readr, tibble va tidyr yordamida

Ma'lumotlar Importi

bo‘yicha qo‘llanma

Rning **tidyverse**i **tibble**da saqlanuvchi **toza ma'lumot**lar asosida qurilgan.

Mazkur qo‘llanmaning old qismi **readr** yordamida Rda matnli fayllarni o‘qishni ko‘rsatadi.

Orqa qismi esa, **tibble** yordamida tibblelarni yaratish va **tidyr** yordamida tozalashni ifodalaydi.

Ma'lumotlarning boshqa turlari

Quyidagi paketlar boshqa turdagi fayllarni import qilish uchun xizmat qiladi

- **haven** - SPSS, Stata va SAS fayl
- **readxl** - excel fayllari (.xls va .xlsx)
- **DBI** - ma'lumotlar ombori
- **jsonlite** - json
- **xml2** - XML
- **httr** - Web API
- **rvest** - HTML (Web Scraping)

Yozish funksiyalari

x R obyektini **path** nomli katalogga yozish:

write_csv(x, path, na = "NA", append = FALSE, col_names = !append**)**
Tibble/df ni vergul bilan ajratilgan faylga

write_delim(x, path, delim = " ", na = "NA", append = FALSE, col_names = !append**)**
Tibble/df ni ixtiyoriy belgi bilan ajratilgan faylga

write_excel_csv(x, path, na = "NA", append = FALSE, col_names = !append**)**
Tibble/df ni excel uchun CSV faylga o‘tkazish

write_file(x, path, append = FALSE**)**
Qatorni faylga o‘girish.

write_lines(x, path, na = "NA", append = FALSE**)**
Qatorli vektorni faylga o‘girish.

write_rds(x, path, compress = c("none", "gz", "bz2", "xz"), ...**)**
Obyektni RDS faylga o‘tkazish.

write_tsv(x, path, na = "NA", append = FALSE, col_names = !append**)**
Tibble/df ni tab bilan ajratilgan faylga o‘tkazish.

O‘qish funksiyalari

Jadval shaklidagi ma'lumotlarni tibblega aylantirish

Ushbu funksiyalar quyidagi umumiy argumentlarga ega:

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

read_csv()
Vergul bilan ajratilgan fayllar.
read_csv("file.csv")

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

→

A	B	C
1	2	3
4	5	NA

read_csv2()
Nuqta-vergul yordamida ajratilgan fayllar.
read_csv2("file2.csv")

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

→

A	B	C
1	2	3
4	5	NA

read_delim(delim, quote = "\"", escape_backslash = FALSE, escape_double = TRUE**)**
Ixtiyoriy belgi bilan ajratilgan fayllar.
read_delim("file.txt", delim = "|")

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

→

A	B	C
1	2	3
4	5	NA

read_fwf(col_positions**)**
O‘zgarmas kenglikli fayllarni o'qish.
read_fwf("file.fwf", col_positions = c(1, 3, 5))

read_tsv()
Tabulyatsiya bilan ajratilgan fayllarni o‘qish. Shuningdek, **read_table().** *read_tsv("file.tsv")*

Foydali argumentlar

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

Namunaviy fayl
write_csv (path = "file.csv", x = read_csv("a,b,c\n1,2,3\n4,5,NA"))

A	B	C
1	2	3
4	5	NA

Sarlavhasiz
*read_csv("file.csv", **col_names = FALSE**)*

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

Sarlavhali
*read_csv("file.csv", **col_names = c("x", "y", "z")**)*

1	2	3
4	5	NA

Qatorlarni tashlash
*read_csv("file.csv", **skip = 1**)*

A	B	C
1	2	3

Kichik to‘plamga o‘qish
*read_csv("file.csv", **n_max = 1**)*

A	B	C
1	2	3
NA	NA	NA

Yo‘q qiymatlar
*read_csv("file.csv", **na = c("4", "5", ".")**)*

Jadval shaklida bo‘lmagan ma'lumotlarni o‘qish

read_file(file, locale = default_locale()**)**
Faylni yagona Stringga o‘qish

read_file_raw(file**)**
Fayli vektor shaklida o‘qish

read_lines(file, skip = 0, n_max = -1L, locale = default_locale(), na = character(), progress = interactive()**)**
Har bir qatorni alohida string sifatida o‘qish.

read_lines_raw(file, skip = 0, n_max = -1L, progress = interactive()**)**
Har bir qatorni alohida vektor sifatida o‘qish

read_log(file, col_names = FALSE, col_types = NULL, skip = 0, n_max = -1, progress = interactive()**)**
Apache stilidagi log fayllar

Ma'lumot parsingi

readr funksiyalari har bir ustunning turini aniqlashga va mos kelsa o‘zgartirishga harakat qiladi (biroq u stringlarni faktorlarga avtomatik tarzda almashtirmaydi).

