

Ввод-вывод данных

Данные на обработку средствами R можно непосредственно задать и программе, используя, например, функцию `c()` или ввести из различных внешних источников:

- текстовых файлов;
- файлов, сохраненных в различных форматах табличного процессора MS Excel;
- специализированных статистических программ (IBM SPSS Statistics, Statistica, STATA и др.);
- баз данных (MS Access, Oracle и др.).

Второй вариант является основным, поскольку позволяет провести исследование сверхобъемных массивов информации, непосредственное создание которых в R-программе крайне затруднительно, да и нерационально, тем более, если есть возможность использовать данные, сформированные в ходе обработки другими программами,

Отметим некоторые особенности импортирования данных из внешних источников:

- в импортируемой таблице не должно быть пустых ячеек; если какие-то значения по тем или иным причинам отсутствуют, вместо них следует ввести NA (нет данных);
- в качестве первой строки в импортируемой таблице рекомендуется ввести заголовки столбцов-переменных; если данная строка отсутствует, то об этом следует указать при описании параметров функции, с помощью которой будет выполняться импорт внешнего последующие строки импортируемой таблицы в качестве первого элемента могут содержать заголовки строк, после которых должны следовать значения переменных;
- в именах столбцов таблицы и заголовках строк не допускаются пробелы; имена обязательно должны начинаться с буквы;
- во избежание проблем, связанных с кодировкой, текстовые элементы в импортируемых файлах рекомендуется создавать с использованием букв *латинского* алфавита;

- подлежащий импортированию файл рекомендуется поместить в текущую папку программы, установленную функцией `setwd()`;
- для облегчения выполнения импорта из внешнего файла рекомендуется преобразовать импортируемую таблицу с данными в простой текстовый файл с расширением `.txt` или `.csv` (хотя это не обязательно).

Вывод результатов выполнения R-программы также может быть осуществлен различным образом:

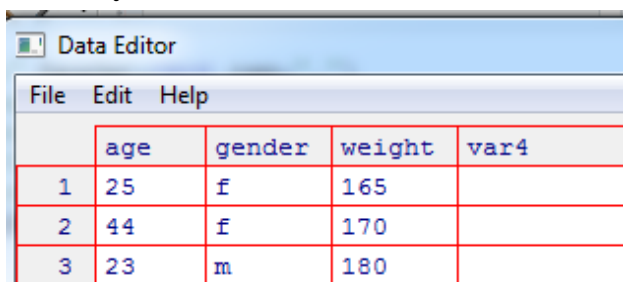
- на экран;
- на печать;
- в файлы различных форматов.

Импорт данных из текстовых файлов

Самый простой способ введения данных – это ввод с клавиатуры. Для этого необходимо создать пустую таблицу данных (или матрицу), указав названия и типы переменных. Затем открывается текстовый редактор функцией `edit()` с этим объектом, вводятся данные и сохраняется результат в виде объекта с данными.

В приведенном ниже примере создается таблица данных с названием `mydata` с тремя переменными: `age` (возраст, числовая), `gender` (пол, текстовая) и `weight` (вес, числовая). Затем откроется текстовый редактор, вносятся данные и сохраняется результат.

```
> mydata <- data.frame(age=numeric(0),
+gender=character(0), weight=numeric(0))
> mydata <- edit(mydata)
```



	age	gender	weight	var4
1	25	f	165	
2	44	f	170	
3	23	m	180	

Рисунок 1 Копия объекта `mydata` открытая в редакторе данных функцией `edit()`

Для импорта данных из внешних источников, представляющих собой

текстовые файлы, применяют функции `read.table()`, `read.csv()` и др. Рассмотрим форматы данных функций.

Функция `read.table()` обеспечивает считывание таблицы данных из внешнего источника – текстового файла:

```
read.table(file, header = FALSE, sep = "", quote = "\"",  
           dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),  
           row.names, col.names, as.is = !stringsAsFactors,  
           na.strings = "NA", colClasses = NA, nrows = -1,  
           skip = 0, check.names = TRUE, fill = !blank.lines.skip,  
           strip.white = FALSE, blank.lines.skip = TRUE,  
           comment.char = "#",  
           allowEscapes = FALSE, flush = FALSE,  
           stringsAsFactors = default.stringsAsFactors(),  
           fileEncoding = "", encoding = "unknown", text, skipNul = FALSE)
```

Раскроем назначение основных параметров.

