生态学和进化中的现代统计

第二天：R语言开发和基本统计分析的实现

张金龙

jinlongzhang01@gmail.com

云南 • 昆明

2013年12月24日

目录

目录

代码的重用：函数

假设mat是输入的群落数据矩阵，即每行表示样地，每列表是物种，要计算每个样地的shannon多样性指数

计算shanon生态位宽度的R代码

```
Bi <- c()
for (i in 1:ncol(mat)) {
nij <- mat[, i]
nij <- nij[nij > 0]
pij <- nij/sum(nij)
Bi[i] <- -sum(pij * log(pij))
} Bi <- as.data.frame(t(Bi))
colnames(Bi) <- colnames(mat)
```

Bi是计算结果。
问题：每次计算shannon生态位宽度，都要拷贝这段代码？
代码的重复拷贝，不但浪费了大量的时间和精力，而且非常容易产生错误，并且难以纠正。

R函数

编写R函数是减少代码重复拷贝，提高程序效率的重要手段。
R可以方便地编写函数，用户编写的函数可以直接调用。
R无需声明变量的类型，这与C,C++等语言不同。

R函数基本结构

函数名<-function(数据,参数1= 默认值,参数2= 默认值,···){
异常处理;
表达式(循环/判别);
return(返回值);
}

R函数

函数举例：度、分、秒转换为时的小数

```
deg2dec <- function (d, m, s) {
if(any(!is.numeric(c(d,m,s)))){
stop("None numeric value find, can not calculate")
}
if (d < 0) {
m = -m
s = -s
}
res = d + m/60 + s/3600
return(res)
}
```

deg2dec(23,56,04)
deg2dec(23,56,"Test")

for循环和while循环

for是循环中最重要的函数之一，在for循环下，只要条件满足，即执行后面花括号{}内的语句,如果是只有一行语句，则花括号可以省略。
for(变量in 向量) 表达式

for的用法

```
dat <- rpois(20,19)
for(i in 1:20) {
print(dat[i])
}
```

while循环是满足条件下，执行某些语句的控制函数。

while的用法

```
i <- 1
while(i<10){
print(i);
i <- i + 1
}
```

流程控制if

if是在条件为真的情况下，执行后面程序的条件控制语句。决定程序的分支和走向

if的用法

if(条件) {表达式}
if(条件) {表达式1} else {表达式2}

if举例

```
p = 0.03 {
if(p <= 0.05) {
print("p <= 0.05")
}
else {
print("p > 0.05")
}
}
```

用warning和stop处理异常

若数据格式等不能满足要求，或者参数设定错误，函数处理往往会给出错误的结果，此时，必须发出警告或及时终止程序，以提高程序的稳健性。

警告的写法

```
if(any(is.na(inputdata)))
inputdata <- na.omit(inputdata)
cat("NAs found in the input data, and have been removed.")
```

终止的写法

```
if(any(is.na(xx))) stop("NAs are not allowed!")
```

missing() 判断某个参数是否缺失，返回值为TRUE或者FALSE

stopifnot() 如果不满足某条件，则中止。

参数的匹配

match.arg的使用

```
center <- function(x, type = c("mean", "median",
"trimmed"))
{
type <- match.arg(type)
switch(type,
mean = mean(x),
median = median(x),
trimmed = mean(x, trim = .1))
}
```

type参数在经过match.arg函数处理后，无需输入全称，即可判断。
如果输入的参数和type预设的名称不符，则触发错误。R给出相应的提示。

变量的作用域

输入一个字符串，如“ABCDE”，返回“EDCBA”

函数举例

```
reverse <- function (x)
{
splitstr <- substring(x, 1:nchar(x), 1:nchar(x))
result <- paste(splitstr[length(splitstr):1], collapse =
"")
return(result)
}
```

x为形参，实际参数是输入的结果，而x的值，仅在函数的花括号内部生效，在调用的过程中创建，其他函数不能直接访问x。随着该函数运行结束，x的值从内存中清除，这就是x的作用域。

返回值

- 1. 返回值表示函数输出的结果。
- 2. 返回值必须是一个对象。如果函数的结果需要返回多个值，可以创建一个list()，并返回该list。
- 3. R默认将最后一行作为返回值。
- 4. 一个函数内部，可能出现若干个return()。如果执行中遇到任意一个return()，函数都将结束，返回return内部的对象。

Debugging:检查函数的错误

一般容易出错地方:

1. 检查拼写，括号和运算符等的中英文切换
2. 参数不匹配
3. 向量或者矩阵等下标出界
4. 逻辑错误（最难发现的错误）。

undebug函数和debug函数

debug和undebug需要配合使用。debug内部放函数名称，之后再运行该函数时，函数会一步一步执行，执行每一步，都需要按回车，中间过程的变量也都可访问，因此方便检查错误。修改完错误后，要用undebug取消debuging.

