

统计软件应用录屏解说

万苓韵 16307110430

目 录

一、Excel 三维气泡图.....	(2)
二、R 创意图示.....	(4)
三、自编函数：用 R 写诗词.....	(9)
四、函数跟踪口述：conf.limits.nc.chisq().....	(12)
五、统计模型几何解说：大数定律及中心极限定理.....	(16)
六、Netlogo 合作模型复现.....	(20)

一、Excel 三维气泡图制作

1 关于气泡图

- 气泡图可用于展示三个变量之间的关系；与散点图类似，绘制时将一个变量放在横轴，另一个变量放在纵轴；不同之处在于：气泡图允许在图表中额外加入一个表示气泡大小的变量，即第三个变量
- 所用数据：第一列中列出 x 值，在相邻列中列出相应的 y 值和气泡大小的值
- 三维气泡图：深度，近大远小

2 制作思路

- 生成随机(x, y, z)数组
- 运用三维旋转矩阵得到旋转后的变量值
- 对 z 值进行转换，用于表示气泡面积
- 利用数据画出气泡图，通过滚动条调节旋转角度

3 操作步骤

1) 生成正态分布伪随机数：

- 运用公式 = NORMINV (RAND(), 0, 1) ##平均值为 0，标准差为 1 的正态分布
- 善用 Ctrl + R 及 Ctrl + D，生成 20 组三维数组

2) 创建旋转矩阵：

- 弧度制转换公式：= degree/180 * PI()
- 三维旋转矩阵公式：

$$\text{绕 z 轴旋转: } \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{绕 x 轴旋转: } \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix}$$

$$\text{绕 y 轴旋转: } \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix}$$

- 矩阵的相乘：= MMULT (array1, array2) ##三个矩阵相乘通过两次 MMULT 实现
按 Ctrl + Shift + Enter 键得到结果矩阵

3) 进一步处理 z 值：

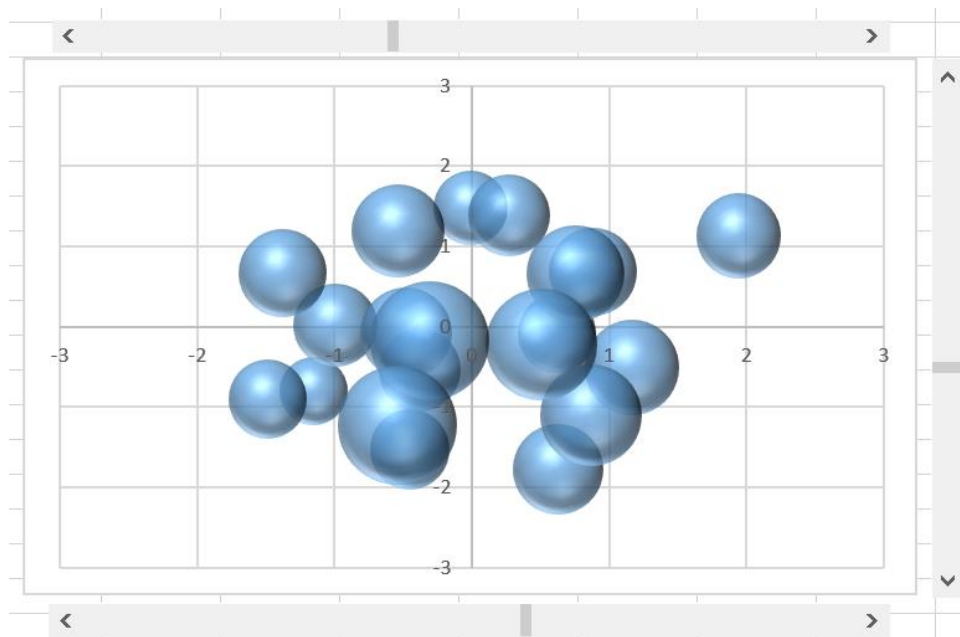
- 将 z 值按从小到大的顺序排序：= SMALL (array, k)
##返回给定区域内第 k 小的数值，此处 k 使用公式 ROW() -1；
选择区域时注意加\$符号固定

- 检索从小到大排序的每一个 z 值对应的原行数：
`= MATCH (lookup_value, lookup_array, 0)` **##要匹配的值，原数值序列，精准匹配**
- 借助 z 值对应行数得到按 z 值大小排列的三维数组数据：
`= INDEX (array, row_num, column_num)`
##返回区域内对应行、对应列的数值；column_num 通过公式 COLUMN()实现
- 将 z 值转换为气泡面积数值：
 - ✓ 需要达到的效果：近大远小
 - ✓ 依据：人眼成像（凸透镜成像规律，倒立缩小的实像，两倍焦距以外）
 - ✓ 若气泡本身大小为 1，则物距为 u 的气泡大小为 $(f/(u-f))^2$
 假设气泡落在 z 轴上 $(-3, 3)$ 范围内，焦距为 3，人眼位于 z 轴+9 处
 面积变换公式： $= (3/(6-z))^2$

4) 气泡图绘制及滚动条控制：

- 选中排序、转换后的 x, y, z 值，插入三维气泡图
- 在开发工具中选择创建滚动条，设置控件格式，规定变化范围为 0 至 360，并将控件链接至对应的旋转角度，拖动滚动条即可控制旋转

5) 效果图：



4 一些有意思的点

- 增加了一个绕 y 轴的旋转维度，实现在三个平面内的旋转
- 对比乱序、按气泡面积大小排序的数据，画出的气泡图在遮挡层次上有差别
##对 z 值按大小进行排序的意义：大气泡放在更上层，符合远近的规律
- z 值（气泡面积值）按比例变化，三维气泡图视觉效果不变
##气泡是基于面积的相对比例呈现的

二、R 画图

demo(graphics)

demo(persp)

