

Задание векторов и матриц в R (RStudio)

Объявление векторов

Напомним, что в языке R во всех именах переменных / объектов / пакетов / функций различаются строчные и прописные буквы, то есть переменные с именами «a» и «A» – разные.

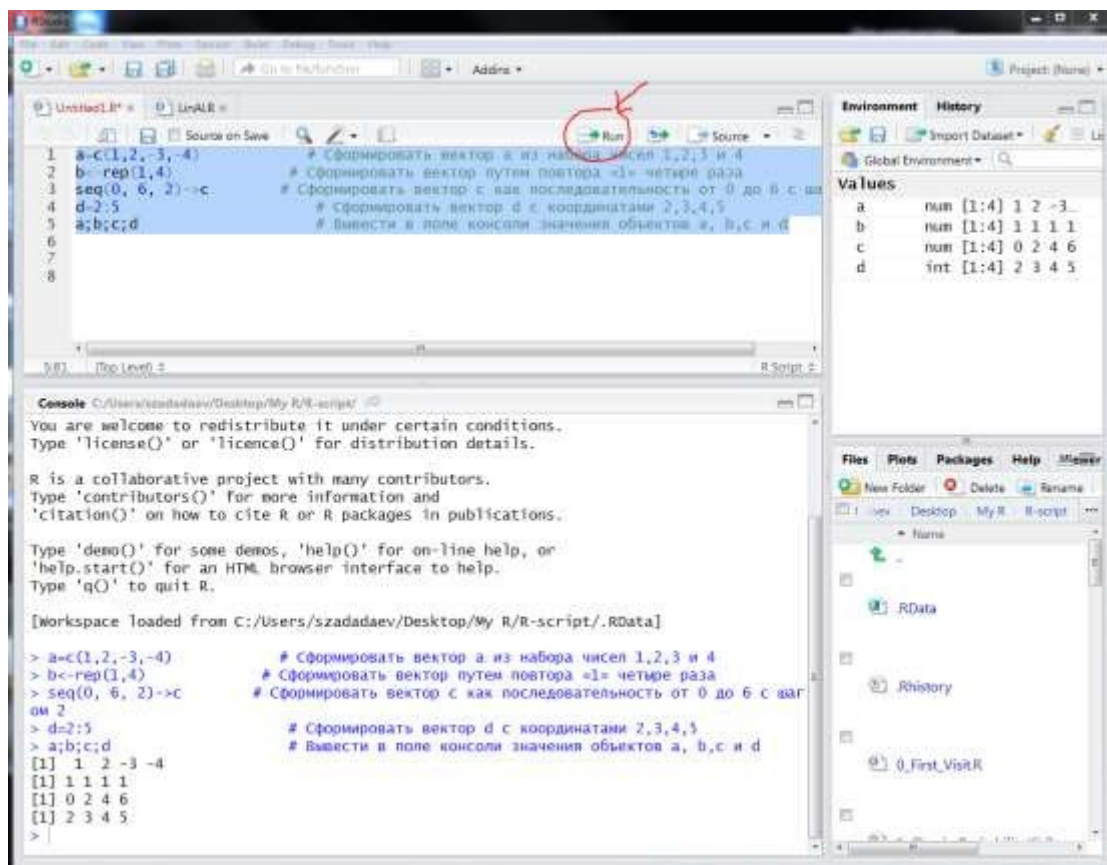
Задание 1. Образовать в R векторы:

$$\vec{a} = \begin{pmatrix} 1 \\ 2 \\ -3 \\ -4 \end{pmatrix}, \vec{b} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \vec{c} = \begin{pmatrix} 0 \\ 2 \\ 4 \\ 6 \end{pmatrix} \text{ и } \vec{d} = \begin{pmatrix} 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}.$$

Решение. Составим следующий текст в левом верхнем поле нового документа (можно просто скопировать эти 5 строчек в область рабочего листа RStudio, но лучше все-таки ввести с клавиатуры):

```
a = c(1, 2, -3, -4)    # Сформировать вектор a из набора чисел 1,2,-3 и -4
b <- rep(1, 4)         # Сформировать вектор b путем повтора «1» четыре раза
seq(0, 6, 2) -> c      # Сформировать вектор c как последовательность от 0 до 6 с шагом 2
d = 2:5               # Сформировать вектор d с координатами 2, 3, 4, 5
a; b; c; d            # Вывести в поле консоли значения объектов a, b, c и d
```

