

Міністерство освіти і науки України
Національний університет «Львівська політехніка»

Звіт
до лабораторної роботи №1
з курсу «Методи і засоби комп'ютерного навчання»
ЗНАЙОМСТВО З СИСТЕМОЮ R

Виконав:
студент гр. СПКм-12
Сергієнко В. Р.
Прийняв
Пукач А.І.

Львів 2014

Мета: Оволодіти навиками по створенню векторів, графіків функцій, матриць, багатовимірних масивів, фреймів даних тощо в системі R.

Короткі теоретичні відомості

При запуску програми RStudio з'являється основне вікно програми, що містить підвікно, яке називається Console. Це командне вікно (консоль), в якому користувач вводить команди, а система друкує результати. В процесі роботи в основному вікні будуть з'являтися інші вікна, тобто вікна редактора скриптів, графічні вікна, в яких відображаються графічні результати виконання команд тощо.

Основним типом даних в системі R є число з плаваючою крапкою. Для відділення дробової частини від цілої використовується крапка. Показник відділяється від мантиси символом *e* або *E*. Наприклад, 31415, +3.1415e4, .31451e+5, 3141500e-2 – різні записи одного і того ж числа 31415.

Вектори в R формуються функцією *c()*. Аргументи цієї функції сутність компоненти вектора. Наприклад, команда `> c(1, 2, 3, 4, 5)`

```
[1] 1 2 3 4 5
```

формує вектор-рядок (1, 2, 3, 4, 5).

Для зображення графіків функцій в R є функція *plot((x, y))*. Функція *barplot(x)* малює стовпцюву діаграму. Функція *pie(x)* будує кругову діаграму. Функція *hist* будує гістограму.

Списки в R – це колекції об'єктів, доступ до якої можна здійснювати за номером або ім'ям. Список може містити об'єкти (компоненти) різних типів, що відрізняє списки від векторів. Компонентами списку можуть бути в тому числі і вектори та інші списки.

Числову матрицю можна створити з числового вектора за допомогою функції *matrix*. Потрібно вказати число рядків *nrow = m* та/або число стовпців *ncol = m*. В результаті елементи з вектора будуть записані в матрицю зазначених розмірів.

Матриці – це окремий випадок багатовимірних масивів. Матриці мають дві розмірності. В загальному випадку масиви можуть мати більше розмірностей. Робота з багатовимірними масивами в R багато в чому аналогічна. Основний спосіб їх створення – функція *array*.

Фрейми даних (data frames) – один з найважливіших типів даних в R, що дозволяє об'єднувати дані різних типів разом. Можна вважати, що фрейм даних – це двовимірна таблиця, в якій (на відміну від числових матриць), різні стовпці можуть містити дані різних типів (але всі дані в одному стовпці мають один тип).

Індивідуальне завдання

1. Згенеруйте вектор довжини 7983, елементами якого є реалізації нормально розподіленої випадкової величини з математичним очікуванням, що дорівнює 1, і стандартним відхиленням, рівним 0.5. Підрахуйте статистичне математичне очікування і стандартну помилку, не використовуючи вбудовані функції та перевірте правильність результату. Підрахуйте .95, .99-квантілі. Дослідіть відхилення статистичного математичного очікування від 1 при зростанні N (N = 1000, 2000, 4000, 8000).

Код скрипту:

```
mas<-rnorm(7983,1,0.50)
sumamas<-sum(mas)
MatemSpod=sumamas/7983
MatemSpod

Szn=mean(mas)
Dys=mean(mas^2)-mean(mas)^2
Dys

OcDys=var(mas)
OcDys

kvantel=quantile(mas,c(0.95,0.99))
kvantel

nmas=matrix(c(1000,2000,4000,8000),nrow=1)
Msp1=matrix(1:4, nrow=1)
Msp2=matrix(1:4, nrow=1)
Ds1=matrix(1:4,nrow=1)
Ds2=matrix(1:4,nrow=1)
for(i in 1:length(nmas)){
  x=rnorm(nmas[1,i],1,0.5)
  suma=sum(x)
  Msp1[1,i]=mean(x)
  Msp2[1,i]=suma/nmas[1,i]
  Ds1[1,i]=var(x)
  Ds2[1,i]=mean(x^2)-mean(x)^2
}
Msp1
Msp2
Ds1
Ds2
```

