



Программирование в среде R

Шевцов Василий Викторович,
директор ДИТ РУДН, shevtsov_vv@rudn.university

Циклы и условия

Логические операторы

Оператор	Описание
>	Больше
>=	Больше или равно
<	Меньше
<=	Меньше или равно
==	Равно
!=	Не равно
&	Логическое И
	Логическое ИЛИ
!	Логическое НЕ

Операторы условия

- Краткий вариант оператора if
 - **if(условие) выражение**
 - **условие** — любой оператор условия ($<$, $>$, $>=$, $<=$, $==$, $!=$), результатом выполнения которого является логический вектор единичной длины. Если значение вектора TRUE, то выполняется выражение.
 - **выражение** — одно или несколько выражений, выполняемых в случае верности условия. Если задано несколько выражений, то они должны быть заключены в фигурные скобки $\{ \}$ и разделяться точкой с запятой (если на одной строке)
 - Оператор возвращает значение выражения в случае верности условия или ничего не возвращает (NULL).

```
> x<-2;y<-3
> if(y>x){z<-x+y;z}
[1] 5
> if(y<x){z<-x+y;z}
> |
```

Операторы условия

- Полный вариант оператора if
 - **if(условие) выражение1 else выражение2**
 - **условие** — любой оператор условия (<, >, >=, <=, ==, !=), результатом выполнения которого является логический вектор единичной длины. Если значение вектора TRUE, то выполняется выражение.
 - **выражение1** — одно или несколько выражений, выполняемых в случае верности условия
 - **выражение2** — одно или несколько выражений, выполняемых в случае ложного условия

```
> if(y<x){z<-x+y;z} else {z<-x*y;z}  
[1] 6
```

Операторы условия

```
> x <- 2
> y <- 3
> if (x > 4 & y < 1) {"да"} else {"нет"}
[1] "нет"
```

```
> x <- 2
> if (x>0) {"больше нуля"} else if (x==0) {"равно нулю"} else {"меньше нуля"}
[1] "больше нуля"
> x <- 0
> if (x>0) {"больше нуля"} else if (x==0) {"равно нулю"} else {"меньше нуля"}
[1] "равно нулю"
> x <- -0
> if (x>0) {"больше нуля"} else if (x==0) {"равно нулю"} else {"меньше нуля"}
[1] "равно нулю"
> x <- -1
> if (x>0) {"больше нуля"} else if (x==0) {"равно нулю"} else {"меньше нуля"}
[1] "меньше нуля"
```

Операторы условия

- оператор **ifelse**

- **ifelse(условие, yes, no)**

- позволяет переменной в зависимости от выполнения (невыполнения) некоторого условия принимать различные значения. Отличие от оператора `if` состоит в том, что здесь условие является логическим вектором любой заданной размерности (зависит от размерности сравниваемых объектов). Порядковый номер элемента логического вектора и его значение определяет какой элемент вектора **yes** или **no** берётся новой переменной.

```
> x <- c(-1,0,1,2)
> ifelse(x>=0,"positive","negative")
[1] "negative" "positive" "positive" "positive"
```

```
> if (x>0) {"больше нуля"} else if (x==0) {"равно нулю"} else {"меньше нуля"}
[1] "меньше нуля"
Warning messages:
1: In if (x > 0) { :
  the condition has length > 1 and only the first element will be used
2: In if (x == 0) { :
  the condition has length > 1 and only the first element will be used
```

Операторы условия

```
> x<-c(1,2,3,4,5)
> y<-c(2,2,4,3,5)
> z<-ifelse(x==y,c(11,12,13,14,15),c(-11,-12,-13,-14,-15));z
[1] -11  12 -13 -14  15
```

если $x[1]==y[1]$ то $yes[1]$ иначе $no[1]$
если $x[2]==y[2]$ то $yes[2]$ иначе $no[2]$
...
если $x[n]==y[n]$ то $yes[n]$ иначе $no[n]$

} z

Операторы условия

- оператор **ifelse**

- Пример. При вычислении sqrt из отрицательных чисел получаем сообщение об ошибке. Используем ifelse для замены отрицательных

```
> x<--2:3
> sqrt(x)
[1]      NaN      NaN 0.000000 1.000000 1.414214 1.732051
warning message:
In sqrt(x) : NaNs produced
> sqrt(ifelse(x>=0,x,NA))
[1]      NA      NA 0.000000 1.000000 1.414214 1.732051
```

