УДК 615.036.2

ОСНОВЫ РАБОТЫ В ПРОГРАММНОЙ СРЕДЕ R ПРИ АНАЛИЗЕ БИОМЕДИЦИНСКИХ ДАННЫХ

© 2018 г. [1]В. Л. Егошин, [2]С. В. Иванов, [3]Н. В. Саввина, [4]Г. Ж. Капанова, [3,5]А. М. Гржибовский

[1]Павлодарский филиал Государственного медицинского университета г. Семей, г. Павлодар, Казахстан;
[2]Первый Санкт-Петербургский государственный медицинский университет им. акад. И. П. Павлова, г. Санкт-Петербург; [3]Северо-Восточный федеральный университет им. М. К. Аммосова, г. Якутск;
[4]Казахский Национальный Университет им. аль-Фараби, г. Алматы, Казахстан;
[5]Северный государственный медицинский университет, г. Архангельск

Статья знакомит с основными принципами работы программной среды R в применении к обработке исследовательских данных. Описаны типы переменных и типы данных, анализируемых программой, описаны алгоритмы импорта, ввода, преобразования данных, вывода результатов анализа, работы с векторами и таблицами. Дано представление о работе программной оболочки RStudio.

Ключевые слова: статистических анализ данных, R, RStudio

BASIC PRINCIPLES OF BIOMEDICAL DATA ANALYSIS IN R

[1]V. L. Egoshin, [2]S. V. Ivanov, [3]N. V. Savvina, [4]G. Zh. Kapanova, [3,5]A. M. Grjibovski

[1]Semey State Medical University, Pavlodar Campus, Pavlodar, Kazakhstan; [2]I. P. Pavlov First St. Petersburg State Medical University, St. Petersburg, Russia; [3]M. K. Ammosov North-Eastern Federal University, Yakutsk, Russia;
[4]Al-Farabi Kazakh National University, Almaty, Kazakhstan; [5]Northern State Medical University, Arkhangelsk, Russia

The article presents basic principles of using R software for biomedical data analysis. Types of variables and main principles of their analysis are described. The algorithms for importing, inputting, transforming data, presenting results, working with vectors and tables are presented. The RStudio software using is presented as well.

Keywords: statistical analysis, R, RStudio

Библиографическая ссылка:

Егошин В. Л., Иванов С. В., Саввина Н. В., Капанова Г. Ж., Гржибовский А. М. Основы работы в программной среде R при анализе исследовательских данных // Экология человека. 2018. № 7. С. 55–64.

Egoshin V. L., Ivanov S. V., Savvina N. V., Kapanova G. Zh., Grjibovski A. M. Basic Principles of Biomedical Data Analysis in R. *Ekologiya cheloveka* [Human Ecology]. 2018, 7, pp. 55-64.

Анализ данных – составная часть любого исследования, так как именно на этапе анализа данных происходит переход от полученных результатов к пониманию имеющейся в них информации и далее к новому научному знанию. Практически ни одно исследование в медицине не может быть рассмотрено как имеющее научную доказательность, если выводы по итогам проведения исследования не подтверждены результатами статистической обработки данных [6]. Широкое внедрение компьютерных программ позволило существенно ускорить процесс анализа исследовательских данных, а также выполнить ряд других функций. В частности, специализированное программное обеспечение помогает подготовить данные для анализа и осуществить его воспроизводимым образом.

В рутинной практике наиболее часто для статистического анализа данных используются такие программы, как SPSS, Statistica, STATA, SAS, MatLab и др. Полноценной бесплатной альтернативой этим коммерческим продуктам является бесплатно распространяемая программная среда R, которая по сути является языком программирования. Созданный более 20 лет назад, этот язык успешно развивается и поддерживается научным сообществом во всех областях научного поиска. Язык R хорошо документирован, и для обучения пользователей существуют интернет-ресурсы [1, 3, 4, 7–9]. О росте популярности языка R свидетельствует тот факт, что в рейтинге языков программирования TIOBE, составляемом на основе частоты поисковых запросов, в апреле 2018 года язык R находился на 12-м месте, причем рейтинг вырос за год на три позиции [10]. В этом рейтинге из числа языков программирования, обычно используемых при анализе данных, его опережает только «Python», но этот факт объясняется тем, что сфера применения «Python» существенно шире сферы применения R. Все вышесказанное позволяет назвать R самым популярным языком программирования в сфере статистических вычислений.

R, являясь высокоуровневым скриптовым языком программирования, представляет собой программную среду, в которой возможно проведение всего процессинга обработки данных – от подготовки до непосредственно анализа и выведения результатов в удобной форме. Работая в R, исследователь вы-

полняет необходимую трансформацию данных, их визуализацию, расчет показателей, создание моделей. Использование набора скриптов и последовательности команд в R обеспечивает воспроизводимость выполненного анализа, что является важной особенностью работы в данной системе.

Ключевой особенностью работы в R является необходимость ввода команд с консоли, что несколько отличается от привычной для многих работы с меню статистических программ, позволяющих в открытой в программе базе данных с помощью выбора пиктограмм анализа проводить обработку результатов исследования. Именно поэтому для повышения удобства работы с R были разработаны различные оболочки – IDE (integrated development environment, интегрированная среда разработчика). Примером такой оболочки является RStudio. R и RStudio являются свободно распространяемыми мультиплатформенными программами, способными работать с операционными системами Windows, MacOS, Linux.

Последнюю версию R рекомендуется загружать с официального сайта проекта http://cran.r-project.org. Программа RStudio (версия для персональных компьютеров) доступна для загрузки на сайте https://www.rstudio.com/products/rstudio/#Desktop. Следует отметить, что выбранные для загрузки дистрибутивы должны соответствовать операционной системе компьютера, на котором они будут установлены. Установка программ R и RStudio осуществляется по обычным правилам, причем первой устанавливается программа R. Соответственно, запуск RStudio начинает работу в программной среде R.