##画图示例

1 基本函数及参数

基本画图函数：

plot()

lines() 在图上加线

abline() 在图上加直线

points() 在图上加点

pie() 饼图

barplot() 柱状图

hist() 直方图

mosaicplot() 马赛克图

boxplot()/vioplot() 箱型图/小提琴图

cdplot() 条件密度图

scatterplot3d() 三维散点图

symbols() 气泡图

等

plot()函数的参数们：

asp = y 轴/x 轴的数值长度比例

type = "b" / "h" / "l" / "p" 数据呈现方式：b 点+线，h 垂直线段，l 线，p 点

lty = 线的类型

pch = 点的类型

lwd = 线的粗细

cex = 点的大小

col = 颜色

main = " " 标题

sub = " " 副标题

cex.axis = 坐标轴数字大小

font.axis = 坐标轴数字字体

xlab / ylab = " " 横纵轴标题

xlim / ylim = c(a,b) 横纵轴数值范围

xaxt / yaxt = "n" 禁用横纵坐标轴刻度线

2 整体图示

```
library(vcd)
attach(Arthritis)
##载入数据
str(Arthritis)
##查看数据: ID 及 Age 为连续变量; Treatment 及 Sex 为 2 水平分类变量, Improved
##为 3 水平分类变量
plot(Arthritis)
##各变量关系散点图

library(car)
scatterplotMatrix(Arthritis, diagonal = "h")
##散点图矩阵
scatterplotMatrix(~ID+Treatment+Age+Improved| Sex)
##不同性别下其他变量的散点图矩阵

Arthritis[,1:5]<-lapply(Arthritis[,1:5],as.numeric)
##把变量类型转换为数值, lapply 对数据集中每个元素实施这一操作并返回列表
str(Arthritis)
plot(as.data.frame(jitter(as.matrix(Arthritis[,2:3]),factor = 0.1)))
plot(as.data.frame(jitter(as.matrix(Arthritis),factor = 0.1)))
##加抖动, factor 设置不同程度
```

3 单个分类变量

```
pie(table(Improved), labels = table(Improved),clockwise = F)
pie(table(Improved),col = c("white","yellow","brown"),labels = table(Improved))
pie(table(Improved),col = rainbow(3),labels = table(Improved))
##table()转换成列联表, 饼图

barplot(table(Improved),legend.text = table(Improved),col = rainbow(3),horiz = T)
##柱状图

mosaicplot(table(Improved))
##马赛克图
```

4 单个连续变量

```
hist(Age)
##直方图
plot(density(Age),ylim = c(0,0.05))
##密度图
```

```

points(jitter(Age),Age*0,pch='|')
  ##加扰动
hist(Age,add = T,probability = T)
  ##加概率直方图

plot(quantile(Age, probs = (p <- seq(0,1,1e-3)) ),p,pch = ".")
  ##各百分位数处的值
points(Age,(rank(Age)-0.5)/length(Age))
plot(qnorm((rank(Age)-0.5)/length(Age)),Age)
  ##正态分布下各百分位处的值与实际值对比
qqnorm(Age);qqline(Age)
  ##QQ 图

```

5 两个分类变量

```

mosaicplot(Treatment~Improved)
mosaicplot(~Treatment+Sex+Improved)
  ##马赛克图

```

6 两个连续变量

```

layout(mat = rbind(c(1,2),c(3,4)))
  ##分割画面

plot(ID,Age,type = "p",pch = 21, col = "red", xaxt = "n", yaxt = "n", lty = 3, ann = F)
  ##散点图
title("Age~ID")
  ##加标题
axis(3,at = ID, labels = ID, col.axis = "red", las = 0)
axis(2,at = Age, labels = Age, col.axis = "red", las = 0)
  ##画坐标轴，las 设置坐标轴文字方向
mtext("Age~ID",side = 4,line = 1,las = 2,col = "blue")
  ##添加文字

plot(Age~ID)
abline(lm(Age~ID), col = "blue", lty = 2)
  ##加线性回归线

plot(Age~ID)
lines(lowess(Age~ID))
  ##加 lowess 回归曲线

```

```
plot(Age~ID,type = "p", pch = "△",col = ifelse(Age>ID,"orange","brown"))
  ##散点图，按条件上色
layout(1)
  ##复原画面

fit <- lm(ID~Age,data = Arthritis)
plot(fit)
plot(cooks.distance(fit))
  ##回归及库克距离图示

plot(Age~ID,type = "p",col = "red",lty = 2, pch = 2, lwd = 2, main = "Age~ID", sub = "Arthritis", xlab
= "ID", ylab = "Age", xlim = c(0,85), ylim = c(20,80),asp = 1.5)
boxplot(ID,horizontal = T,add = T, at = 20)
boxplot(Age,horizontal = F,add = T, at = -5)
library(vioplot)
vioplot(ID,horizontal = T,add = T,at = 80)
vioplot(Age,horizontal = F,add = T,at = 90)
  ##箱型图和小提琴图， horizontal 横纵向， add 添加至原图， at 位置
```

7 分类变量+连续变量

用分类解释连续:

```
plot(Age~Sex)
boxplot(Age~Sex,outline = T,outpch = 10)
  ##箱型图，离群值处理

library(sm)
sm.density.compare(Age,Sex)
  ##不同 Sex 水平下 Age 的密度图
legend("topleft",inset = 0.05, title = "Sex", c("Male","Female"),lty = c(1,2),pch = c("*","#"),col =
c("red","green"),cex = 0.8)
  ##在左上角加标签，inset 设置偏离最角落的百分比
```

用连续解释分类:

```
cdplot(Improved ~ Age, col = c("pink","orange","white"))
  ##条件密度图，logistic 回归的结果
points(Improved ~ Age,col = "blue")

plot(Age,Improved)
text(Age,Improved,Age,pos = 4,cex = 0.7,col = "blue")
  ##加数据标签，pos 设置加在点的哪一边
```

8 其他图示

```
library(scatterplot3d)
```

```
scatterplot3d(ID, Age, Improved, type = "h", highlight.3d = T)
```

##三维散点图，highlight.3d 以涂色深浅表示 z 的远近

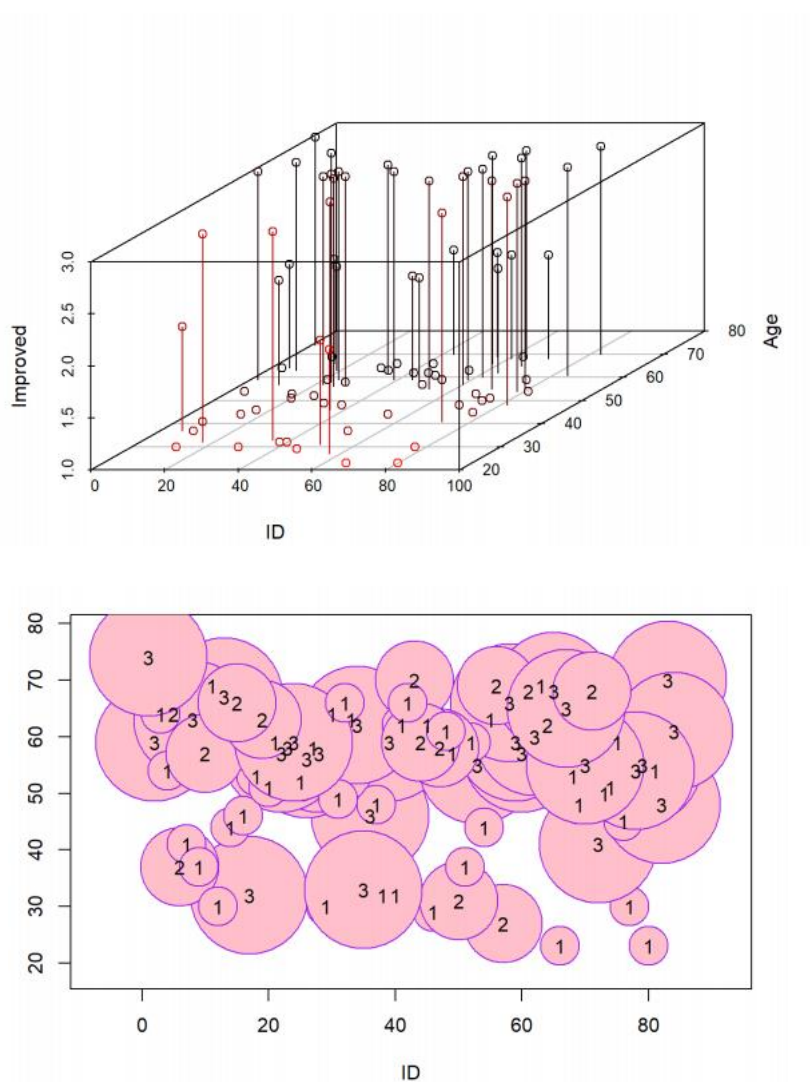
```
symbols(ID, Age, Improved, inches = 0.5, fg = "purple", bg = "pink")
```

```
text(ID, Age, labels = as.numeric(Improved), cex = 1)
```