- операции с матрицами

```
> x<-rep(c(1,2,3),3);x
[1] 1 2 3 1 2 3 1 2 3
> x<-matrix(x,3,3);x
      [,1] [,2] [,3]
[1,]    1    1    1
[2,]    2    2    2
[3,]    3    3    3
> z<-ifelse(x>1,"больше","меньше");z
      [,1] [,2] [,3]
[1,] "меньше" "меньше" "меньше"
[2,] "больше" "больше" "больше"
[3,] "больше" "больше" "больше"
> z<-ifelse(x>1,c(1:9),c(11:19));z
      [,1] [,2] [,3]
[1,]   11   14   17
[2,]    2    5    8
[3,]    3    6    9
```

Операторы цикла. for

- **for(переменная in последовательность) выражение**

Оператор цикла. Пока переменная находится в рамках заданной числовой последовательности, выполняется выражение (или блок выражений).

```
> x<-1:10;y<-10:1
> for(i in 1:10){if (x[i]>y[i]) {print("больше")} else {print("меньше, равно")}}
[1] "меньше, равно"
[1] "меньше, равно"
[1] "меньше, равно"
[1] "меньше, равно"
[1] "меньше, равно"
[1] "больше"
[1] "больше"
[1] "больше"
[1] "больше"
[1] "больше"
```

```
> z<-integer(10)
> for(i in 1:10){if (x[i]>y[i]) z[i]<-x[i] else z[i]<-y[i]};z
[1] 10 9 8 7 6 6 7 8 9 10
```

```
> for (i in c(-1,0,1,2)){print (i)}
```

```
[1] -1
[1] 0
[1] 1
[1] 2
```

```
> x <- c(-1,0,1,2)
```

```
> for (i in x){print (i)}
```

```
[1] -1
[1] 0
[1] 1
[1] 2
```

Операторы цикла. for

```
> for (i in 1:nrow(mtcars)){print(mtcars$mpg[i])}
```

```
[1] 21  
[1] 21  
[1] 22.8  
[1] 21.4  
[1] 18.7  
[1] 18.1  
[1] 14.3  
[1] 24.4  
[1] 22.8  
[1] 19.2  
[1] 17.8  
[1] 16.4  
[1] 17.3  
[1] 15.2  
[1] 10.4  
[1] 10.4  
[1] 14.7
```

```
> for (i in 1:nrow(mtcars)){if (mtcars$cyl[i]==4) {print(mtcars$mpg[i])}}
```

```
[1] 22.8  
[1] 24.4  
[1] 22.8  
[1] 32.4  
[1] 30.4  
[1] 33.9  
[1] 21.5  
[1] 27.3  
[1] 26  
[1] 30.4  
[1] 21.4
```

Операторы цикла. for

```
> View (df)
> df <- mtcars
> for (i in 1:nrow(df)) {
+   if (df$vs[i]==0){df$vs2[i] <- "v"} else {df$vs2[i] <- "s"}
+
+ }
> View (df)
```

	mpg	cyl	displacement	hp	drat	wt	qsec	vs	am	gear	carb	vs2
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4	v
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4	v
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1	s
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1	s
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2	v
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1	s
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4	v
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2	s
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2	s

Операторы цикла. for

```
> df$vs3 <- ifelse(df$vs==0,"v","s")  
> View (df)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb	vs2	vs3
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4	v	v
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4	v	v
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1	s	s
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1	s	s
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2	v	v
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1	s	s
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4	v	v
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2	s	s

```
> df$vs==0
```

```
[1] TRUE TRUE FALSE FALSE TRUE FALSE TRUE FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE  
[18] FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE FALSE TRUE FALSE TRUE TRUE TRUE FALSE
```

Операторы цикла. while

- **while(условие) выражение**

Пока выполняется условие вычисляется выражение, как только результатом условия становится FALSE, выходим из цикла.