Debug和Undebug的用法

Debug示例

```
print.mat2 <- function(x){
Ncol <- ncol(x); Nrow <- nrow(x)
temp <- c()
k = 1
for(i in 1:Nrow) {
for(j in 1:Ncol){
temp[k] <- x[i,j]
k = k + 1
}
}
return(temp)
}
debug(print.mat2)
x1 <- matrix(1:20, 4, 5)
print.mat2(x1)
```

练习1

(1)编写一个函数，将39度20分52秒转换成十进制。如果用户输入的字符不在合理的范围内，要给出警报。如果用户输入了字符串，要告知输入的数据类型有误。
(2)编写一个函数，输入一个字符串，该函数要告知该字符串含有多少个字母，每个字母分别有多少个？如果输入的字符串不包括字母，要告诉用户，该字符串不包括字母。

目录

通过Rcpp调用C语言和C++函数

1 安装Rcpp程序包，install.packages("Rcpp")
2 安装Rtools，并配制好启动路径。电脑>属性>高级系统设置>高级>环境变量>系统变量>路径
3 在R console 中运行library(Rcpp) ，之后调用Rcpp.package.skeleton函数，

Rcpp.package.skeleton

```
Rcpp.package.skeleton( "test" )
```

在生成的R包skeleton中，可以找到rcpp_hello_world.cpp文件，可以尝试添加几个函数，修改成下页中的样式（注意以下代码为C++文件,扩展名为cpp)

Rcpp举例调用时C++文件举例

Rcpp文件编辑

```
////////////////////// C++文件开始/////////////////////
#include "rcpp_hello_world.h"
#include <Rcpp.h>
using namespace Rcpp;
using namespace std;
// 注意#include <Rcpp.h> 是调用Rcpp的必要条件
RcppExport SEXP intVec1a(SEXP vec) {
Rcpp::NumericVector vec2(vec);
int prod = 1;
for (int i=0; i<vec2.size(); i++) {
prod *= vec2[i];
}
return (wrap(prod));
}
```

目录

编写程序包的需要

- 随着个人编写R函数的积累，零散的R脚本文件，很快不能满足函数调用的需要，人们迫切需要为R函数编写帮助文件。同时也在帮助文件中，可以更好的描述算法，参数特征，等其他注意事项。
- R程序包可以用来分享数据和函数等。
- 正如能够编写R函数，是数量掌握R基本操作的标志，能开发R程序包，是个人运用R到达一个新的阶段的标志。

编写程序包主要步骤

- 1. R函数编写
- 2. R，Rtools和MikTeX（或CTeX）的安装与配置：添加到启动路径
- 3. R包框架的生成
- 4. 填写帮助文件以及编辑Description文件
- 5. 程序包创建

需要的工具

Rtools + MikTeX
可以分别在以下网址下载
(1)Rtools:
http://cran.r-project.org/bin/windows/Rtools/
Rtools 中有生成R程序包以及检查的重要工具，内置gcc等编译器，用来编译其他计算机语言所写的函数
(2)MikTeX:
http://miktex.org/
用来编译帮助文件Rd files. 中文版本的LaTeX则可以考虑用CTeX（http://www.ctex.org/CTeXDownload/）

生成R包的框架

package.skeleton

package.skeleton(name = “mypackage”, list = ls())
(1)Read-and-delete-me
包括如何创建R包的指南,看完之后需要删除
(2)DESCRIPTION
对R包的简要介绍。
(3)r文件夹
存放的是.r文件，即各函数的源代码。
(4)man文件夹
存放的是Rd文件，也就是R帮助的源代码. Rd 文件，需要用TEX文件写成

DESCRIPTION

DESCRIPTION文件

```
Package:  KIB
Type:  Package
Title:  A package skeleton for testing
Version:  1.0
Date:  2013-12-23
Author:  Jinlong Zhang
Maintainer:  Jinlong Zhang <jinlongzhang01@gmail.com>
Description:  Package developed at Kunming Institute of
Botany
License:  GPL-2
LazyLoad:  yes
Depends:  vegan
Suggests:  ade4
```

Rd文件：R的帮助文件

(1).rd文件是帮助文件的源代码, 是用LaTeX语言书写的。填写Rd files需要对LaTeX有基本的了解。
(2)对于函数、数据，都已经生成了对应的.rd文件
注意: 其中title是必须填写的内容。而有些项是可以删除的。同时要注意：在Rd文件中，不要出现非ASCII码字符，否则在Rcmd check中将不能通过。

Rd文件需要填写的内容

似曾相识？

title{ }	标题
description{ }	函数功能
arguments{ }	函数使用方法
details{ }	处理细节
value{ }	返回值
references{ }	参考文献
author{ }	作者
examples{ }	运行实例

