

tbl\_df

подмножество  
строк

подмножество  
столбцов

упорядочить  
строки

уникальные  
комбинации

добавить  
столбец

tidyr

summarise и  
group\_by

# Продвинутая манипуляция с датафреймами: пакеты dplyr и tidyr

Г. Мороз

# Данные

В данной презентации все примеры будут приводиться на примере датасета из работы [Chi-kuk 2007] (доступна по ссылке <http://goo.gl/MKfSc6>). В работе исследовались речь 7 гомосексуальных и 7 гетеросексуальных носителей кантонского диалекта языка юэ. В датасете есть следующие переменные:

- долгота  $s$  (`s.duration.ms`)
- долгота гласных (`vowel.duration.ms`)
- среднее значение ЧОТ (`average.fo.Hz`)
- диапазон ЧОТ (`fo.range.Hz`)
- сколько носителей воспринимает говорящего как гомосексуала (`perceived.as.homo`)
- сколько носителей воспринимает говорящего как гетеросексуала (`perceived.as.hetero`)
- ориентация говорящего (`orientation`)
- возраст говорящего (`age`)

презентация доступна: <http://idrv.ms/1PMoiZj>

# tbl\_df

Первая важная функция пакет dplyr это формат tbl\_df, который позволяет выводить на экран значительно более удобным способом, а в остальном это все тот же датафрейм:

```
homo <- read.csv("http://goo.gl/Zjr9aF") # скачиваем данные
library(dplyr) homo <- tbl_df(homo) # преобразуем данные
head(homo) # первые шесть значений
```

```
# A tibble: 6 x 10
  speaker s.duration.ms vowel.duration.ms average.f0.Hz
  <fctr>    <dbl>          <dbl>          <dbl>
1 A         61.40         112.60         119.51
2 B         63.90         126.49         100.29
3 C         55.08         126.81         114.90
4 D         78.11         119.17         126.61
5 E         64.71          93.68         130.76
6 F         67.00         127.87         150.79
```

```
# ... with 6 more variables: f0.range.Hz <dbl>,
# perceived.as.homo <int>, perceived.as.hetero <int>,
# perceived.as.homo.percent <dbl>, orientation <fctr>,
# age <int>
```

# Подмножество строк по условию

Какие информанты старше 28?

base R

```
homo[homo$age > 28, ]
```

dplyr

```
homo %>%  
  filter(age > 28)
```

# A tibble: 7 x 10

	speaker	s.duration.ms	vowel.duration.ms	average.f0.Hz
	<fctr>	<dbl>	<dbl>	<dbl>
1	A	61.40	112.60	119.51
3	C	55.08	126.81	114.90
4	D	78.11	119.17	126.61
6	F	67.00	127.87	150.79
5	J	59.59	121.01	123.90
6	K	62.94	137.37	119.48
7	N	57.67	118.02	121.48

```
# ... with 6 more variables: f0.range.Hz <dbl>,  
# perceived.as.homo <int>, perceived.as.hetero <int>,  
# perceived.as.homo.percent <dbl>, orientation <fctr>,  
# age <int>
```

Оператор `%>%` называется конвейер (по-английски pipe) и вставляется при помощи сочетания клавиш **Ctrl+Shift+m**.

презентация доступна: <http://idrv.ms/iPMoiZj>

# Подмножество строк по номеру

Какие информанты хранятся под номером 3, 4, 5, 6, 7?

base R

homo[3:7, ]

dplyr

homo %>%  
slice(3:7)

# A tibble: 5 x 10

	speaker	s.duration.ms	vowel.duration.ms	average.f0.Hz
	<fctr>	<dbl>	<dbl>	<dbl>
1	C	55.08	126.81	114.90
2	D	78.11	119.17	126.61
3	E	64.71	93.68	130.76
4	F	67.00	127.87	150.79
5	N	57.67	118.02	121.48

# ... with 6 more variables: f0.range.Hz <dbl>,  
# perceived.as.homo <int>, perceived.as.hetero <int>,  
# perceived.as.homo.percent <dbl>, orientation <fctr>,  
# age <int>

## Подмножество столбцов по названию

Выделяет столбцы с информантов, ориентацией и возрастом.

base R

dplyr

```
cbind.data.frame(homo$speaker, homo$age)
```

```
homo %>%  
  select(speaker, age)
```

```
# A tibble: 14 x 2
```

	speaker <fctr>	age <int>
1	A	30
2	B	19
3	C	29
4	D	36
5	E	27
6	F	33
7	G	28
8	H	22
9	I	22
10	J	40
11	K	30
12	L	25
13	M	20
14	N	29

# Подмножество столбцов по номеру

Как выбрать столбцы под номером 8, 9, 10?

