统计作业——R语言（1）

康珣

30 September 2016

## 阅读内容(读书笔记)

* R语言编程艺术
* 1-5章

1. 取模运算：%\*%
2. R语言与其他编程语言，不同的是下标从1开始。
3. Str显示R对象的内部结构
4. 数据框就是数据库中二维表的形式
5. 调用帮助，查看函数、对象，使用？加上被查询内容
6. 循环补齐，在一定情况下，自动延长向量。
7. +、-、\*、/等运算符实际上为函数：“+”（2,3）
8. 向量相乘\*，实际上是元素相乘
9. 冒号运算符优先级高于加减
10. rep函数中，参数each交替重复、times连续重复
11. 不论是向量还是矩阵，若想加快运算速度，提高性能，最好预先分配内存空间，避免使用runs <- c(runs,i)重复分配空间。
12. NA有模式，计算时不会自动跳过，需要设置参数na.rm 为TRUE，NULL没有模式，计算时会自动跳过。
13. 选择函数返回符合条件的元素的下标
14. 矩阵默认按列填充
15. 设置参数drop防止意外降低维度，如：矩阵中提取一行，子矩阵变为向量类型。
16. 同模式：向量（维度1）、矩阵（维度2）、数组（维度3）

不同模式：列表、数据框

1. 若为对象中变量添加名称，除了下标访问，都可以通过名称访问。

## 练习一:

* 找到1000以下，所有能被3或5整除的数，将它们相加

res <- NULL

for(i in 1:999){

if(i %% 3 == 0 || i %% 5 == 0)

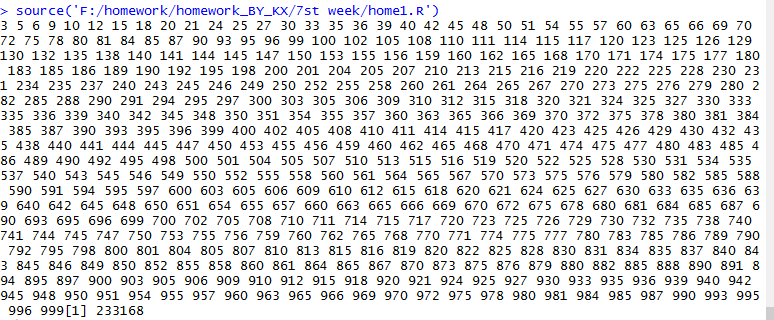
res <- c(res,i)

}

cat(res)

print (sum(res))

结果：



## 练习二:

* 找到4000000以下的斐波纳契数列，将其中的偶数进行求和

findodd <- function(x){

a <- 0

b <- 1

odd <- c(a,b)

c <- a + b

sumodd <- 0

while(c < x){

odd <- c(odd,c)

if(c %% 2 == 0){

sumodd = sumodd + c

}

a <- b

b <- c

c <- a + b

}

cat(odd)

print (sumodd)

}

findodd(4000000)

结果：

C:\Users\sherl\AppData\Roaming\Tencent\Users\1070746603\QQ\WinTemp\RichOle\7F5[OLWL([_B8JHCEK%TTBA.png

## 练习三:

* 已知一对兔子每一代可以生育n对，初始有x对成小兔子，y对未成年兔子，那么在m月后一共有多少兔子（斐波纳契数列进化版，代码尽量写成函数形式） ### 给出一组条件：x=1,y=1,n=3,m=5，求m月后的兔子数量。

count <- function(x, y, n, m){

num[1] <- x + y

num[2] <- num[1] + x \* n

for(i in 3:m+1){

num[i] <- num[i-1] + num[i-2] \* n

}

print(num[m+1])

}

count(1,1,3,5)

结果：

C:\Users\sherl\AppData\Roaming\Tencent\Users\1070746603\QQ\WinTemp\RichOle\)O$$`VBWMIKW36(NZI4031S.png

## 小练习四:

根据ISLR包中Wage数据集进行如下计算 -

wage <- Wage$wage

age <- Wage$age

education <- Wage$education

health <- Wage$health

health\_ins <- Wage$health\_ins

* 计算wage的均值和标准差，计算上侧和下侧四分位数

mean(wage) #均值

sd(wage) #标准差

quantile(wage, 0.25) #四分之一分位数

quantile(wage, 0.75) #四分之三分位数

* 计算wage和age之间的相关性

cor(age,wage)