Если теперь в RStudio выделить этот фрагмент и нажать мышкой кнопку Run, то снизу на листе консоли отобразятся действия всех выделенных команд, в том числе, и значения образованных четырех векторов:



Впрочем, текущие значения данных объектов автоматически отображаются на правом верхнем листе «Environment». Выделенный текст можно было запустить на компиляцию и с клавиатуры, нажав одновременно Ctrl + Enter. Если нажать это сочетание без предварительного выделения текста кода, то на компиляцию будет отправлена вся строка, где стоит курсор мыши, так называемая *построчечная* компиляция.

Из приведенных примеров нетрудно вспомнить, что оператор присваивания в R допускает три формы: (=), (->) и (<-); оператор c(...) формирует вектор из перечисляемых внутри аргументов (чисел), оператор rep(...) реализует повторы, seq(...) – задает последовательность, а знак решетки (#) является оператором комментария.

Заметим, что лучше не использовать для имени переменной символ «с», т.к. под символом «с» зарезервирована команда объединения в вектор c(...).

Замечание. Не забудьте, что символ «с» одинаков в русской и латинской раскладках и при неаккуратном использовании раскладок ошибку очень сложно заметить, особенно в

тех случаях, когда на компьютере используется автоматический переключатель клавиатуры.

Обращение к координатам векторов с целью их считывания или изменения здесь вполне естественно:

```
a[2] <- 10 # Записать во вторую координату ранее объявленного вектора a число 10
```

При наборе длинных выражений или имен функций полезно нажать клавишу Tab, по которой RStudio предложит для выбора возможные продолжения фразы или доступные аргументы набранной функции.

Вообще, нажимайте Tab почаще, чтобы избежать рутинных наборов с клавиатуры.

Укажем еще один вариант задания векторов – наследование из массивов:

```
x <- array(2, dim = c(1,10)); x # Объявляем одномерный массив x из двоек  
s <- as.vector(x); s           # Объявляем вектор s, составленный из элементов массива x
```

После выполнения этих двух команд компилятор выдаст в левом нижнем окне консоли сообщения о формировании массива *x* и вектора *s*:

```
> x <- array(2,dim = c(1,10)); x # Объявляем одномерный массив x из двоек  
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]  
[1,]    2    2    2    2    2    2    2    2    2    2  
> s <- as.vector(x); s           # Объявляем вектор s, составленный из элементов  
массива x  
[1] 2 2 2 2 2 2 2 2 2 2
```

Кстати, массив здесь может быть не только одномерным.

Объявление матриц

В пакете R существует достаточно большое количество способов образования матриц, укажем наиболее популярные.

Задание 2. Образовать в R три матрицы:

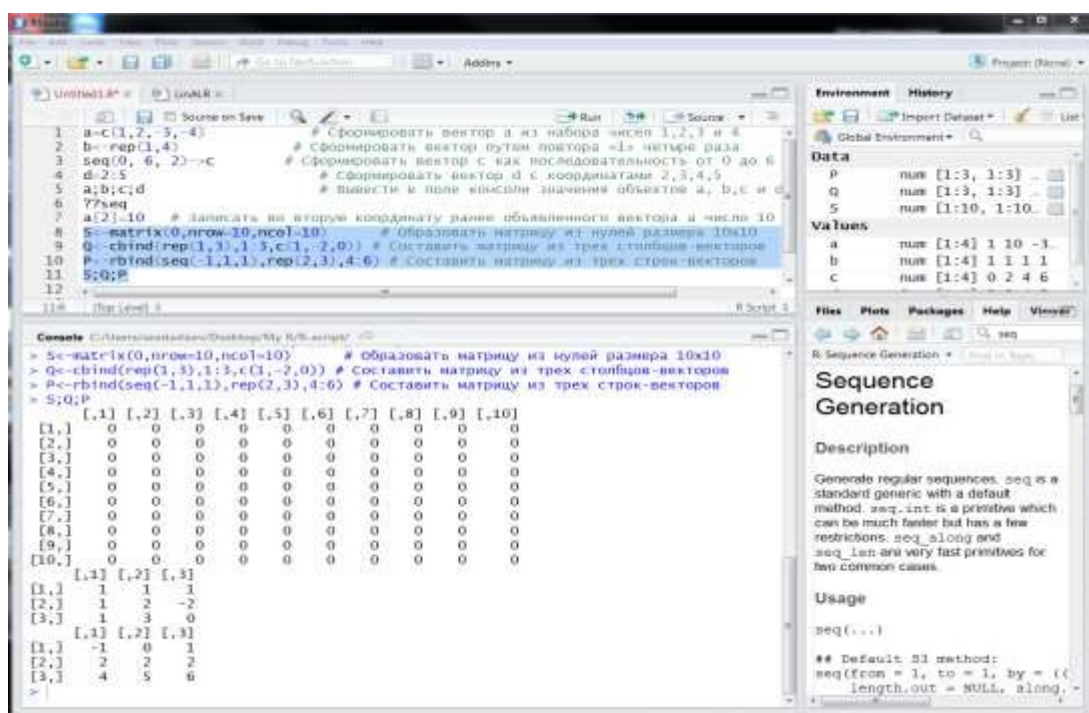
S — нулевая матрица, размера (10×10) ;