Quyida keltirilgan xabar, natijadagi har bir ustunning turini quyidagicha ifodalaydi.

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

age bu butun son

sex bu belgi

earn bu double toifali son

1. Muammoni aniqlash **problems()**
   x <- read_csv("file.csv"); problems(x)

2. Parsingga col_ funksiya bilan ko‘rsatma berish
   - **col_guess()** - odatiy holda
   - **col_character()**
   - **col_double()**
   - **col_euro_double()**
   - **col_datetime(**format = ""**).** Shuningdek, **col_date(**format = ""**)** va **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. Yoki, belgi sifatida o'qib, parse_ funksiyalari bilan parsing qilish.
   - **parse_guess(**x, na = c("", "NA"), locale = default_locale()**)**
   - **parse_character(**x, na = c("", "NA"), locale = default_locale()**)**
   - **parse_datetime(**x, format = "", na = c("", "NA"), locale = default_locale()**)** va **parse_date()** hamda **parse_time()**
   - **parse_double(**x, na = c("", "NA"), locale = default_locale()**)**
   - **parse_factor(**x, levels, ordered = FALSE, na = c("", "NA"), locale = default_locale()**)**
   - **parse_integer(**x, na = c("", "NA"), locale = default_locale()**)**
   - **parse_logical(**x, na = c("", "NA"), locale = default_locale()**)**
   - **parse_number(**x, na = c("", "NA"), locale = default_locale()**)**

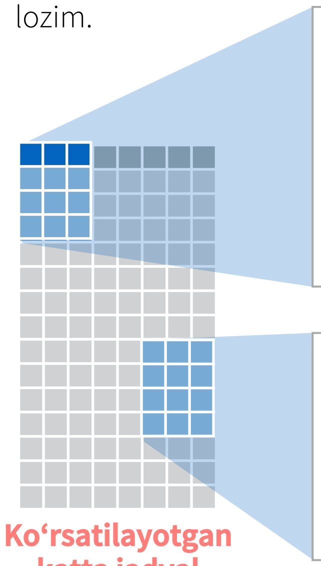
   x$A <- parse_number(x$A)

browseVignettes(package = c("readr", "tibble", "tidyr")) da yanada ko‘proq o‘rganing • readr 1.1.0 • tibble 1.2.12 • tidyr 0.6.0 • Yangilangan: 2017-01

Tibble - mukammallashgan data frame

tibble paketi jadval ma'lumotlarini saqlash uchun data frame vorisi bo‘lgan tibble klassini taqdim qiladi. U quyidagilari bilan afzal:

- **Ko‘rsatish** - Tibble chop qilinganda, undagi ma'lumot qisqa shaklda ekranga sig‘adigan qilib chiqariladi.
- **Kichik to‘plamga ajratish** - [har doim tibble qaytaradi, [[va $ doimo vektor qaytaradi.
- **Qisman mos kelish mavjud emas** - to‘plamga ajratishda ustunning to‘liq nomini keltirishinggiz lozim.



```
# 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>
```

tibble shaklida

```
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 ]
```

Ko‘rsatilayotgan katta jadval

data frame shaklida

- Odatiy holni quyidagicha moslash mumkin:

  options(tibble.print_max = n, tibble.print_min = m, tibble.width = Inf**)**

- Ma'lumotlarni ko‘rish **View(**x, title**)** yoki **glimpse(**x, width = NULL, …**)**
- Data framega o‘tkazish **as.data.frame()** (ba'zi eski paketlar uchun zarur)

tibbleni 2 xil usulda yaratish

tibble(…)
Ustunlar bo‘yicha
tibble(x = 1:3,
y = c("a", "b", "c"))