Первый позиционный параметр `file` представляет собой строку, содержащую полный путь и импортируемому файлу, либо является собственно именем файла, находящемся в текущей папке.

Ключевой параметр `header` может принимать значение `TRUE` (истина) или `FALSE` (ложь, по умолчанию). Если его значение истинно, следовательно, импортируемая таблица из внешнего файла содержит строку заголовков столбцов. Если заголовки столбцов представляют собой числа, к ним добавляется литера, чтобы имя начиналось с буквы.

Параметр `sep` позволяет задать разделитель между данными в строке импортируемой таблицы. Если значение `sep = ""`, разделителем является один или несколько пробелов, знак табуляции или возврат каретки.

Параметр `dec` определяет знак – разделитель целой и дробной части числа. По умолчанию разделителем является точка (`dec = "."`); для таблиц, подготовленных в России и большинстве европейских стран – запятая (`dec = ","`).

Значением параметра `row.names` является вектор, содержащий имена строк таблицы. Если в импортируемой таблице есть заголовок и первая строка содержит на одно поле меньше, чем количество столбцов, первый столбец во второй и последующих строках воспринимается, как имя строки. Если имена строк отсутствуют, строки нумеруются.

Параметр `col.names` указывает на вектор имен переменных. Если имена переменных не определены, по умолчанию они формируются из

символа V, за которым следует номер столбца.

Параметр fill в случае истинного значения позволяет заполнять более короткие строки импортируемой таблицы пустыми полями.

Пример

Пусть имеется текстовый файл Primer01.txt, в котором содержится таблица с числовыми данными. Полный путь к файлу известен. Заголовки строк отсутствуют. Необходимо выполнить ввод числовых значений в R-программу с одновременным выводом данных на экран.

Решение

Текст скрипта:

Ввод и вывод на экран числовой таблицы из текстового файла Primer01.txt

по полному пути к файлу

```
> chem<-read.table (file = "D:/R/data/Primer01.txt", sep=" ",header = TRUE)
> chem
  year wheat barley oats rye corn
1 1980   5.9    4.4  4.1 3.8    1
2 1981   5.8    4.4  4.3 3.7    1
3 1982   6.2    4.9  4.4 4.1    2
4 1983   6.4    4.7  4.3 3.7    3
5 1984   7.7    5.6  4.9 4.7    3
```

Рисунок 2 Фрагмент таблицы вывода на экран по полному пути файла

Пример

Пусть имеется текстовый файл Primer01.txt, находящийся в текущей папке и в котором содержится таблица с числовыми данными. Заголовки строк отсутствуют. Необходимо выполнить ввод числовых значений в R-программу с одновременным выводом данных на экран.

Решение

Текст скрипта:

Ввод и вывод на экран числовой таблицы из текстового файла Primer01.txt , находящегося в текущей папке.

Протокол выполнения скрипта:

```
> setwd("D:/R/data")
> chem<-read.table (file = "Primer01.txt", sep=" ",header = TRUE)
> chem
  year wheat barley oats rye corn
1 1980   5.9    4.4  4.1 3.8    1
2 1981   5.8    4.4  4.3 3.7    1
3 1982   6.2    4.9  4.4 4.1    2
4 1983   6.4    4.7  4.3 3.7    3
5 1984   7.7    5.6  4.9 4.7    3
```

Рисунок 3 Фрагмент вывода на экран числовой таблицы

Пример

Пусть в текстовом файле Primer01.txt требуется после ввода

импортируемой таблицы в среду R выполнить замену разделителя целой и дробной части чисел с запятой на точку и сохранить полученный результат в таблице.