После установки программы R пользователь получает базовый набор пакетов, позволяющий выполнять большой объём разнообразных действий. При этом в сообществе R продолжается разработка новых пакетов, которые позволяют проводить еще больший объем манипуляций с данными и вычислений. Для установки нового пакета необходимо в консоли ввести команду *install.packages('название пакета')*, причем название пакета необходимо заключить в кавычки (кавычки в R могут использоваться как двойные, так и одинарные). Поскольку рассматриваемый шаблон работы в R предполагает использование пакета «tidyverse», его необходимо загрузить, использовав команду *install.packages('tidyverse')*. После загрузки пакета его включение для работы осуществляется командой library (название пакета), причем в данном случае название пакета должно быть без кавычек. Например, пакет «tidyverse» будет включен в работу командой *library(tidyverse)*.

Основы ввода данных в R

Ввод данных в консоли осуществляется после знака «>» и завершается нажатием на Enter, а при необходимости продолжить строку используется комбинация Shift + Enter.

Для присвоения названия какому-то объекту в R рекомендуется использовать сочетание знаков «<» и «-» без пробела между ними («<-»), как это показано на рис. 1 (листинг 1).

Листинг 1
```
x<-3
x
## [1] 3
```

Рис. 1. Вывод данных в R (листинг 1)

В самом простом применении R может использоваться как калькулятор – рис. 2 (листинг 2).

Листинг 2
```
# Сложение
2+3
## [1] 5
# Вычитание
4-3
## [1] 1
# Умножение
2*3
## [1] 6
# Возведение в степень
3**2
## [1] 9
# Деление
14/3
## [1] 4.666667
# Целочисленное деление
14%/%3
## [1] 4
# Модуль (остаток от деления)
14%%3
## [1] 2
```

Рис. 2. Использование R в качестве калькулятора (листинг 2)

Особенности переменных в R

В R, как и в других языках программирования, для обозначения объектов, содержащих данные, для последующего выполнения необходимых действий используются определенные переменные. Пример присвоения значений переменным представлен на рис. 3 (листинг 3).

Листинг 3
```
# присвоение значений переменным
a<-14
b<-3
# сложение
a+b
## [1] 17
(a-b) **2
## [1] 121
```

Рис. 3. Присвоение значений переменным a и b (листинг 3)

Следует учесть, что язык R является регистрозависимым, поэтому, например, переменные «Alpha» и «alpha» будут расценены программой как разные. При этом названия переменных не могут начинаться с цифр, специальных знаков и совпадение названий переменных с названиями функций нежелательно. Существуют руководства по стилю языка R, их основная цель – сделать скрипты более читаемыми. Следует обратить внимание, что знак «#» используется для того, чтобы ввести необходимые комментарии, в

том числе для удобства понимания вводимых и выводимых данных.

В R используются следующие типы данных:

– Logical (TRUE & FALSE, T & F) – логический тип данных.

– Numeric – числовой тип, который допускает арифметические операции.

– Integer (L) – целочисленные данные, частный случай данных Numeric.

– Character – строковый тип данных.

– Complex – особый тип данных для операций с комплексными числами.

В среде R используются следующие типы объектов для хранения данных:

– векторы (vector);

– матрицы (matrix);

– массивы (array);

– таблицы данных (data frame);

– списки (list).

Для работы с данными, закодированными в рамках объектов для хранения данных, используются различные функции. В любом языке программирования роль функций состоит в преобразовании данных на входе в данные на выходе. В R существуют функции базового пакета, функции дополнительных пакетов и пользовательские функции.

Векторный тип хранения данных при работе в среде R применяется достаточно широко (в векторе содержатся данные одного типа). Функция seq() позволяет создавать последовательности чисел – рис. 4 (листинг 4).

Листинг 4

```
# создаем числовой вектор
x<- c(3, 4, 7)
x
## [1] 3 4 7
is.vector(x)
## [1] TRUE
x <- 3 : 10
x
## [1] 3 4 5 6 7 8 9 10
# Используем функцию seq()
x <- seq(from = 2, to = 20, by = 2)
x
## [1] 2 4 6 8 10 12 14 16 18 20
x<- seq(2, 10, length = 12)
x
## [1] 2.000000 2.727273 3.454545 4.181818
4.909091 5.636364 6.363636
## [8] 7.090909 7.818182 8.545455 9.272727
10.000000
```

Рис. 4. Создание последовательности чисел (листинг 4)

Если вектор является одномерным набором данных, то матрица – это уже двумерный, а массив – многомерный набор однотипных данных. На рис. 5 (листинг 5) показаны способы создания матриц.

Листинг 5

```
m<-matrix(1:12, nrow =3)
m
## [,1] [,2] [,3] [,4]
## [1,] 1 4 7 10
## [2,] 2 5 8 11
## [3,] 3 6 9 12
ma <-matrix(letters[1:12], nrow =4,
dimnames =list(c('row1', 'row2', 'row3', 'row4'),
c('c1', 'c2', 'c3')))
ma
## c1 c2 c3
## row1 "a" "e" "i"
## row2 "b" "f" "j"
## row3 "c" "g" "k"
## row4 "d" "h" "l"
mrb <-rbind(c(3, 5, 7), c(4, 6, 8))
mrb
## [,1] [,2] [,3]
## [1,] 3 5 7
## [2,] 4 6 8
mrb <-cbind(c(3, 5, 7), c(4, 6, 8))
mrb
## [,1] [,2]
## [1,] 3 4
## [2,] 5 6
## [3,] 7 8
```

Рис. 5. Способы создания матриц (листинг 5)

Таблицы данных являются основным способом хранения данных и создаются при импорте данных в R из внешних источников. В разных столбцах таблицы могут храниться разные типы данных, но в одном отдельно взятом столбце должны находиться данные только одного типа. Таблицу данных можно создать с использованием функции data.frame() – рис. 6 (листинг 6).

Листинг 6

```
set.seed(123)
options(digits =1)
df <-data.frame(id = LETTERS[1:6],
age =runif(6, 24, 30),
height =rnorm(6, 170, 2))
df
## id age height
## 1 A 26 170
## 2 B 29 170
## 3 C 26 173
## 4 D 29 171
## 5 E 30 167
## 6 F 24 169
```

Рис. 6. Создание таблиц данных (листинг 6)

Данные одного столбца в таблице данных по сути представляют собой вектор, наименование которого включает в себя название таблицы данных и название переменной, соединенные знаком «$»– рис. 7 (листинг 7).