R程序包的编译

开始>运行>cmd>
1 编译成.tar.gz源程序包，该程序包用于检查和提交到CRAN

创建Linux source code包

Rcmd build mypackage

2 检查程序包

检查

Rcmd check mypackage

3 编译成二进制的zip包，供Windows使用

生成Windows程序包

Rcmd INSTALL −−build mypackage

目录

数据的平均取值:期望

若X为离散型随机变量，其概率分布为

$$P(X = x_k) = p_k, (k = 1, 2, ,)$$

则称

$$x_1 p_1 + x_2 p_2 + ... + x_k p_k + ... = \sum x_k p_k$$

为随机变量X的数学期望,简称期望,记为$E(X)$ 即

$$E(X) = \sum x_k p_k$$

若X为连续型随机变量，其概率密度为$f(x)$，则X的数学期望为

$$E(X) = \int_{-\infty}^{+\infty} x f(x) dx$$

期望体现了随机变量取值的“平均”，有时也称其为均值。
这里的均值，一般用μ表示。

数据的离散程度：方差

为了表示数据的离散程度，我们引入方差的概念，表示数据与其期望值距离的总和。
设X为随机向量，如果

$$E[X - E(X)]^2$$

存在，则称它为X的方差(Variance)。
并定义其平方根为标准差:

$$\sigma = \sqrt{E[X - E(X)]^2}$$

因此，方差，常用σ^2表示。

概率密度分布函数

- 为什么要了解概率密度分布？
- 不同的概率密度，在统计检验, 线性模型和广义线性模型的连接函数中, 找到对应的概率分布，是进行相应推断的或者建模的基础。
- 不同的类型的事件的发生情况或者测量的数据，符合不同的概率密度分布，在使用时，需要加以甄别。
- 例如某一段时间内发生事故的次数，符合泊松分布。
  某次考试，学生的考试成绩，或一个地区的人均寿命，符合正态分布。
  群落内个体的存活情况，常符合Weibull分布等。

生态学中常用概率密度

binom()	二项分布
norm()	正态分布
pois()	泊松分布
t()	学生t分布
f()	Fisher's F 分布
chisq()	Chi-Square 分布
unif()	均匀分布
weibull()	Weibull分布
lnorm()	对数正态Lognormal分布
beta()	Beta 分布
gamma()	Gamma 分布

其他概率密度

cauchy()	Cauchy 分布
exp()	Exponential 分布
hyper()	超几何分布Hypergeometric 分布
logis()	Logistic分布
nbinom()	Negative binomial分布
signrank()	Wilcoxon signed rank statistic分布
wilcox()	Wilcoxon rank sum分布

二项分布

如果投掷一枚硬币，其出现正面的概率为p，那么重复投掷n次后，正面为k次的概率是多少？这个仅有成功或者失败两种结局的分布就是二项分布(binomial distribution)，又称伯努利分布，为纪念瑞士数学家雅各·伯努利而命名。

$$p(k) = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}$$

$$(k = 1, 2, ,)$$

正态分布

正态分布又称高斯分布，最早由德国数学家高斯推导出来，因此得名。正态分布中在统计学中占据着非常重要的位置，在数学、工程、物理等也都有非常重要的应用。统计学中很多分布是从正态分布中推导出来，很多假设检验，需要以数据的正态分布作为前提。
正态分布常记为：