* 计算education,health,health\_ins的交互频数表

health.table <- with(Wage,table(education,health,health\_ins))

health.table

* 计算不同education的wage均值以及相应人数

tapply(wage,education,mean) #对不同education使用mean

tapply(wage,education,length) #不同education的wage均值以及相应人数

* 计算不同education和wage的联列表

wage\_edu <- tapply(wage,education,table)

wage\_edu #不同education和wage的联列表

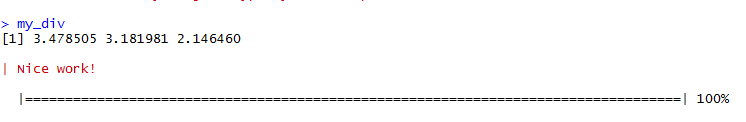
四个练习须附上相关代码

## Swirl Exercise

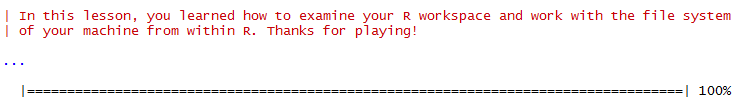
* R Programming
* 截图最后的结果

install.packages("swirl")  
library(swirl)  
swirl()

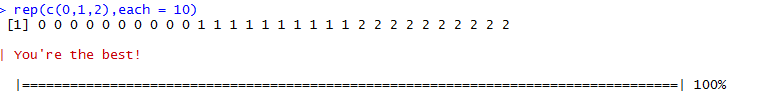
1



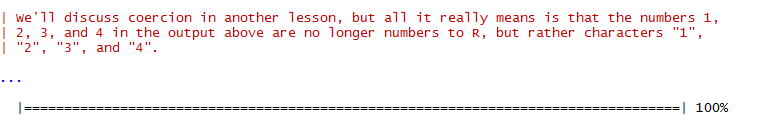
2



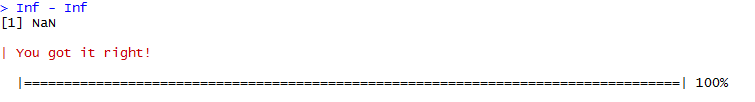
3



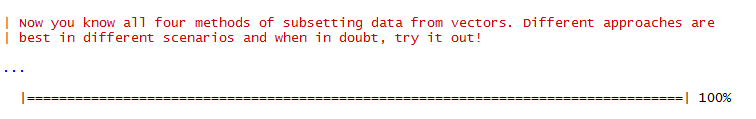
4



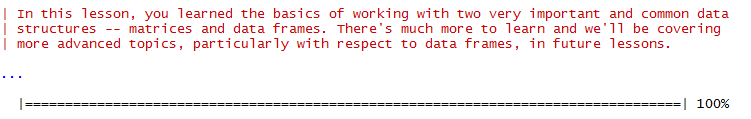
5



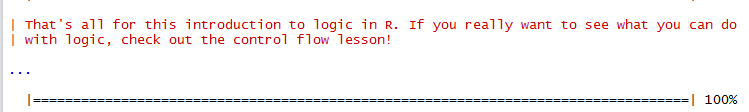
6



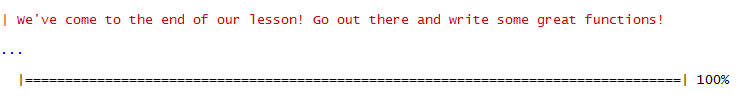
7



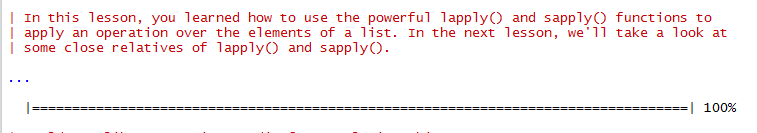
8



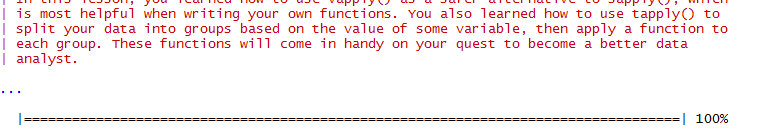
9



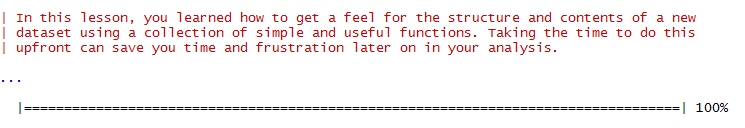
10



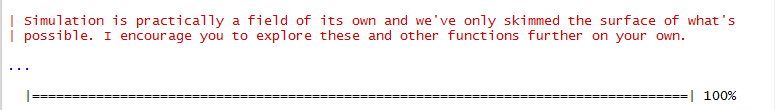
11



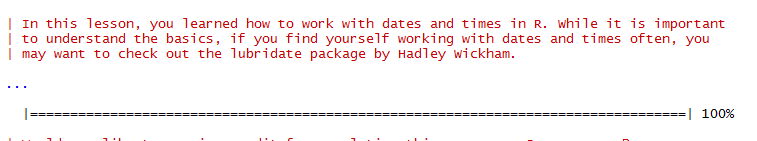
12



13



14



15

