

<https://intro2r.com/> Chapter 3

CSCI 297b, Spring 2023

April 20, 2023

R basic data types

Numeric data are numbers that contain a decimal.

Integers are whole numbers.

Logical data take on the value of either TRUE or FALSE. There's also another special type of logical called NA to represent missing values.

Character data are used to represent string values. You can think of character strings as something like a word (or multiple words). A special type of character string is a factor, which is a string but with additional attributes (like levels or an order). We'll cover factors later.

R basic data types

```
1 num <- 2.2
2 class(num)
3 ## [1] "numeric"
4
5 char <- "hello"
6 class(char)
7 ## [1] "character"
8
9 logi <- TRUE
10 class(logi)
11 ## [1] "logical"
```

R basic data types

```
1 is.numeric(num)
2 ## [1] TRUE
3
4 is.character(num)
5 ## [1] FALSE
6
7 is.character(char)
8 ## [1] TRUE
9
10 is.logical(logi)
11 ## [1] TRUE
```

R type conversion

```
1 # coerce numeric to character
2 class(num)
3 ## [1] "numeric"
4 num_char <- as.character(num)
5 num_char
6 ## [1] "2.2"
7 class(num_char)
8 ## [1] "character"
9
10 # coerce character to numeric!
11 class(char)
12 ## [1] "character"
13 char_num <- as.numeric(char)
14 ## Warning: NAs introduced by coercion
```

Scalars and Vectors

- A vector with a single element is called a scalar.
- Vectors can contain any single type.
- You can't mix types in a vector.
- NA can mix with any type.



scalar



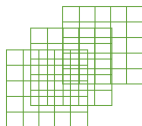
vector

Matrices and arrays

- A matrix is a vector with additional attributes called *dimensions*.
- Arrays are multidimensional matrices.
- Matrices and arrays can contain only a single type.
- They may also contain NAs.



matrix



array

Creating matrices and arrays

```
1 my_mat <- matrix(1:16, nrow = 4, byrow = TRUE)
2 my_mat
3 ##      [,1] [,2] [,3] [,4]
4 ## [1,]    1    2    3    4
5 ## [2,]    5    6    7    8
6 ## [3,]    9   10   11   12
7 ## [4,]   13   14   15   16
8
9 my_array <- array(1:16, dim = c(2, 4, 2))
10 my_array
11 ## , , 1
12 ##
13 ##      [,1] [,2] [,3] [,4]
14 ## [1,]    1    3    5    7
15 ## [2,]    2    4    6    8
16 ##
17 ## , , 2
18 ##
19 ##      [,1] [,2] [,3] [,4]
20 ## [1,]    9   11   13   15
21 ## [2,]   10   12   14   16
```


Optional row and column names

```
1 rownames(my_mat) <- c("A", "B", "C", "D")
2 colnames(my_mat) <- c("a", "b", "c", "d")
3 my_mat
4 ##      a  b  c  d
5 ## A   1  2  3  4
6 ## B   5  6  7  8
7 ## C   9 10 11 12
8 ## D  13 14 15 16
```

Transpose a matrix

```
1 my_mat_t <- t(my_mat)
2 my_mat_t
3 ##      A B  C  D
4 ## a  1  5  9 13
5 ## b  2  6 10 14
6 ## c  3  7 11 15
7 ## d  4  8 12 16
```

Diagonal elements

```
1 my_mat_diag <- diag(my_mat)
2 my_mat_diag
3 ## [1]  1  6 11 16
```

Matrix arithmetic

```
1 mat.1 <- matrix(c(2, 0, 1, 1), nrow = 2)
2                                     # notice that the matrix has been filled
3                                     # column-wise by default
4 mat.1
5 ##      [,1] [,2]
6 ## [1,]    2    1
7 ## [2,]    0    1
8
9 mat.2 <- matrix(c(1, 1, 0, 2), nrow = 2)
10 mat.2
11 ##      [,1] [,2]
12 ## [1,]    1    0
13 ## [2,]    1    2
14
15 mat.1 + mat.2                      # matrix addition
16 ##      [,1] [,2]
17 ## [1,]    3    1
18 ## [2,]    1    3
19 mat.1 * mat.2                      # element by element products
20 ##      [,1] [,2]
21 ## [1,]    2    0
22 ## [2,]    0    2
```

Matrix multiplication

```
1 mat.1 <- matrix(c(2, 0, 1, 1), nrow = 2)
2 mat.1
3 ##      [,1] [,2]
4 ## [1,]    2    1
5 ## [2,]    0    1
6
7 mat.2 <- matrix(c(1, 1, 0, 2), nrow = 2)
8 mat.2
9 ##      [,1] [,2]
10 ## [1,]    1    0
11 ## [2,]    1    2
12
13
14 mat.1 %*% mat.2          # matrix multiplication
15 ##      [,1] [,2]
16 ## [1,]    3    2
17 ## [2,]    1    2
```