Протокол выполнения скрипта:

```
> chem<-read.table (file = "D:/R/data/Primer01.txt", sep="," ,header = TRUE, dec = ",")
> chem
  year wheat barley oats rye corn
1 1980   5.9    4.4  4.1 3.8    1
2 1981   5.8    4.4  4.3 3.7    1
3 1982   6.2    4.9  4.4 4.1    2
4 1983   6.4    4.7  4.3 3.7    3
5 1984   7.7    5.6  4.9 4.7    3
```

Рисунок 4 Фрагмент вывода на экран числовой таблицы с на точку

Применение буфера обмена при импорте данных

Если перед выполнением соответствующей операции, находясь в среде любой программы – Excel, Word, Access и др., скопировать в Буфер обмена операционной системы какие-то данные, например, таблицу, то легко и безошибочно эти данные будут перемещены в среду R.

Причем ошибки не возникнут, даже если в таблице используются буквы русского алфавита и имеются другие ограничения (например, целая часть от дробной отделяется запятой, а не точкой). Ошибки будут возникать только в случае пробелов в текстовых названиях и когда копируются данные в Буфер обмена из файлов *.csv.

Пример

Пусть имеется таблица MS Excel. В числовых данных дробная часть от целой отделена запятой. Требуется с использованием Буфера обмена ввести данную таблицу в среду R. Рассмотреть возможные варианты решения, в том числе, если таблица размещена в текстовом документе MS Word.

Решение

Первый вариант: в окне редактора RStudio наберем требуемые операторы:

```
> Data<-read.table("clipboard",h=TRUE,dec = ",",sep = "\t")
> Data
  x15.26..14.84..0.871..5.763..3.312..2.221
1 14.88, 14.57, 0.8811, 5.554, 3.333, 1.018
2 14.29, 14.09, 0.905, 5.291, 3.337, 2.699
3 13.84, 13.94, 0.8955, 5.324, 3.379, 2.259
4 16.14, 14.99, 0.9034, 5.658, 3.562, 1.355
5 14.38, 14.21, 0.8951, 5.386, 3.312, 2.462
```

Рисунок 5 Протокол выполнения скрипта

Второй вариант: в окне редактора RStudio наберем требуемые операторы:

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

	v1
1	15.26, 14.84, 0.871, 5.763, 3.312, 2.221
2	14.88, 14.57, 0.8811, 5.554, 3.333, 1.018
3	14.29, 14.09, 0.905, 5.291, 3.337, 2.699
4	13.84, 13.94, 0.8955, 5.324, 3.379, 2.259
5	16.14, 14.99, 0.9034, 5.658, 3.562, 1.355
6	14.38, 14.21, 0.8951, 5.386, 3.312, 2.462

Рисунок 6 Протокол выполнения скрипта

Если исходная таблица представлена в текстовом документе MS Word (или в файле другого формата), то допустимы ее копирование а Буфер обмена и отправка на выполнение в окне редактора RStudio команд:

```
> read.table("clipboard")
      v1 v2 v3 v4 v5
1 1357 23 14 45 66
```

Рисунок 7 Протокол выполнения скрипта

```
> Data<-read.table("clipboard",h=FALSE,dec = ".",sep = "",skip=1)
> dim(Data)
[1] 10 7
> Data
```

	v1	v2	v3	v4	v5	v6	v7
1	19,2	19,1	21,9	23,1	23,5	25,7	27,3
2	21,0	20,3	23,5	24,6	25,1	26,9	27,3
3	17,5	16,4	17,7	18,1	18,6	20,4	21,5
4	16,3	16,3	16,8	17,9	17,8	19,2	20,2
5	20,0	18,3	20,2	21,3	20,9	22,5	25,5
6	20,6	20,4	22,4	23,6	23,7	26,1	26,4
7	28,3	27,7	29,8	30,8	31,0	33,4	37,5
8	20,7	20,4	22,6	23,9	24,4	25,9	26,8
9	16,6	16,7	18,1	19,5	19,8	22,2	24,7
10	14,7	14,8	16,5	17,0	16,5	18,4	20,1

```
> Data[2,]
      v1 v2 v3 v4 v5 v6 v7
2 21,0 20,3 23,5 24,6 25,1 26,9 27,3
> Data[1,3]
[1] 21,9
Levels: 16,5 16,8 17,7 18,1 20,2 21,9 22,4 22,6 23,5 29,8
>
```

Рисунок 8 Протокол выполнения скрипта

Анализ протоколов импорта табличных данных с использованием Буфера обмена показывает, что данный вариант является наиболее приемлемым, так как позволяет работать с каждым элементом введенной таблицы за счет сохранения результатов импортирования в фреймовом объекте.