##气泡图，z 值的大小决定气泡大小，fg/bg 为气泡轮廓/填充颜色，text 加标签

```
detach(Arthritis)
```

结果节选：（完整请见录屏链接的结果 PDF）



三、自编函数：用 R 写诗词

1 基本介绍

想法来源：

用编程来创作诗词文章乃至歌曲是很有意思的事，现在也有很多方法来实现，尤其在深度学习方面有很多发展。不过这次的诗词自编函数还是基于一个比较简单的逻辑，主要是想试用一下 R 里面的 `jiebaR` 包（功能包括分词、统计词频、词性等等）。“创作”诗词所采用的方式参照了知乎：<https://zhuanlan.zhihu.com/p/25446637>；而文本库素材来自 GitHub 上的一个古典中文语料库（繁体）：<https://github.com/rime-aca/corpus>。

编写逻辑：

- 通过文本库来确定诗词里常出现的词汇及意象
 - ✓ 导入文本库，利用分词功能拆分成具体词汇并统计词频
 - ✓ 词云的可视化展示
- 给出想要创作的诗词体例示范，再用文本库的内容进行结构内的替换
 - ✓ 对范例进行分词及词性分析
 - ✓ 以文本库中同词性、同词长的类似词汇作为素材，随机替换
 - ✓ 按原结构补充标点

实现内容：

自由选择文本库及体例示范，进行模仿创作。

2 代码解读

```
library(jiebaR)
```

```
library(wordcloud2)
```

```
sucai <- "C:\\Users\\mylos\\Desktop\\复旦\\课程\\心理\\统计软件应用\\函数\\corpus-master\\宋词三百首.txt"
```

```
sucai <- "C:\\Users\\mylos\\Desktop\\复旦\\课程\\心理\\统计软件应用\\函数\\corpus-master\\唐诗三百首.txt"
```

```
fanli <- "画堂晨起，来报雪花坠。高卷帘栊看佳瑞，皓色远迷庭砌。盛气光引炉烟，素草寒生玉佩。应是天仙狂醉，乱把白云揉碎。"
```

```
fanli <- "桂魄初生秋露微，轻罗已薄未更衣。银箏夜久殷勤弄，心怯空房不忍归。"
```

```
shici <- function(sucai, fanli){ ## 需给出素材（文件位置）和范例
```

```
  sc <- scan(sucai, sep = "\n", what = "", encoding = "UTF-8") ## 导入文本库，编码方式是便于使用中文的 UTF-8
```

```
  wk = worker() ## jiebaR 中的分词引擎
```

```
  analysis <- as.data.frame(table(wk[sc])) ## 处理文件格式
```

```
  analysis <- analysis[order(-analysis$Freq),] ## 根据词频重新排序
```

```

names(analysis)<-c("word","freq") ## 更改变量名
analysis$word <- as.character(analysis$word) ## 转换为字符型数据
wordcloud2(analysis) ## 做词云，观察是否有异常
wordcloud2(analysis[analysis$freq>1 & analysis$freq < 300 & nchar(analysis$word) == 1,]) ##
对合适频率的词长为 1-3 的词分别做词云
wordcloud2(analysis[analysis$freq>1 & analysis$freq < 300 & nchar(analysis$word) == 2,])
wordcloud2(analysis[analysis$freq>1 & analysis$freq < 300 & nchar(analysis$word) == 3,])

tagger <- worker("tag") ## jiebaR 中分析词性的工具
fanli_1 <- tagger <= fanli
cixing <- attributes(fanli_1)$names ## 提取范例的词性
example <- subset(analysis,freq>1 & nchar(word)<=3 & freq<300) ## 在文本库中找出频率合
适、词长不大于 3 的词
example_2 <- tagger <= example$word ## 进行词性分析

empty <- ""
for (i in 1:length(fanli_1)){ ## 对范例中的每个词进行同等词性、词长的替换
  temp_file <- example_2[attributes(example_2)$name == cixing[i]] ## 筛选同词性
  temp_file <- temp_file[nchar(temp_file) == nchar(fanli_1[i])] ## 筛选同词长
  empty <- paste0(empty, sample(temp_file,1)) ## 每个词对应抓取一个替换词，无间隔拼
接
}

fanli_2 <- substring(fanli, 1:nchar(fanli), 1:nchar(fanli)) ## 把范例字符串分割为单个字符串
result <- empty
for (i in 1:length(fanli_2)){ ## 通过结果的分割、再拼接加入对应位置的标点
  if (fanli_2[i] == "，"){
    result <- paste0(substr(result,1,i-1),"，",substr(result,i,nchar(result)))
  }
  if (fanli_2[i] == "。"){
    result <- paste0(substr(result,1,i-1),"。",substr(result,i,nchar(result)))
  }
}
print(result) ## 输出结果
}

shici(sucaifanli)

```

输出结果示例：

[1] "故鄉雲淡，一簾枝頭叩。白晝堂囀石州慢，春衫畫簾半轉。紅情彭元遜照，中酒殘醉言

語。歌罷數枝淚滿，卻自梅子知道。"

[1]"空相舊香，相呼雨聲收。高前事燼最關情，人歸凌波路簇。無端江神子渾，漏永春山秋千。人去霓裳前度，再自陰晴飛來。"

[1]"三絃淒切，驚秋眉峰認。遲紗窗囀最關情，玉殿洞仙歌轉。疏影濃如酒酸，朱戶看盡朱顏。紅萼豆蔻夢短，竟憑朱顏分明。"