$$Q = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & -2 \\ 1 & 3 & 0 \end{pmatrix} \quad \text{и} \quad P = \begin{pmatrix} -1 & 0 & 1 \\ 2 & 2 & 2 \\ 4 & 5 & 6 \end{pmatrix}.$$

Решение. Составим следующий текст программы:

```
S <- matrix(0, nrow=10, ncol=10) # Образовать матрицу из нулей размера 10x10
Q <- cbind(rep(1,3), 1:3, c(1,-2,0)) # Составить матрицу из трех столбцов-векторов
P <- rbind(seq(-1,1,1), rep(2,3), 4:6) # Составить матрицу из трех строк-векторов
S; Q; P # Вывести в поле консоли значения объектов S, Q и P
```

Обратите внимание на разницу формирования матриц в последних двух случаях `rbind` и `cbind` (*row* – строка, *col* – столбец):



Отметим, что числовые массивы фактически уже являются матрицами:

```
x <- array(2,dim = c(3,5)); x # Объявляем одномерный массив x из двоек
s <- as.matrix(x); s # Объявляем одномерный массив x вектором s
```

```
> x <- array(2,dim = c(3,5)); x # Объявляем одномерный массив x из двоек
      [,1] [,2] [,3] [,4] [,5]
[1,]    2    2    2    2    2
[2,]    2    2    2    2    2
[3,]    2    2    2    2    2
> s <- as.matrix(x); s # Объявляем одномерный массив x вектором s
      [,1] [,2] [,3] [,4] [,5]
[1,]    2    2    2    2    2
[2,]    2    2    2    2    2
[3,]    2    2    2    2    2
```

Тем не менее, наиболее удобным и универсальным оказывается способ, основанный на конвертации матриц из текстовых файлов или из MS Excel.

Задание 3. Образовать в Excel матрицу:

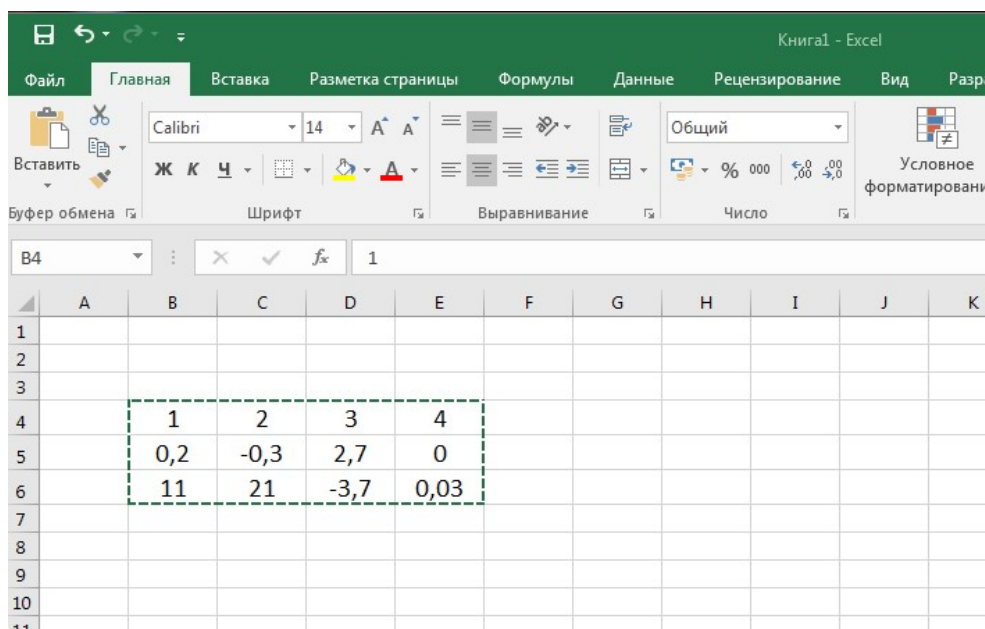
$$W = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 0,2 & -0,3 & 2,7 & 0 \\ 11 & 21 & -3,7 & 0,03 \end{pmatrix},$$

