Вероятнсти и статистика с R

Асен Чорбаджиев

October 12, 2017

1 Увод.

1.1 Среда за програмиране и език

R е език и среда за статистически изчисления и графика. Той е разработен в Bell Laboratories, като GNU проект на езика S. Езикът R е скриптов език за пресмятания в специална среда, написана предимно на C/C++ и Фортран. По-късно са добавени фунционалности програмирани на R. Първоначално R е бил предназначен основно за Unix, но в момента няма ограничение за операционната система.

Използването на R може да става чрез:
- скриптове с файлове с разширение R
- Използване на IDE - RGui, RStudio, Eclipse, Visual Studio.
- Вграждане в програми от други езици, като C/C++ и Java (Не е тема на курса)

Средата R се инсталира с минимална функционалност налична в пакета base. Останалите статистически функционалности и примерни данни са разделени в библиотеки. Те са достъпни чрез mirror locations - CRANE(Comprehensive R Archive Network). Тяхното първоначално инсталиране става със специална команда или през специално меню в IDE. След това исталираните библиотеки могат да се ползват, чрез зареждане през меню на студио или команда. По същият начин може да бъде ползван help за всяка функция и библиотека.

За програмиране и изпълнение на програмен код през shell се използва разширение на файлове $'*.R'$. Символ за отделяне на коментари е $'\#'$.

Упражнения:

- Отваряне на RStudio

- *help()*
- *sessionInfo()*
- Да се инсталира пакет *UsingR*. Да се намери в Help какво представлява пакета. Да се използват команда *install.packages(); library()*
- Команда *ls()*
- Използване на меню от RStudio
- Създаване на $'*.R'$ файл за упражнение 1.

1.2 Основни типове данни

Основното предназначение на R е за статистическа обработка на данни. Затова структурите от данни са оптимизирани за работа с вектори. Инициализирането на вектор x за структура от даннии или математическа функция става посредством функцията x=c(x1,x2 ,...) и оператор за присвояване '='.

Операции за достъп до елементи:

- скалар чрез индекс: x[i]; i >= 1
- подмножество от i до j чрез индекс: x[i:j]; i,j>=1
- подмножество чрез изключване: x[-i] ; x[-(i:j)]
- конкатенация на вестори: x=c(x1,x2)
- чрез операции за сравняване $>, <, ==, \&, |, ...$: x[x>1]
- достъп до индекси: which(x==3)
- автоматизирано създаване на редица: seq(); rep()
- размер: length()
- операции със скалар/вектор: +,-,*,/,^
- основни функции за манипулации на вектори: sum(); mean(); sort(); min(); max(); range(); cumsum()

Основните типове променливи са числени (integer, numeric, double, complex), логически (TRUE/FALSE/NULL), чар (character), категориращи (factor). Всеки тип има нулев pointer NULL (включително логическите!! за разлика от C++). За разлика от C++, съществува стойност за липсващи данни - NA и невъзможни числени резултати, като деление на 0 - NaN.

Проверките за типа на променливите става с помощта на функциите 'is.'. Дефинирането на тип на променливи с 'as.'

- Проверката за нулев pointer става с помощта на функцията is.null(x).
- Проверката за нулева стойност става с функцията is.na(x).
- Проверка за тип - is.numeric(x), is.integer(x), is.character(x)...
- Дефиниране на тип: as.numeric(x), as.integer(x), as.character(x)...

1.3 Матрици

Матрици в R се създават чрез функцията matrix(data, nrow, ncol, byrow). Например:

$$A = matrix(\\ c(2,\ 4,\ 3,\ 1,\ 5,\ 7),\\ nrow = 2,\\ ncol = 3,\\ byrow = TRUE)$$

Съществува възможност за начална инициализация чрез примитивната функция as.matrix(data) - създава по подразбиране обект от тип matrix(ncol=1). Проверката за това дали даден обект M е матрица става с функцията is.matrix(M). Операции за достъп до елементи:

- скалар чрез индекс: M[i,j]; i,j >= 1
- колона или стълб: M[i,]; x[,i]; i>=1
- селектиране на определени стълбове и колони: M[c(i,j),]; M[,c(i,j)]l i,j>=1
- добавяне на колони и редове - cbind(M,col), rbind(M,row)
- размерност dim(M), nrow(M), ncol()
- dimnames() - функция и параметър в matrix() за поставяне на имена на колоните и стълбовете. Например, за създадената по-горе матрица A, поставянето едновременно на имена на редовете и колоните става по следния начин:
  dimnames(A) = list(
  c("row1", "row2"), # row names
  c("col1", "col2", "col3")) # column names

- достъп до колони и редове с име - x["row_name","col_name"]
- операции със скалар/вектор/матрица: +,-,*, /,^, %*%
- обръщане и транспониране - solve(M), t(M)
- достъп до главения диагонал на матрицата. diag(M)

Упражнения:

- **Зад. 1.** Какво прави следната функция:
  v=matrix(c("a", "b", "c", "2", "2", "3"), TRUE)
- **Зад. 2.** Има две матрици:
  mat1 <- matrix(1:6, 2)
  mat2 <- matrix(c(rep(1, 3), rep(2, 3)), 2, byrow = T)
  Какво извеждат следните операции:
  solve(mat1[, 2:3])
  mat1*mat2
  mat1 %*% t(mat2)
- **Зад. 3.** Нека е даден вектор x = (8,3,8,7,15,9,12,4,9,10,5,1). Да се:
  - създадете матрица 6 реда и 2 колони
  - да се поставят имена на колоните "c1", "c2"
  - намерете максимум и минимум
  - добавете нов нов ред от случайни числа между 1:20 в края на матрицата. Използвайте функцията runif()
  - умножете втората колона с 2 и след това всеки елемент съберете с 6.
- **Зад. 4.** Да се създаде матрица от вероятностите за получаване на ези от n=10 хвърляния на монета. В първата колона е вектора на честотите μ/n, където μ=0:10 е броят на получаване на ези. Във втората редица са вероятностите ($C_n^\mu/2^n$), изчислени с функцията choose(). Да се начертае графика с функцията plot().