Lists

- Notice the double bracket `[[]]` for list items.

```
1 list_1 <- list(c("black", "yellow", "orange"),
2               c(TRUE, TRUE, FALSE, TRUE, FALSE, FALSE),
3               matrix(1:6, nrow = 3))
4 list_1
5 ## [[1]]
6 ## [1] "black" "yellow" "orange"
7 ##
8 ## [[2]]
9 ## [1] TRUE TRUE FALSE TRUE FALSE FALSE
10 ##
11 ## [[3]]
12 ##      [,1] [,2]
13 ## [1,]    1    4
14 ## [2,]    2    5
15 ## [3,]    3    6
```

List elements can be named

```
1 list_2 <- list(colours = c("black", "yellow", "orange"),
2               evaluation = c(TRUE, TRUE, FALSE,
3                             TRUE, FALSE, FALSE),
4               time = matrix(1:6, nrow = 3))
5 list_2
6 ## $colours
7 ## [1] "black" "yellow" "orange"
8 ##
9 ## $evaluation
10 ## [1] TRUE TRUE FALSE TRUE FALSE FALSE
11 ##
12 ## $time
13 ##      [,1] [,2]
14 ## [1,]    1    4
15 ## [2,]    2    5
16 ## [3,]    3    6
```

List elements can be renamed using `names`

```
1 names(list_1) <- c("colours", "evaluation", "time")
2 list_1
3 ## $colours
4 ## [1] "black" "yellow" "orange"
5 ##
6 ## $evaluation
7 ## [1] TRUE TRUE FALSE TRUE FALSE FALSE
8 ##
9 ## $time
10 ##      [,1] [,2]
11 ## [1,]    1    4
12 ## [2,]    2    5
13 ## [3,]    3    6
```


Data frames

treat	nitrogen	block	height	weight	leafarea	shootarea	flowers
tip	medium	1	7.5	7.62	11.7	31.9	1
tip	medium	1	10.7	12.14	14.1	46.0	10
tip	medium	1	11.2	12.76	7.1	66.7	10
tip	medium	1	10.4	8.78	11.9	20.3	1
tip	medium	1	10.4	13.58	14.5	26.9	4
tip	medium	1	9.8	10.08	12.2	72.7	9
notip	low	2	3.7	8.10	10.5	60.5	6
notip	low	2	3.2	7.45	14.1	38.1	4
notip	low	2	3.9	9.19	12.4	52.6	9
notip	low	2	3.3	8.92	11.6	55.2	6
notip	low	2	5.5	8.44	13.5	77.6	9
notip	low	2	4.4	10.60	16.2	63.3	6

- Most used data structure for real world data.
- Each row contains an individual **observation**.
- Each column contains a measured **variable**.
- Each column is a vector of a single type.
- Columns can be different types.

Data frames

treat	nitrogen	block	height	weight	leafarea	shootarea	flowers
tip	medium	1	7.5	7.62	11.7	31.9	1
tip	medium	1	10.7	12.14	14.1	46.0	10
tip	medium	1	11.2	12.76	7.1	66.7	10
tip	medium	1	10.4	8.78	11.9	20.3	1
tip	medium	1	10.4	13.58	14.5	26.9	4
tip	medium	1	9.8	10.08	12.2	72.7	9
notip	low	2	3.7	8.10	10.5	60.5	6
notip	low	2	3.2	7.45	14.1	38.1	4
notip	low	2	3.9	9.19	12.4	52.6	9
notip	low	2	3.3	8.92	11.6	55.2	6
notip	low	2	5.5	8.44	13.5	77.6	9
notip	low	2	4.4	10.60	16.2	63.3	6

- Each row is an individual petunia plant.
- treat and nitrogen are **categorical** variables (factors)
- treat has 2 levels, tip and notip
- nitrogen has 3 levels, low, medium, high
- height, weight, leafarea, and shootarea are numeric
- flowers is an integer
- block uses integers, but should be treated as a factor

Data frames

treat	nitrogen	block	height	weight	leafarea	shootarea	flowers
tip	medium	1	7.5	7.62	11.7	31.9	1
tip	medium	1	10.7	12.14	14.1	46.0	10
tip	medium	1	11.2	12.76	7.1	66.7	10
tip	medium	1	10.4	8.78	11.9	20.3	1
tip	medium	1	10.4	13.58	14.5	26.9	4
tip	medium	1	9.8	10.08	12.2	72.7	9
notip	low	2	3.7	8.10	10.5	60.5	6
notip	low	2	3.2	7.45	14.1	38.1	4
notip	low	2	3.9	9.19	12.4	52.6	9
notip	low	2	3.3	8.92	11.6	55.2	6
notip	low	2	5.5	8.44	13.5	77.6	9
notip	low	2	4.4	10.60	16.2	63.3	6

- This type of data is known as **rectangular** or **tidy** data.
- Each column must have the same number of observations.
- Missing data must have NAs in their position.
- Spreadsheets are often NOT tidy.