[1]"輕寒新綠日日百，畫簾半嫩終重到。煙村芳酒庾信舞，中酒多情夢見謾。"

[1]"暗解微寒後期階，開朱戶好忽無計。清景晝永楊柳咽，過盡音信濃睡怕。"

3 注意点及其它

- 实际上可以不局限于诗词，只要有适合的文本库和范例就可以（也可以混搭），但关于加标点的算法主要还是针对有特定体制规则的文本。相比参考内容中完全框定某种格式而言已经做了改进，但还是不够自由。
- 词长的限制需要格外注意，有时对范例的分词可能会有没怎么分开的情况，就会出现找不到对应词而报错 NULL 的状况。需要根据所用的文本库和范例特点进行调整，也可以对 jiebaR 的词典进行一些编辑。
- Debug 真的很难！找分词结果和词长限制不合适的 bug 找了很久，加标点的部分也比较困难。本来想用 supply 之类的函数实现一次写出 n 首，但总是会有奇怪的错误，暂时还是没能实现。
- 最初想用 jiebaR 做的是骗人算命测试，类似于输入名字或昵称出来一些描述或一句话一首歌，也就是网上常有的实际规则不明可能是随机的小测试。但最后因为没能想出一个比较有意思的规则而搁浅了，就先做了这个诗词的函数。
- **一点补充：**发现一个新问题，对于唐诗而言还有更复杂的押韵，可能需要通过文本库的进一步筛选匹配来实现。

四、函数跟踪：conf.limits.nc.chisq()

1 代码解读

函数来自 MBESS 包

补充一下非中心参数 **ncp** 的解释：用于和中心分布的区分，代表检验统计量的均值偏离在原假设为真情况下的均值的程度

可通俗理解为与原分布偏离了几个标准差；**ncp** 变化，**p** 值随之改变

```
function (Chi.Square = NULL, conf.level = 0.95, df = NULL, alpha.lower = NULL,
         alpha.upper = NULL, tol = 1e-09, Jumping.Prop = 0.1) {
```

判断输入的参数是否正确

```
if (Jumping.Prop <= 0 | Jumping.Prop >= 1) ## Jumping.Prop 需在 0,1 之间
  stop("The Jumping Proportion ('Jumping.Prop') must be between zero and one.")
```

```
if (is.null(Chi.Square))
  stop("Your 'Chi.Square' is not correctly specified.")
```

```
if (Chi.Square < 0) ## 需给出卡方值且不小于 0
  stop("Your 'Chi.Square' is not correctly specified.")
```

```
if (is.null(df))
  stop("You must specify the degrees of freedom ('df').")
```

if (is.null(alpha.lower) & is.null(alpha.upper) & is.null(conf.level)) ## 不可以不设定置信极限

```
  stop("You need to specify the confidence interval parameters.")
```

```
if (!(is.null(alpha.lower) | is.null(alpha.upper)) & !is.null(conf.level))
  stop("You must specify only one method of defining the confidence limits.")
```

不可以同时用两种方法设定置信极限（注意即使不用 **conf.level** 也需另外设置为 **NULL**）

```
if (is.null(conf.level)) {
  if (alpha.lower < 0 | alpha.upper < 0)
    stop("The upper and lower confidence limits must be larger than 0.")
}
```

```
if (!is.null(conf.level)) {
  if (conf.level >= 1 | conf.level <= 0)
    stop("Your confidence level ('conf.level') must be between 0 and 1.")
```

alpha.lower <- alpha.upper <- (1 - conf.level)/2 ## 将 **conf.level** 转化为 **alpha.lower** 及 **alpha.upper** 以进行计算

```
}
```

```
if (alpha.lower == 0)
```

```
  LL <- 0
```

```
if (alpha.upper == 0)
```

```
  UL <- Inf
```

```

FAILED <- NULL
if (alpha.lower > 0) {
  LL.0 <- 0.01  ## 设定初始 ncp 值
  Diff <- pchisq(q = Chi.Square, df = df, ncp = LL.0) -
    (1 - alpha.lower)  ## 在该 ncp 值下与我们所希望对应的 p 值的差异
  if (pchisq(q = Chi.Square, df = df, ncp = LL.0) < (1 - alpha.lower)) {
    FAILED <- if (pchisq(q = Chi.Square, df = df, ncp = 0) < 1 - alpha.lower)
      LL.0 <- 1e-08
    if (pchisq(q = Chi.Square, df = df, ncp = LL.0) < 1 - alpha.lower)
      FAILED <- TRUE  ## ncp 较小时 p 值较大，若仍无法达到 1-alpha.lower 的标准，则
    fail 并给出 warning，后续直接把 LL 设定为 0
    if (FAILED == TRUE)
      warning("The size of the effect combined with the degrees of freedom is too small to
        determine a lower confidence limit for the 'alpha.lower' (or the (1/2)(1-'conf.level') symmetric)
        value specified (set to zero).", call. = FALSE)
  }
  if (is.null(FAILED)) {
    LL.1 <- LL.2 <- LL.0
    while (Diff > tol) {  ## 迭代循环，ncp 变大，p 变小，Diff 变小，直至小于 tol
      LL.2 <- LL.1 * (1 + Jumping.Prop)
      Diff <- pchisq(q = Chi.Square, df = df, ncp = LL.2) - (1 - alpha.lower)
      LL.1 <- LL.2
    }
    LL.1 <- LL.2 / (1 + Jumping.Prop)  ## 此时 LL.1 刚好使 Diff > tol，LL.2 刚好使 Diff < tol
    LL.Bounds <- c(LL.1, (LL.1 + LL.2) / 2, LL.2)
    Diff <- pchisq(q = Chi.Square, df = df, ncp = LL.Bounds[2]) - (1 - alpha.lower)
    while (abs(Diff) > tol) {  ## 通过取中间值的方式（二分法）不断缩小使 Diff 刚好可以
    小于 tol 的范围，迭代收敛
      Diff.1 <- pchisq(q = Chi.Square, df = df, ncp = LL.Bounds[1]) - (1 - alpha.lower) > tol
      Diff.2 <- pchisq(q = Chi.Square, df = df, ncp = LL.Bounds[2]) - (1 - alpha.lower) > tol
      Diff.3 <- pchisq(q = Chi.Square, df = df, ncp = LL.Bounds[3]) - (1 - alpha.lower) > tol  ##
      总是更大的 ncp 更容易得到 false 的结果，必然在 1 和 2，或 2 和 3 的区间之一发生变化
      if (Diff.1 == TRUE & Diff.2 == TRUE & Diff.3 == FALSE) {
        LL.Bounds <- c(LL.Bounds[2], (LL.Bounds[2] + LL.Bounds[3]) / 2, LL.Bounds[3])
      }
      if (Diff.1 == TRUE & Diff.2 == FALSE & Diff.3 == FALSE) {
        LL.Bounds <- c(LL.Bounds[1], (LL.Bounds[1] + LL.Bounds[2]) / 2, LL.Bounds[2])
      }  ## 判断在哪个区间发生变化，再把此区间继续二分，不断收敛
      Diff <- pchisq(q = Chi.Square, df = df, ncp = LL.Bounds[2]) - (1 - alpha.lower)
    }
  }
}