Листинг 7

```
df
## id age height
## 1 A 26 170
## 2 B 29 170
## 3 C 26 173
## 4 D 29 171
## 5 E 30 167
## 6 F 24 169
df$height
## [1] 170 170 173 171 167 169
is.vector(df$height)
## [1] TRUE
```

Рис. 7. Вывод вектора из таблицы данных (листинг 7)

В отличие от вышеперечисленных вариантов хранения данных, списки предназначены для хранения данных разных типов разного размера.

Индексирование в R

Индексирование является одним из способов доступа к данным в R. Первое значение в объекте данных индексируется единицей, а значение индекса выводится в квадратных скобках – рис. 8 (листинг 8). Индексация в таблицах данных аналогична индексации в матрицах.

Листинг 8

```
# индексация в векторе
v<-seq(2, 14, 3)
v
## [1] 2 5 8 11 14
length(v)
## [1] 5
v[3]
## [1] 8
v[c(1, 3)]
## [1] 2 8
# индексация в матрице
m <-matrix(1:15, nrow =3)
m
## [,1] [,2] [,3] [,4] [,5]
## [1,] 1 4 7 10 13
## [2,] 2 5 8 11 14
## [3,] 3 6 9 12 15
# вернуть значения первого ряда
m[1, ]
## [1] 1 4 7 10 13
# вернуть значения второго столбца
m[, 2]
## [1] 4 5 6
# вернуть значение ячейки в первом ряду во втором столбце
m[1, 2]
## [1] 4
```

Рис. 8. Индексирование (листинг 8)

Использование логических операторов

В R используются логические операторы !=(НЕ), & (И), | (ИЛИ), = (точно равно), а также >, >=, <, <= и круглые скобки – рис. 9 (листинг 9).

Листинг 9

```
# в векторе
set.seed(123)
v <-round(runif(6, 12, 21), 2)
v
## [1] 15 19 16 20 20 12
v >17
## [1] FALSE TRUE FALSE TRUE TRUE FALSE
v[v >17]
## [1] 19 20 20
# в таблице данных
set.seed(123)
df <-data.frame(id = letters[1:6], value =
round(runif(6, 12, 21), 2))
df$value
## [1] 15 19 16 20 20 12
df$value >17
## [1] FALSE TRUE FALSE TRUE TRUE FALSE
df[df$value >17,]
## id value
## 2 b 19
## 4 d 20
## 5 e 20
```

Рис. 9. Использование логических операторов (листинг 9)

Работа с пропущенными значениями

В любом исследовательском наборе данных при достаточно большом объеме выборок, как правило, встречаются пропущенные значения. При выявлении пропущенных значений их необходимо в обязательном порядке учесть, причем на усмотрение исследователя возможно выполнение следующих действий: удаление столбцов и строк с пропущенными значениями, учет пропущенных значений при расчете показателей или подбор значений («imputation»).

Пропущенные значения («NA») в таблице данных могут быть выявлены при использовании функции summary(name_dataframe), и в результате её выполнения можно будет увидеть количество пропущенных значений по столбцам – рис. 10 (листинг 10).

Листинг 10

```
options(digits =3)
n <-9
set.seed(3456789)
df_mice <-data.frame(x1 =sample(c('a', 'b', 'c'),
n, replace =TRUE),
x2 =sample(c(runif(5, 12, 45), NA), n, replace
=TRUE),
x3 =sample(c(rnorm(5), NA), n, replace =TRUE),
x4 =rnorm(n, 2, 1))

df_mice
## x1 x2 x3 x4
## 1 c 32.2 -1.468 4.072
## 2 b 18.5 0.353 0.615
## 3 a NA -1.468 3.160
## 4 c NA 1.570 0.649
## 5 c 27.9 NA 3.177
## 6 c 15.9 NA 2.740
## 7 b 15.9 1.570 1.407
## 8 b NA 0.353 2.823
## 9 c NA 2.576 2.930
summary(df_mice)
## x1 x2 x3 x4
## a:1 Min. :15.9 Min. :-1.468 Min. :0.62
## b:3 1st Qu.:15.9 1st Qu.:-0.557 1st Qu.:1.41
## c:5 Median :18.5 Median : 0.353 Median :2.82
## Mean :22.1 Mean : 0.498 Mean :2.40
## 3rd Qu.:27.9 3rd Qu.: 1.570 3rd Qu.:3.16
## Max. :32.2 Max. : 2.576 Max. :4.07
## NA's :4 NA's :2
any(is.na(df_mice))
## [1] TRUE
```

Рис. 10. Выявление пропущенных значений (листинг 10)

Удаление строк и столбцов может быть выполнено с использованием индексирования – рис. 11 (листинг 11).

Листинг 11

```
# создадим таблицу данных с NA
df_cr <-data.frame(x1 =c(2, 3, NA, 6, 7),
x2 =c(5, 3, 6, 4, 3),
x3 =c(NA, 2, NA, NA, NA),
x4 =c(6, 3, NA, 2, 11))

df_cr
## x1 x2 x3 x4
## 1 2 5 NA 6
## 2 3 3 2 3
## 3 NA 6 NA NA
## 4 6 4 NA 2
## 5 7 3 NA 11
summary(df_cr)
```

```
## x1 x2 x3 x4
## Min. :2.00 Min. :3.0 Min. :2 Min. : 2.00
## 1st Qu.:2.75 1st Qu.:3.0 1st Qu.:2 1st Qu.:
2.75
## Median :4.50 Median :4.0 Median :2 Median :
4.50
## Mean :4.50 Mean :4.2 Mean :2 Mean : 5.50
## 3rd Qu.:6.25 3rd Qu.:5.0 3rd Qu.:2 3rd Qu.:
7.25
## Max. :7.00 Max. :6.0 Max. :2 Max. :11.00
## NA's :1 NA's :4 NA's :1
# удалим третью строку
df_cr <-df_cr[-3,]

df_cr
## x1 x2 x3 x4
## 1 2 5 NA 6
## 2 3 3 2 3
## 4 6 4 NA 2
## 5 7 3 NA 11
# удалим третий столбец
df_cr$x3 <-NULL

df_cr
## x1 x2 x4
## 1 2 5 6
## 2 3 3 3
## 4 6 4 2
## 5 7 3 11
```

Рис. 11. Удаление строк и столбцов (листинг 11)

Учет пропущенных значений при расчете показателей обеспечивается включением параметра na.rm = TRUE в используемую функцию, например, таким образом: *mean(name_vector, na.rm = TRUE)* – рис. 12 (листинг 12).