Импорт из EXCEL-файлов

Следует отметить, что целесообразнее перед запуском RStudio запускать загрузчик RGui. Такой сценарий обеспечит более широкие возможности по включению в разрабатываемые R-скрипты инструментария различных дополнительных пакетов из многочисленных внешних источников хранения – так называемых зеркал, физически расположенных в разных уголках мира.

Если предполагается в R-скрипте непосредственная обработка файлов, подготовленных в MS Excel (с расширением .xlsx для версий табличного процессора, начиная с 2007 года, или .xls – для более ранних версий), требуется предварительно скачать и установить пакет Ms*.

Перед установкой пакета xlsx, а также некоторых других пакетов из репозитория R, следует убедиться, что на компьютере установлены язык и вычислительная платформа Java, работающие на любой архитектуре – от стационарных компьютеров до мобильных устройств, и используемые при выполнении различных онлайн-приложений, обеспечивая их функциональность, быстродействие, безопасность и надежность.

Бесплатно загрузить Java можно, например, с сайта <https://www.java.com/ru/download/>, выбрав соответствующую операционную систему.

Скачать и установить пакет xlsx можно как в окне RStudio, последовательно выполнив команды `install.packages("xlsx")` и `library(xlsx)`, либо в окне загрузчика RGui. Последний вариант является более предпочтительным, так как позволяет избежать ситуаций, когда отсутствует доступ к репозиторию облачного хранилища CRAN (этому могут быть разные причины), из-за чего могут быть получены некорректные результаты инсталляции.

Рассмотрим последовательность действий в окне среды RStudio.

Вначале необходимо определить зеркало, из которого будет выбираться и скачиваться нужный пакет:

Кнопки Packages / Install.

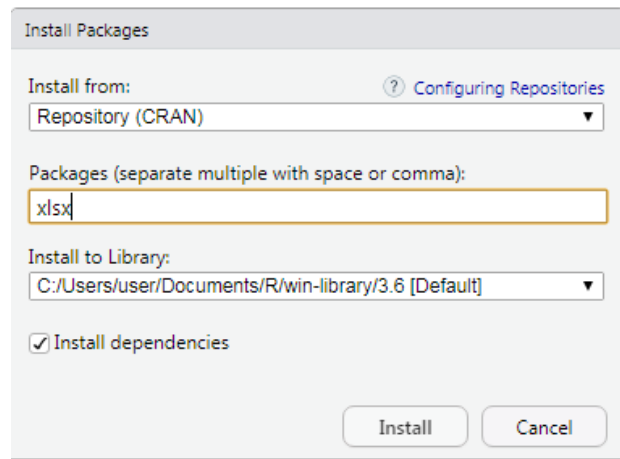


Рисунок 9 Пример скачивания и установки пакета xlsx

Начнется процесс инсталляции.

```
package 'xlsx' successfully unpacked and MD5 sums checked
The downloaded binary packages are in
  C:/Users/user/AppData/Local/Temp/Rtmpw0BZs3/downloaded_packages
> |
```

Рисунок 10 Завершение процесса инсталляции пакета xlsx

После рассмотрения и выполнения на компьютере указанных предварительных операций перейдем к описанию применения возможностей пакета xlsx.

Для непосредственного импорта в среду R нужных листов из Excel-файлов с расширениями .xlsx либо .xls используется функция read.xlsx():

```
read.xlsx(
  xlsxFile,
  sheet = 1,
  startRow = 1,
  colNames = TRUE,
  rowNames = FALSE,
  detectDates = FALSE,
  skipEmptyRows = TRUE,
  skipEmptyCols = TRUE,
  rows = NULL,
  cols = NULL,
  check.names = FALSE,
  sep.names = ".",
  namedRegion = NULL,
  na.strings = "NA",
  fillMergedCells = FALSE
```


)

Позиционный параметр `xlsxFile` представляет собой строковую константу либо переменную, содержащую полный путь к имени Excel файла или имя Excel-файла из рабочей директории, либо URL-адрес Excel-файла.