```

```

    LL <- LL.Bounds[2] ## 循环收敛至 tol 的范围内后，赋中间值
  }
}
if (!is.null(FAILED))
  LL <- 0
if (alpha.upper > 0) { ## 与设定 alpha.lower 的逻辑类似，注意此处直接用 alpha.upper 而
  不用 1 减
    FAILED.Up <- NULL
    UL.0 <- LL + 0.01
    Diff <- pchisq(q = Chi.Square, df = df, ncp = UL.0) - alpha.upper
    if (Diff < 0)
      UL.0 <- 1e-08
    Diff <- pchisq(q = Chi.Square, df = df, ncp = UL.0) - alpha.upper
    if (Diff < 0) {
      FAILED.Up <- TRUE
      warning("The size of the effect combined with the degrees of freedom is too small to
determine an upper confidence limit for the 'alpha.upper' (or (1/2)(1-'conf.level') symmetric)
value specified.", call. = FALSE)
    }
    if (is.null(FAILED.Up)) {
      UL.1 <- UL.2 <- UL.0
      while (Diff > tol) {
        UL.2 <- UL.1 * (1 + Jumping.Prop)
        Diff <- pchisq(q = Chi.Square, df = df, ncp = UL.2) - alpha.upper
        UL.1 <- UL.2
      }
      UL.1 <- UL.2/(1 + Jumping.Prop)
      UL.Bounds <- c(UL.1, (UL.1 + UL.2)/2, UL.2)
      Diff <- pchisq(q = Chi.Square, df = df, ncp = UL.Bounds[2]) - alpha.upper
      while (abs(Diff) > tol) {
        Diff.1 <- pchisq(q = Chi.Square, df = df, ncp = UL.Bounds[1]) - alpha.upper > tol
        Diff.2 <- pchisq(q = Chi.Square, df = df, ncp = UL.Bounds[2]) - alpha.upper > tol
        Diff.3 <- pchisq(q = Chi.Square, df = df, ncp = UL.Bounds[3]) - alpha.upper > tol
        if (Diff.1 == TRUE & Diff.2 == TRUE & Diff.3 == FALSE) {
          UL.Bounds <- c(UL.Bounds[2], (UL.Bounds[2] + UL.Bounds[3])/2, UL.Bounds[3])
        }
        if (Diff.1 == TRUE & Diff.2 == FALSE & Diff.3 == FALSE) {
          UL.Bounds <- c(UL.Bounds[1], (UL.Bounds[1] + UL.Bounds[2])/2, UL.Bounds[2])
        }
      }
      Diff <- pchisq(q = Chi.Square, df = df, ncp = UL.Bounds[2]) - alpha.upper
    }
  }
}

```

```

    }
    UL <- UL.Bounds[2]
  }
  if (!is.null(FAILED.Up))
    UL <- NA
}
## 结果的输出: 设定的置信极限及其对应的 ncp 值, 上下限范围
if (alpha.lower > 0 & alpha.upper > 0)
  return(list(Lower.Limit = LL, Prob.Less.Lower = alpha.lower,
              Upper.Limit = UL, Prob.Greater.Upper = alpha.upper))
if (alpha.lower == 0 & alpha.upper > 0)
  return(list(Conf.Interval.type = "one-sided", Lower.Limit = 0,
              Upper.Limit = UL, Prob.Greater.Upper = alpha.upper))
if (alpha.lower > 0 & alpha.upper == 0)
  return(list(Conf.Interval.type = "one-sided", Lower.Limit = LL,
              Prob.Less.Lower = alpha.lower, Upper.Limit = Inf))
}

```

2 结果示例

```
conf.limits.nc.chisq(Chi.Square = 30, df = 15, conf.level = NULL, alpha.lower = 0.05, alpha.upper = 0)
```

```
$Conf.Interval.type
```

```
[1] "one-sided"
```

```
$Lower.Limit
```

```
[1] 3.130205
```

```
$Prob.Less.Lower
```

```
[1] 0.05
```

```
$Upper.Limit
```

```
[1] Inf
```

```
conf.limits.nc.chisq(Chi.Square = 30, df = 15, conf.level = 0.95)
```

```
$Lower.Limit
```

```
[1] 1.407074
```

```
$Prob.Less.Lower
```

```
[1] 0.025
```

```
$Upper.Limit
```

```
[1] 38.87651
```

```
$Prob.Greater.Upper
```

```
[1] 0.025
```

五、R 中的大数定律及中心极限定理解说

1 基本介绍

大数定律：

在试验不变的条件下，重复试验足够多次，随机事件的频率会趋近于它的概率，而样本也更能代表总体。其表现了随机现象的性质之一：平均结果的稳定性。

关于强大数定律和弱大数定律：

<https://stats.stackexchange.com/questions/2230/convergence-in-probability-vs-almost-sure-convergence?noredirect=1&lq=1>

<https://www.zhihu.com/question/21110761>

弱大数定律在强大数定律之前被发现，两者都反映的是：随着样本数的增大，样本的平均数更能估计总体的平均数。区别在于：弱大数定律表示样本均值“依概率收敛”于总体均值；而强大数定律证明了样本均值可以“以概率为 1 收敛”于总体均值，几乎处处收敛，比前者更强。