Листинг 12

```
v <-c(3, 5, NA, 6, 7, 12)
mean(v)
## [1] NA
mean(v, na.rm =TRUE)
## [1] 6.6
```

Рис. 12. Учет пропущенных значений (листинг 12)

Подбор значений может быть выполнен с использованием функций дополнительных пакетов. Для этого создадим таблицу данных, содержащую «NA» – рис. 13 (листинг 13).

Листинг 13

```
options(digits =3)
n <-9
set.seed(3456789)
df_mice <-data.frame(x1 =sample(c('a', 'b', 'c'),
n, replace =TRUE),
x2 =sample(c(runif(5, 12, 45), NA), n, replace
=TRUE),
x3 =sample(c(rnorm(5), NA), n, replace =TRUE),
x4 =rnorm(n, 2, 1))

df_mice
## x1 x2 x3 x4
## 1 c 32.2 -1.468 4.072
## 2 b 18.5 0.353 0.615
## 3 a NA -1.468 3.160
## 4 c NA 1.570 0.649
## 5 c 27.9 NA 3.177
## 6 c 15.9 NA 2.740
## 7 b 15.9 1.570 1.407
## 8 b NA 0.353 2.823
## 9 c NA 2.576 2.930
```

Рис. 13. Создание таблицы данных с пропущенными значениями (листинг 13)

Далее используем функции пакета *«mice»: name_dataframe <- complete(mice(name_dataframe))* – рис. 14 (листинг 14). На рисунке видно, как значения «NA» были заменены на числовые.

Листинг 14

```
library(mice)
df_imp <-complete(mice(df_mice))
# полученный результат
df_imp
## x1 x2 x3 x4
## 1 c 32.2 -1.468 4.072
## 2 b 18.5 0.353 0.615
## 3 a 27.9 -1.468 3.160
## 4 c 18.5 1.570 0.649
## 5 c 27.9 0.353 3.177
## 6 c 15.9 1.570 2.740
## 7 b 15.9 1.570 1.407
## 8 b 15.9 0.353 2.823
## 9 c 18.5 2.576 2.930
```

Рис. 14. Использование подбора значений (листинг 14)

Если пропущенные значения были кодированы не как «NA», эти значения обязательно требуется заменить на «NA» – рис. 15 (листинг 15).

Листинг 15

```
df_999<-data.frame(x1 =c(5, 2, 4, 999),
x2 =c(3, 999, 5, 5),
x3 =c(999, 2, 1, 3))
df_999
## x1 x2 x3
## 1 5 3 999
## 2 2 999 2
## 3 4 5 1
## 4 999 5 3
df_999[df_999==999] <-NA
df_999
## x1 x2 x3
## 1 5 3 NA
## 2 2 NA 2
## 3 4 5 1
## 4 NA 5 3
```

Рис. 15. Замена пропущенных значений на кодировку «NA» (листинг 15)

Преобразование данных с использованием функций пакета «tidyverse»

Для примера преобразования данных были использована случайная выборка из Архангельского областного регистра родов с небольшой модификацией абсолютных значений переменных [5].

Так как файл с данными представлен в формате sav, для его импорта потребуется функция из пакета «foreign» – рис. 16 (листинг 16).

Листинг 16

```
library(foreign)
df <-read.spss("Simulated_sample.sav", to.data.
frame =TRUE)
```

Рис. 16. Установка пакета foreign и открытие файла с данными (листинг 16).

Далее изучим структуру файла с данными — рис. 17 (листинг 17 представлен с сокращениями).

Листинг 17

```
## 'data.frame': 2000 obs. of 27 variables:
## $ ID : num 117796 117775 117767 117719
117713 ...
## $ Marital_status : Factor w/ 4 levels
"Unmarried","Married",..: 4 3 2 2 2 2 2 2 2 2
...
## $ Education : Factor w/ 6 levels
"None","Primary (class 1-9)",..: 6 4 5 5 3 5 5
4 4 4 ...
## $ Paternal_age : num 29 29 35 33 38 27 29
30 36 32 ...
## $ Maternal_height : num 165 156 170 177 168
160 150 168 173 176 ...
## $ Maternal_weight : num 59 52 72 73 74 49
54 63 55 61 ...
## $ Gestational_age : num 39 39 38 40 40 40
39 40 39 38 ...
## $ Vitamins_before_pregnancy : Factor w/ 2
levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
## $ Vitamis_during_pregnancy : Factor w/ 2
levels "no","yes": 2 1 1 2 1 2 2 2 2 1 ...
## $ Folic_acid_before_pregnancy : Factor w/ 2
levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
## $ Folic_acid_during_pregnancy : Factor w/ 2
levels "no","yes": 2 1 1 2 1 2 1 2 1 1 ...
## $ Smoking_before_pregnancy : Factor w/ 2
levels "no","yes": 1 2 1 1 1 1 1 1 1 1 ...
## $ N_cigarettes_day_before : num NA 5 NA NA
NA NA NA NA NA NA ...
## $ Smoking_during_pregnancy : Factor w/ 2
levels "no","yes": 1 2 1 1 1 1 1 1 1 1 ...
## $ N_cigarettes_during_pregnancy: num NA 5 NA
NA NA NA NA NA NA NA ...
## $ Alcohol_abuse : Factor w/ 2 levels
"no","yes": 1 1 1 1 1 1 1 1 1 1 ...
## $ Delivery_type : Factor w/ 3 levels
"Spontaneous",..: 1 1 3 1 1 1 1 1 1 1 ...
## $ Infant_sex : Factor w/ 3 levels
"Male","Female",..: 1 2 2 2 2 2 1 1 1 1 ...
## $ Apgar1 : num 7 8 7 7 7 7 6 8 8 9 ...
## $ Apgar5 : num 8 8 8 8 8 8 7 9 9 9 ...
## $ Birth_defect : Factor w/ 2 levels
"no","yes": 1 1 1 1 1 1 1 1 1 1 ...
## $ Year_of_birth : num 2013 2013 2013 2013
2013 ...
## $ Maternal_age : num 32 24 34 26 38 28 26
23 32 32 ...
## $ Preeclampsia : num 0 0 0 0 0 0 0 0 0 0
...
## $ BIrthweight : num 3835 2920 3190 3850 3330
...
## $ Birthlength : num 56 50 52 54 52 54 51
50 54 55 ...
## $ Anemia : num 1 1 0 0 1 1 0 0 1 1 ...
```

Рис. 17. Структура файла с данными (листинг 17)

По результатам изучения листинга 17 можно сказать, что в наборе данных имеются 2 000 наблюдений с 27 переменными.