**base R**

`homo[, 8:10]`

**dplyr**

`homo %>%  
select(8:10)`

# A tibble: 14 x 3

	perceived.as.homo.percent	orientation	age
	<dbl>	<fctr>	<int>
1	0.28	hetero	30
2	0.80	hetero	19
3	0.36	homo	29
4	0.60	homo	36
5	0.40	homo	27
6	0.68	homo	33
7	0.80	hetero	28
8	0.84	hetero	22
9	0.80	homo	22
10	0.32	homo	40
11	0.84	homo	30
12	0.32	hetero	25
13	0.36	hetero	20
14	0.16	hetero	29

презентация доступна: <http://idrv.ms/1PMoiZj>

# Подмножество столбцов по названию

Выделяет столбцы от одного до другого.

**dplyr**

homo %>%

`select(speaker:average.f0.Hz)`

# можно комбинировать оба способа

# A tibble: 14 x 4

	speaker	s.duration.ms	vowel.duration.ms	average.f0.Hz
	<fctr>	<dbl>	<dbl>	<dbl>
1	A	61.40	112.60	119.51
2	B	63.90	126.49	100.29
3	C	55.08	126.81	114.90
4	D	78.11	119.17	126.61
5	E	64.71	93.68	130.76
6	F	67.00	127.87	150.79
7	G	65.39	147.52	128.96
8	H	62.46	120.13	105.26
9	I	60.45	140.44	109.86
10	J	59.59	121.01	123.90
11	K	62.94	137.37	119.48
12	L	53.31	112.05	146.20
13	M	45.13	133.74	155.34
14	N	57.67	118.02	121.48



# Упорядочить строки по значению столбцов

Упорядочивает строки сначала по ориентации, потом по возрасту.

**base R**

**dplyr**

```
homo[order(homo$orientation, homo$age), ]
```

```
homo %>%  
  arrange(orientation, age)
```

# A tibble: 14 x 3

# выбраны только важные столбцы

	speaker <fctr>	orientation <fctr>	age <int>
1	B	hetero	19
2	M	hetero	20
3	H	hetero	22
4	L	hetero	25
5	G	hetero	28
6	N	hetero	29
7	A	hetero	30
8	I	homo	22
9	E	homo	27
10	C	homo	29
11	K	homo	30
12	F	homo	33
13	D	homo	36
14	J	homo	40

# Упорядочить строки по значению столбцов

В обратном порядке.

**base R**

```
homo[order(-homo$age), ]
```

# A tibble: 14 x 2

	speaker <fctr>	age <int>
1	J	40
2	D	36
3	F	33
4	A	30
5	K	30
6	N	29
7	C	29
8	G	28
9	E	27
10	L	25
11	H	22
12	I	22
13	M	20
14	B	19

**dplyr**

```
homo %>%  
  arrange(desc(age))
```

# выбран только важный столбец

# Уникальные комбинации строк

Уникальные значения строк в столбце ориентация:

**base R**

```
unique(homo$orientation)
```

```
[1] hetero homo  
Levels: hetero homo
```

**dplyr**

```
homo %>%  
  distinct(orientation)
```

```
orientation  
1 hetero  
2 homo
```

# Уникальные комбинации строк

В случае, если хочется выбрать несколько столбцов, средства R становятся сложнее:

## base R

```
unique(homo[c("orientation "perceived.as.homo")])
```

## dplyr

```
homo %>%  
  distinct(orientation, perceived.as.homo)
```

# Добавить и преобразовать столбцы

Добавляет столбцы с минимумом и максимумом  $f_0$  каждого носителя:

## base R

```
homo$f0.min <- homo$average.f0.Hz - homo$f0.range.Hz/2  
homo$f0.max <- homo$average.f0.Hz + homo$f0.range.Hz/2
```

## dplyr

```
homo %>%  
  mutate(  
    f0.min = average.f0.Hz - f0.range.Hz/2,  
    f0.max = average.f0.Hz + f0.range.Hz/2)
```

# Добавить и преобразовать столбцы

tbl\_df

подмножество  
строк

подмножество  
столбцов

упорядочить  
строки

уникальные  
комбинации

добавить  
столбец

tidyr

summarise и  
group\_by

# A tibble: 14 x 3

	speaker	f0.min	f0.max
	<fctr>	<dbl>	<dbl>
1	A	93.26	145.76
2	B	43.29	157.29
3	C	63.30	166.50
4	D	97.21	156.01
5	E	112.06	149.46
6	F	129.79	171.79
7	G	69.86	188.06
8	H	77.41	133.11
9	I	61.66	158.06
10	J	68.05	179.75
11	K	75.68	163.28
12	L	117.30	175.10
13	M	105.09	205.59
14	N	102.78	140.18

# выбраны только важные столбцы

# tidyr

В основе пакета tidyr лежит понятие Tidy Data:

- каждая переменная — колонка
- каждое наблюдение — строка
- все данные, связанные с одними тем же типом измерения, собраны в одну таблицу

Именно такой формат *ожидают* большинство статистических функций и функций машинного обучения.

## df.short

	consonant	initial	intervocalic	final
1	stops	123	57	30
2	fricatives	87	77	69
3	affricates	73	82	12
4	nasals	7	78	104

## df.long

	consonant	position	number
1	stops	initial	123
2	fricatives	initial	87
3	affricates	initial	73
4	nasals	initial	7
5	stops	intervocalic	57
6	fricatives	intervocalic	77
7	affricates	intervocalic	82
8	nasals	intervocalic	78
9	stops	final	30
10	fricatives	final	69
11	affricates	final	12
12	nasals	final	104



# Short format $\Leftrightarrow$ Long format: gather() and spread()

```
df.short <- data.frame(
  consonant = c("stops", "fricatives", "affricates", "nasals"),
  initial = c(123, 87, 73, 7),
  intervocal = c(57, 77, 82, 78),
  final = c(30, 69, 12, 104))
```

```
library(tidyr)
```

```
df.long <- # short to long
  df.short %>%
  gather(position, number, initial:final)
```

```
df.short <- # long to short
  df.long %>%
  spread(position, number)
```

## Пакет dplyr: summarise и group\_by

Функция summarise() (или summarise()) позволяет получить любые описательные статистики, которые доступны в R, в целом не отличается от них практически ничем.

```
homo %>%  
  summarise(min(age), mean(s.duration.ms))
```

```
# A tibble: 1 x 2  
  min(age) mean(s.duration.ms)  
  <int>      <dbl>  
1      19      61.22429
```

## Пакет dplyr: summarise и group\_by

В сочетании с командой group\_by(), которая группирует данные по какому-то параметру/параметрам, данная функция становится мощным инструментом.

```
homo %>%  
  group_by(orientation)  
  summarise(count = n())
```

```
# A tibble: 2 x 2  
  orientation count  
    <fctr>    <int>  
1   hetero      7  
2    homo      7
```

```
homo %>%  
  group_by(orientation)  
  summarise(mean(s.duration.ms))
```

```
# A tibble: 2 x 2  
  orientation mean(s.duration.ms)  
    <fctr>          <dbl>  
1   hetero    58.46571  
2    homo    63.98286
```

В base R подобное можно сделать при помощи функции aggregate():

```
aggregate(speaker~orientation, length, data = homo)
```

tbl\_df

подмножество  
строк

подмножество  
столбцов

упорядочить  
строки

уникальные  
комбинации

добавить  
столбец

tidyr

summarise и  
group\_by

# Спасибо за внимание!

Пишите письма

[agricolamz@gmail.com](mailto:agricolamz@gmail.com)