Результати виконання скрипту:

```
> MatemSpod
[1] 1.01336
> Dys
[1] 0.2530851
> OcDys
[1] 0.2531168
> kvantel
      95%      99%
1.851900 2.214745
> Msp1
      [,1]      [,2]      [,3]      [,4]
[1,] 1.007596 1.008497 0.9908564 0.9987392
> Msp2
      [,1]      [,2]      [,3]      [,4]
[1,] 1.007596 1.008497 0.9908564 0.9987392
> Ds1
      [,1]      [,2]      [,3]      [,4]
[1,] 0.2482894 0.2495071 0.2541076 0.2511954
> Ds2
      [,1]      [,2]      [,3]      [,4]
[1,] 0.2480411 0.2493823 0.2540441 0.251164
```

2. Створіть фрейм даних з $N = 15$ записів з наступними полями: *Nrow* – номер запису, *Name* – ім'я користувача, *BirthYear* – рік народження, *EmployYear* – рік прийому на роботу, *Salary* – зарплата, де *Nrow* змінюється від 1 до N , *Name* задається довільно, *BirthYear* розподілений рівномірно на відрізку $[1960, 1990]$, *EmployYear* розподілений рівномірно на відрізку $[BirthYear + 18, 2010]$, *Salary* для працівників молодше 1975 р.н. визначається за формулою $Salary = (\ln(2011 - EmployYear) + 1) * 800$, для решти – $Salary = (\log_2(2011 - EmployYear) + 1) * 800$. Підрахуйте число співробітників із зарплатою, більшою 1500. Додайте в таблицю поле, що відповідає сумарному прибуткового податку (ставка 13%), виплачується працівником за час роботи в організації, якщо його зарплата за кожен рік нараховувалася відповідно до формул для *Salary*, де замість 2011 потрібно послідовно підставити кожен рік роботи працівника в організації.

Код скрипту:

```
N=15
Number=c(1:N)
Name=c("A","B","C","D","E","F",
"G","H","I","J","K",
"L","M","N","O")
Birth=seq(1960,1990,len=N)
Empl=seq((Birth[1]+18),2010,len=N)
Salary=matrix(1:15,nrow=N)
for(i in 1:N){
  if (Birth[i]<1970){
    Salary[i]=(log(2011-Empl[i])+1)*800
  }
  else {
    Salary[i]=(log(2011-Empl[i])+1)/log(2)*810
  }
}
p=data.frame(Number,Name,Birth,Empl,Salary)
count=0
for(i in 1:N){
  if(Salary[i]>1500) {
    count=count+1
  }
}
Stavka=matrix(1:15,nrow=N)
Suma=matrix(1:15,nrow=N)
for (i in 1:N){
  if((2011-Empl[i])<5)
  {
    Stavka[i]=Salary[i]*0.13
  }else {
    Stavka[i]=Salary[i]*0.2
  }
  Suma[i]=Salary[i]+Stavka[i]
```

```
}  
p=data.frame(p,Stavka, Suma)  
p
```

Результати виконання:

```
> p  
  Number Name   Birth   Empl  Salary  Stavka  Suma  
1      1    A 1960.000 1978.000 3597.206 719.4412 4316.647  
2      2    B 1962.143 1980.286 3539.782 707.9565 4247.739  
3      3    C 1964.286 1982.571 3477.916 695.5831 4173.499  
4      4    D 1966.429 1984.857 3410.861 682.1722 4093.033  
5      5    E 1968.571 1987.143 3337.667 667.5334 4005.200  
6      6    F 1970.714 1989.429 4757.733 951.5467 5709.280  
7      7    G 1972.857 1991.714 4626.846 925.3692 5552.215  
8      8    H 1975.000 1994.000 4479.428 895.8856 5375.313  
9      9    I 1977.143 1996.286 4310.691 862.1382 5172.829  
10     10   J 1979.286 1998.571 4113.410 822.6819 4936.092  
11     11   K 1981.429 2000.857 3875.921 775.1841 4651.105  
12     12   L 1983.571 2003.143 3577.527 715.5054 4293.032  
13     13   M 1985.714 2005.429 3175.801 635.1603 3810.962  
14     14   N 1987.857 2007.714 2558.711 332.6324 2891.343  
15     15   O 1990.000 2010.000 1168.583 151.9158 1320.499
```

3. Напишіть функцію, яка приймає на вхід числовий вектор x і число розбиття інтервалу 10 та виконує наступне: знаходить мінімальне і максимальне значення елементів вектора x_{min} та x_{max} , розділячи отриманий відрізок $[x_{min}; x_{max}]$ на k рівних інтервалів і підраховує число елементів вектора, що належать кожному інтервалу. Далі побудувати графік, де по осі абсцис – середини інтервалів, по осі ординат – число елементів вектора, що належать інтервалу, розділене на загальне число точок. Проведіть експеримент на даній функції, де x – вектор довжини 5000, згенерований з експоненціально розподіленої випадкової величини, $k = 500$. Наближення до якого графіка ми отримуємо в результаті при великій кількості точок і числі розбивань?