Далее получим ряд суммарных показателей — описательную статистику [2]. Функция «summary» позволяет для количественных данных получить среднее арифметическое, медиану, первый и третий квартили, минимальное и максимальное значения, а для категориальных и порядковых данных — количество наблюдений для каждого значения. Также будет подсчитано количество «пустых» значений («NA») — рис. 18 (листинг 18).

Листинг 18

```
summary(df)
##  ID  Marital_status  Education
##  Min.  : 304  Unmarried  : 249  None  : 1
##  1st Qu.: 6116  Married  :1390  Primary (class
1-9) :126
##  Median : 16852  Cohabitant: 357  Secondary
(class 10-11):301
##  Mean  : 28952  Other  : 4  Technical School :874
##  3rd Qu.: 37446  Higher education :690
##  Max.  :117796  Unknown  : 5
##  NA's  : 3
##  Paternal_age  Maternal_height  Maternal_weight
Gestational_age
##  Min.  :17.00  Min.  :135.0  Min.  : 36.00  Min.
:24.00
##  1st Qu.:26.00  1st Qu.:160.0  1st Qu.: 55.00
1st Qu.:38.00
##  Median :30.00  Median :164.0  Median : 61.00
Median :39.00
##  Mean  :30.75  Mean  :163.7  Mean  : 63.32  Mean
:38.77
##  3rd Qu.:35.00  3rd Qu.:168.0  3rd Qu.: 70.00
3rd Qu.:40.00
##  Max.  :79.00  Max.  :184.0  Max.  :122.00  Max.
:42.00
##  NA's  :253  NA's  :10  NA's  :29  NA's  :12
##  Vitamins_before_pregnancy  Vitamis_during_
pregnancy
##  no  :1955  no  : 927
##  yes  : 39  yes :1068
##  NA's: 6  NA's: 5
##
##  Folic_acid_before_pregnancy  Folic_acid_during_
pregnancy
##  no  :1958  no  : 953
##  yes  : 37  yes :1042
##  NA's: 5  NA's: 5
##
##  Smoking_before_pregnancy  N_cigarettes_day_
before  Smoking_during_pregnancy
##  no  :1491  Min.  : 0.00  no  :1535
##  yes  : 348  1st Qu.: 5.00  yes  : 301
##  NA's: 161  Median :10.00  NA's: 164
##  Mean  :10.44
##  3rd Qu.:15.00
##  Max.  :40.00
##  NA's  :1713
##  N_cigarettes_during_pregnancy  Alcohol_abuse
Delivery_type
##  Min.  : 0.000  no  :1984  Spontaneous  :1351
##  1st Qu.: 5.000  yes: 16  Induced  : 215
##  Median :10.000  Caesarean section: 423
##  Mean  : 8.629  NA's  : 11
##  3rd Qu.:10.000
##  Max.  :20.000
##  NA's  :1747
##  Infant_sex  Apgar1  Apgar5  Birth_defect
##  Male  :1033  Min.  : 0.000  Min.  : 0.000  no
:1911
##  Female  : 967  1st Qu.: 7.000  1st Qu.: 8.000
yes  : 74
##  Undetermined: 0  Median  : 8.000  Median  :
8.000  NA's: 15
##  Mean  : 7.497  Mean  : 8.302
##  3rd Qu.: 8.000  3rd Qu.: 9.000
##  Max.  :10.000  Max.  :10.000
##  NA's  :13  NA's  :18
##  Year_of_birth  Maternal_age  Preeclampsia
BIrthweight
##  Min.  :2012  Min.  :15.0  Min.  :0.000  Min.  : 620
##  1st Qu.:2012  1st Qu.:25.0  1st Qu.:0.000  1st
Qu.:3090
##  Median  :2013  Median  :28.0  Median  :0.000
Median  :3430
```

```
## Mean :2013 Mean :28.6 Mean :0.035 Mean :3371
## 3rd Qu.:2014 3rd Qu.:32.0 3rd Qu.:0.000 3rd
Qu.:3740
## Max. :2014 Max. :46.0 Max. :1.000 Max. :5930
## NA's :1
## Birthlength Anemia
## Min. :28.00 Min. :0.0000
## 1st Qu.:51.00 1st Qu.:0.0000
## Median :53.00 Median :1.0000
## Mean :52.19 Mean :0.5135
## 3rd Qu.:54.00 3rd Qu.:1.0000
## Max. :66.00 Max. :1.0000
## NA's :2
```

Рис. 18. Описательная статистика для файла с данными (листинг 18)

Манипуляции с данными в R включают в себя выбор столбцов, выбор строк по условию, сортировку в столбцах, создание новых переменных, группировку данных и получение суммарных показателей. Для данных манипуляций используются функции пакета «tidyverse».

Концепция использования пакета «tidyverse» предполагает использование оператора «%>%». Этот оператор указывает, что объект с данными на выходе является объектом с данными на входе следующей функции. Без использования оператора «%>%» выполняются следующие действия: сначала создается новая таблица данных, далее функцией «head» выводятся первые шесть строк таблицы – рис. 19 (листинг 19).

Листинг 19

```
library(tidyverse)
df_1<-select(df, Marital_status, Education,
Maternal_age)
head(df_1)
## Marital_status Education Maternal_age
## 1 Other Unknown 32
## 2 Cohabitant Technical School 24
## 3 Married Higher education 34
## 4 Married Higher education 26
## 5 Married Secondary (class 10-11) 38
## 6 Married Higher education 28
```

Рис. 19. Использование функции «head» (листинг 19)

Код скрипта, выполняющего те же действия, но с использованием оператора «%>%» выглядит так, как представлено на рис. 20 (листинг 20).