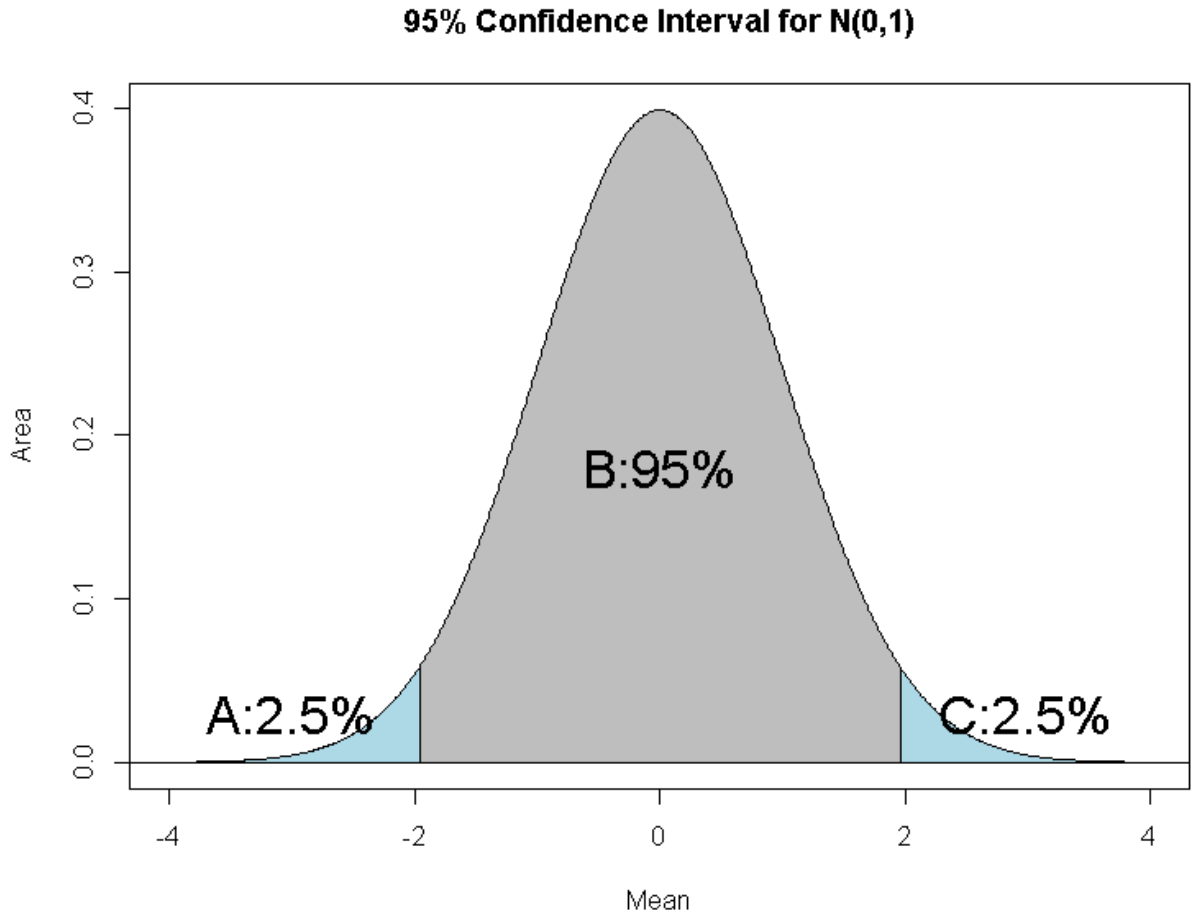
$$N(\mu, \sigma^2)$$

其概率密度的公式为

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

其中μ 为期望，在样本数据中，为均值，σ^2为方差。

标准正态分布



标准正态概率密度分布的分区，95%置信区间

正态分布的95%概率区间R代码

```
x = seq(-4,4, by = 0.01)
plot(x,dnorm(x),type="l", ylab = "Area", xlab = "Mean")
abline(h=0, col="black");
x1 = seq(-5, -1.96, 0.01)
x2 = seq(1.96, 5, 0.01)
x3 = seq(-1.96, 1.96, 0.01)
y1 = dnorm(x1)
y2 = dnorm(x2)
y3 = dnorm(x3)
polygon(c(x1, qnorm(0.025)), c(y1, 0), col = "lightblue")
polygon(c(x2, qnorm(0.975)), c(y2, 0), col = "lightblue")
polygon(c(qnorm(0.025), x3, qnorm(0.975)), c(0, y3, 0), col
= "grey")
text(x = -3, y = 0.03, "A:2.5%", cex = 2)
text(x = 0, y = 0.18, "B:95%", cex = 2)
text(x = 3, y = 0.03, "C:2.5%", cex = 2)
title("95% Confidence Interval for N(0,1)")
```

卡方分布

若$Z_1 + Z_2, Z_n$服从标准正态分布,

$$X^2 \equiv Z_1^2 + Z_2^2 + \cdots + Z_n^2$$

则，X^2遵循卡方分布，其概率密度为

$$p(x^2) = \frac{1}{2^{(n/2)}\Gamma(\frac{n}{2})}(x^2)^{(n-2)/2}e^{-x^2/2}$$

其中Γ为伽玛函数。

学生t分布

如果Z服从标准正态分布，X^2服从卡方分布，

$$t \equiv \frac{Z}{\sqrt{\frac{X^2}{n}}}$$

则t服从t分布。
t分布的概率密度为

$$p(t) = \frac{\Gamma(\frac{n+1}{2})}{\sqrt{\pi n}\Gamma(\frac{n}{2})(1+\frac{t^2}{n})^{(n+1)/2}}$$

学生t分布，主要用于小样本量，一组，或者两组数据的均值比较。
由英国的Willam S. Gosset在1908年最早推导出来，并以Student的笔名发表，故名学生t分布。