```
> x<-5
> while(x<10){print(x);x<-x+1}
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
```

```
> x<-5
> while(x<10){if(x==7){break};print(x);x<-x+1}
[1] 5
[1] 6
```

```
> x<-5
> while(x<10){if(x==7){x<-x+1;next};print(x);x<-x+1}
[1] 5
[1] 6
[1] 8
[1] 9
```

Операторы цикла. repeat

- **repeat** выражение

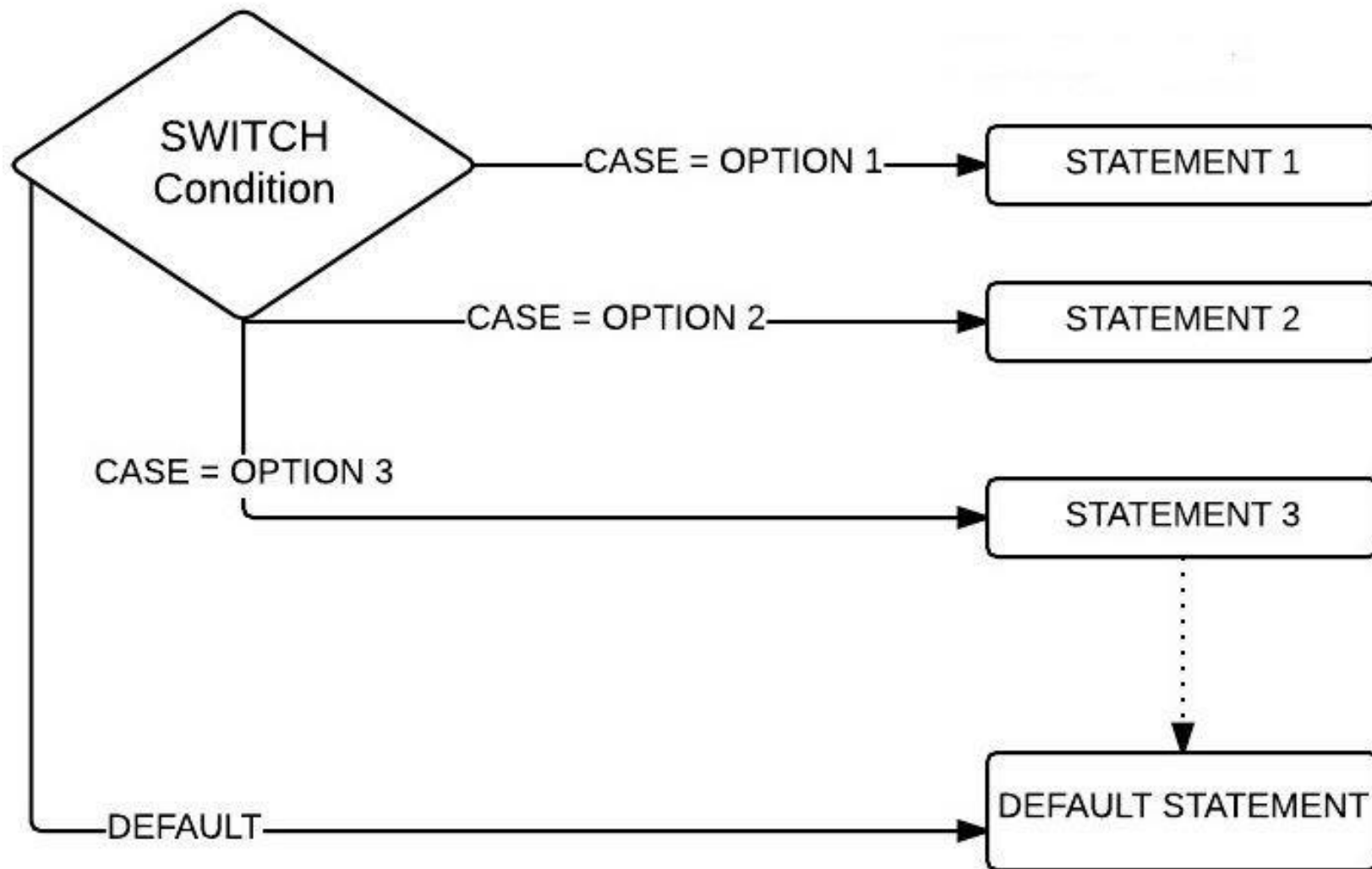
repeat создаёт бесконечный цикл, в котором вычисляется выражение (или блок выражений). Для выхода из этого цикла нужно использовать **break**, а также условный оператор в качестве одного из выражений.