Ikkalasi ham ushbu tibbleni yaratadi

tribble(…)
Satrlar bo‘yicha
tribble(
~x, ~y,
1, "a",
2, "b",
3, "c")

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

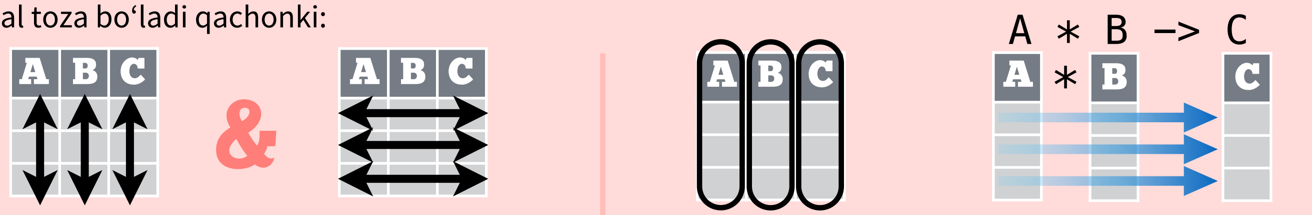
as_tibble(x, …**)** data frameni tibblega aylantirish

enframe(x, name = "name", value = "value"**)** Vektorni "name" va "value" ustunli tibblega o‘zgartirish.

is_tibble(x**)** x tibbleligini tekshirish.

tidyr yordamida ma'lumotni tozalash

Toza ma'lumot bu jadvalli ma'lumotlarni tashkillashtirish yo‘li. U paketlararo o‘zgarmas tuzilmani ta'minlaydi. Jadval toza bo‘ladi qachonki:



Har bir **o‘zgaruvchi** o‘z **ustunida** bo‘lsa

Har bir **kuzatuv, holat** o‘z **satrida** bo‘lsa

O‘zgaruvchilardan vektor sifatida foydalanish mumkin

Vektorli operatsiyalarda satrlar o‘zgarmasa

Ma'lumotlar shaklini o‘zgartirish - jadvalda ma'lumot keltirilishini o‘zgartirish

gather() va **spread()** dan jadval ma'lumotlarini qayta taqsimlashda qo‘llash mumkin. Bunda, key : value juftligidan foydalaniladi.

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

Gather, ustun nomlarini **key** ustuniga, satr qiymatlarini **value** ustuniga ko‘chiradi.

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**)**

Spread, **key** ustunining unikal qiymatlarini ustun nomlariga aylantiradi va **value** ustuni qiymatlarini yangi ustunga joylaydi.

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

key value

→

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

spread(table2, type, count)

Yo‘q qiymatlar ustida ishlash

drop_na(data, ...**)**

NAdan iborat satrlarni tashlab yuborish.

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")**)**

NAlarni eng yaqin qiymatga o‘zgartirish

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(), ...**)**

Ustunga ko‘ra almashtirish.

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), x2)

Jadvallarni kengaytirish - qiymatlar to‘plami bilan jadval yaratish

complete(data, ..., fill = list()**)**

…da keltirilgan o‘zgaruvchilarning mavjud bo‘lmagan qiymatlar to‘plamini asosiy ma'lumotlarga qo‘shish

complete(mtcars, cyl, gear, carb)

expand(data, ...**)**

…da keltirilgan o‘zgaruvchilarning mumkin bo‘lgan qiymatlaridan iborat yangi tibble yaratish

expand(mtcars, cyl, gear, carb)

Ajratish va birlashtirish

Ushbu funksiyalardan yacheykalarni alohida qiymatlarga ajratish yoki birlashtirish uchun foydalaning.

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

Ustundagi har bir yacheykani ajratib bir nechta ustunlar hosil qilish.

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**)**

Ustundagi har bir yacheykani ajratib, bir nechta qator yaratish. Shuningdek, **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**)**

Bir nechta ustunlarni yagona ustunga birlashtirish.

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 = "")

RStudio® RStudio, Inc. ning savdo belgisi • CC BY RStudio info@rstudio.com • 844-448-1212 • rstudio.com

browseVignettes(package = c("readr", "tibble", "tidyr")) da yanada ko‘proq o‘rganing • readr 1.1.0 • tibble 1.2.12 • tidyr 0.6.0 • Yangilangan: 2017-01