F分布

若X_1和X_2分别为自由度为n_1和n_2的卡方随机变量，则

$$F \equiv \frac{X_1^2/n_1}{X_2^2/n_2}$$

该分布遵循n_1和n_2的F分布。即F-分布的随机变量是两个卡方分布变量的比率。
其概率密度遵循

$$p(f) = \frac{\Gamma(\frac{n_1+n_2}{2})}{\Gamma(\frac{n_1}{2})\Gamma(\frac{n_2}{2})}(\frac{n_1}{n_2})^{n_1/2} f^{(n_1-2)/2}(1+\frac{n_1}{n_2}f)^{-(n_1+n_2)/2}$$

F分布可用于ANOVA，也用于似然比率检验(Likelihood Ratio Test, LRT)。

回顾随机数的几个函数

runif()	生成均匀分布的随机数
dunif()	均匀分布的概率密度函数
punif()	均匀分布的累计概率
qunif()	均匀分布的分位数
rnorm()	生成正态分布的随机数
dnorm()	正态分布的概率密度
pnorm()	正态分布的累计概率
qnorm()	正态分布的分位数

随机数生成的控制set.seed()

- 随机数之所以重要，是因为其在数值计算，最优化方法，randomization，蒙特卡罗方法，贝叶斯推断等多方面，都有非常广泛的应用。
- 目前为止，计算机程序上可以生成的随机数，大多为伪随机数。由于计算机运算数位的限制，生成随机数的种子的取值等，并不能令所得的随机数序列完全随机，而是极大限度的接近随机。
- 生成随机数，一般需要从一个随机数种子开始，如计算机系统的时间，再通过一些公式，既可以获得。
- 因此，如果要生成一系列随机数，并且，让这一个序列的随机数，在不同场合都完全一致，则需要为随机数的种子赋值。
- set.seed()设置随机数种子

目录

应用于向量的基本统计函数

mean(x)	平均值
median(x)	中位数
var(x)	方差
sd(x)	标准差
cov(x,y)	协方差
cor(x,y)	相关系数
min(x)	最小值
max(x)	最大值
range(x)	极差
quantile(x)	分位数

信息的汇总：summary

summary()

summary为泛型函数，针对对象的不同类型，可以打印出不同的结果。
如lm，则打印出各因子的系数，以及显著性，
若为dataframe，则给出平均值，中位数等。

summary(runif(50))

summary输出线性模型等的结果等。

两组数的均值比较:t检验或Wilcoxon检验

两组数之间的均值比较，一般用t检验

t.test()

t检验需要满足的条件：

- 独立性：各数据之间完全独立
- 正态性：数据取自一个正态的总体，可以用shapiro.test
- 方差齐性：两组的方差不能有显著性差异，可以用bartlett.test或levene.test{lawstat }

r中的t检验P值经过Welch修正，对方差齐性的要求降低。

在数据无法满足上述要求时，要用非参数的方法：Wilcoxon rank-sum test

wilcox.test()

非参数检验无需满足上述要求，但敏感度可能较t检验稍差。两个样本均值的非参数比较，又称Mann-Whitney检验。

检验正态性的两种方法

1 检验正态性的两种方法Shapiro-Wilk检验
shapiro.test(rnorm(100, mean = 5, sd = 3))
2 卡方检验方法QQ plot Quantile against quantile 利用数据分位数的值，和理论值进行比较
chisq.test()
并不仅仅有这两种方法!
检验学生成绩是否符合正态分布