中心极限定理：

对于一个样本量足够大的随机抽样而言，即使样本原分布不是正态分布，样本均值的分布也近似服从正态分布。随着样本量增大，样本均值的分布会越来越接近正态分布。

计算随机变量均值的标准化变量：

$$Y_n = \frac{X_k - \mu}{\sigma/\sqrt{n}}$$

ggplot：

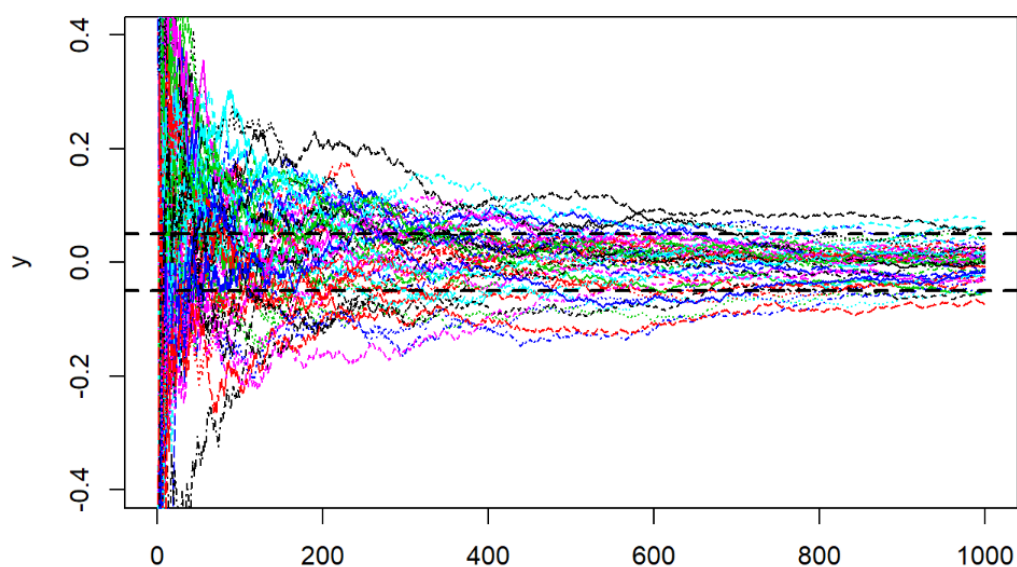
ggplot2 是一个用于绘制数据图形的 R 软件包。它的基础是一套描述所有统计图形深层特性的语法规则。其特色是按图层绘图，首先绘制原始数据，然后在其上不断添加图形注释和统计汇总结果。这种绘图方式有利于结构化思维。

参考：<https://zhuanlan.zhihu.com/p/19792783>

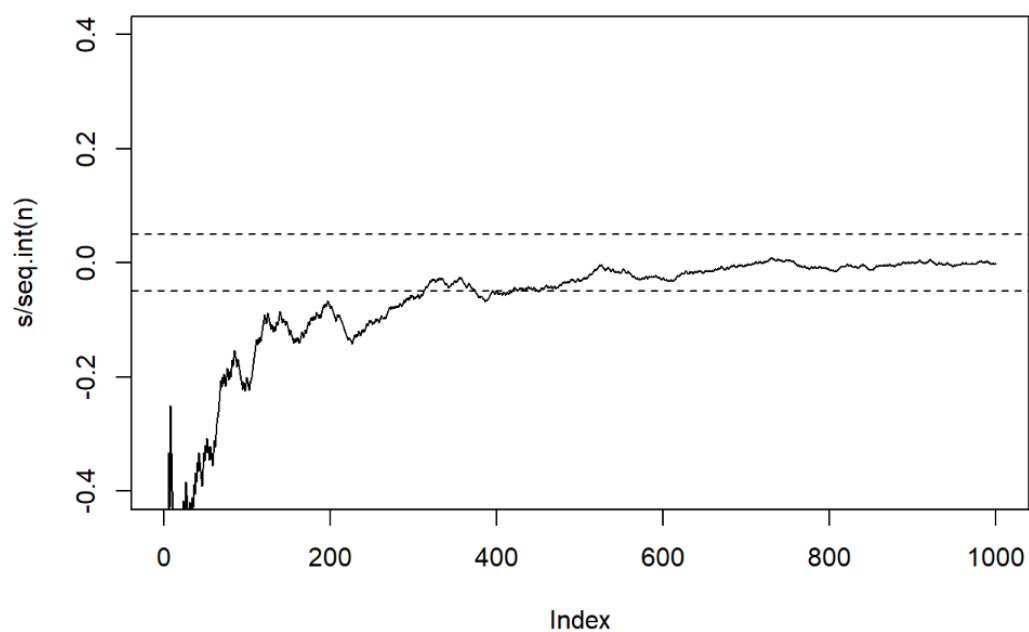
2 解读

大数定律：

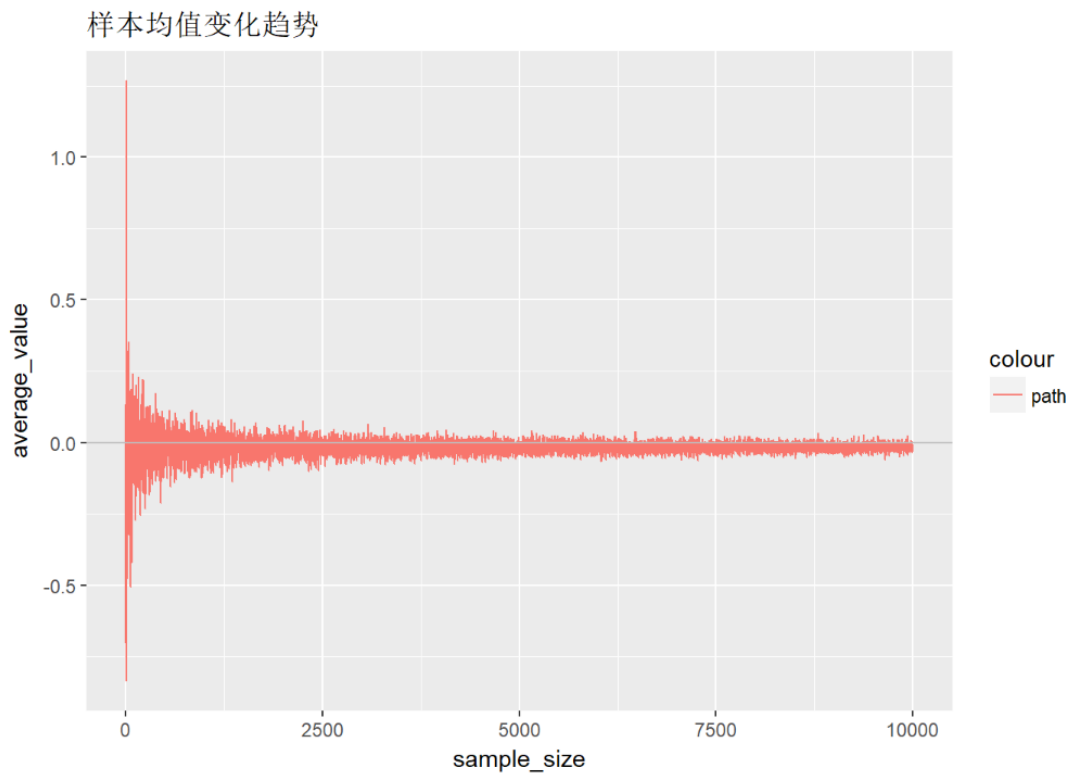
弱大数定律：每条线代表一个数列，虚线表示一个非常小的区间。总的来说每个数列都越来越趋近 0，且大部分时候不会超过虚线所表示的小边界，但偶尔会有一些线超过虚线、然后再回到虚线之内。