```
> x<-5  
> repeat{print(x); if(x==8){break}; x<-x+1}  
[1] 5  
[1] 6  
[1] 7  
[1] 8  
.
```

Операторы. switch

- **switch(управляющее выражение, альтернативные действия)**
- **позволяет выполнять одну из нескольких операций в зависимости от результатов управляющего выражения. Управляющее выражение возвращает:**
 - либо целое число (от 1 до числа альтернатив), которое является номером выполняемого действия;
 - либо символьную переменную (строку), соответствующую имени выполняемой операции.
- **Если возвращаемое контрольным выражением значение не соответствует ни номеру выполняемой операции, ни имени, то результатом оператора switch будет NULL.**
- **Оператор switch не является самостоятельным, используется либо внутри функций, либо внутри других управляющих конструкций**

Операторы. switch



Операторы. switch

```
> x<-numeric(5)
> for(i in 1:5){x[i]<-switch(i,11,12,13,14,15)};x
[1] 11 12 13 14 15
> for(i in 1:5){x[i]<-switch(i,"11","12","13","14","15")};x
[1] "11" "12" "13" "14" "15"
> for(i in 1:5){x[i]<-switch(i,print("11"),print("12"),print("13"),print("14"),print("15"))};x
[1] "11"
[1] "12"
[1] "13"
[1] "14"
[1] "15"
[1] "11" "12" "13" "14" "15"
> for(i in 1:5){x[i]<-switch(i,{a<-10;print(a)},print("12"),print("13"),print("14"),print("15"))};x
[1] 10
[1] "12"
[1] "13"
[1] "14"
[1] "15"
[1] "10" "12" "13" "14" "15"
```

Операторы. switch

```
> x<-letters[1:5]
> for(i in x){switch(i,"a"=print("A"),"b"=print("B"),"c"=print("C"),print("default"))}
[1] "A"
[1] "B"
[1] "C"
[1] "default"
[1] "default"
```

```
switch (caseSwitch)
{
  case 1:
    Console.WriteLine("Case 1");
    break;
  case 2:
    Console.WriteLine("Case 2");
    break;
  default:
    Console.WriteLine("Default case");
    break;
}
```

Simple CASE expression:

```
CASE input_expression
  WHEN when_expression THEN result_expression [ ...n ]
  [ ELSE else_result_expression ]
END
```

Searched CASE expression:

```
CASE
  WHEN Boolean_expression THEN result_expression [ ...n ]
  [ ELSE else_result_expression ]
END
```

Описательные статистики

mpg	расход топлива (количества миль на галлон топлива)
cyl	кол-во цилиндров
disp	объем двигателя
hp	мощность двигателя (лошадиные силы)
drat	передаточное число заднего моста
wt	вес
qsec	значение времени разгона
vs	тип двигателя (v-образный, рядный)
am	тип коробки передач
gear	кол-во передач
carb	число карбюраторов

Преобразование в факторы

```
> df <- mtcars
> str(df)
'data.frame':   32 obs. of  11 variables:
 $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl : num   6  6  4  6  8  6  8  4  4  6 ...
 $ disp: num  160 160 108 258 360 ...
 $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num   3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt  : num   2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num  16.5 17 18.6 19.4 17 ...
 $ vs  : num   0  0  1  1  0  1  0  1  1  1 ...
 $ am  : num   1  1  1  0  0  0  0  0  0  0 ...
 $ gear: num   4  4  4  3  3  3  3  4  4  4 ...
 $ carb: num   4  4  1  1  2  1  4  2  2  4 ...

> df$vs <- factor(df$vs, labels = c("V","S"))
> df$am <- factor(df$am, labels = c("auto","manual"))
> str(df)
'data.frame':   32 obs. of  11 variables:
 $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl : num   6  6  4  6  8  6  8  4  4  6 ...
 $ disp: num  160 160 108 258 360 ...
 $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num   3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt  : num   2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num  16.5 17 18.6 19.4 17 ...
 $ vs  : Factor w/ 2 levels "V","S": 1 1 2 2 1 2 1 2 2 2 ...
 $ am  : Factor w/ 2 levels "auto","manual": 2 2 2 1 1 1 1 1 1 1 ...
 $ gear: num   4  4  4  3  3  3  3  4  4  4 ...
 $ carb: num   4  4  1  1  2  1  4  2  2  4 ...
```

Статистики

```
> median(df$mpg)
[1] 19.2
> mean(df$mpg)
[1] 20.09062
> sd(df$mpg)
[1] 6.026948
> range(df$mpg)
[1] 10.4 33.9
```

```
> mean(df$mpg[df$cy1==6])
[1] 19.74286
> mean(df$mpg[df$cy1==6 & df$vs=="V"])
[1] 20.56667
> sd(df$hp[df$cy1!= 3 & df$am=="auto"])
[1] 53.9082
```

Статистики