Constructing data frames

```
1 p.height <- c(180, 155, 160, 167, 181)
2 p.weight <- c(65, 50, 52, 58, 70)
3 p.names <- c("Joanna", "Charlotte", "Helen", "Karen", "Amy")
4
5 dataf <- data.frame(height = p.height,
6                     weight = p.weight,
7                     names = p.names)
8 dataf
9 ##      height weight      names
10 ## 1      180      65      Joanna
11 ## 2      155      50  Charlotte
12 ## 3      160      52        Helen
13 ## 4      167      58        Karen
14 ## 5      181      70         Amy
```

- Column names are taken from the constructor.
- Can be changed with `names`.
- Numbers at left are row names automatically produced by R, not another column.
- If the vectors are not the same length R will (quietly) cycle!

Structure of data frames

```
1 dim(dataf)    # 5 rows and 3 columns
2 ## [1] 5 3
3
4 str(dataf)
5 ## 'data.frame':    5 obs. of  3 variables:
6 ## $ height: num  180 155 160 167 181
7 ## $ weight: num   65  50  52  58  70
8 ## $ names : chr  "Joanna" "Charlotte" "Helen" "Karen" ...
```

- Gives type, dimensions, column names, types, a few values.
- Convenient to place this in a comment block in your code when dealing with a data frame, for reference.
- R has automatically made names a character vector, not a factor.

Automatically convert strings to factors

```
1 p.height <- c(180, 155, 160, 167, 181)
2 p.weight <- c(65, 50, 52, 58, 70)
3 p.names <- c("Joanna", "Charlotte", "Helen", "Karen", "Amy")
4
5 dataf <- data.frame(height = p.height,
6                     weight = p.weight,
7                     names = p.names,
8                     stringsAsFactors = TRUE)
9 str(dataf)
10 ## 'data.frame':    5 obs. of  3 variables:
11 ## $ height: num  180 155 160 167 181
12 ## $ weight: num   65  50  52  58  70
13 ## $ names : Factor w/ 5 levels "Amy","Charlotte",...: 4 2 3
    5 1
```

Preparing data for import

- Easiest way to enter data is use Excel or LibreOffice Calc.
- Save data in tab-delimited or comma-delimited file.
- Keep column headings short.
- No spaces in column headings.
- Avoid special characters, e.g., mm^2
- No empty cells! Use NAs.
- Make sure it's tidy!
- **Beware!** <https://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-1044-7>

	A	B	C	D	E	F	G	H	I	J
1	test	nitrogen	block	height	weight	volume	shootarea	flowers		
2	np	medium	1	7.5	7.62	11.7	31.6	1		
3	np	medium	1	10.7	12.14	14.1	40	10		
4	np	medium	1	11.2	12.70	7.1	68.7	10		
5	np	medium	1	10.4	8.70	11.9	20.5	1		
6	np	medium	1	10.4	13.00	14.5	28.9	4		
7	np	medium	1	9.8	10.00	12.2	72.7	9		
8	np	medium	1	8.9	10.11	13.2	43.1	7		
9	np	medium	1	9.4	10.20	14	28.5	6		
10	np	medium	2	10.4	10.40	10.5	57.8	5		
11	np	medium	2	12.5	13.40	10.1	38.5	8		
12	np	medium	2	10.4	13.10	11.1	68.8	12		
13	np	medium	2	11	11.50	12.6	31.5	6		
14	np	medium	2	7.1	8.10	29.6	9.7	2		
15	np	medium	2	8	11.20	13	18.4	3		
16	np	medium	2	9	10.2	10.8	60.1	6		
17	np	medium	2	4.5	12.50	13.4	14.4	6		
18	np	high	1	12.6	18.00	18.6	54	9		
19	np	high	1	10	18.07	18.9	60.5	3		
20	np	high	1	10	13.20	15.8	142.7	12		
21	np	high	1	8.5	14.33	13.2	91.4	5		
22	np	high	1	14.1	19.12	13.1	113.2	13		
23	np	high	1	10.1	15.40	12.6	77.2	12		
24	np	high	1	8.5	17.82	20.5	54.4	3		

Importing

```
1 flowers <- read.table(file = 'data/flower.txt',  
2                       header = TRUE, sep = "\t",  
3                       stringsAsFactors = TRUE)
```

- Forward slash works on ALL systems.
- header=TRUE means the first line is variable names.
- sep="\t" for tab-delimited files
- sep=", " for comma-delimited files

