





# data.table

Владимир Волохонский  
Специалист по исследованиям



-Я каждый раз злюсь и никак не могу  
этот синтаксис усвоить!

- Да. А потом ты не понимаешь, как  
можно писать иначе.



data.table – это пакет  
library(data.table)



`data.table` – это `data.frame`

Быстрый и умный `data.frame`

```
DF<-data.frame(a=c(1,2,3),b=1)
```

```
DT<-data.table(a=c(1,2,3),b=1)
```

## Плюсы

- Производительность
- Лаконичность
- Код легко читать
- Обратная совместимость с `data.frame`-ориентированными пакетами

## Минусы

- Путаница с `data.frame` при написании кода
- Диалектность
- Код трудно читать

```
> library(data.table)
> data("airquality")
> DT<-data.table(airquality)
> str(DT)
Classes 'data.table' and 'data.frame': 153 obs. of 6 variables:
 $ Ozone   : int  41 36 12 18 NA 28 23 19 8 NA ...
 $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
 $ Wind    : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
 $ Temp    : int  67 72 74 62 56 66 65 59 61 69 ...
 $ Month   : int  5 5 5 5 5 5 5 5 5 5 ...
 $ Day     : int  1 2 3 4 5 6 7 8 9 10 ...
- attr(*, ".internal.selfref")=<externalptr>
> |
```

Синтаксис с \$ работает по прежнему

```
> head(DT$Ozone)
[1] 41 36 12 18 NA 28
```

Синтаксис [,] кардинально меняется

DT[**фильтр строк**,**выражение**,**параметр**]

Выражение – это не фильтр колонок!

```
> DT[1,5]
[1] 5
```



DT[фильтр строк]

Долой повторения!

Вместо `subDF<-DF[DF$Ozone>36 & DF$Wind<8,]`

Теперь `subDT<-DT[Ozone>36 & Wind<8]`



DT[,выражение]

Два типа выражений

Вывод:

```
Oz<-DT[,Ozone] #вектор!  
subDT<-DT[,.(Ozone,Wind,S=Solar.R*1000)]
```

Изменение:

```
DT[,S:=Solar.R*1000]
```



>>

4

>>

3

>>

2

>>

1

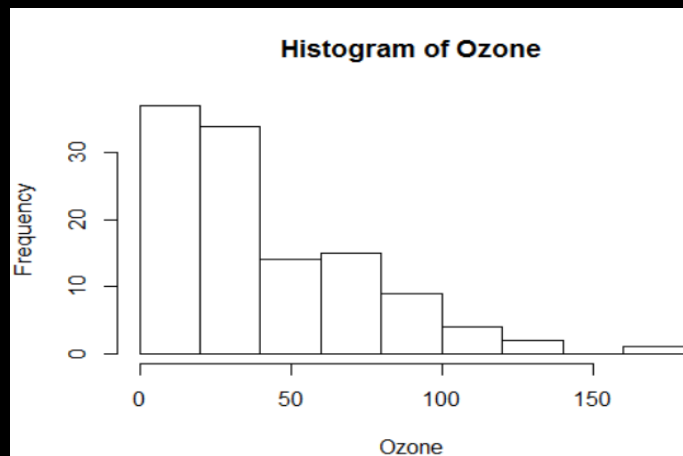
>>

0

>>

DT[,выражение]

```
> DT[,{hist(Ozone)  
+   print("Удивительно, но можно и так")}]  
[1] "Удивительно, но можно и так"  
[1] "Удивительно, но можно и так"
```



DT[,выражение]

Два типа выражений

Вывод:

```
Oz<-DT[,Ozone] #вектор!  
subDT<-DT[,.(Ozone,Wind,S=Solar.R*1000)]
```

Изменение:

```
DT[,S:=Solar.R*1000]
```



>>



4



>>



3



>>



>>



2



>>



1



>>



0



>>



>>



>>



>>



>>





DT[,выражение]

Можно делать цепочки!