```
> # Арифметическая средняя:  
> mean(mtcars$mpg)  
[1] 20.09062  
> # Медиана  
> median(mtcars$mpg)  
[1] 19.2  
> # Дисперсия:  
> var(mtcars$mpg)  
[1] 36.3241  
> # Стандартное отклонение:  
> sd(mtcars$mpg)  
[1] 6.026948  
> # Минимальное значение:  
> min(mtcars$mpg)  
[1] 10.4  
> # Максимальное значение:  
> max(mtcars$mpg)  
[1] 33.9  
> # расчет стандартной ошибки среднего  
> sd(mtcars$mpg)/sqrt(length(mtcars$mpg))  
[1] 1.065424
```

$$S_{\bar{x}} = \frac{S}{\sqrt{n}}$$

Статистики

```
> # Квантили
> quantile(mtcars$mpg)
  0%    25%    50%    75%   100%
10.400 15.425 19.200 22.800 33.900
> quantile(mtcars$mpg, probs = seq(0,1,0.1))
  0%   10%   20%   30%   40%   50%   60%   70%   80%   90%  100%
10.40 14.34 15.20 15.98 17.92 19.20 21.00 21.47 24.08 30.09 33.90
```

x	numeric vector whose sample quantiles are wanted, or an object of a class for which a method has been defined (see also ‘details’). NA and NaN values are not allowed in numeric vectors unless na.rm is TRUE.
probs	numeric vector of probabilities with values in $[0, 1]$. (Values up to $2e-14$ outside that range are accepted and moved to the nearby endpoint.)
na.rm	logical; if true, any NA and NaN's are removed from x before the quantiles are computed.
names	logical; if true, the result has a names attribute. Set to FALSE for speedup with many probs.
type	an integer between 1 and 9 selecting one of the nine quantile algorithms detailed below to be used.

aggregate

```
> aggregate(x=df$hp, by=list(df$vs), FUN=mean)
  Group.1      x
1      V 189.7222
2      S  91.35714
> res <- aggregate(x=df$hp, by=list(df$vs), FUN=mean)
> is.data.frame(res)
[1] TRUE
```

	Group.1	x
1	V	189.72222
2	S	91.35714

```
> colnames(res)
[1] "Group.1" "x"
> colnames(res) <- c("vs", "am")
```

	vs	am
1	V	189.72222
2	S	91.35714

Группировка

По одной переменной

```
> aggregate(hp ~ vs, df, mean)
```

	vs	hp
1	V	189.72222
2	S	91.35714

По двум переменным

```
> aggregate(hp ~ vs + am, df, mean)
```

	vs	am	hp
1	V	auto	194.16667
2	S	auto	102.14286
3	V	manual	180.83333
4	S	manual	80.57143

```
> aggregate(x = df[,-c(8,9)], by = list(df$am), FUN=median)
```

	Group.1	mpg	cyl	disp	hp	drat	wt	qsec	gear	carb
1	auto	17.3	8	275.8	175	3.15	3.52	17.82	3	3
2	manual	22.8	4	120.3	109	4.08	2.32	17.02	4	2

```
> aggregate(x = df[,c(1,3)], by=list(df$am, df$vs), FUN=sd)
```

	Group.1	Group.2	mpg	disp
1	auto	V	2.774396	71.82349
2	manual	V	4.008865	95.23362
3	auto	S	2.471071	49.13072
4	manual	S	4.757701	18.80213

пакет psych. describe