Код скрипту:

```
kil=500
x=seq(1,10,len=kil)
y=dexp(x)
plot(x,y)
min_max=function(x)
{ mas=matrix(1:2,nrow=1)
  mas[1]=min(x)
  mas[2]=max(x)
  return (mas)
}
myfun=function(x,arg2)
{
  minimym=min_max(x)
  newx=seq(minimym[1],minimym[2],len=arg2/10)
  xs=seq(minimym[1],minimym[2],len=arg2/10)
  newy=seq(minimym[1],minimym[2],len=arg2/10)
  for (i in 1:(length(newx)-1))
  {
    newy[i]=0
    for(j in 1:length(x))
    {
      if ((x[j]>newx[i])&&(x[j]<newx[i+1]))
      {
        newy[i]=newy[i]+1
      }
    }
    xs[i]=(newx[i]+newx[i+1])/2
  }
  newy[arg2/10]=2
  plot(xs,newy/arg2)
  p=data.frame(xs,newy)
  return (p)
```

```

}
m_m=min_max(y)
m_m
m=myfun(y,kil)
m

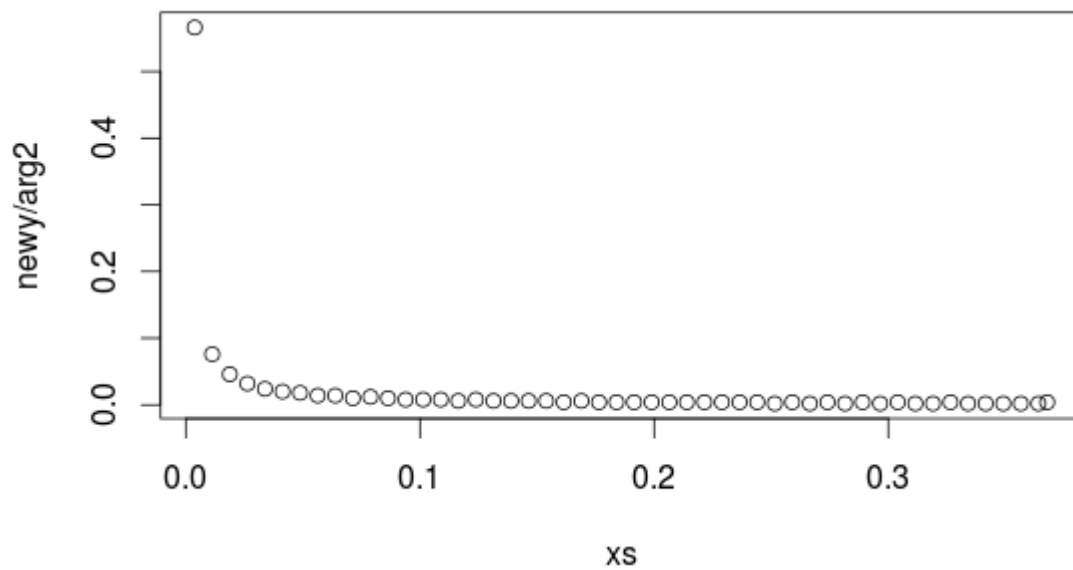
```

Результати виконання:

```

> m
      xs newy
1 0.003798809 283
2 0.011305626 38
3 0.018812443 23
4 0.026319260 16
5 0.033826077 12
6 0.041332894 10
7 0.048839712 9
8 0.056346529 7
9 0.063853346 7
10 0.071360163 5
11 0.078866980 6
12 0.086373797 5
13 0.093880615 4
14 0.101387432 4
15 0.108894249 4
16 0.116401066 3
17 0.123907883 4
18 0.131414700 3
19 0.138921518 3
20 0.146428335 3
21 0.153935152 3
22 0.161441969 2
23 0.168948786 3
24 0.176455603 2
25 0.183962421 2
26 0.191469238 2
27 0.198976055 2
28 0.206482872 2
29 0.213989689 2
30 0.221496506 2
31 0.229003324 2
32 0.236510141 2
33 0.244016958 2
34 0.251523775 1
35 0.259030592 2
36 0.266537409 1
37 0.274044227 2
38 0.281551044 1
39 0.289057861 2
40 0.296564678 1
41 0.304071495 2
42 0.311578312 1
43 0.319085130 1
44 0.326591947 2
45 0.334098764 1
46 0.341605581 1
47 0.349112398 1
48 0.356619215 1
49 0.364126033 1
50 0.367879441 2