Листинг 20

```
library(tidyverse)
df %>%
select(Marital_status, Education, Maternal_
age) %>%
head()
## Marital_status Education Maternal_age
## 1 Other Unknown 32
## 2 Cohabitant Technical School 24
## 3 Married Higher education 34
## 4 Married Higher education 26
## 5 Married Secondary (class 10-11) 38
## 6 Married Higher education 28
```

Рис. 20. Использование оператора «%>%» (листинг 20)

Таким образом, использование оператора «%>%» позволяет исключить промежуточные этапы, делая код скрипта более компактным.

Выбор столбцов делается с использованием функции «select» – рис. 21 (листинг 21).

Листинг 21

```
library(tidyverse)
df %>%
select(Marital_status, Education, Maternal_
age) %>%
head()
## Marital_status Education Maternal_age
## 1 Other Unknown 32
## 2 Cohabitant Technical School 24
## 3 Married Higher education 34
## 4 Married Higher education 26
## 5 Married Secondary (class 10-11) 38
## 6 Married Higher education 28
```

Рис. 21. Использование функции «select» (листинг 21)

Код скрипта может содержать несколько функций, использующихся при манипуляциях с данными. Для выбора данных по значению отдельных переменных используется функция «filter» – рис. 22 (листинг 22).

Листинг 22

```
library(tidyverse)
# одно условие для отбора
df %>%
select(Marital_status, Education, Maternal_
age) %>%
filter(Maternal_age >30) %>%
head()
## Marital_status Education Maternal_age
## 1 Other Unknown 32
## 2 Married Higher education 34
## 3 Married Secondary (class 10-11) 38
## 4 Married Technical School 32
## 5 Married Technical School 32
## 6 Married Higher education 35
# два условия для отбора
df %>%
select(Marital_status, Education, Maternal_
age) %>%
filter(Maternal_age >30&Education == 'Higher
education') %>%
head()
## Marital_status Education Maternal_age
## 1 Married Higher education 34
## 2 Married Higher education 35
## 3 Married Higher education 35
## 4 Married Higher education 32
## 5 Married Higher education 31
## 6 Married Higher education 35
# использование в фильтре оператора %in% для вы-
бора нескольких
# категориальных значений из одной переменной
df %>%
select(Marital_status, Education, Maternal_
age) %>%
filter(Maternal_age >30&
Education %in%c('Higher education', 'Technical
School') ) %>%
head()
## Marital_status Education Maternal_age
## 1 Married Higher education 34
## 2 Married Technical School 32
## 3 Married Technical School 32
## 4 Married Higher education 35
## 5 Married Higher education 35
## 6 Married Higher education 32
```

Рис. 22. Использование функции «filter» (листинг 22)

Сортировка в столбцах проводится с использованием функции «arrange» – рис. 23 (листинг 23).

Листинг 23
```
library(tidyverse)
df %>%
select(Marital_status, Education, Maternal_
age) %>%
filter((Maternal_age >30&Maternal_age <=40) &
Education == 'Higher education'&
Marital_status != 'Married') %>%
arrange(Maternal_age) %>%
head()
## Marital_status Education Maternal_age
## 1 Unmarried Higher education 31
## 2 Unmarried Higher education 31
## 3 Unmarried Higher education 31
## 4 Unmarried Higher education 31
## 5 Cohabitant Higher education 31
## 6 Cohabitant Higher education 31
# сортировка в порядке убывания с использованием
функции desc()
df %>%
select(Marital_status, Education, Maternal_
age) %>%
filter((Maternal_age >30&Maternal_age <=40) &
Education == 'Higher education'&
Marital_status != 'Married') %>%
arrange(desc(Maternal_age)) %>%
head()
## Marital_status Education Maternal_age
## 1 Unmarried Higher education 39
## 2 Cohabitant Higher education 39
## 3 Unmarried Higher education 38
## 4 Cohabitant Higher education 38
## 5 Cohabitant Higher education 38
## 6 Cohabitant Higher education 37
```

Рис. 23. Использование функции «arrange» (листинг 23)

Создание новых переменных в пакете «tidyverse» выполняется с использованием функции «mutate» – рис. 24 (листинг 24).

Листинг 24
```
library(tidyverse)
# создание новой количественной переменной из
имеющихся данных
df %>%
mutate(Maternal_BMI = Maternal_weight/(Maternal_
height *.01)**2) %>%
select(Maternal_height, Maternal_weight, Maternal_
BMI) %>%
arrange(Maternal_BMI) %>%
head()
## Maternal_height Maternal_weight Maternal_BMI
## 1 168 41 14.52664
## 2 174 45 14.86326
## 3 172 46 15.54895
## 4 158 39 15.62250
## 5 167 44 15.77683
## 6 151 36 15.78878
```

Рис. 24. Использование функции «mutate» (листинг 24)

В следующем примере приведен код создания новой бинарной переменной с использованием функции «ifelse». Функция «ifelse» проверяет логическое значение переменной и возвращает значение, соответствующее «TRUE» или «FALSE». Функция «table» возвращает распределение категориальных данных в переменной – рис. 25 (листинг 25).

Листинг 25
```
# создание новой бинарной переменной
df %>%
mutate(Marital_status_2 =ifelse(Marital_status ==
'Married',
'Married', 'Other')) %>%
select(Marital_status_2) %>%
table()
## .
## Married Other
## 1390 610
```

Рис. 25. Использование функций «ifelse» и «table» (листинг 25)

Количественная переменная может быть преобразована в категориальную, например, возраст – в возрастные группы. Функция «addmargins» позволяет получить суммарные выражения по строкам и столбцам в таблице – рис. 26 (листинг 26).

Листинг 26
```
# создание новой категориальной переменной путем
«разрезания» имеющейся количественной переменной
df %>%
mutate(M_age_group =factor(cut(Maternal_age,
breaks =c(14, 18, 30 , 40, 50),
labels =c('<18', '19-30', '31-40', '>40')))) %>%
select(Marital_status, M_age_group) %>%
table() %>%
addmargins()
## M_age_group
## Marital_status <18 19-30 31-40 >40 Sum
## Unmarried 11 166 65 7 249
## Married 12 854 509 15 1390
## Cohabitant 8 239 108 2 357
## Other 0 1 2 1 4
## Sum 31 1260 684 25 2000
```

Рис. 26. Использование функции «addmargins» (листинг 26)

Для группировки и получения суммарных показателей используются функции «group_by» и «summarise». Функция n() выводит количество наблюдений – рис. 27 (листинг 27).