Загрузить ее в R и образовать из первых трех ее столбцов матрицу A, а из последнего столбца – матрицу (столбец) B.

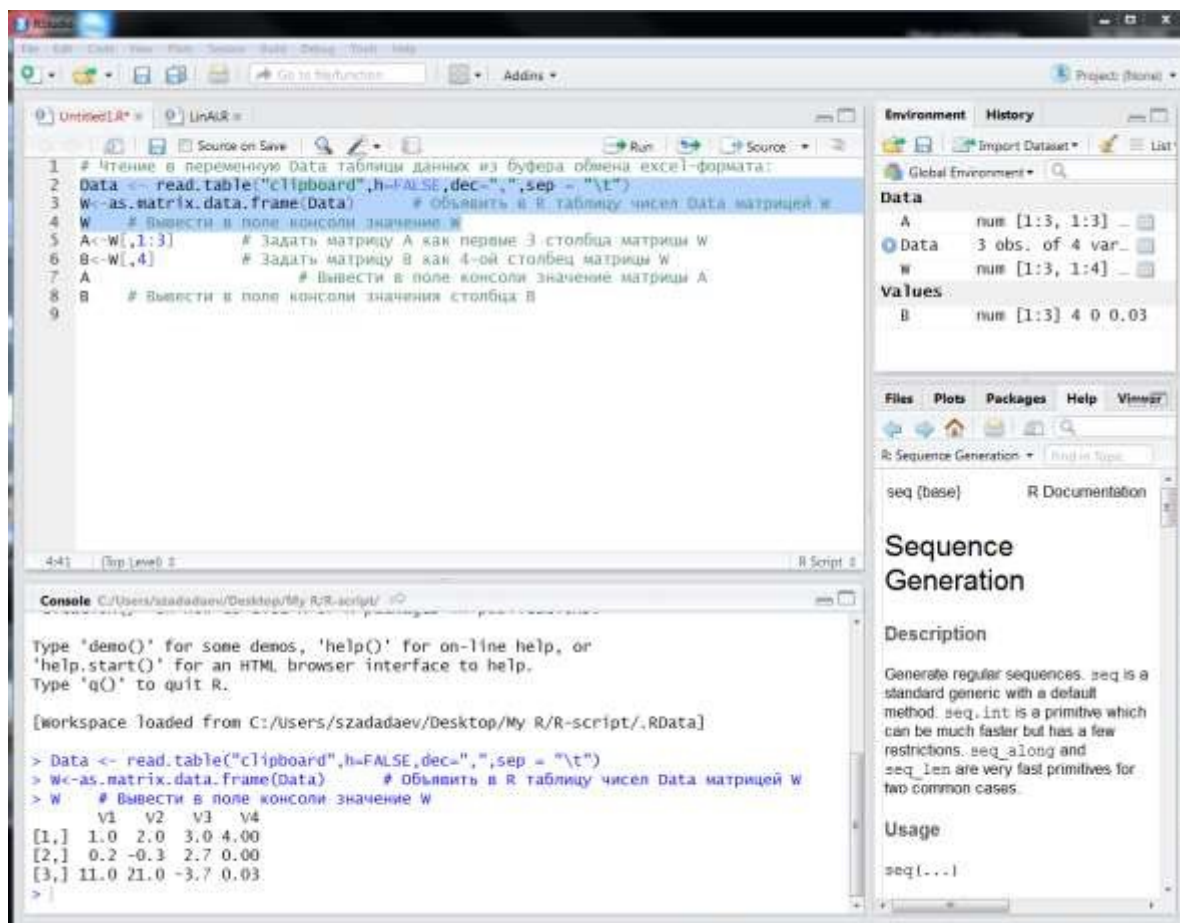
Решение. Скопируем на рабочий лист R-код «считывания» из буфера обмена данных Excel-формата:

```
# Чтение в переменную Data таблицы данных из буфера обмена excel-формата:  
Data <- read.table("clipboard", h=FALSE, dec=".", sep = "\t")  
W <- as.matrix.data.frame(Data) # Объявить в R таблицу чисел Data матрицей W  
W # Вывести в поле консоли значение W
```

