

Задание 1. Тема: Знакомство с системой анализа данных R. Данные в R.

Теоретическая часть

- 1) Ознакомиться с интерфейсом программы RGui. Консоль, скрипты, окружение, окна графиков, помощи. Прямые расчеты в консоли, создание элементарных скриптов. Определить рабочую директорию и изменить ее на свою папку.

```
> getwd()
[1] "C:/Users/polos/OneDrive/Документы"
> setwd("C:/Users/polos/OneDrive/Рабочий стол/Практика")
> getwd()
[1] "C:/Users/polos/OneDrive/Рабочий стол/Практика"
> |

> x <- 7
> y <- 4
> z <- 1
> x + y - z
[1] 10
> (x * y) / z
[1] 28
> (x ** y) % z
..
```

- 2) Переменные в R. Типы переменных: числовые, символьные, логические. Матрицы. Списки. Операции над переменными: арифметические, логические. Отбор данных с помощью фильтров, поиск значений в массивах. Ввести числовой, строковый и логический вектор. Выбрать любой отдельный элемент вектора. Ввести матрицу произвольного размера. Определить отдельные элементы матрицы. Ввести массив и выполнить поиск значения.

```
> 2 != 3
[1] TRUE
> x <= y
[1] FALSE
> numVector <- c(1,2,3,4,5,6)
> strVector <- c("S", "T", "A", "S")
> numVector
[1] 1 2 3 4 5 6
> strVector
[1] "S" "T" "A" "S"
> strVector[2]
[1] "T"
> numVector[0]
numeric(0)
> numVector[1]
[1] 1
> |
> numMatrix <- matrix(numVector, nrow=2, byrow=TRUE)
> numMatrix
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
> |
```

```

> numMatrix[1][1]
[1] 1
> numMatrix[2][1]
[1] 4
> |

> binary_search <- function(number) {
+ numArray <- seq(1, 100, 3)
+ mid <- length(numArray) %/% 2
+ low <- 0
+ high <- (length(numArray) - 1)
+ while (numArray[mid] != number & low <= high) {
+ if (number > numArray[mid]) {
+ low <- (mid + 1)}
+ else {
+ high <- (mid - 1)}
+ mid <- (low + high) %/% 2}
+ if (low > high) {
+ print("No value")}
+ else {
+ print(mid)}
+ }
> seq(1, 100, 3)
[1] 1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55
[20] 58 61 64 67 70 73 76 79 82 85 88 91 94 97 100
> binary_search(4)
[1] 2
> binary_search(5)
[1] "No value"
> binary_search(97)
[1] 33
> binary_search(25)
[1] 9
> binary_search(24)
[1] "No value"

```

3) Готовые функции в R: предназначение, аргументы. Примеры базовых функций: расчет среднего, стандартного отклонения и др. Ниже приведено описание некоторых математических функций, применяемых к векторам.

- **sin(x), cos(x), tan(x), asin(x), acos(x), atan(x), atan2(x)** – Тригонометрические функции.
- **exp(x), log(x), log10(x), log(x, base)** – Экспонента и логарифм.
- **max(x), min(x)** - Поиск максимального/минимального значения.
- **range(x)**- Поиск минимума и максимума, т.е. c(min(x), max(x)).
- **sum(x)** - Сумма элементов вектора.
- **prod(x)** - Произведение элементов вектора.
- **mean(x)** - Среднее арифметическое элементов вектора.
- **median(x)** - Медиана (средний по величине элемент) вектора.
- **quantile(x, probs = seq(0, 1, 0.25))** - Подсчет выборочных квантилей указанных уровней.

- 4) Функции для выделения данных с заданными критериям: **subset**, **with** и др.
- 5) Работа с таблицами данных **data.frame**. Таблица данных по структуре напоминает расширенную матрицу, однако данные в разных колонках могут быть различного типа. R содержит встроенные базы данных, которые можно использовать для обучения.

data() # загрузка всех доступных баз данных.

help(datasetname) # информация о конкретной базе данных datasetname

name <-data.frame(datasetname) # Преобразование данных в фрейм данных
name

Способы идентификации элементов фрейма данных **name**

1. **name** # отображение всей таблицы данных
2. **names(name)** # отображение названий столбцов
3. **colnames(name)** # отображение названий столбцов
4. **rownames(name)** # отображение названий строк
5. **dim(name))** # размер таблицы данных
6. **head(name)** # отображение первых 6 строк
7. **tail(name)** # отображение последних 6 строк
8. **str(name)** # просмотр структуры таблицы
9. **summary(name)** # для базового стат. описания таблицы
10. **name[,]** - относительно "универсальная формула" для выбора элементов таблицы. После имени таблицы пространство внутри квадратных скобок разделяется на две части запятой. Все, что находится до запятой, относится к строчкам, все что после - к столбцам. Если надо выделить несколько элементов, то используют функцию **c(...)**: внутри скобок перечисление элементов через запятую.
11. **name[,c("col.name1", " col.name2")]** # отображение столбцов с именами "col.name1", " col.name2"
12. **name\$col.name** # отображение столбца с именем col.name
13. **name [3:6,]** # отображение строк с 3-ей по 6-ю.