Листинг 27
```
df %>%
group_by(Delivery_type) %>%
summarise(m_Birthweight =mean(BIrthweight, na.rm
=TRUE),
sd_Birthweight =sd(BIrthweight, na.rm =TRUE),
n =n())
## # A tibble: 4 x 4
## Delivery_type m_Birthweight sd_Birthweight n
## <fct><dbl><dbl><int>
## 1 Spontaneous 3415 523 1351
## 2 Induced 3361 534 215
## 3 Caesarean section 3230 789 423
## 4 <NA> 3496 590 11
```

Рис. 27. Использование функций «group_by», «summarise» и «n()» (листинг 27)

Импорт данных в R

Следует отметить, что программная среда R не очень подходит для ввода данных, поэтому ввод данных целесообразнее осуществлять в других программах, например в электронных таблицах MS Excel, SPSS и проч.

Данные для анализа должны быть подготовлены в виде таблицы, в которой все ячейки должны быть заполнены. Названия столбцов таблицы при импорте в R превратятся в названия переменных, поэтому названия столбцов должны быть указаны на английском языке и соответствовать требованиям R – не начинаться с цифр и специальных знаков, должны быть информативными и не очень большими по количеству знаков. Для того, чтобы в ячейках таблицы не было пустых значений, в такую ячейку вводится значение «NA» («not available»).

Предлагаемый в данной публикации вариант работы в R рекомендует распространенный метод подготовки данных в электронных таблицах, например в MS Excel. Подходящим форматом готовой к импорту таблицы является csv («comma separated value»), но возможен импорт данных и из других типов файлов.

Импорт в R может быть выполнен с использованием функции R *read.csv('имя файла с расширением')*, причем имя файла с расширением должно быть указано в кавычках.

Например, если пользователь создал файл с данными example.csv, в R его можно импортировать командой *df <- read.csv('example.csv')*. В этой записи созданная переменная df – это *data.frame (таблица данных)*, созданная в среде R, дальнейшее обращение будет осуществляться к этому созданному объекту. Если файл создавался в русскоязычной версии MS Excel, то разделителем является точка с запятой. Для импорта такого файла необходимо использовать функцию *read.csv2('имя файла с расширением')*. Следует учесть, что при импорте данных из файлов других форматов могут потребоваться функции из других пакетов.

Если в ходе работы файл подвергался изменениям и эти изменения потребовалось сохранить, это действие можно выполнить командой *write.csv(df, 'example_mod.csv')*. Изменения сохранятся в новом файле example_mod.csv, который можно будет использовать в дальнейшей работе.

Программа RStudio

Как уже было сказано ранее, данная программа предназначена для повышения удобства работы с данными и проведения анализа, так как она формализует многие процедуры, ввод которых в виде скриптов может оказаться достаточно трудоемким.

При запуске RStudio на экране можно увидеть четыре окна, размеры которых могут изменяться (рис. 28).

Левое нижнее окно называется «Console», работа в данном окне соответствует работе в программе R. Левое верхнее окно – «Editor», в нём создаются скрипты и другие документы при выполнении анализа данных. Правое верхнее окно включает разделы «Environment» и «History». В разделе «Environment» помещаются сведения о созданных переменных и загруженных в программу наборах данных. Правое нижнее окно включает разделы «Files», «Plots», «Packages», «Help», «Viewer». В разделе «Files» пользователю становятся доступны файлы рабочей папки, графики при их создании выводятся в разделе «Plots», при этом созданные графики могут быть сохранены в отдельных файлах. Раздел «Packages» предназначен для установки новых пакетов (приложений, расширяющих базисные возможности программы) и обновление имеющихся пакетов. Меню программы RStudio находится выше окон программы и предоставляет много возможностей, освобождая от необходимости ввода команд в консоли.

При решении конкретных задач анализа данных работу с ними в RStudio рекомендуется начинать с создания проекта, при этом может быть создана новая рабочая папка, в которую пользователь может поместить все необходимые для работы файлы.