DT[Wind>20][,m:=5][,Wind/m]

DT[,параметр]

by expression  
очень простая агрегация!

```
> DT[,mean(Ozone,na.rm=T),by=Month]
  Month      V1
1:     5 23.61538
2:     6 29.44444
3:     7 59.11538
4:     8 59.96154
5:     9 31.44828
> |
```

DT[,параметр]

by expression

прямо внутри data.table

DT[,MonthOzone:=mean(Ozone,na.rm=T),  
by=Month]

	Ozone ↕	Solar.R ↕	Wind ↕	Temp ↕	Month ↕	Day ↕	MonthOzone ↕
30	115	223	5.7	79	5	30	23.61538
31	37	279	7.4	76	5	31	23.61538
32	NA	286	8.6	78	6	1	29.44444
33	NA	287	9.7	74	6	2	29.44444
34	NA	242	16.1	67	6	3	29.44444

DT[,параметр]

Можно фильтровать внутри выражения  
DT[,MonthOzone:=mean(Ozone[Wind>9],  
na.rm=T),by=Month]

	Ozone	Solar.R	Wind	Temp	Month	Day	MonthOzone
30	115	223	5.7	79	5	30	17.75000
31	37	279	7.4	76	5	31	17.75000
32	NA	286	8.6	78	6	1	30.25000
33	NA	287	9.7	74	6	2	30.25000
34	NA	242	16.1	67	6	3	30.25000

DT[,параметр]

Это не то же самое!

DT[Wind>9, MonthOzone:=mean(Ozone,  
na.rm=T), by=Month]

	Ozone ↕	Solar.R ↕	Wind ↕	Temp ↕	Month ↕	Day ↕	MonthOzone ↕
30	115	223	5.7	79	5	30	NA
31	37	279	7.4	76	5	31	NA
32	NA	286	8.6	78	6	1	NA
33	NA	287	9.7	74	6	2	30.25
34	NA	242	16.1	67	6	3	30.25





```
DT[,параметр]
```

```
.SD и .SDcol
```

Это фильтр колонок!

```
vars<-c("Ozone","Wind")  
DT[,.SD,.SDcol=vars]
```

Логические и числовые векторы-  
фильтры тоже можно использовать

DT[,параметр]

Сложный случай...

vars<-c("Ozone","Wind")

DT[,lapply(.SD,mean,na.rm=T),by=Month,.  
SDcol=vars]

DT[,c(vars):=lapply(.SD,as.character),.SDc  
ol=vars]

```
> DT[,lapply(.SD,mean,na.rm=T),by=Month,.SDcol=vars]
  Month   Ozone   Wind
1:     5 23.61538 11.622581
2:     6 29.44444 10.266667
3:     7 59.11538  8.941935
4:     8 59.96154  8.793548
5:     9 31.44828 10.180000
```

DT[,.N,параметр]      DT[,.I,параметр]

```
> DT[,.N,by=Month]
```

	Month	N
1:	5	31
2:	6	30
3:	7	31
4:	8	31
5:	9	30

```
> DT[,.I[Wind>15]]
```

```
[1] 9 18 22 25 34 48 113 129 135 148
```

```
> DT[Wind>15,.I]
```

```
[1] 1 2 3 4 5 6 7 8 9 10
```

```
> DT[,.(n=.N,sol=mean(Solar.R,na.rm=T)),by=list(Month,d=ifelse(Day>10,1,0))]
```

	Month	d	n	sol
1:	5	0	10	172.6250
2:	5	1	21	184.9474
3:	6	0	10	249.9000
4:	6	1	20	160.3000
5:	7	0	10	233.3000
6:	7	1	21	208.4762
7:	8	0	10	156.7143
8:	8	1	21	176.9048
9:	9	0	10	188.4000
10:	9	1	20	156.9500

merge и keys

У data.table могут быть ключевые поля.

И тогда можно делать так:

```
setkey(DT,Month)
setkey(DT2,Month)
DT[DT2]
```

Что эквивалентно

```
merge(DT,DT2,by="Month",all.y=T)
```

имена и типы ключей для merge должны совпадать!

## Нюансы

- В `data.table` не поддерживается тип данных времени `POSIXlt` (`strptime`)
- `unique(DT)` может давать неожиданный результат. Чтобы результат был ожидаемым, надо делать `unique(DT,by=NULL)`
  - Есть команда `setnames`