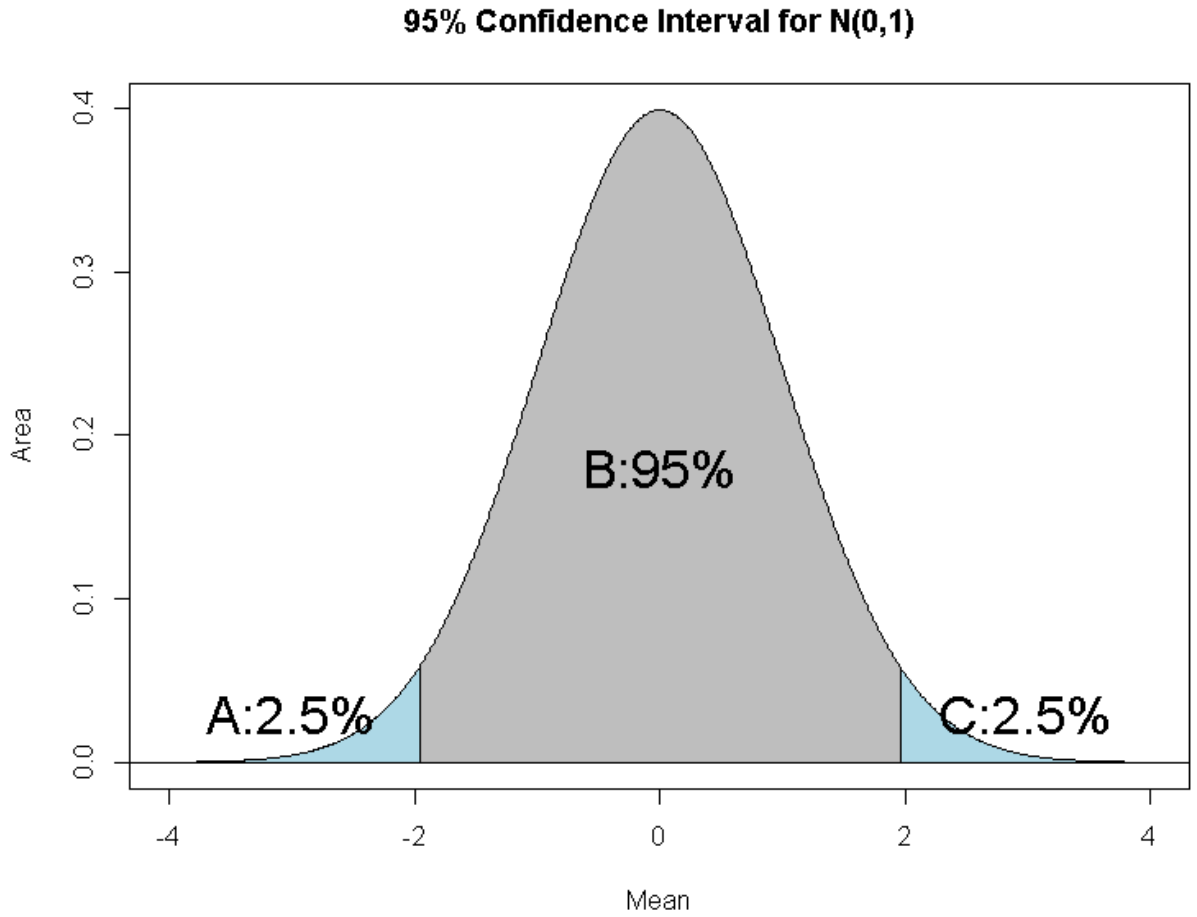
卡方检验正态性

```
X <- c( 25, 45, 50, 54, 55, 61, 64, 68, 72, 75, 75,
78, 79, 81, 83, 84, 84, 84, 85, 86, 86, 86,
87, 89, 89, 89, 90, 91, 91, 92, 100)
A<-table(cut(X, br=c(0,69,79,89,100)))
p<-pnorm(c(70,80,90,100), mean(X), sd(X))
p<-c(p[1], p[2]-p[1], p[3]-p[2], 1-p[3])
chisq.test(A,p=p)
```

P值

- P值是给定当前样本数据的前提下，零假设发生的概率。
  例如，通过计算，某个统计量位于概率分布的0.025或者0.975的区间之外，就是意味着，这件事发生的可能性非常小。
- 该事件为小概率事件，就可以在0.05的显著水平上否定零假设。
  可以通过假设数据符合某概率分布进行计算，也可以通过事件发生的频率进行计算。

标准正态分布



标准正态概率密度分布的分区，95%置信区间

Type I and Type II Error

- 若零假设事实上成立，但是统计检验结果不支持零假设，称为第一类错误。
- 若零假设事实上不成立，但是统计结果支持零假设，称为第二类错误。
- 这两类错误的发生，和数据本身，以及统计方法的敏感性密切相关。

F检验

F检验是为纪念著名统计学家Fisher博士而命名。Fisher在1920年代发明了F检验。
F检验中，计算F统计量，必须满足F分布，也就是意味着，必须要满足正态性假设。
举例：X, Y两组数据，平均值有无差别？

F检验举例

```
var.test()
X<-c(78.1,72.4,76.2,74.3,77.4,78.4,76.0,75.5,76.7,77.3)
Y<-c(79.1,81.0,77.3,79.1,80.0,79.1,79.1,77.3,80.2,82.1)
var.test(X,Y)
```

相关性检验

1. Pearson's correlation：观测值之间独立，并取自一个正态的总体
2. Kendall's tau：非参数方法
3. Spearman's rho：非参数方法

cor和cor.test

```
cor(x, y = NULL, use = "everything",
method = c("pearson", "kendall", "spearman"))
cor.test(x, y,
alternative = c("two.sided", "less", "greater"),
method = c("pearson", "kendall", "spearman"),
exact = NULL, conf.level = 0.95, continuity = FALSE, ...)
```

Kendall's tau和Spearman's rho的区别，参见吴喜之教授的《统计学:从数据到结论》

拟合优度检验Goodness-of-Fit Test

chisq.test()
常用于计算某数据是否符合统计概率
如孟德尔豌豆实验的数据，是否符合分离或者自由组合概率分布。
大麦的杂交后代芒性状的比例无芒：长芒：短芒=9：3：4,而实际观测值为335：125：160 ,检验观测值是否符合理论假设？
命令：
chisq.test(c(335, 125, 160), p=c(9,3,4)/16)