Назначение основных ключевых параметров:

`sheetindex` – задает имя или номер импортируемого листа Excel-файла;

`startRow` – определяет номер строки, начиная с которой будет осуществляться импорт данных; пустые строки в верхней части файла всегда пропускаются;

`colNames` – при значении `TRUE` устанавливает, что первая строка данных будет использоваться в качестве имен столбцов;

`rowNames` – при значении `TRUE` устанавливает, что первый столбец данных будет использоваться в качестве имен строк;

`detectDates` – при значении `TRUE` будет предпринята попытка распознать и преобразовать даты;

`skipEmptyCols` – при значении `TRUE` предписывает пропускать пустые столбцы;

`rows` – числовой вектор, содержащий номера строк, подлежащие импорту; при значении `NULL` импортируются все строки;

`cols` – числовой вектор, содержащий номера столбцов, подлежащие импорту; при значении `NULL` импортируются все столбцы;

`check.names` – при значении `TRUE` имена переменных во фрейме данных проверяются на предмет их синтаксической допустимости именам переменных.

В качестве примечания, напомним, что наличие в таблицах имен строк либо столбцов в русскоязычном варианте может вызвать из-за используемой на компьютере таблицы кодировки появление нечитаемых надписей. Во избежание такой проблемы следует в файлах использовать имена, набранные латиницей.

Пример

Пусть имеется Excel-файл `Зарплата_ЦФО.xls`, в котором на листе 2 содержится таблица. Известен полный путь к имени файла. Имена строк в таблице записаны латиницей, за исключением одного значения. Собственно таблица начинается с 4-й строки.

Используя функцию `readxlsx()`, выполнить ввод таблицы в среду R, присвоив ее значения таблице данных. Вывести результат импорта на экран.

В случае отсутствия RJava воспользоваться командами:

Установите 64 бит Java из <https://java.com/en/download/manual.jsp> .

Затем в windows cmd запустить

```
setx PATH "C:\Program Files\Java\jre1.8.0_211\bin\server;%PATH%"
```

(убедитесь, что ваш путь правильный).

Затем в RStudio запустить

```
Sys.setenv(JAVA_HOME="")
```

Протокол выполнения кода:

```
> Sys.setenv(JAVA_HOME="")
> library(rJava)
предупреждение:
пакет 'rJava' был собран под R версии 3.6.3
> library(xlsx)
предупреждение:
пакет 'xlsx' был собран под R версии 3.6.3
> getwd()
[1] "C:/Users/user/Documents"
> setwd("D:\\R\\data")
> read.xlsx("voenvuz.xls",sheetIndex = 1)
Ошибка в loadWorkbook(file, password = password) :
cannot find voenvuz.xls
> read.xlsx("voenvuz.xlsx",sheetIndex = 1)
  Name Age Height weight Blood.group Rhesus.factor NA.
1  Ivan  23   178    80         2             +   <NA>
2  Peter 18   169    62         1             -   <NA>
3  Oleg  22   185    77         2             +   <NA>
4  Sergey 19   182    73         2             -   <NA>
```

Рисунок 11 Протокол выполнения скрипта

Импорт из файлов *.csv

При работе в R с файлами формата *.csv отрываються более широкие возможности представлен и я обрабатываемых текстовых данных на национальных языках. Кроме того, csv-файлы занимают существенно меньший объем, что снижает нагрузку на требования памяти, так как это фактически текстовые файлы, в которых данные разделяются запятой, а не пробелами или знаками табуляции, а дробная часть чисел отделяется от целой части точкой.

Файлы *.csv можно создать при работе в Excel, если при сохранении выбрать данный формат. Однако прежде чем сохранять Excel-файл в формате *.csv, необходимо в параметрах операционной системы задать необходимые изменения значений разделителей в числовых данных.

Для этого в операционной системы Windows 10 следует выполнить операции:

Кнопка Пуск → Кнопка Параметры → Группа Время и язык→

Ссылка Регион и язык → Ссылка Дополнительные параметры даты и времени, региональные параметры → Ссылка Изменение форматов даты, времени и чисел → Кнопка Дополнительные параметры.., → В окне Настройка формата на вкладке Числа (рис, 4.14) изменить разделитель целой и дробной части на точку, разделитель элементов списка на запятую → Кнопка ОК → Кнопка ОК

После указанных действий можно сохранить Excel-файл в формате *.csv. Для этого при сохранении файла следует указать его новый тип – CSV UTF-8 (разделитель – запятая) (*.csv).

