Импорт данных : : ШПАРГАЛКА

В R **tidyverse** построена вокруг **опрятных данных**, хранящихся в **tibble** (развитие data frame).

На лицевой стороне описано, как считывать текстовые файлы в R с помощью **readr**.

На обратной стороне описано, как создавать tibble с **tibble** и опрятные данные с **tidyr**.

ДРУГИЕ ТИПЫ ДАННЫХ

Используйте следующие пакеты для импорта файлов другого типа

- **haven** - файлы SPSS, Stata и SAS
- **readxl** - excel файлы (.xls и .xlsx)
- **DBI** - базы данных
- **jsonlite** - json
- **xml2** - XML
- **httr** - Web API
- **rvest** - HTML (данные из Интернета)

Сохранение данных

Сохраняйте **x**, объект R, в **path**, путь к файлу, с помощью:

Файл с разделителем запятой
write_csv(x, path, na = "NA", append = FALSE, col_names = !append**)**

Файл с произвольным разделителем
write_delim(x, path, delim = " ", na = "NA", append = FALSE, col_names = !append**)**

CSV для excel
write_excel_csv(x, path, na = "NA", append = FALSE, col_names = !append**)**

Строка в файл
write_file(x, path, append = FALSE**)**

Строковый вектор в файл, один элемент на строку
write_lines(x, path, na = "NA", append = FALSE**)**

Объект в файл RDS
write_rds(x, path, compress = c("none", "gz", "bz2", "xz"), ...**)**

Файл с разделителем табуляцией
write_tsv(x, path, na = "NA", append = FALSE, col_names = !append**)**

Считывание табличных данных

Эти функции имеют общие аргументы:

read_*(file, col_names = TRUE, col_types = NULL, locale = default_locale(), na = c("", "NA"), quoted_na = TRUE, comment = "", trim_ws = TRUE, skip = 0, n_max = Inf, guess_max = min(1000, n_max), progress = interactive()**)**

a,b,c
1,2,3
4,5,NA

A	B	C
1	2	3
4	5	NA

Файлы с разделителем запятой
read_csv("file.csv"**)**
Для создания file.csv исполните:
write_file(x = "a,b,c\n1,2,3\n4,5,NA", path = "file.csv")

a;b;c
1;2;3
4;5;NA

A	B	C
1	2	3
4	5	NA

Файлы с разделителем точкой с запятой
read_csv2("file2.csv"**)**
write_file(x = "a;b;c\n1;2;3\n4;5;NA", path = "file2.csv")

a|b|c
1|2|3
4|5|NA

A	B	C
1	2	3
4	5	NA

Файлы с произвольным разделителем
read_delim("file.txt", delim = "|"**)**
write_file(x = "a|b|c\n1|2|3\n4|5|NA", path = "file.txt")

a b c
1 2 3
4 5 NA

A	B	C
1	2	3
4	5	NA

Файлы с фиксированной шириной
read_fwf("file.fwf", col_positions = c(1, 3, 5)**)**
write_file(x = "a b c\n1 2 3\n4 5 NA", path = "file.fwf")

Файлы с разделителем табуляцией
read_tsv("file.tsv"**)** Также **read_table().**
write_file(x = "a\tb\tc\n1\t2\t3\n4\t5\tNA", path = "file.tsv")

ПОЛЕЗНЫЕ АРГУМЕНТЫ

a,b,c
1,2,3
4,5,NA

Файл для примеров
write_file("a,b,c\n1,2,3\n4,5,NA","file.csv")
f <- "file.csv"

A	B	C
1	2	3
4	5	NA

Без заголовка
read_csv(f, **col_names = FALSE**)

x	y	z
A	B	C
1	2	3
4	5	NA

С указанием заголовка
read_csv(f, **col_names = c("x", "y", "z")**)

1	2	3
4	5	NA

Пропуск строк
read_csv(f, **skip = 1**)

A	B	C
1	2	3

Считывание части
read_csv(f, **n_max = 1**)

A	B	C
NA	2	3
4	5	NA

Пропущенные значения
read_csv(f, **na = c("1", ".")**)

Считывание нетабличных данных

Чтение файла в одну строку
read_file(file, locale = default_locale()**)**

Чтение строк файла в отдельные строки
read_lines(file, skip = 0, n_max = -1L, na = character(), locale = default_locale(), progress = interactive()**)**

Чтение логовых файлов Apache
read_log(file, col_names = FALSE, col_types = NULL, skip = 0, n_max = -1, progress = interactive()**)**

Чтение файла в raw вектор
read_file_raw(file**)**

Чтение строк файла в raw векторы
read_lines_raw(file, skip = 0, n_max = -1L, progress = interactive()**)**

Типы данных

Функции readr угадывают типы столбцов и преобразуют их, если считают уместным (но НИКОГДА строки в факторы автоматически).