Но прежде чем запускать этот код на компиляцию, сформируем в файле Excel требуемую числовую матрицу W , выделим ее и скопируем в буфер обмена комбинацией Ctrl+c:



Теперь в RStudio выделяем наш код и запускаем Run.



В результате переменная W содержит нашу Excel-матрицу.

Замечание. Обратите внимание, что десятичным разделителем в Excel является запятая, а в R – точка. Об этом мы особо указали в аргументах функции чтения:

```
Data <- read.table("clipboard",h=FALSE,dec="," ,sep = "\t")
```

$h = \text{FALSE}$ – отключить заголовки столбцов;

$dec=","$ – десятичным разделителем в данных является запятая (в excel);

$sep = "\t"$ – разделителем чисел в данных является символ табуляции (в excel).

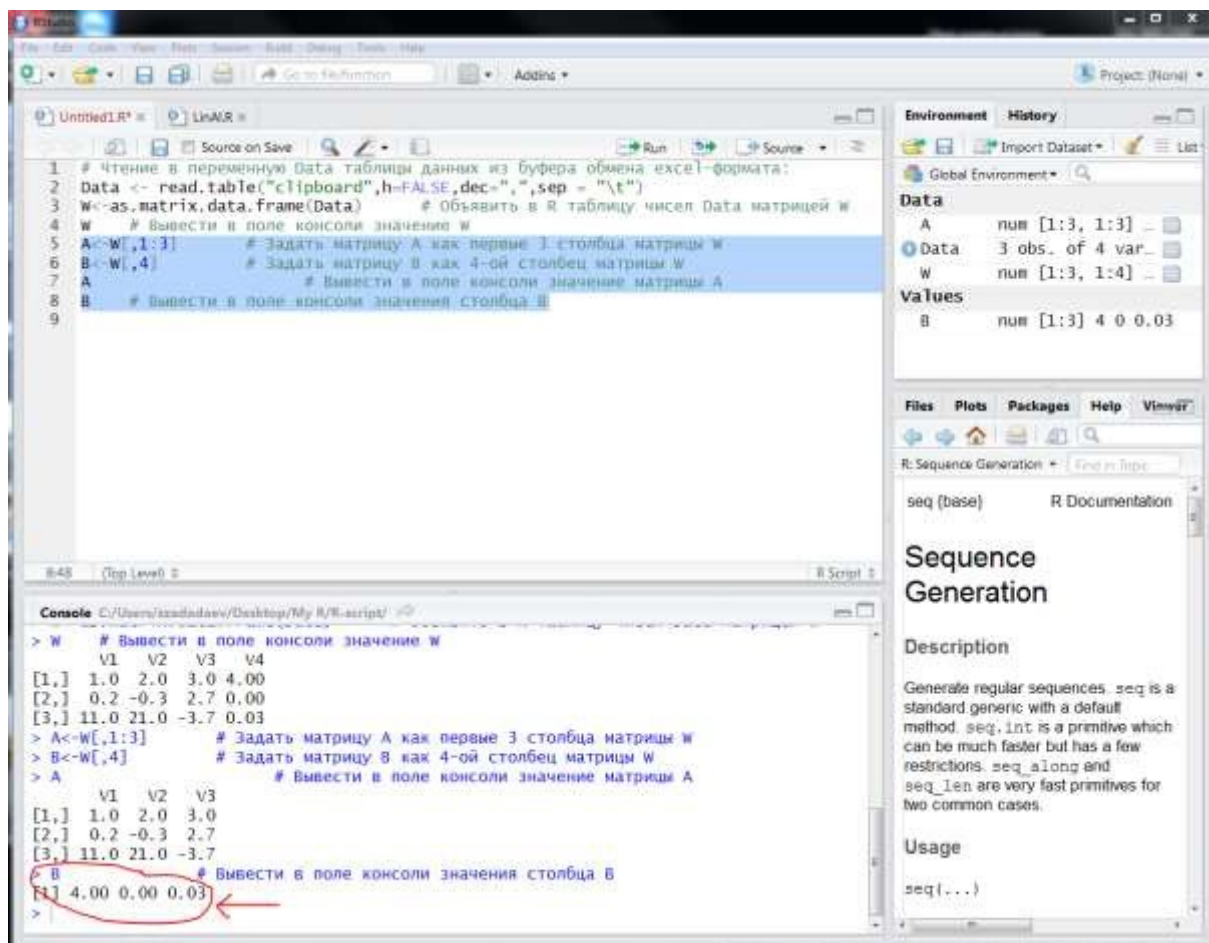
Далее нам остается из уже имеющейся матрицы W образовать требуемые матрицы A и B (B – фактически будет являться вектором):

```

A <- W[,1:3] # Задать матрицу A как первые 3 столбца матрицы W
B <- W[,4]   # Задать матрицу B как 4-ой столбец матрицы W
A           # Вывести в поле консоли значение матрицы A
B           # Вывести в поле консоли значения столбца B