练习2，绘制正态分布概率密度曲线

绘制正态分布概率密度曲线

目录

线性模型

$$y_i = \alpha + \beta x_i + \varepsilon_i$$

为什么称为线性模型？因为在坐标系中，该函数的图像为一条直线。

lm()	建立线性模型
summary()	查看结果
aov()	方差分析
abline()	添加回归拟合直线
predict()	预测给定x的y值
fitted()	对于x处y的理论值
resid()	残差，没有能够拟合的部分

残差是否正态分布，是确定模型拟合度好坏的重要标志。

若有多个变量，只是在高维空间中的一条直线。

线性模型输出结果I

表示调用的命令：参见公式部分**Call:**
lm(formula = weight ~ group - 1)
残差
Residuals:
各变量的系数以及统计性检验
Coefficients:
Estimate Std. Error t value Pr(¿—t—)
groupCtl 5.0320 0.2202 22.85 9.55e-15 *
groupTrt 4.6610 0.2202 21.16 3.62e-14 ***
—**

线性模型输出结果II

显著性标签
Signif. codes: 0 ‘*’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘’ 1**
模型总体的显著性检验残差**Residual standard error: 0.6964 on 18 degrees of freedom**
方差解释率**Multiple R-squared: 0.9818,** 调整后的方差解释率**Adjusted R-squared: 0.9798**
F统计量模型是否显著。**F-statistic: 485.1 on 2 and 18 DF, p-value: ¡ 2.2e-16**

线性模型之一：方差分析

方差分析是在线性模型中，分析各变量对响应变量y的影响大小，。
“原理为：把因变量的值随着自变量的不同取值而得到的变化进行分解，使得每一个自变量都有一份贡献，最后剩下无法用已知的原因解释的则看成随机误差的贡献。然后用各自变量的贡献和随机误差的贡献进行比较（F检验），以判断该自变量的不同水平是否对因变量的变化有显著贡献。输出就是F-值和检验的一些p-值。”
吴喜之教授《统计学：从数据到结论》

anova()
aov()
summary()

R的公式：向量之间的基本关系

+	各变量独立作用，不考虑交互
:	只考虑变量间的交互作用
*	考虑变量的独立作用和交互作用
-	去掉某一部分
I()	括号中可放数学表达式

R公式举例

a+b	a 和b独立的作用
a:b	a 和b 的交互作用
a*b	相加和交互作用(等价于a+b+a:b)
poly(a, n)	a的n价多项式
y∼x-1	表示过原点的线性回归
y∼1	拟合截距

注意：不要在公式内进行数据下标操作，更不要用中文为变量命名，而让中文出现在公式中。这些都将导致不能预知的错误。

TukeyHSD多重比较

若方差分析结果为显著，人们往往还想知道各组数据之间，是否差异显著。TukeyHSD即可做多重比较。

```
TukeyHSD()
fit <- aov(len~ supp+factor(dose), data=ToothGrowth)
# TukeyHSD 仅适用于各组观测值相同的情况
TukeyHSD(fit) # For all terms
TukeyHSD(fit, "factor(dose)") # 仅显示dose
plot(TukeyHSD(fit, "factor(dose)")) # Plot method for T
```

agricolae程序包提供了多重比较的若干种方法，如LSE，LSD等, 并且可以标注多重比较显著性的ABCD等。

模型诊断

建立模型后，需要对模型进行判断，看是否合理。模型诊断用到的函数主要包括：

fitted.values()	给定变量值后的模型估计值
residuals()	残差
rstandard()	标准化之后的残差
rstudent()	标准化之后的残差
qqnorm()	正态分位数QQ图
qqline()	正态理论预测线
plot.lm()	绘制线性模型的诊断结果

线性回归

lm的应用

```
set.seed(12345)
x = 1:40 + 3*rnorm(40)
y = 1:40 + rnorm(40)
dat <- data.frame(x, y)
plot(y ~ x, data = dat)
lmodel <- lm(y ~ x)
summary(lmodel)
abline(lmodel, col = 2)
text(10, 35, "Adjusted expression(R^2) = 0.9258")
```