Importing

```
1 > str(flowers)
2 'data.frame':   96 obs. of  8 variables:
3 $ treat      : Factor w/ 2 levels "notip","tip": 2 2 2 2 2 2 2 2 2 2 ...
4 $ nitrogen   : Factor w/ 3 levels "high","low","medium": 3 3 3 3 3 3 3 3 3 3 ...
5 $ block      : int   1 1 1 1 1 1 1 1 2 2 ...
6 $ height     : num   7.5 10.7 11.2 10.4 10.4 9.8 6.9 9.4 10.4 12.3 ...
7 $ weight     : num   7.62 12.14 12.76 8.78 13.58 ...
8 $ leafarea   : num   11.7 14.1 7.1 11.9 14.5 12.2 13.2 14 10.5 16.1 ...
9 $ shootarea  : num   31.9 46 66.7 20.3 26.9 72.7 43.1 28.5 57.8 36.9 ...
10 $ flowers    : int    1 10 10 1 4 9 7 6 5 8 .
```

- After importing check structure of data frame.
- treat and nitrogen have been converted to factors.
- Usually very helpful to include this as a comment block in your script.

Specialized import functions

```
1 # import .csv file with sep = "\t" and header = FALSE
2 flowers <- read.table(file = 'data/flower.txt')
3
4 # import .csv file with sep = ","
5 # and header = TRUE
6 flowers <- read.csv(file = 'data/flower.csv')
7
8 # import .csv file with dec = "," and sep = ";"
9 # and header = TRUE
10 flowers <- read.csv2(file = 'data/flower.csv')
11
12 # import tab delim file with sep = "\t" and header = TRUE
13 flowers <- read.delim(file = 'data/flower.txt')
```

- Avoid importing from spreadsheet files (.xls etc.).

Import problems

```
1 Error in file(file, "rt") : cannot open the connection
2 In addition: Warning message:
3 In file(file, "rt") :
4   cannot open file 'flower.txt': No such file or directory
```

- Spelling mistakes.
- Wrong working directory.
- Forgot the extension (`.txt`, `.csv`, *etc.*).

Forgot header = TRUE

```
1 flowers_bad <- read.table(file = 'data/flower.txt',
2                             sep = "\t")
3 str(flowers_bad)
4 ## 'data.frame':    97 obs. of  8 variables:
5 ## $ V1: chr  "treat" "tip" "tip" "tip" ...
6 ## $ V2: chr  "nitrogen" "medium" "medium" "medium" ...
7 ## $ V3: chr  "block" "1" "1" "1" ...
8 ## $ V4: chr  "height" "7.5" "10.7" "11.2" ...
9 ## $ V5: chr  "weight" "7.62" "12.14" "12.76" ...
10 ## $ V6: chr  "leafarea" "11.7" "14.1" "7.1" ...
11 ## $ V7: chr  "shootarea" "31.9" "46" "66.7" ...
12 ## $ V8: chr  "flowers" "1" "10" "10" ...
```

- All of our variables are character (or factor).
- The first value of each variable is the column name.
- R has provided default names, V1, V2, V3, ...

Alternative loader functions

```
1 library(readr)
2 # import white space delimited files
3 all_data <- read_table(file = 'data/flower.txt',
4                        col_names = TRUE)
5
6 # import comma delimited files
7 all_data <- read_csv(file = 'data/flower.txt')
8
9 # import tab delimited files
10 all_data <- read_delim(file = 'data/flower.txt',
11                       delim = "\t")
12
13 # or use
14 all_data <- read_tsv(file = 'data/flower.txt')
```

- readr is from the **tidyverse** collection of packages.
- Many of the arguments are the same as read.table
- Returns a tibble, which is very similar to a data frame.

Packages for large datasets

- `read.table`
- `ff`
- `bigmemory`

Wrangling data frames

```
1 flowers <- read.table(file = 'data/flower.txt', header = TRUE, sep = "\t")
2 str(flowers)
3 ## 'data.frame':    96 obs. of  8 variables:
4 ## $ treat      : chr  "tip" "tip" "tip" "tip" ...
5 ## $ nitrogen   : chr  "medium" "medium" "medium" "medium" ...
6 ## $ block      : int   1 1 1 1 1 1 1 2 2 ...
7 ## $ height     : num   7.5 10.7 11.2 10.4 10.4 9.8 6.9 9.4 10.4 12.3 ...
8 ## $ weight     : num   7.62 12.14 12.76 8.78 13.58 ...
9 ## $ leafarea   : num   11.7 14.1 7.1 11.9 14.5 12.2 13.2 14 10.5 16.1 ...
10 ## $ shootarea : num   31.9 46 66.7 20.3 26.9 72.7 43.1 28.5 57.8 36.9 ...
11 ## $ flowers    : int    1 10 10 1 4 9 7 6 5 8 ...
```