Редактирование элементов таблицы

14. Добавление строк

Для этого необходимо создать новую структуру - список (**list**), В список по порядку внести параметры, совпадающие со структурой таблицы (в кавычках пишут нечисловые типы данных):

```
row1<-list()
```

```
name<- rbind(name,row1) # присоединение новой строки к таблице name
```

15. Добавление столбцов

```
ID <- c(1:n) # n- общее количество строк в таблице.
```

```
name<-cbind(name,ID) # объединение вектора с таблицей, с использованием  
функцией cbind (от англ. column bind) name$ID <- as.character(name$ID) #  
поменять тип данных нового столбца на символьный.
```

16. Удаление строк и столбцов

new.name<-name[,] - Поставьте минус перед номером столбца или номером строки, которую собираетесь удалить. Если таких элементов несколько, используйте функцию **c(...)**: внутри скобок перечисление элементов через запятую.

17. Изменение имен столбцов и данных в ячейках:

```
names(name)[6] <- "new.name" # Переименование колонок. Для этого нужно  
вызвать функцию names, написать в параметрах функции имя таблицы и  
номер столбца, и присвоить ему новое имя.
```

```
name[ , "Name"] <- "....." # Изменение данные в ячейках таблицы. В  
квадратных скобках прописываем координаты нужной ячейки (до запятой -  
строка, после запятой - столбец) и присваиваем новое значение.
```

```
18.name$col.name<-as.factor(name$col.name) # изменение типа данных  
таблицы.
```

19. **subset ()** # вывода значений с использованием какого-то условия В функции **subset ()** указывается название датафрейма и условие по которому будут отобраны значения. Например:

```
subset(name, gender == 'female'))
```

```
subset(name, score > 3.5))
```

6) Система пакетов (расширений) в R: установка, загрузка пакетов, система справки. Пакеты, содержащие данные.

Практическая часть.

1. Создать n -мерный числовой вектор ($n = 10 + N, N$ – номер студента по списку).

```
> vector <- c(sample(1:100, 15, replace=T))
> vector
[1] 93 27 19 74 94 67 29 11 11 76 42 32 72 79 49

> secondVector <- c(sample(1:100, 15, replace=T))
> secondVector
[1] 95 84 52 90 86 18 89 7 23 54 45 52 72 50 67
```

2. Применить указанные в п.3 (теоретическая часть) функции к числовому вектору.

```
> cbind(vector, secondVector)
      vector secondVector
[1,]      93          95
[2,]      27          84
[3,]      19          52
[4,]      74          90
[5,]      94          86
[6,]      67          18
[7,]      29          89
[8,]      11           7
[9,]      11          23
[10,]     76          54
[11,]     42          45
[12,]     32          52
[13,]     72          72
[14,]     79          50
[15,]     49          67

> rbind(vector, secondVector)
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
vector    93  27  19  74  94  67  29  11  11
secondVector 95  84  52  90  86  18  89  7  23
      [,10] [,11] [,12] [,13] [,14] [,15]
vector     76   42   32   72   79   49
secondVector 54   45   52   72   50   67

> max(vector)
[1] 94
> max(secondVector)
[1] 95
> |

> range(vector)
[1] 11 94
> range(secondVector)
[1] 7 95

> sum(vector)
[1] 775
> prod(vector)
[1] 2.221194e+24
> mean(vector)
[1] 51.66667
```

```
> median(vector)
[1] 49
> quantile(vector, probs = seq(0, 1, 0.25))
 0%  25%  50%  75% 100%
 11   28   49   75   94
.
```

3. Для числового вектора, созданного в п. 1 создать новый вектор, в котором будут выведены только те значения, которые будут удовлетворять следующему условию: значение вектора должны быть больше среднего значения. Использовать п.4 (теоретическая часть).

```
> newVector <- subset(vector, vector > mean(vector))
> newVector
[1] 93 74 94 67 76 72 79
> vector
[1] 93 27 19 74 94 67 29 11 11 76 42 32 72 79 49
> mean(vector)
[1] 51.66667
.
```