```
> df <- mtcars
> df$vs <- factor(df$vs, labels = c("V","S"))
> df$am <- factor(df$am, labels = c("auto","manual"))
> describe(x = df)
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
mpg	1	32	20.09	6.03	19.20	19.70	5.41	10.40	33.90	23.50	0.61	-0.37	1.07
cyl	2	32	6.19	1.79	6.00	6.23	2.97	4.00	8.00	4.00	-0.17	-1.76	0.32
disp	3	32	230.72	123.94	196.30	222.52	140.48	71.10	472.00	400.90	0.38	-1.21	21.91
hp	4	32	146.69	68.56	123.00	141.19	77.10	52.00	335.00	283.00	0.73	-0.14	12.12
drat	5	32	3.60	0.53	3.70	3.58	0.70	2.76	4.93	2.17	0.27	-0.71	0.09
wt	6	32	3.22	0.98	3.33	3.15	0.77	1.51	5.42	3.91	0.42	-0.02	0.17
qsec	7	32	17.85	1.79	17.71	17.83	1.42	14.50	22.90	8.40	0.37	0.34	0.32
vs*	8	32	1.44	0.50	1.00	1.42	0.00	1.00	2.00	1.00	0.24	-2.00	0.09
am*	9	32	1.41	0.50	1.00	1.38	0.00	1.00	2.00	1.00	0.36	-1.92	0.09
gear	10	32	3.69	0.74	4.00	3.62	1.48	3.00	5.00	2.00	0.53	-1.07	0.13
carb	11	32	2.81	1.62	2.00	2.65	1.48	1.00	8.00	7.00	1.05	1.26	0.29

```
> describe(x = df[, -c(8,9)])
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
mpg	1	32	20.09	6.03	19.20	19.70	5.41	10.40	33.90	23.50	0.61	-0.37	1.07
cyl	2	32	6.19	1.79	6.00	6.23	2.97	4.00	8.00	4.00	-0.17	-1.76	0.32
disp	3	32	230.72	123.94	196.30	222.52	140.48	71.10	472.00	400.90	0.38	-1.21	21.91
hp	4	32	146.69	68.56	123.00	141.19	77.10	52.00	335.00	283.00	0.73	-0.14	12.12
drat	5	32	3.60	0.53	3.70	3.58	0.70	2.76	4.93	2.17	0.27	-0.71	0.09
wt	6	32	3.22	0.98	3.33	3.15	0.77	1.51	5.42	3.91	0.42	-0.02	0.17
qsec	7	32	17.85	1.79	17.71	17.83	1.42	14.50	22.90	8.40	0.37	0.34	0.32
gear	8	32	3.69	0.74	4.00	3.62	1.48	3.00	5.00	2.00	0.53	-1.07	0.13
carb	9	32	2.81	1.62	2.00	2.65	1.48	1.00	8.00	7.00	1.05	1.26	0.29

пакет psych. describeBy

```
> describeBy(x = df[,-c(8,9)], group = df$vs)
```

Descriptive statistics by group

group: V													
	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
mpg	1	18	16.62	3.86	15.65	16.42	2.97	10.40	26.00	15.60	0.48	-0.05	0.91
cyl	2	18	7.44	1.15	8.00	7.62	0.00	4.00	8.00	4.00	-1.74	1.94	0.27
disp	3	18	307.15	106.77	311.00	308.52	72.65	120.30	472.00	351.70	-0.26	-1.06	25.16
hp	4	18	189.72	60.28	180.00	186.81	48.18	91.00	335.00	244.00	0.45	-0.15	14.21
drat	5	18	3.39	0.47	3.18	3.37	0.32	2.76	4.43	1.67	0.74	-0.73	0.11
wt	6	18	3.69	0.90	3.57	3.68	0.50	2.14	5.42	3.28	0.54	-0.43	0.21
qsec	7	18	16.69	1.09	17.02	16.75	0.85	14.50	18.00	3.50	-0.71	-0.80	0.26
gear	8	18	3.56	0.86	3.00	3.50	0.00	3.00	5.00	2.00	0.90	-1.07	0.20
carb	9	18	3.61	1.54	4.00	3.44	1.48	2.00	8.00	6.00	1.17	1.33	0.36

group: S													
	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
mpg	1	14	24.56	5.38	22.80	24.34	6.00	17.80	33.90	16.10	0.41	-1.40	1.44
cyl	2	14	4.57	0.94	4.00	4.50	0.00	4.00	6.00	2.00	0.85	-1.36	0.25
disp	3	14	132.46	56.89	120.55	127.11	61.82	71.10	258.00	186.90	0.80	-0.49	15.21
hp	4	14	91.36	24.42	96.00	92.00	32.62	52.00	123.00	71.00	-0.24	-1.61	6.53
drat	5	14	3.86	0.51	3.92	3.86	0.26	2.76	4.93	2.17	-0.28	0.46	0.14
wt	6	14	2.61	0.72	2.62	2.63	0.95	1.51	3.46	1.95	-0.17	-1.68	0.19
qsec	7	14	19.33	1.35	19.17	19.24	1.02	16.90	22.90	6.00	0.86	1.25	0.36
gear	8	14	3.86	0.53	4.00	3.83	0.00	3.00	5.00	2.00	-0.17	-0.09	0.14
carb	9	14	1.79	1.05	1.50	1.67	0.74	1.00	4.00	3.00	1.13	-0.03	0.28

list > data.frame