The \$ notation

```
1 flowers$height
2 [1] 7.5 10.7 11.2 10.4 10.4 9.8 6.9 9.4 10.4 12.3 10.4
3 [12] 11.0 7.1 6.0 9.0 4.5 12.6 10.0 10.0 8.5 14.1 10.1
4 [23] 8.5 6.5 11.5 7.7 6.4 8.8 9.2 6.2 6.3 17.2 8.0
5 [34] 8.0 6.4 7.6 9.7 12.3 9.1 8.9 7.4 3.1 7.9 8.8
6 [45] 8.5 5.6 11.5 5.8 5.6 5.3 7.5 4.1 3.5 8.5 4.9
7 [56] 2.5 5.4 3.9 5.8 4.5 8.0 1.8 2.2 3.9 8.5 8.5
8 [67] 6.4 1.2 2.6 10.9 7.2 2.1 4.7 5.0 6.5 2.6 6.0
9 [78] 9.3 4.6 5.2 3.9 2.3 5.2 2.2 4.5 1.8 3.0 3.7
10 [89] 2.4 5.7 3.7 3.2 3.9 3.3 5.5 4.4
11
12 f_height <- flowers$height
13 mean(f_height)
14 ## [1] 6.839583
15 summary(f_height)
16 ##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
17 ##      1.200   4.475   6.450   6.840   9.025  17.200
18
19 mean(flowers$height)
20 ## [1] 6.839583
21 summary(flowers$height)
22 ##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
23 ##      1.200   4.475   6.450   6.840   9.025  17.200
```


Positional indexes

```
1 flowers[1, 4]
2 ## [1] 7.5
3
4 # this would give you the same
5 flowers$height[1]
6 ## [1] 7.5
```

Positional indexes

```
1 flowers[1:10, 1:4]
2 ##      treat nitrogen block height
3 ## 1      tip    medium      1    7.5
4 ## 2      tip    medium      1   10.7
5 ## 3      tip    medium      1   11.2
6 ## 4      tip    medium      1   10.4
7 ## 5      tip    medium      1   10.4
8 ## 6      tip    medium      1    9.8
9 ## 7      tip    medium      1    6.9
10 ## 8      tip    medium      1    9.4
11 ## 9      tip    medium      2   10.4
12 ## 10     tip    medium      2   12.3
```

Positional indexes

```
1 flowers[c(1, 5, 12, 30), c(1, 3, 6, 8)]
2 ##      treat block leafarea flowers
3 ## 1      tip      1      11.7      1
4 ## 5      tip      1      14.5      4
5 ## 12     tip      2      12.6      6
6 ## 30     tip      2      11.6      5
```

Empty index means "all of them"

```
1 flowers[1:8, ]
2 ##      treat  nitrogen  block  height  weight  leafarea  shootarea  flowers
3 ## 1    tip    medium    1      7.5    7.62    11.7      31.9        1
4 ## 2    tip    medium    1     10.7   12.14    14.1      46.0       10
5 ## 3    tip    medium    1     11.2   12.76     7.1      66.7       10
6 ## 4    tip    medium    1     10.4    8.78    11.9      20.3        1
7 ## 5    tip    medium    1     10.4   13.58    14.5      26.9        4
8 ## 6    tip    medium    1      9.8   10.08    12.2      72.7        9
9 ## 7    tip    medium    1      6.9   10.11    13.2      43.1        7
10 ## 8    tip    medium    1      9.4   10.28    14.0      28.5        6
```

Negative numbers mean "not these"

```
1 flowers[-(1:85), -c(4, 7, 8)]
2 ##      treat  nitrogen  block  weight  leafarea
3 ## 86 notip      low      1    6.01    17.6
4 ## 87 notip      low      1    9.93    12.0
5 ## 88 notip      low      1    7.03     7.9
6 ## 89 notip      low      2    9.10    14.5
7 ## 90 notip      low      2    9.05     9.6
8 ## 91 notip      low      2    8.10    10.5
9 ## 92 notip      low      2    7.45    14.1
10 ## 93 notip      low      2    9.19    12.4
11 ## 94 notip      low      2    8.92    11.6
12 ## 95 notip      low      2    8.44    13.5
13 ## 96 notip      low      2   10.60    16.2
```

Can also use column names

```
1 flowers[1:5, c("treat", "nitrogen", "leafarea")]
2 ##      treat nitrogen leafarea
3 ## 1    tip    medium    11.7
4 ## 2    tip    medium    14.1
5 ## 3    tip    medium     7.1
6 ## 4    tip    medium    11.9
7 ## 5    tip    medium    14.5
```

- More readable and portable than
`flowers[1:5, c(1, 2, 6)]`

Logical indexes

```
1 big_flowers <- flowers[flowers$height > 12, ]
2 big_flowers
3 ##      treat nitrogen block height weight leafarea shootarea flowers
4 ## 10     tip    medium     2   12.3   13.48     16.1     36.9         8
5 ## 17     tip      high     1   12.6   18.66     18.6     54.0         9
6 ## 21     tip      high     1   14.1   19.12     13.1    113.2        13
7 ## 32     tip      high     2   17.2   19.20     10.9     89.9        14
8 ## 38     tip      low      1   12.3   11.27     13.7     28.7         5
```

