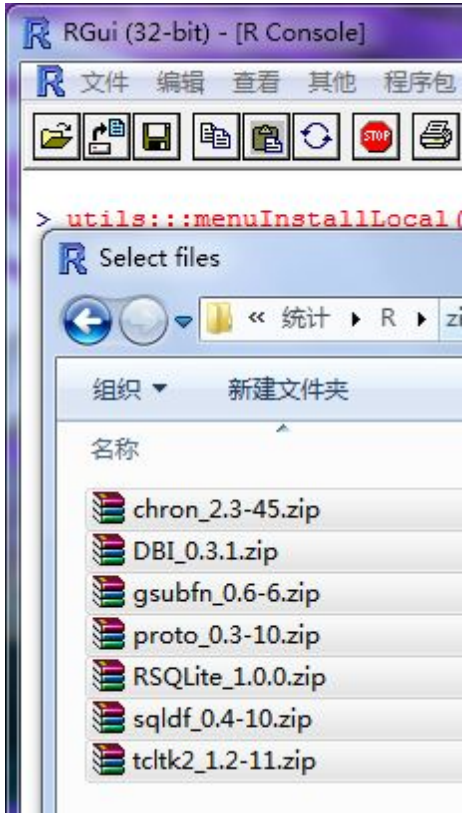
扩展R–自定义安装扩展包

例:安装sqldf相关包，并用sql语句在mtcars中找出重量大于4吨的车的相关信息。

install.packages(‘sqldf‘)

library(sqldf)

newdf </ sqldf(‘select * from mtcars where wt ı 4 order by mpg‘,row.names=TRUE)




```
> newdf <- sqldf("select * from mtcars where wt > 4 order by mpg"
+ ,row.names=TRUE)
> newdf
                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
```

扩展R–扩展包概览

扩展包	描述
foreign	读取Minitab、S、SAS、SPSS、Stata等软件存储的数据。
ggplot2	语法全面而连贯的绘图系统，便于创建新颖、高级的图形。
kernlab	线性和非线性的SVM工具库，提供更多的核函数。
playwith	用于编辑R图形，以及互动式的用户图形界面。
RCurl	与libcurl库中HTTP协议交互的接口。
reshape	提供一系列处理、聚会及管理数据的工具。
ROCR	使用简便的ROC绘图工具。
RODBC	ODBC数据库访问接口。
sqldf	提供对R中数据框的SQL操作。
tm	提供一系列文本挖掘函数，处理非结构化文本。
xlsx	读写和格式化xlsx文件。
XML	解析和生成XML的工具。
e1071	线性和非线性的SVM工具库。
...	...

扩展R–参考资料

- R 导论（丁国徽译）
- R in Action R语言实战
- R for Beginners Chinese Edition 2.0
- 统计建模与R软件（薛毅、陈立萍）
- Machine learning for hackers 机器学习实用案例解析
- 统计学习方法（李航）

谢谢