Introduction to R Cheat Sheet

by josi68 via cheatography.com/197242/cs/41550/

Arithmetics

Addition	+
Subtraction	-
Multiplication	*
Division	/
Modulo	%%
Exponentiation	^

Modulo returns the remainder of the division of the number to the left by the number on its right, for example 5 modulo 3 or 5 %% 3 is 2.

Comparison operators

Less than	<
More than	>
Less than or equal to	<=
Greater than or equal to	>=
Equal to each other	==
Not equal to each other	!=

Selecting by comparison

```
# Poker and roulette winnings from Monday to Friday:
poker_ vector <- c(140, -50, 20, -120, 240)
roulet te_ vector <- c(-24, -50, 100, -350, 10)
days_v ector <- c("M ond ay", " Tue sda y", " Wed nes day ", " Thu rsd ay", " Fri day ")
names( pok er_ vector) <- days_v ector
names( rou let te_ vector) <- days_v ector
# Which days did you make money on roulette?
select ion _vector <- roulet - te_ vector > 0
# Select from roulet te_ vector these days
roulet te_ win nin g_days <- roulet te_ vec tor [se lec tio - n_v ector]
```

Data Types

Decimal values	4.5	Numerics
Whole numbers	4	Integers
Boolean values	TRUE / FALSE	Logical
Text / String	"Text"	Characters

Show the data type: class(*data*)

Lists

Create a list	my_list <- list(element1, element2)
Give names to the list items	my_list <- list(name1 = your_comp1, name2 = your_comp2)

```
# Adapt list() call to give the components names
my_list <- list(vec = my_vector,
mat = my_matrix,
df = my_df)

#or if the list was alread created
names(my_list) <- c("vec", "mat", "df")
```

Selecting components in a list

One way to select a component is using the numbered position of that component. For example, to "grab" the first component of shining_list you type shining_list[[1]]

A quick way to check this out is typing it in the console. Important to remember: to select elements from vectors, you use single square brackets: [ ]. Don't mix them up!

You can also refer to the names of the components, with [[ ]] or with the $ sign. Both will select the data frame representing the reviews:

shining_list[["reviews"]]

shining_list$reviews

Selecting components in a list (cont)

Besides selecting components, you often need to select specific elements out of these components. For example, with shining_list[[2]][1] you select from the second component, actors (shining_list[[2]]), the first element ([1]). When you type this in the console, you will see the answer is Jack Nicholson.

Vector Basics

Assign value to variable	my_var <- 4
Numeric vector	numeric_vector <- c(1, 10, 49)
Character_vector	character_vector <- c("a", "b", "c")
Boolean vector	boolean_vector <- c(TRUE, FALSE, TRUE)
Naming a vector	names(numeric_vector) <- c("Jack", "Jill", "Johanna")
Sum of the elements in the vector	sum(vector_name)
Select element 3 of the vector	element <- vector_name[3]
Select elements 2, 3, 4, 5 of the vector	elements <- vector_name[3:5]

Factors

```
# Animals - Turn vector character elements into nominal factors
animal s_v ector <- c("E lep - han t", " Gir aff e", " Don - key ", " Hor se")
factor _an ima ls_ vector <- factor (an ima ls_ vector)
factor _an ima ls_ vector
```

Factors (cont)

```
> # Temperature - Turn vector character elements into ordinal factors
temperature_vector <- c("High", "Low", "High","Low", "Medium")
factor_temperature_vector <- factor(temperature_vector, order = TRUE, levels = c("Low", "Medium", "High"))
factor_temperature_vector
```

When factors are ordinal::
order = TRUE

To give the order of the ordinal factors:
levels = c("Low", "Medium", "High"))

Data Frames

Show the first couple of lines	head(data_frame)
Show the last couple of lines	tail(data:frame)
Summarize data frame (min, max, median, quartiles)	summary(data_frame)
Structure (nr. obs, var., names, data type...)	str(data_frame)

unlike matrixes, df can have different types of data - BUT all variables need to have the same length (unlike for lists)

Create data frame from vectors + select values

```
# Definition of vectors
name <- c("M erc ury ", " Ven - us", " Ear th",
                    " Mar s",
" Jup ite r", " Sat urn ",
                    " Ura - nus ", " Nep tun e")
type <- c("T err estrial planet ",
                    " Ter res - trial planet ",
```

Create data frame from vectors + select values (cont)

```
>           "Terrestrial planet",
          "Terrestrial planet", "Gas giant",
          "Gas giant", "Gas giant", "Gas giant")
diameter <- c(0.382, 0.949, 1, 0.532,
              11.209, 9.449, 4.007, 3.883)
rotation <- c(58.64, -243.02, 1, 1.03,
              0.41, 0.43, -0.72, 0.67)
rings <- c(FALSE, FALSE, FALSE, FALSE,
TRUE, TRUE, TRUE, TRUE)
# Create a data frame from the vectors
planets_df <- data.frame(name, type,
diameter, rotation, rings)
# Select first 5 values of diameter column
planets_df[1:5, "diameter"]
# Select the rings variable from planets_df
rings_vector <- planets_df$rings
# Select planets with diameter < 1
subset(planets_df, subset = diameter <1)
```

Order the data

In data analysis you can sort your data according to a certain variable in the dataset. In R, this is done with the help of the function order().
order() is a function that gives you the ranked position of each element when it is applied on a variable, such as a vector for example:

```
a <- c(100, 10, 1000)
order(a)
[1] 2 1 3
```

Order the data (cont)

10, which is the second element in a, is the smallest element, so 2 comes first in the output of order(a). 100, which is the first element in a is the second smallest element, so 1 comes second in the output of order(a).
This means we can use the output of order(a) to reshuffle a:

```
a[order(a)]
[1] 10 100 1000
```

Matrices

Construct Matrix with 3 rows that contain the numbers 1 to 9	matrix(1:9, byrow = TRUE, nrow = 3)
From Vector to Matrix	Matrix_names <- matrix(vector_name, byrow = TRUE, nrow = 3)
Totals for each row of a matrix	rowSums(my_matrix)
Total for each row of a matrix	colSums()
Adding columns	big matrix <- cbind(vector1, matrix1)
Adding rows	rbind
Select all elements of the first column	matrix[,1]
Select all elements of the first row	matrix[1,]
Select 2nd element of 3rd column	matrix[2,3]
Create matrix with the data on the rows 1, 2, 3 and columns 2, 3, 4.	matrix[1:3,2:4]

Matrices (cont)

Average of the matrix elements	mean(matrix_name)
Summary of Matrix (and other stuff)	summary(matrix_name)

The argument byrow indicates that the matrix is filled by the rows. If we want the matrix to be filled by the columns, we just place byrow = FALSE

all data in a matrix should be of the same type. Otherwise, create a data frame

Naming a Matrix

```
# Box office Star Wars (in
millions!)
new_hope <- c(460.998, 314.4)
empire _st rikes <- c(290.475,
247.900)
return _jedi <- c(309.306,
165.8)
# Construct matrix
star_w ars _matrix <- matrix -
(c( new _hope, empire _st rikes,
return _jedi), nrow = 3, byrow =
TRUE)
# Vectors region and titles,
used for naming
region <- c("U S", " non -US ")
titles <- c("A New Hope", "The
Empire Strikes Back", " Return
of the Jedi")
# Name the columns with region
colnam es( sta r_w ars _ma trix)
<- region
# Name the rows with titles
rownam es( sta r_w ars _ma trix)
<- titles
# Print out star_w ars _matrix
star_w ars _matrix
```