Logical indexes

```
1 nit_high <- flowers[flowers$nitrogen == "high", ]
2 nit_high
3 ##      treat  nitrogen  block  height  weight  leafarea  shootarea  flowers
4 ## 17     tip      high      1    12.6   18.66     18.6      54.0         9
5 ## 18     tip      high      1    10.0   18.07     16.9      90.5         3
6 ## 19     tip      high      1    10.0   13.29     15.8     142.7        12
7 ## 20     tip      high      1     8.5   14.33     13.2      91.4         5
8 ## 21     tip      high      1    14.1   19.12     13.1     113.2        13
9 ## 22     tip      high      1    10.1   15.49     12.6      77.2        12
10 ....
```


Logical indexes

```
1 low_notip_high6 <- flowers[flowers$height >= 6 &
2                             flowers$nitrogen == "medium" &
3                             flowers$treat == "notip", ]
4 low_notip_high6
5 ##      treat  nitrogen  block  height  weight  leafarea  shootarea  flowers
6 ##  51 notip    medium     1     7.5   13.60     13.6    122.2      11
7 ##  54 notip    medium     1     8.5   10.04     12.3    113.6       4
8 ##  61 notip    medium     2     8.0   11.43     12.6     43.2      14
```

Logical indexes using subset

```
1 tip_med_2 <- subset(flowers, treat == "tip" &
2                       nitrogen == "medium" &
3                       block == 2)
4 tip_med_2
5 ##      treat nitrogen block height weight leafarea shootarea flowers
6 ## 9      tip   medium    2   10.4  10.48    10.5     57.8        5
7 ## 10     tip   medium    2   12.3  13.48    16.1     36.9        8
8 ## 11     tip   medium    2   10.4  13.18    11.1     56.8       12
9 ## 12     tip   medium    2   11.0  11.56    12.6     31.3        6
10 ## 13     tip   medium    2    7.1   8.16    29.6      9.7        2
11 ## 14     tip   medium    2    6.0  11.22    13.0     16.4        3
12 ## 15     tip   medium    2    9.0  10.20    10.8     90.1        6
13 ## 16     tip   medium    2    4.5  12.55    13.4     14.4        6
```

- We don't need to use `flowers$treat`

Logical indexes using subset and select

```
1 tipplants <- subset(flowers ,
2                       treat == "tip" &
3                       nitrogen == "medium" &
4                       block == 2,
5                       select = c("treat", "nitrogen", "leafarea"))
6 tipplants
7 ##      treat  nitrogen  leafarea
8 ##  9      tip    medium     10.5
9 ## 10      tip    medium     16.1
10 ## 11      tip    medium     11.1
11 ## 12      tip    medium     12.6
12 ## 13      tip    medium     29.6
13 ## 14      tip    medium     13.0
14 ## 15      tip    medium     10.8
15 ## 16      tip    medium     13.4
```

- Selecting both rows and columns.

Ordering data frames

```
1 height_ord <- flowers[order(flowers$height), ]
2 height_ord
3 ##      treat nitrogen block height weight leafarea shootarea flowers
4 ## 68 notip      high     1   1.2  18.24    16.6    148.1      7
5 ## 62 notip    medium     2   1.8  10.47    11.8    120.8      9
6 ## 86 notip      low     1   1.8   6.01    17.6     46.2      4
7 ## 72 notip      high     1   2.1  19.15    15.6    176.7      6
8 ## 63 notip    medium     2   2.2  10.70    15.3     97.1      7
9 ## 84 notip      low     1   2.2   9.97     9.6     63.1      2
10 ## 82 notip      low     1   2.3   7.28    13.8     32.8      6
11 ## 89 notip      low     2   2.4   9.10    14.5     78.7      8
12 ## 56 notip    medium     1   2.5  14.85    17.5     77.8     10
13 ## 69 notip      high     1   2.6  16.57    17.1    141.1      3
14 ## 76 notip      high     2   2.6  18.88    16.4    181.5     14
15 ## 87 notip      low     1   3.0   9.93    12.0     56.6      6
16 ## 42  tip      low     2   3.1   8.74    16.1     39.1      3
17 ...
```

Ordering data frames

```
1 leafarea_ord <- flowers[order(flowers$leafarea , decreasing = TRUE), ]
2 leafarea_ord
3 ##      treat nitrogen block height weight leafarea shootarea flowers
4 ## 70 notip      high     1  10.9  17.22    49.2    189.6      17
5 ## 13  tip      medium    2   7.1   8.16    29.6     9.7       2
6 ## 24  tip      high     1   6.5  17.13    24.1    147.4       6
7 ## 65 notip      high     1   8.5  22.53    20.8    166.9      16
8 ## 23  tip      high     1   8.5  17.82    20.5     54.4       3
9 ## 66 notip      high     1   8.5  17.33    19.8    184.4      12
10 ## 73 notip      high     2   4.7  13.42    19.8    124.7       5
11 ## 80 notip      high     2   5.2  17.70    19.1    181.1       8
12 ## 17  tip      high     1  12.6  18.66    18.6     54.0       9
13 ## 49 notip      medium    1   5.6  11.03    18.6     49.9       8
14 ## 78 notip      high     2   9.3  18.75    18.4    181.1      16
15 ...
```