```

Обратите внимание, что вектор B для экономии места консоли представлен в виде строки, хотя, в действительности, он заявлен как вектор:



В этом нетрудно убедиться, если попытаться отправить матричные результаты расчетов обратно в Excel. Это легко сделать так же через буфер обмена с помощью строки:

```
# Скопировать в буфер обмена объект B в Excel-формате:
write.table(B,"clipboard",quote=FALSE, col.names = FALSE, row.names = FALSE, sep = "\t",
dec=".",")
```

Теперь остается не забыть скопировать записанное содержимое буфера обмена обратно в Excel комбинацией Ctrl+v:

3				
4	1	2	3	4
5	0,2	-0,3	2,7	0
6	11	21	-3,7	0,03
7				
8				
9				
10	4			
11	0			
12	0,03			

Замечание. Вместо "clipboard" во всех функциях чтения/записи может быть указан путь к диску с именем текстового файла. Например, чтобы записать данные переменной Data в текстовый файл aaa.txt необходимо ввести команду:

```
# Запись в файл data.txt содержимого таблицы (матрицы) Data
write.table(Data,"C:/Users/aaa.txt",col.names = TRUE, row.names = FALSE, quote=FALSE, sep
= "\t",dec=",")
```

Здесь важно, что при указании пути к файлу должны использоваться символы (/) или (\\) вместо обычных (\). Такой файл можно открыть и в MS Excel, указав в качестве разделителя знак табуляции. Если вместо пути указано только имя файла, результат будет записан в рабочую директорию RStudio.

Замечание. Получить путь к файлу можно следующим образом: укажите на требуемый файл мышью и, удерживая нажатыми Ctrl+Shift, кликните правой клавишей мыши. Из развернувшегося меню необходимо выбрать пункт «Копировать как путь» – теперь в буфере обмена находится путь к файлу, его можно скопировать в нужное место кода RStudio по Ctrl+v и не забыть заменить все символы (\) на (/) или на (\\).

Замечание. Не забываем, что для более детального изучения всех возможностей каких-либо операторов или пакетов языка R необходимо нажать клавишу F1 при поставленном курсоре перед нужной командой или составить команду по следующему правилу:

```
??seq или help(seq)           # Вызов справки, например, по функции seq.
```

Задания для самостоятельной работы

1. Образовать и вывести на экран вектор a, содержащий 50 одинаковых координат, равных 3.

2. Образовать и вывести на экран вектор b , содержащий 100 координат 1,2,...,100.
3. Образовать квадратную матрицу A , состоящую из последовательности чисел 1,2,...,16, которая заполняет матрицу
 - а) по столбцам;
 - б) по строкам.

Подсказка: при вызове команды «matrix» использовать специальный параметр «byrow», о котором можно узнать из справки по функции «matrix».

4. Решить предыдущие упражнения 1–3 с помощью MS Excel и буфера обмена.
5. Образовать и вывести на экран консоли матрицу Гильберта G размера (10x10), элементы которой равны сумме соответствующих строки и столбца: $G_{ij} = i + j$;
6. Образовать в R единичную матрицу размера (40x40) и перенести ее через буфер обмена в excel- таблицу.