强大数定律：黑线表示一个随机数列，这个数列在 n 达到某个值后进入了一个规定的小边界（用虚线表示），且之后不会再超出虚线所表示的边界（超出这个边界的概率是 0）。

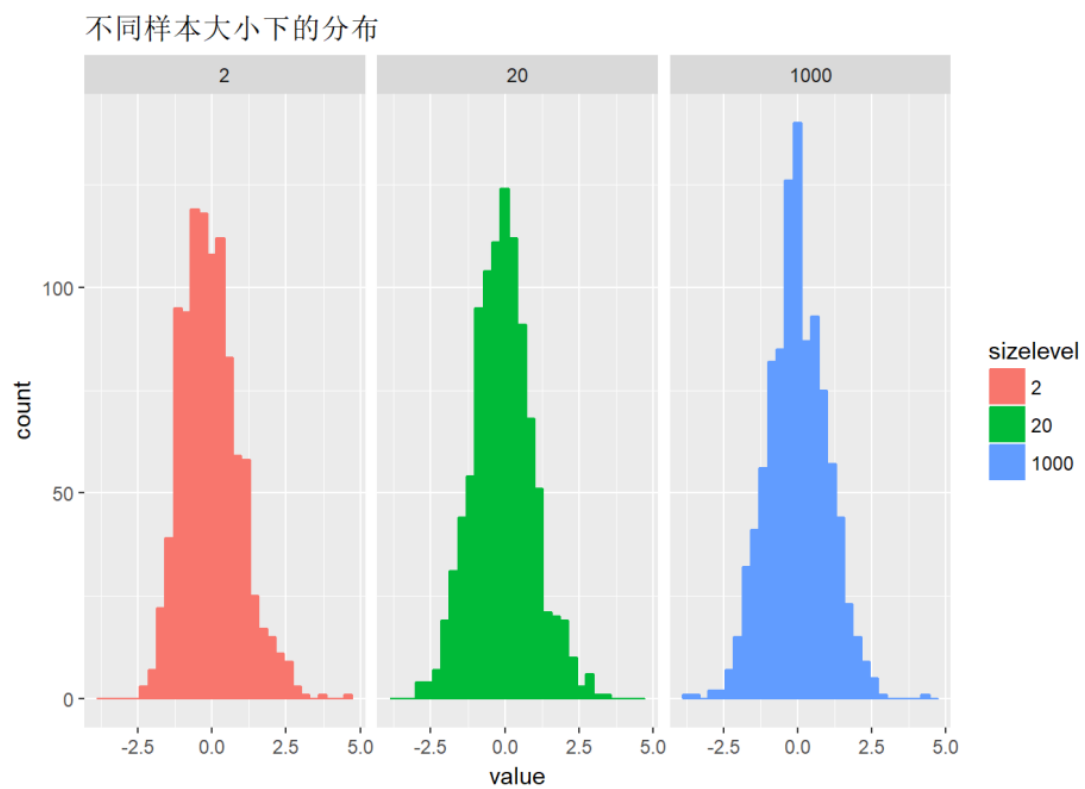


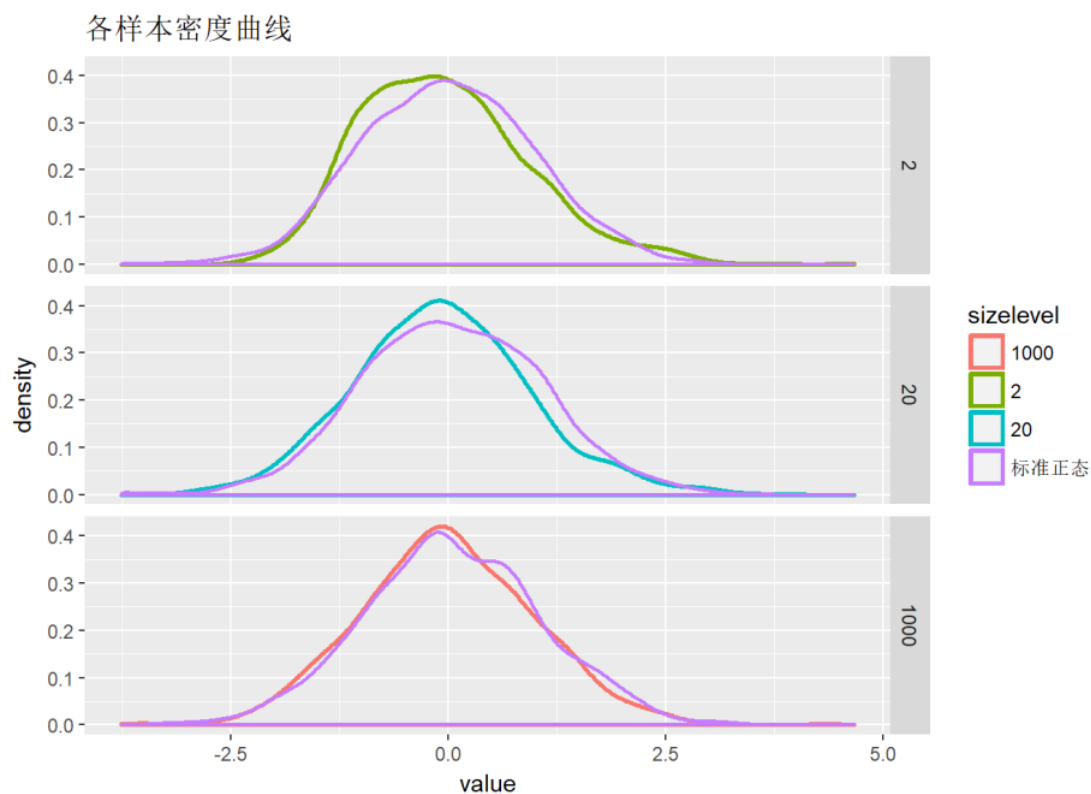
两种条件下，虚线所表示的边界都可以是任意小，总存在使线保持在边界内的 n 值。对于强大数定律而言，超过 n 值之后，线超出这个边界的概率就是 0 了。



小样本时样本均值十分不稳定,重复多次后样本均值收敛明显,基本收敛于总体的均值。

中心极限定理:





当样本量更大时，与标准正态的密度曲线拟合得更好。

3 问题及其它

- 关于 `ggplot` 的报错 “Error: Aesthetics must be either length 1 or the same as the data (300): x, colour” 仍然不太清楚，需要解决。
- 标准正态画不标准也是未解之谜（后续：把抽样次数调大了似乎好一些）。中心极限定理的图示不是特别明显，或许和分布、样本量的选择有关。
- 对强弱大数定律、`ggplot` 的画图及分面方式有了一点新了解。

六、Netlogo 合作模型复现

1 基本介绍

Netlogo:

用来对自然和社会现象进行仿真的可编程建模环境。

合作模型:

参见 <http://ccl.northwestern.edu/netlogo/models/Cooperation>。

- 一个进化生物学模型，和“利他主义”、“分蛋糕”同属于单元“利他主义和合作习惯的演变：学习进化中的复杂性”。
- 在其中，主体（奶牛）竞争自然资源（草）。得到更多草的奶牛更容易繁殖，因此在进化上更有优势。模型中包括两种奶牛，贪婪型和合作型。模型显示了这两种不同的策略在进化过程中的竞争（假定每头奶牛的行为策略是不变的）。
- 具体过程：每一步，每头牛吃掉当前所在 patch 的一片草。草的高度有一个阈值，在阈值之上时草的生长速度远高于在阈值之下时。贪婪型的奶牛只要有草就吃，不顾对草生长的影响；而合作型的奶牛不会吃掉高度阈值以下的草，为群体的食物资源发展考虑。

2 参数及指标

关于奶牛:

- **initial-cows** 初始奶牛数：设置起始的奶牛数量。（0-100，初始值 20）
- **cooperative-probability** 合作概率：设置初始合作型奶牛的比例。（0-1，初始值 0.50）
- **stride-length** 步长：奶牛每一步向前移动的距离。此值增加时，奶牛会更频繁地转移到其他 patch。（0-0.3，初始值 0.08）
- **metabolism** 代谢：奶牛每走一步消耗的能量。如果奶牛能量低于 0，它就会死亡。（0-99，初始值 6）
- **reproduction-threshold** 繁殖阈值：奶牛为了能够繁殖而必须具有的能量。牛的能量达到此值即可繁殖。（0-200，初始值 102）
- **reproduction-cost** 繁殖成本：奶牛每次繁殖时消耗的能量。（0-99，初始值 54）
- **按时间计数的图像及监视器**：plot count cooperative-cows; plot count greedy-cows

关于草:

- **grass-energy** 草的能量：奶牛每吃一次草能获得的能量。（0-200，初始值 51）
- **max-grass-height** 草的最大高度：草生长的高度上限。（2-40，初始值 10）
- **low-high-threshold** 草的生长阈值：草的高度高于或等于此值时，草以高增长速度生长；低于此值时，以低增长速度生长。（0-40，初始值 5）
- **high-growth-chance** 高增长几率：高于生长阈值的草将重新生长的百分比几率。该值越低，合作型奶牛和贪婪型奶牛的行为之间的差异就越小。（0-1，初始值 0.77）
- **low-growth-chance** 低增长几率：低于生长阈值的草将重新生长的百分比几率。该值越

高，合作型奶牛和贪婪型奶牛的行为之间的差异就越小。（0-1，初始值 0.30）

3 代码解读

思路整理：

- 牛：吃草、移动、繁殖；能量变化
- 草：生长、变色；高度变化

turtles-own [energy] ;;设置属于海龟（牛）的变量：能量

patches-own [grass-height] ;;设置属于瓦片（草）的变量：高度

breed [cooperative-cows cooperative-cow]

breed [greedy-cows greedy-cow] ;;定义牛的种类

to setup ;;初始化

clear-all ;;清屏

setup-cows ;;牛的初始设定

ask patches [;;草的初始设定

set grass-height max-grass-height

grass-color

]

reset-ticks ;;重置时钟计数

end

to setup-cows

set-default-shape turtles "cow" ;;设定牛的图形

create-turtles initial-cows [;;画初始的牛

setxy random-xcor random-ycor ;;随机分布位置

set energy metabolism * 4 ;;设定初始能量

ifelse (random-float 1.0 < cooperative-probability) [;;设定牛的比例及其颜色

set breed cooperative-cows

set color red - 1.5

][

set breed greedy-cows

set color blue - 0.5

]

]

end

to grass-color

set pcolor scale-color green grass-height 0 (2 * max-grass-height) ;;根据草的高度设定颜色

end

;; scale-color 用法:

scale-color color number range1 range2 返回明暗与 number 成正比的 color 颜色。

若 range1 < range2, 则 number 越大, 颜色越亮。

number < range1 时为最暗的 color 颜色, number > range2 时为最亮的 color 颜色。

若 range1 > range2, 则相反。

to go

ask turtles [

move

eat

reproduce

]

ask patches [

grass-grow

grass-color

]

tick ;;时钟计数器

end

to move

rt random 360 ;;右转任意角度

fd stride-length ;;移动一个步长

set energy energy - metabolism ;;消耗代谢能量, 若能量不足则死亡

if energy < 0 [die]

end

to eat

ifelse breed = cooperative-cows [;;根据种类设定吃草策略

eat-cooperative

][

if breed = greedy-cows

[eat-greedy]

]

end

to eat-cooperative ;;合作型奶牛只在草高于生长阈值时吃草

if grass-height > low-high-threshold [

set grass-height grass-height - 1

set energy energy + grass-energy

```

    ]
end

to eat-greedy ;;贪婪型奶牛有草就吃
    if grass-height > 0 [
        set grass-height grass-height - 1
        set energy energy + grass-energy
    ]
End

to reproduce
    if energy > reproduction-threshold [ ;;当能量足够繁殖时，繁殖一头牛
        set energy energy - reproduction-cost
        hatch 1 [set energy metabolism * 4] ;; hatch number [ commands ]表示创建一个新海龟并
        执行指令
    ]
end

to grass-grow
    ifelse ( grass-height >= low-high-threshold) [ ;;在阈值以上/以下的草分别高/低速生长
        if random-float 1.0 <= high-growth-chance
            [ set grass-height grass-height + 1 ]
    ][
        if random-float 1.0 <= low-growth-chance
            [ set grass-height grass-height + 1 ]
    ]
    if grass-height > max-grass-height ;;控制草的最高高度
        [ set grass-height max-grass-height ]
end

```

4 测试结果

- 在初始设定下，只改变合作型奶牛的比例、初始奶牛数，发现只有当几乎没有贪婪型奶牛时，合作型奶牛才能在进化中胜出。
- 更改代谢：当消耗为 0 时，两种奶牛都增长，合作型奶牛最终数量稳定，而贪婪型继续增加；增加代谢，当代谢较低时贪婪型奶牛胜出，较高时合作型奶牛胜出，更高时两者均灭亡，贪婪型更早灭亡。生存条件比较艰难时，合作型的策略更有利。
- 更改吃草获得的能量：收益很少时，两者都消亡，合作型占优；收益大时，贪婪型占优。
- 繁殖成本过大时，贪婪型所得的实际利益不高，合作型占优。
- 环境条件制约、行为成本及收益对两种策略优劣的影响，更多仿真结果参见文献：合作

行为的 NetLogo 计算机仿真研究（耿柳娜，李艳，2011）。

效果图：