Ordering data frames

```
1 block_height_ord <- flowers[order(flowers$block, flowers$height), ]
2 block_height_ord
3 ##      treat  nitrogen  block  height  weight  leafarea  shootarea  flowers
4 ## 68 notip      high      1      1.2  18.24      16.6      148.1        7
5 ## 86 notip      low       1      1.8   6.01      17.6       46.2        4
6 ## 72 notip      high      1      2.1  19.15      15.6      176.7        6
7 ## 84 notip      low       1      2.2   9.97       9.6       63.1        2
8 ## 82 notip      low       1      2.3   7.28      13.8       32.8        6
9 ## 56 notip     medium      1      2.5  14.85      17.5       77.8       10
10 ## 69 notip      high      1      2.6  16.57      17.1      141.1        3
11 ....
12 ## 38  tip      low       1     12.3  11.27      13.7       28.7        5
13 ## 17  tip      high      1     12.6  18.66      18.6       54.0        9
14 ## 21  tip      high      1     14.1  19.12      13.1      113.2       13
15 ## 62 notip     medium      2      1.8  10.47      11.8      120.8        9
16 ## 63 notip     medium      2      2.2  10.70      15.3       97.1        7
17 ## 89 notip      low       2      2.4   9.10      14.5       78.7        8
18 ## 76 notip      high      2      2.6  18.88      16.4      181.5       14
19 ## 42  tip      low       2      3.1   8.74      16.1       39.1        3
20 ## 92 notip      low       2      3.2   7.45      14.1       38.1        4
21 ....
```

Ordering data frames

```
1 block_revheight_ord <- flowers[order(flowers$block, -flowers$height), ]
2 block_revheight_ord
3 ##      treat  nitrogen  block  height  weight  leafarea  shootarea  flowers
4 ## 21    tip      high     1    14.1  19.12     13.1    113.2     13
5 ## 17    tip      high     1    12.6  18.66     18.6     54.0     9
6 ## 38    tip      low      1    12.3  11.27     13.7     28.7     5
7 ## 3     tip    medium     1    11.2  12.76      7.1     66.7    10
8 ## 70 notip      high     1    10.9  17.22     49.2    189.6    17
9 ## 2     tip    medium     1    10.7  12.14     14.1     46.0    10
10 ## 4     tip    medium     1    10.4   8.78     11.9     20.3     1
11 ## 5     tip    medium     1    10.4  13.58     14.5     26.9     4
12 ## 22    tip      high     1    10.1  15.49     12.6     77.2    12
13 ....
14 ## 72 notip      high     1     2.1  19.15     15.6    176.7     6
15 ## 86 notip      low      1     1.8   6.01     17.6     46.2     4
16 ## 68 notip      high     1     1.2  18.24     16.6    148.1     7
17 ## 32    tip      high     2    17.2  19.20     10.9     89.9    14
18 ## 10    tip    medium     2    12.3  13.48     16.1     36.9     8
19 ## 25    tip      high     2    11.5  23.89     14.3    101.5    12
20 ## 47    tip      low      2    11.5   8.72     10.2     28.3     6
21 ## 12    tip    medium     2    11.0  11.56     12.6     31.3     6
22 .....
```

Ordering data frames

```
1 revheight_ord <- flowers[order(-xtfrm(flowers$nitrogen), flowers$height), ]
2 revheight_ord
3 ##      treat  nitrogen  block  height  weight  leafarea  shootarea  flowers
4 ## 62 notip    medium     2     1.8   10.47    11.8     120.8        9
5 ## 63 notip    medium     2     2.2   10.70    15.3     97.1        7
6 ## 56 notip    medium     1     2.5   14.85    17.5     77.8       10
7 ## 53 notip    medium     1     3.5   12.93    16.6    109.3        3
8 ## 58 notip    medium     2     3.9    9.07     9.6     90.4        7
9 ## 64 notip    medium     2     3.9   12.97    17.0     97.5        5
10 ## 52 notip    medium     1     4.1   12.58    13.9    136.6       11
11 ....
12 ## 2      tip    medium     1    10.7   12.14    14.1     46.0       10
13 ## 12     tip    medium     2    11.0   11.56    12.6     31.3        6
14 ## 3      tip    medium     1    11.2   12.76     7.1     66.7       10
15 ## 10     tip    medium     2    12.3   13.48    16.1     36.9        8
16 ## 86 notip     low       1     1.8    6.01    17.6     46.2        4
17 ## 84 notip     low       1     2.2    9.97     9.6     63.1        2
18 ## 82 notip     low       1     2.3    7.28    13.8     32.8        6
19 ## 89 notip     low       2     2.4    9.10    14.5     78.7        8
20 ....
21 ## 39     tip     low       1     9.1    8.96     9.7     23.8        3
22 ## 37     tip     low       1     9.7    6.49     8.1     18.0        3
23 ## 47     tip     low       2    11.5    8.72    10.2     28.3        6
24 ## 38     tip     low       1    12.3   11.27    13.7     28.7        5
25 ## 68 notip     high      1     1.2   18.24    16.6    148.1        7
26 ## 72 notip     high      1     2.1   19.15    15.6    176.7        6
27 ## 69 notip     high      1     2.6   16.57    17.1    141.1        3
28 ....
```