Сообщение описывает типы столбцов в результате.

```
## Parsed with column specification:
## cols(
##   age = col_integer(),
##   sex = col_character(),
##   earn = col_double()
## )
```

age - целое число
sex - строка
earn - число double

1. Используйте **problems()** для диагностики проблем
x <- read_csv("file.csv"); problems(x)

2. Используйте col_ функцию для разбора
- **col_guess()** - по умолчанию
- **col_character()**
- **col_double()**, **col_euro_double()**
- **col_datetime(**format = ""**)** Также **col_date(**format = ""**)**, **col_time(**format = ""**)**
- **col_factor(**levels, ordered = FALSE**)**
- **col_integer()**
- **col_logical()**
- **col_number()**, **col_numeric()**
- **col_skip()**

```
x <- read_csv("file.csv", col_types = cols(
  A = col_double(),
  B = col_logical(),
  C = col_factor()))
```

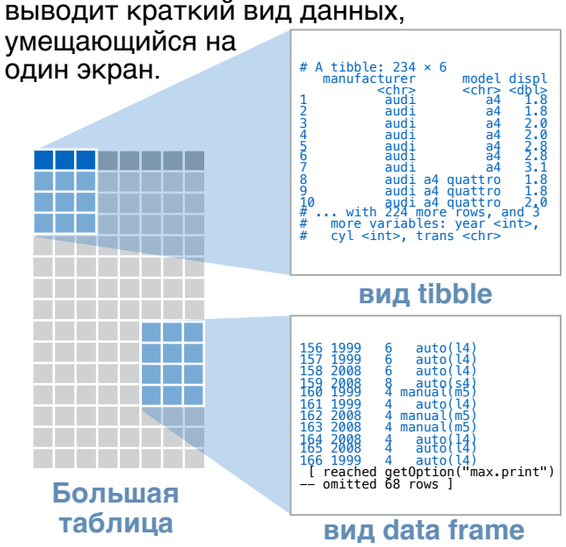
3. По-другому: считывайте как символьные векторы и разбирайте с parse_ функцией
- **parse_guess()**
- **parse_character()**
- **parse_datetime()** Также **parse_date()** и **parse_time()**
- **parse_double()**
- **parse_factor()**
- **parse_integer()**
- **parse_logical()**
- **parse_number()**

x$A <- parse_number(x$A)

Tibble - развитие data frame

Пакет **tibble** вводит новый S3 класс для хранения табличных данных, tibble. Tibble наследует класс data frame, но улучшает три направления:

- **Выбор элементов** - [всегда возвращает новый tibble, [[и $ - вектор.
- **Нет частичного соответствия** - Вы должны использовать полные имена столбцов при выборе элементов.
- **Отображение** - При печати tibble, R выводит краткий вид данных, умещающийся на один экран.




```
# A tibble: 234 × 6
   manufacturer      model displ
          <chr>      <chr> <dbl>
1          audi         a4   1.8
2          audi         a4   1.8
3          audi         a4   2.0
4          audi         a4   2.0
5          audi         a4   2.8
6          audi         a4   2.8
7          audi         a4   3.1
8          audi a4 quattro   1.8
9          audi a4 quattro   1.8
10         audi a4 quattro   2.0
# ... with 224 more rows, and 3
#   more variables: year <int>,
#   cyl <int>, trans <chr>
```

```
156 1999   6   auto(l4)
157 1999   6   auto(l4)
158 2008   6   auto(l4)
159 2008   8   auto(s4)
160 1999   4 manual(m5)
161 1999   4   auto(l4)
162 2008   4 manual(m5)
163 2008   4 manual(m5)
164 2008   4   auto(l4)
165 2008   4   auto(l4)
166 1999   4   auto(l4)
  [ reached getOption("max.print")
-- omitted 68 rows ]
```

- Управление видом по умолчанию опциями:
  options(tibble.print_max = n, tibble.print_min = m, tibble.width = Inf**)**
- Просмотр полных данных: **View()** или **glimpse()**
- Возврат к data frame: **as.data.frame()**

СОЗДАНИЕ TIBBLE ДВУМЯ СПОСОБАМИ

tibble(…**)**
Создает по столбцам.
tibble(x = 1:3, y = c("a","b","c"))

tribble(…**)**
Создает по строкам.
tribble(~x, ~y, 1, "a", 2, "b", 3, "c")

Оба создают этот tibble

```
A tibble: 3 × 2
      x     y
  <int> <chr>
1     1     a
2     2     b
3     3     c
```

as_tibble(x, …**)**
Конвертирует data frame в tibble.

enframe(x, name = "name", value = "value"**)**
Конверт. именованный вектор в tibble

is_tibble(x**)** Проверяет, явл. ли x tibble.

Опрятные данные с tidyr

Опрятные данные - это способ организации табличных данных. Это устанавливает структуру данных, согласованную между пакетами.