4. Выбрать одну из встроенных в R баз данных и преобразование данные базы в фрейм данных. Для полученного фрейма данных выполнить все задания из подпунктов 1-18 пункта 5 (теоретическая часть).

```
> newFrame <- data.frame(women)
> newFrame
  height weight
1     58   115
2     59   117
3     60   120
4     61   123
5     62   126
6     63   129
7     64   132
8     65   135
9     66   139
10    67   142
11    68   146
12    69   150
13    70   154
14    71   159
15    72   164
.

> names(newFrame)
[1] "height" "weight"
> colnames(newFrame)
[1] "height" "weight"

> rownames(newFrame)
[1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11" "12" "13" "14" "15"
> dim(newFrame)
[1] 15  2
> head(newFrame)
  height weight
1     58   115
2     59   117
3     60   120
4     61   123
5     62   126
6     63   129
.
```

```

> tail(newFrame)
  height weight
10     67    142
11     68    146
12     69    150
13     70    154
14     71    159
15     72    164
> str(newFrame)
'data.frame':   15 obs. of  2 variables:
 $ height: num  58 59 60 61 62 63 64 65 66 67 ...
 $ weight: num  115 117 120 123 126 129 132 135 139 142 ...
> summary(newFrame)
   height      weight
Min.   :58.0   Min.   :115.0
1st Qu.:61.5   1st Qu.:124.5
Median :65.0   Median :135.0
Mean   :65.0   Mean    :136.7
3rd Qu.:68.5   3rd Qu.:148.0
Max.   :72.0   Max.    :164.0
.
> newFrame[,c("height", "weight")]
  height weight
1     58    115
2     59    117
3     60    120
4     61    123
5     62    126
6     63    129
7     64    132
8     65    135
9     66    139
10    67    142
11    68    146
12    69    150
13    70    154
14    71    159
15    72    164
.
> newFrame$height
[1] 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72

> newFrame[3:6,]
  height weight
3     60    120
4     61    123
5     62    126
6     63    129
.

```

```
> row1 <- list(76, 170)
> newFrame <- rbind(newFrame, row1)
> newFrame
  height weight
1      58    115
2      59    117
3      60    120
4      61    123
5      62    126
6      63    129
7      64    132
8      65    135
9      66    139
10     67    142
11     68    146
12     69    150
13     70    154
14     71    159
15     72    164
16     76    170
```

```
> ID <- c(1,16)
> newFrame <- cbind(newFrame, ID)
```

```
> newFrame
  height weight ID
1      58    115  1
2      59    117 16
3      60    120  1
4      61    123 16
5      62    126  1
6      63    129 16
7      64    132  1
8      65    135 16
9      66    139  1
10     67    142 16
11     68    146  1
12     69    150 16
13     70    154  1
14     71    159 16
15     72    164  1
16     76    170 16
```



```

> new.newFrame <- newFrame[,-3]
> new.newFrame
  height weight
1      58    115
2      59    117
3      60    120
4      61    123
5      62    126
6      63    129
7      64    132
8      65    135
9      66    139
10     67    142
11     68    146
12     69    150
13     70    154
14     71    159
15     72    164
16     76    170

> names(new.newFrame)[2] <- "WEIGHT"
> new.newFrame[1,"height"] <- "111"
> new.newFrame
  height WEIGHT
1     111    115
2      59    117
3      60    120
4      61    123
5      62    126
6      63    129
7      64    132
8      65    135
9      66    139
10     67    142
11     68    146
12     69    150
13     70    154
14     71    159
15     72    164
16     76    170

> new.newFrame$WEIGHT <- as.factor(new.newFrame$WEIGHT)
> subset(new.newFrame, height > 60)
  height WEIGHT
4      61    123
5      62    126
6      63    129
7      64    132
8      65    135
9      66    139
10     67    142
11     68    146
12     69    150
13     70    154
14     71    159
15     72    164
16     76    170

```

5. Выбрать один из столбцов и найти его среднее значение.

```

> newFrame
  height weight
1      58    115
2      59    117
3      60    120
4      61    123
5      62    126
6      63    129
7      64    132
8      65    135
9      66    139
10     67    142
11     68    146
12     69    150
13     70    154
14     71    159
15     72    164
> mean(newFrame[,c("weight")])
[1] 136.7333

```

6. Создать новый dataframe и сохранить в нее значения тех строк, для которых значение больше среднего из пункта 5.

```

> new.newFrame <- subset(newFrame, newFrame$weight > mean(newFrame[,c("weight")]))
> new.newFrame
  height weight
9      66    139
10     67    142
11     68    146
12     69    150
13     70    154
14     71    159
15     72    164

```