```
> desr1 <- describeBy(x = df[,-c(8,9)], group = df$vs, mat=TRUE)  
> View(desr1)
```



	item	group1	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
mpg1	1	V	1	18	16.616667	3.8606994	15.6500	16.418750	2.9652000	10.400	26.000	15.600	0.4848484	-0.04617697	0.9099756
mpg2	2	S	1	14	24.557143	5.3789782	22.8000	24.341667	6.0045300	17.800	33.900	16.100	0.4055612	-1.40081941	1.4375924
cyl1	3	V	2	18	7.4444444	1.1490263	8.0000	7.625000	0.0000000	4.000	8.000	4.000	-1.7397531	1.94499359	0.2708281
cyl2	4	S	2	14	4.571429	0.9376145	4.0000	4.500000	0.0000000	4.000	6.000	2.000	0.8488760	-1.36173469	0.2505880
disp1	5	V	3	18	307.150000	106.7652193	311.0000	308.525000	72.6474000	120.300	472.000	351.700	-0.2554760	-1.05725636	25.1648035
disp2	6	S	3	14	132.457143	56.8932430	120.5500	127.108333	61.8244200	71.100	258.000	186.900	0.8026718	-0.48933244	15.2053588
hp1	7	V	4	18	189.722222	60.2815019	180.0000	186.812500	48.1845000	91.000	335.000	244.000	0.4528693	-0.14510499	14.2084863
hp2	8	S	4	14	91.357143	24.4244743	96.0000	92.000000	32.6172000	52.000	123.000	71.000	-0.2399009	-1.60915113	6.5277153
drat1	9	V	5	18	3.392222	0.4739515	3.1800	3.366875	0.3187590	2.760	4.430	1.670	0.7404542	-0.73394358	0.1117115
drat2	10	S	5	14	3.859286	0.5057890	3.9200	3.861667	0.2594550	2.760	4.930	2.170	-0.2754065	0.46046904	0.1351778
wt1	11	V	6	18	3.688556	0.9040308	3.5700	3.676875	0.5003775	2.140	5.424	3.284	0.5425973	-0.43390250	0.2130821

пакет psych. describeBy

```
> desr1 <- describeBy(x = df[,-c(8,9)], group = df$vs, mat=TRUE, digits = 2, fast=TRUE)
> View(desr1)
```

	item	group1	vars	n	mean	sd	min	max	range	se
mpg1	1	V	1	18	16.62	3.86	10.40	26.00	15.60	0.91
mpg2	2	S	1	14	24.56	5.38	17.80	33.90	16.10	1.44
cyl1	3	V	2	18	7.44	1.15	4.00	8.00	4.00	0.27
cyl2	4	S	2	14	4.57	0.94	4.00	6.00	2.00	0.25
disp1	5	V	3	18	307.15	106.77	120.30	472.00	351.70	25.16
disp2	6	S	3	14	132.46	56.89	71.10	258.00	186.90	15.21
hp1	7	V	4	18	189.72	60.28	91.00	335.00	244.00	14.21
hp2	8	S	4	14	91.36	24.42	52.00	123.00	71.00	6.53
drat1	9	V	5	18	3.39	0.47	2.76	4.43	1.67	0.11
drat2	10	S	5	14	3.86	0.51	2.76	4.93	2.17	0.14
wt1	11	V	6	18	3.69	0.90	2.14	5.42	3.28	0.21
wt2	12	S	6	14	2.61	0.72	1.51	3.46	1.95	0.19

Группировка по двум переменным

```
> describeBy(x = df$qsec, group = list(df$vs, df$am), mat=TRUE, digits = 2, fast=TRUE)
```

	item	group1	group2	vars	n	mean	sd	min	max	range	se
X11	1	V	auto	1	12	17.14	0.80	15.41	18.00	2.59	0.23
X12	2	S	auto	1	7	19.97	1.46	18.30	22.90	4.60	0.55
X13	3	V	manual	1	6	15.80	1.09	14.50	17.02	2.52	0.45
X14	4	S	manual	1	7	18.70	0.95	16.90	19.90	3.00	0.36

Спасибо за внимание!



Шевцов Василий Викторович

shevtsov_vv@rudn.university
+7(903)144-53-57