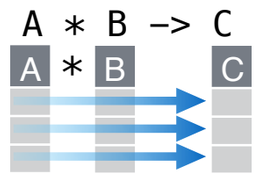
Таблица опрятна, если:

Каждая **переменная - столбец**

Каждое **наблюдение - строка**

В опрятных данных:

Легко обращаться к переменным, как к векторам



Сохр. наблюдения при векторизованных операциях

Форматирование данных

Изменение расположения элементов в таблице

Используйте **gather()** и **spread()** для реорганизации элементов таблицы.

gather(data, key, value, ..., na.rm = FALSE, convert = FALSE, factor_key = FALSE**)**

Перемещает имена столбцов в столбец **key**, собирая значения столбцов в общий столбец **value**.

table4a

country	1999	2000
A	0.7K	2K
B	37K	80K
C	212K	213K

→

country	year	cases
A	1999	0.7K
B	1999	37K
C	1999	212K
A	2000	2K
B	2000	80K
C	2000	213K

key value

gather(table4a, `1999`, `2000`, key = "year", value = "cases")

spread(data, key, value, fill = NA, convert = FALSE, drop = TRUE, sep = NULL**)**

Перемещает уникальные значения столбца **key** в имена столбцов, распространяя значения столбца **value** по новым столбцам.

table2

country	year	type	count
A	1999	cases	0.7K
A	1999	pop	19M
A	2000	cases	2K
A	2000	pop	20M
B	1999	cases	37K
B	1999	pop	172M
B	2000	cases	80K
B	2000	pop	174M
C	1999	cases	212K
C	1999	pop	1T
C	2000	cases	213K
C	2000	pop	1T

→

country	year	cases	pop
A	1999	0.7K	19M
A	2000	2K	20M
B	1999	37K	172M
B	2000	80K	174M
C	1999	212K	1T
C	2000	213K	1T

key value

spread(table2, type, count)

Работа с пропущенными значениями

drop_na(data, ...**)**
Убирает строки с NA в столбцах из …

x

x1	x2
A	1
B	NA
C	NA
D	3
E	NA

→

x1	x2
A	1
D	3

drop_na(x, x2)

fill(data, ..., .direction = c("down", "up")**)**
Заменяет NA в столбцах из … крайними не-NA значениями.

x

x1	x2
A	1
B	NA
C	NA
D	3
E	NA

→

x1	x2
A	1
B	1
C	1
D	3
E	3

fill(x, x2)

replace_na(data, replace = list(), ...**)**
Заменяет NA по столбцам.

x

x1	x2
A	1
B	NA
C	NA
D	3
E	NA

→

x1	x2
A	1
B	2
C	2
D	3
E	2

replace_na(x, list(x2 = 2))

Расширение таблиц - быстрое создание таблиц с комб-ми значений

complete(data, ..., fill = list()**)**
Добавляет к данным недостающие комбинации значений переменных из …
complete(mtcars, cyl, gear, carb)

expand(data, ...**)**
Создает новый tibble со всеми комбинациями значений переменных из …
expand(mtcars, cyl, gear, carb)

Деление ячеек

Функции для разделения или комбинирования ячеек в отдельные значения.

separate(data, col, into, sep = "[^[:alnum:]]+", remove = TRUE, convert = FALSE, extra = "warn", fill = "warn", ...**)**

Разделяет ячейки столбца по отдельным столбцам.

table3

country	year	rate
A	1999	0.7K/19M
A	2000	2K/20M
B	1999	37K/172M
B	2000	80K/174M
C	1999	212K/1T
C	2000	213K/1T

→

country	year	cases	pop
A	1999	0.7K	19M
A	2000	2K	20M
B	1999	37K	172
B	2000	80K	174
C	1999	212K	1T
C	2000	213K	1T

separate(table3, rate, into = c("cases", "pop"))

separate_rows(data, ..., sep = "[^[:alnum:].]+", convert = FALSE**)**

Разделяет ячейки столбца по отдельным строкам. Также **separate_rows_()**.

table3

country	year	rate
A	1999	0.7K/19M
A	2000	2K/20M
B	1999	37K/172M
B	2000	80K/174M
C	1999	212K/1T
C	2000	213K/1T

→

country	year	rate
A	1999	0.7K
A	1999	19M
A	2000	2K
A	2000	20M
B	1999	37K
B	1999	172M
B	2000	80K
B	2000	174M
C	1999	212K
C	1999	1T
C	2000	213K
C	2000	1T

separate_rows(table3, rate)

unite(data, col, ..., sep = "_", remove = TRUE**)**

Объединяет ячейки нескольких столбцов в один столбец.

table5

country	century	year
Afghan	19	99
Afghan	20	0
Brazil	19	99
Brazil	20	0
China	19	99
China	20	0

→

country	year
Afghan	1999
Afghan	2000
Brazil	1999
Brazil	2000
China	1999
China	2000

unite(table5, century, year, col = "year", sep = "")