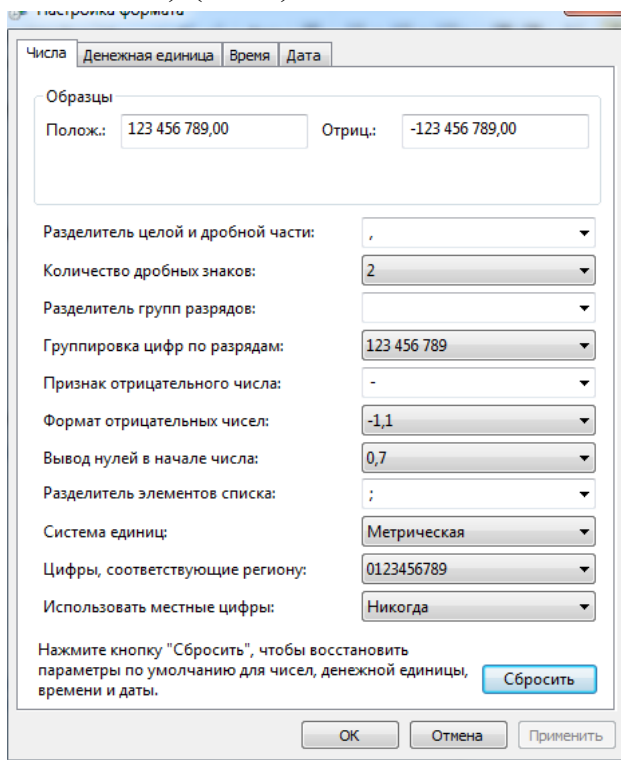


Рисунок 12 Окно изменения параметров разделителей числовых данных

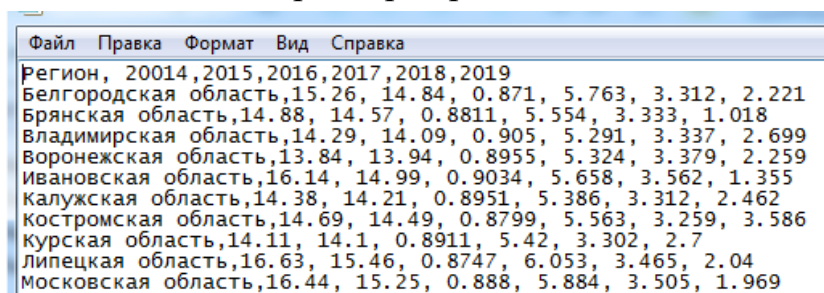


Рисунок 13 Представление Excel-файла в формате *.csv, открытого в блокноте

Для импорта таблиц в среду R из файлов с расширением *.csv используется функция read.csv():

```
read.csv(file, header = TRUE, sep = ",", quote="\"", dec = ".", fill = TRUE,
comment.char = "",...)
```

Назначение параметров данной функции во многом совпадает с

назначением параметров функции read.tablet).

Пример

Пусть имеется файл test2.csv (рис. 11), содержащий информацию о среднемесячной заработной плате работников всех отраслей по регионам Центрального федерального округа РФ за период с 2014 – 2019 гг. Полный путь к файлу известен. Имена строк записаны кириллицей без пробелов. Имена столбцов указаны годами.