```



4. Точковим методом найменших квадратів апроксимувати задану таблицю лінійним та квадратичним поліномами:

x_i	$2+n$	$3+n$	$4+n$	$5+n$
y_i	$7+n$	$5+n$	$8+n$	$7+n$

Код скрипту:

```

k=6
x=c(2+k,3+k,4+k,5+k)
y
y=c(7+k,5+k,8+k,7+k)
Polinom2=function(x1,y1,n)
{
  a1=((sum(x1*y1)/n)-(sum(x1)/n)*(sum(y1)/n))/((sum(x1^2)/n)-
(sum(x1)/n)^2)
  a0=(sum(y1)/n)-a1*(sum(x1)/n)
  s=c(a1,a0)
  return (s)
}
lin=Polinom2(x,y,k)
lin

Polinom4=function(x1,y1,n)
{
  xx4=sum(x1^4)
  xx3=sum(x1^3)
  xx2=sum(x1^2)
  x2y=sum(x1^2*y1)
  xx1=sum(x1)
  xy=sum(x1*y1)
  yy=sum(y1)
  d=xx4*xx2*n+xx1*xx3*xx2+xx1*xx3*xx2-xx2^3-xx1^2*xx4-xx3^2*n
  d1=x2y*xx2*n+xx1*xy*xx2+xx1*xx3*yy-yy*xx2^2-xx1^2*x2y-n*xx3*xy
  d2=xx4*xy*n+yy*xx3*xx2+xx1*x2y*xx2-xx2^2*xy-yy*xx1*xx4-xx3*x2y*n
  d3=xx4*xx2*yy+xx1*xx3*x2y+xy*xx3*xx2-xx2^2*x2y-xy*xx1*xx4-xx3^2*yy
  a=d1/d
  b=d2/d
  c=d3/d
  s=c(a,b,c)
  return (s)
}
kv=Polinom4(x,y,k)
kv
y2=lin[1]*x+lin[2]

```

```

y3=kv[1]*x^2+kv[2]*x+kv[3]
plot (x, y, type = "n", main = "Aproximation")
colors = c("blue","red","black")
lines (x, y, type="b", col = colors[1])
lines (x, y2, type="o", col = colors[2])
lines (x, y3, type="l", col = colors[3])

```

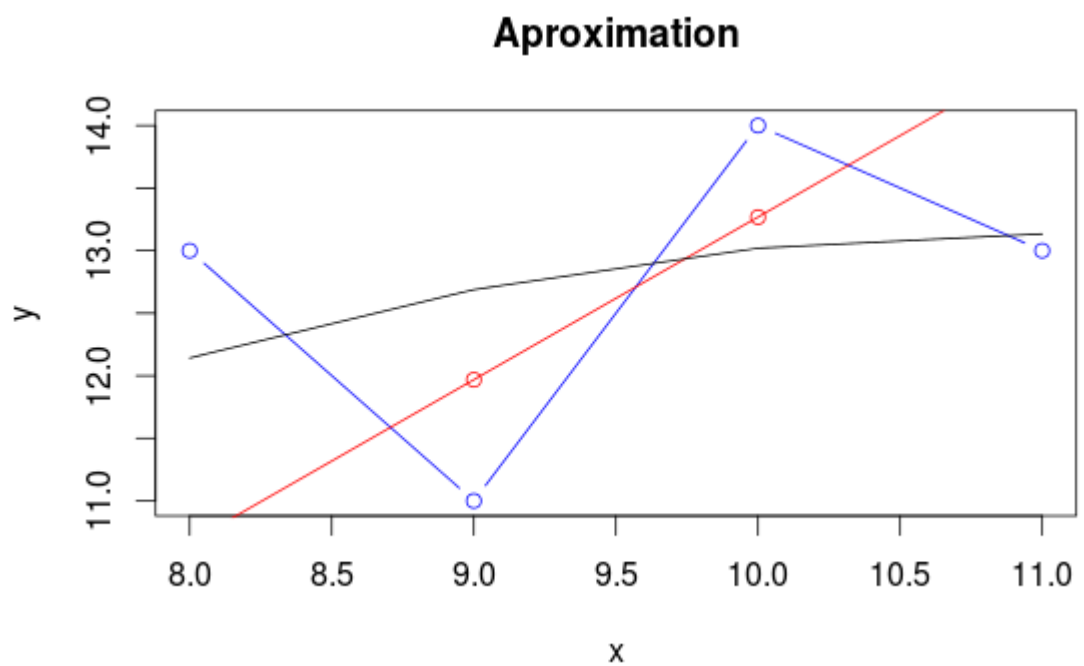
Результати виконання:

```

x [1]  8  9 10 11
y [1] 13 11 14 13

lin [1] 1.3005319 0.2632979
kv [1] -0.10783872 2.37949276 0.00804132

```



Висновок

На лабораторній роботі ознайомився із системою R, та навчився створювати вектори та графіки функцій, виконав індивідуальне завдання.