Adjusted R square

随着变量数的增多，R方上升的很快，其实这并不一定和模型本身有关联。
为了降低这一影响，统计学家提出了Adjusted R square。
Adjusted R square甚至可以是负数
Adjusted R square 和R square关系如下

$$\bar{R}^2 = 1 - (1 - R^2)\frac{n-1}{n-p-1}$$

其中p为变量的数目，n为样本数

AIC: Akaike Information Criterion

日本统计学家赤池弘次在1971年提出赤池系数用来权衡所估计模型的复杂度和该模型拟合数据的优良性。
寻找可以最好地解释数据但包含最少参数的模型。
假设条件为模型的误差服从独立正态分布。
让n为观察数，RSS为剩余平方和，那么AIC变为：

$$AIC = 2k + nln(RSS/n)$$

更一般的AIC,

$$AIC = 2k - 2ln(LogLik)$$

多个变量的逐步回归举例

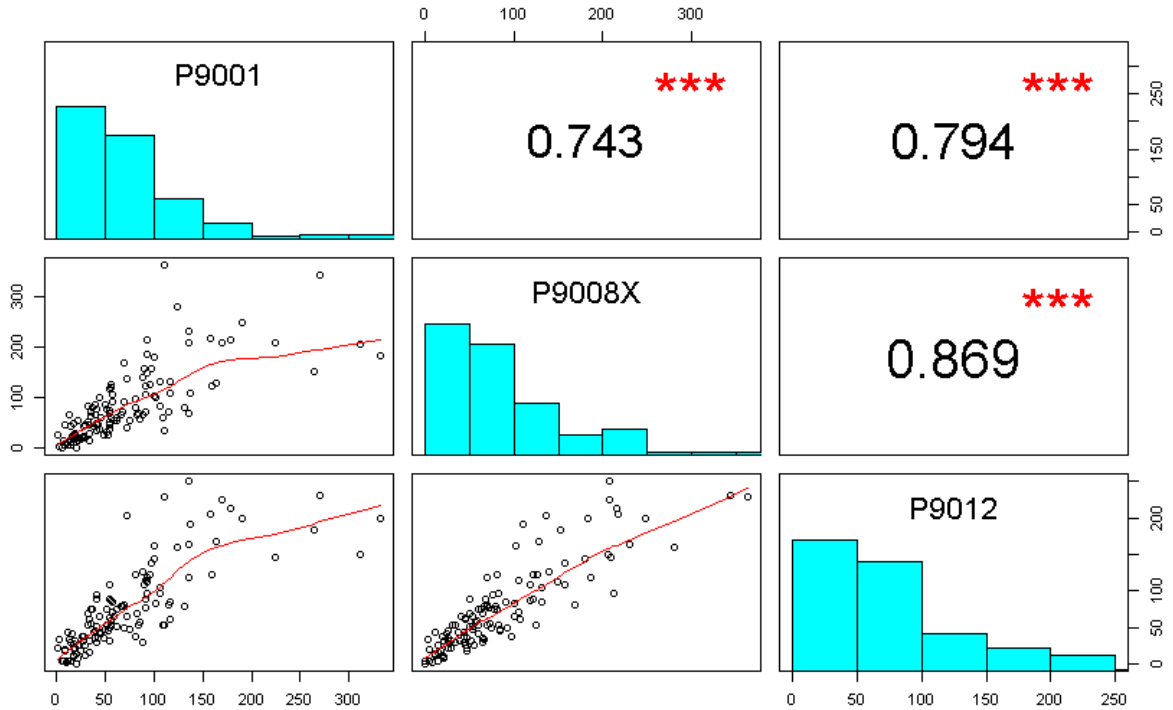
在变量较多的情况下，常常需要将影响不显著的变量剔除出模型，此时就涉及到多个模型比较的问题。

逐步回归举例

```
cement<-data.frame(
X1=c( 7, 1, 11, 11, 7, 11, 3, 1, 2, 21, 1, 11, 10),
X2=c(26, 29, 56, 31, 52, 55, 71, 31, 54, 47, 40, 66, 68),
X3=c( 6, 15, 8, 8, 6, 9, 17, 22, 18, 4, 23, 9, 8),
X4=c(60, 52, 20, 47, 33, 22, 6, 44, 22, 26, 34, 12, 12),
Y =c(78.5, 74.3, 104.3, 87.6, 95.9, 109.2, 102.7, 72.5,
93.1,115.9, 83.8, 113.3, 109.4))
lm.sol<-lm(Y ~ X1+X2+X3+X4, data=cement)
summary(lm.sol)
lm.step<-step(lm.sol)
summary(lm.step)
```

pairs查看多个变量的分布

pairs() 用来查看多元变量
从pairs进一步衍生出诸如hydropairs



hydroTSM的hydropairs

图中的趋势线，用lowess()函数获得

多项式回归

多项式回归：拟合二次或高次曲线？

```
y <- c(8,6,4,3,2,2,3.2,5,7,9)
x <- 1:10
res <- lm(y ~ x+I(x^2))
a <- seq(min(x),max(x),len=100)
```

在X的区间内均匀生成100个点

```
b <- predict(res,data.frame(x=a))
```

预测各x点处y的值

```
plot(y ~ x)
lines(a, b)
```

lines函数用来连接各个点

练习3

1. 运行逐步线性回归的R代码
2. 如何筛选到最优的模型？

谢谢！

敬请批评指正！