Используя функцию read.csv(), выполнить ввод таблицы в среду R. Сохранить таблицу в переменной-фрейме. Вывести различные варианты результатов ввода данных.

```
1 setwd("~/G:/R/data")
2 dat=read.csv("test2.csv",header=TRUE,sep=",")
3 dat# вывод данных таблицы
4 dat[4,5]#вывод данных Воронежской области в 2017 году
5 dat[3]#вывод данных в 2015 году
6 dat[2,]#вывод данных Брянской области по годам
```

Рисунок 14 Текст кода

```
      . . . . .
      Регион x20014 x2015  x2016 x2017 x2018 x2019
1  Белгородская область 15.26 14.84 0.8710 5.763 3.312 2.221
2    Брянская область 14.88 14.57 0.8811 5.554 3.333 1.018
3  Владимирская область 14.29 14.09 0.9050 5.291 3.337 2.699
4  Воронежская область 13.84 13.94 0.8955 5.324 3.379 2.259
5  Ивановская область 16.14 14.99 0.9034 5.658 3.562 1.355
6  Калужская область 14.38 14.21 0.8951 5.386 3.312 2.462
7  Костромская область 14.69 14.49 0.8799 5.563 3.259 3.586
8    Курская область 14.11 14.10 0.8911 5.420 3.302 2.700
9  Липецкая область 16.63 15.46 0.8747 6.053 3.465 2.040
10 Московская область 16.44 15.25 0.8880 5.884 3.505 1.969
> dat[4,5]#вывод данных Воронежской области в 2017 году
[1] 5.324
> dat[3]#вывод данных в 2015 году
x2015
1 14.84
2 14.57
3 14.09
4 13.94
5 14.99
6 14.21
7 14.49
8 14.10
9 15.46
10 15.25
> dat[2,]#вывод данных Брянской области по годам
      Регион x20014 x2015  x2016 x2017 x2018 x2019
2  Брянская область 14.88 14.57 0.8811 5.554 3.333 1.018
```

Рисунок 15 Протокол выполнения скрипта

Данные газодобычи

По ссылке, указанной ниже получить .xlsx файл с данными из открытого репозитория github:

<https://raw.githubusercontent.com/qwerty29544/RpracticeBook/master/2Data/01FlatTables/GAZ.csv>

Таблица EXCEL формата содержит данные о добыче нефти главной российской компанией добытчика данного вида ресурсов. Данные имеют следующую структуру:

1. Дата замера добычи со скважины.
2. Давление в трубе (МПа).
3. Температура установки (°C).
4. Добыча газа (кубометр в сутки).
5. Конденсат (кубометр в сутки).
6. Вода (кубометр в сутки).
7. ID скважины.
8. Куст скважины.
9. Группа скважины.

Задание 1.

1. Произвести импорт данных из файла формата EXCEL. Важно чтобы в результате экспорта все типы данных таблицы воспроизводились, как положено. Дата замера должна быть выражена через стандартный тип данных date в R.
2. Произвести очистку таблицы данных от строк с пустыми значениями признаков.
3. Перевести температуру установки в единицы измерения по Кельвину, удалив при этом первоначальный столбец температуры.
4. Преобразовать данные полей ID, Куст, Группа в ординальный формат данных.
5. Получить новые безразмерные поля, полученные на основе вычисления по старым полям:
 - отношение добычи газа к конденсату;
 - отношение добычи газа к добыче воды;
 - отношение добычи воды к добыче конденсата.
6. Отфильтровать данные измерений и получить выборку добычи на кустах за 2018 год.
7. Получить подвыборку данных измерений с куста по ID = 111
8. Вывести все ID скважин, добыча воды в которых никогда не превышала 2 м³/сут.
9. Вывести ID скважин, суммарная добыча в день в которых не опускалась ниже 1000 м³/сут. Если данному условию удовлетворяют все, или не удовлетворяет ни одна из скважин, то подобрать значение суммарной добычи в день по кусту, в котором окажутся только 3 наименования.
10. Вывести название группы кустов, которая была самой результативной по добыче газа по результатам 2018 года.
11. Вывести название куста, который был самым результативным по

добыче воды по результатам 2018 года.

12. Вывести название куста, в котором среднее отношение добычи нефти к добыче воды было самым наибольшим за всё время наблюдений.

Контрольные вопросы

1. Охарактеризуйте способы ввода данных из внешних источников
2. Каким образом можно ввести информацию в среду R из текстовых файлов
3. Каковы особенности представления файлов в формате .csv
4. Какие функции используются для импорта данных из текстовых файлов
5. Каковы особенности импорта данных из текстовых файлов
6. В чем преимущества импорта данных из Буфера обмена
7. Каковы особенности импорта данных из файлов Excel