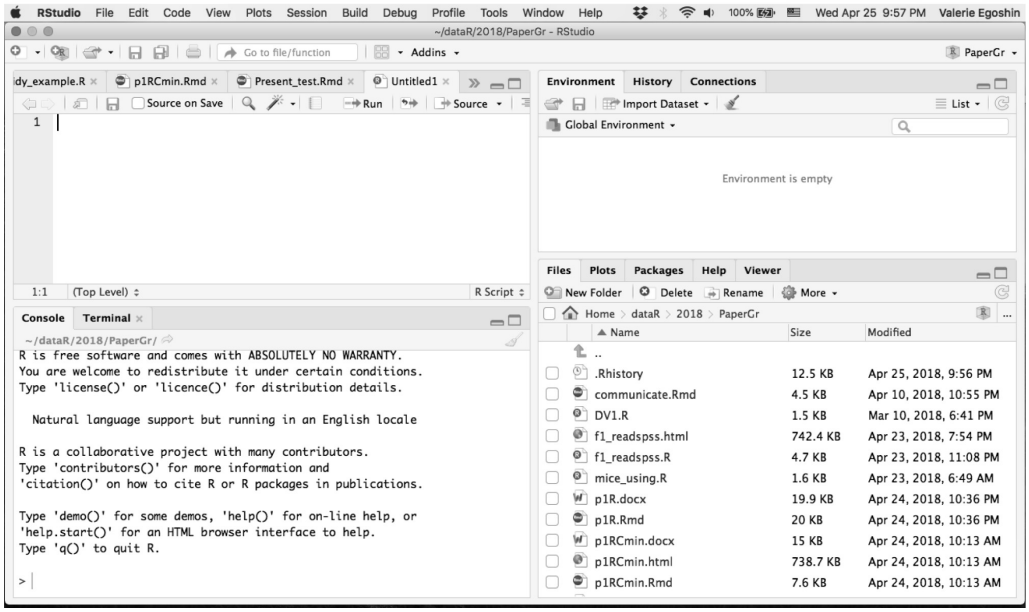

Рис. 28. Общий вид программы RStudio

Воспроизводимость исследований в RStudio обеспечивается работой в панели «Editor». На рис. 29 показано меню выбора типа документа после нажатия на знак «крест в зеленом круге», после которого выполняется выбор из предлагаемых в меню программы вариантов, из которых наиболее часто используются первые три типа. Так, вариант «Script» может быть рекомендован при небольшом объеме комментариев, выполняемых по ходу анализа данных. Варианты «Notebook» и «Markdown» предполагают возможность создания полноценных документов, включающих текст с разметкой и блоки, в которых выполняются вычисления и создаются графики, а также выводятся диаграммы и результаты вычислений. Запуск команды на исполнение из «Editor» может быть осуществлена выбором опции «Run». После завершения работы по анализу данных (создания скрипта или другого документа) может быть сформирован файл в форматах HTML, MS Word, PDF путем выбора из меню «File» – «Knit Document». Создаваемые скрипты могут также быть в дальнейшем использованы как шаблоны.

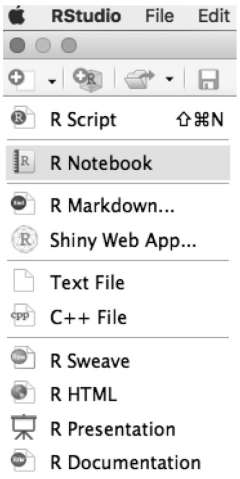

Рис. 29. Выбор типа документа в RStudio

Данные для просмотра в RStudio можно вывести в дополнительном окне после ввода *View(name_dataset)*. Для импорта данных команды необходимо вводить команды в окне «Editor», создавая скрипт или другой документ.

Таким образом, представленные в данном материале возможности программной среды R, несмотря на их краткость, на наш взгляд, позволяют показать, насколько полезным будет для пользователя их применения в практической исследовательской работе.

В последующих материалах будут рассмотрены вопросы применения R для визуализации данных, описательной, аналитической статистики и других методов статистического анализа данных.

Список литературы

1. *Буховец А. Г., Москалев П. В., Богатова В. П., Бирючинская Т. Я.* Статистический анализ данных в системе R: учебное пособие. Воронеж: ВГАУ, 2010. 124 с.

2. *Гржибовский А. М., Унгуряну Т. Н., Горбатова М. А.* Описательная статистика с использованием пакетов статистических программ SPSS и STATA. // Наркология. 2017. № 4. С. 36–51.

3. *Кабаков Р. И.* R в действии. Анализ и визуализация данных в программе R / пер. с англ. П. А. Волковой. М.: ДМК Пресс, 2014. 588 с.

4. *Мастицкий С. Э., Шитиков В. К.* Статистический анализ и визуализация данных с помощью R. М.: ДМК Пресс, 2015. 496 с.

5. *Усынина А. А., Одланд И. О., Пылаева Ж. А., Пастбина И. М., Гржибовский А. М.* Регистр родов Архангельской области как важный информационный ресурс для науки и практического здравоохранения // Экология человека. 2017. № 2. С. 58–64.

6. *Холматова К. К., Харькова О. А., Гржибовский А. М.* Классификация научных исследований в здравоохранении // Экология человека. 2016. № 1. С. 57–64.

7. *Crawley M. J.* The R Book. 2nd ed. Wiley, 2013.

8. *Grolemund G., Wickham H.* R for data science. URL: from: http://r4ds.had.co.nz (дата обращения: 26.04.2018).

9. R: анализ и визуализация данных. URL: http://r-analytics.blogspot.ru (дата обращения: 26.04.2018).

10. TIOBE Index for April 2018. URL: https://www.tiobe.com/tiobe-index/ (дата обращения 26.04.2018).

References

1. Buhovec A. G., Moskalev P. V., Bogatova V. P., Biryuchinskaya T. Y. *Statisticheskii analiz dannyh v sisteme R: uchebnoe posobie* [Statistical data analysis using R: the textbook]. Voronezh, 2010, 124 p.

2. Grjibovski A. M., Unguryanu T. N., Gorbatova M. A. [Descriptive statistics using SPSS and STATA software. *Narkologiya* [Narcology]. 2017, 4, pp. 36-51. [In Russian]

3. Kabakov R. I. *R v deystvii. Analiz i vizualizatsiya dannykh v programme R* [R in action: data analysis and visualization using R software], per. s angl. P. A. Volkov. Moscow, 2014, 588 p.

4. Mastickiy S. E., Shitikov V. K. *Statisticheskiy analiz i vizualizatsiya dannykh s pomoshch'yu R* [Data statistical analysis using R]. Moscow, 2015, 496 p.

5. Usynina A. A., Odland J.Ø., Pylaeva Zh. A., Pastbina I. M., Grjibovski A. M. Arkhangelsk County Birth Registry as an Important Source of Information for Research and Healthcare. *Ekologiya cheloveka* [Human Ecology]. 2017, 2, pp. 58-64. [In Russian]

6. Kholmatova K. K., Kharkova O. A., Grjibovski A. M. Types of Research in Health Sciences. *Ekologiya cheloveka* [Human Ecology]. 2016, 1, pp. 57-64. [In Russian]

7. Crawley M. J. *The R Book*. 2nd ed. Wiley, 2013.

8. Grolemund G., Wickham H. *R for data science*. Available from: http://r4ds.had.co.nz (accessed: 26.04.2018).

9. R: анализ и визуализация данных. Available from: http://r-analytics.blogspot.ru (accessed: 26.04.2018).

10. TIOBE Index for April 2018. Available from: https://www.tiobe.com/tiobe-index/ (accessed: 26.04.2018).

Контактная информация:

Гржибовский Андрей Мечиславович – доктор медицины, заведующий ЦНИЛ СГМУ, г. Архангельск; профессор Северо-Восточного федерального университета, г. Якутск; почетный профессор ГМУ г. Семей (Казахстан); почетный доктор МКТУ, г. Туркестан (Казахстан)

Адрес: 163000, г. Архангельск, Троицкий проспект, д. 51, офис 1252

Тел.: +79214717053

E-mail: Andrej.Grjibovski@gmail.com