Ordering data frames

```
1 flowers$nitrogen <- factor(flowers$nitrogen ,
2                             levels = c("low", "medium", "high"))
3 nit_ord <- flowers[order(flowers$nitrogen),]
4 nit_ord
5 ##      treat  nitrogen  block  height  weight  leafarea  shootarea  flowers
6 ## 33    tip      low      1      8.0    6.88      9.3      16.1      4
7 ## 34    tip      low      1      8.0   10.23     11.9     88.1      4
8 ## 35    tip      low      1      6.4    5.97      8.7      7.3      2
9 ## 36    tip      low      1      7.6   13.05      7.2     47.2      8
10 ## 37    tip      low      1      9.7    6.49      8.1     18.0      3
11 ## 38    tip      low      1     12.3   11.27     13.7     28.7      5
12 ....
13 ## 93 notip      low      2      3.9    9.19     12.4     52.6      9
14 ## 94 notip      low      2      3.3    8.92     11.6     55.2      6
15 ## 95 notip      low      2      5.5    8.44     13.5     77.6      9
16 ## 96 notip      low      2      4.4   10.60     16.2     63.3      6
17 ## 1     tip     medium     1      7.5    7.62     11.7     31.9      1
18 ## 2     tip     medium     1     10.7   12.14     14.1     46.0     10
19 ## 3     tip     medium     1     11.2   12.76      7.1     66.7     10
20 ....
21 ## 61 notip     medium     2      8.0   11.43     12.6     43.2     14
22 ## 62 notip     medium     2      1.8   10.47     11.8    120.8      9
23 ## 63 notip     medium     2      2.2   10.70     15.3     97.1      7
24 ## 64 notip     medium     2      3.9   12.97     17.0     97.5      5
25 ## 17    tip      high      1     12.6   18.66     18.6     54.0      9
26 ## 18    tip      high      1     10.0   18.07     16.9     90.5      3
27 ## 19    tip      high      1     10.0   13.29     15.8    142.7     12
28 ## 20    tip      high      1      8.5   14.33     13.2     91.4      5
29 .....
```

Adding rows to a data frame

```
1 # rbind for rows
2 df1 <- data.frame(id = 1:4, height = c(120, 150, 132, 122),
3                   weight = c(44, 56, 49, 45))
4 df1
5 ##      id height weight
6 ## 1    1    120     44
7 ## 2    2    150     56
8 ## 3    3    132     49
9 ## 4    4    122     45
10
11 df2 <- data.frame(id = 5:6, height = c(119, 110),
12                  weight = c(39, 35))
13 df2
14 ##      id height weight
15 ## 1    5    119     39
16 ## 2    6    110     35
```

Adding rows to a data frame

```
1 df_rcomb <- rbind(df1 , df2)
2 df_rcomb
3 ##      id height weight
4 ## 1    1    120     44
5 ## 2    2    150     56
6 ## 3    3    132     49
7 ## 4    4    122     45
8 ## 5    5    119     39
9 ## 6    6    110     35
```

Adding columns to a data frame

```
1 df3 <- data.frame(id = 1:4, height = c(120, 150, 132, 122),
2                   weight = c(44, 56, 49, 45))
3 df3
4 ##      id height weight
5 ## 1    1    120     44
6 ## 2    2    150     56
7 ## 3    3    132     49
8 ## 4    4    122     45
9
10 df4 <- data.frame(location = c("UK", "CZ", "CZ", "UK"))
11 df4
12 ##      location
13 ## 1          UK
14 ## 2          CZ
15 ## 3          CZ
16 ## 4          UK
```

Adding columns to a data frame

```
1 df_ccomb <- cbind(df3 , df4)
2 df_ccomb
3 ##      id height weight location
4 ## 1    1    120     44         UK
5 ## 2    2    150     56         CZ
6 ## 3    3    132     49         CZ
7 ## 4    4    122     45         UK
```

Adding computed columns to a data frame

```
1 df_rcomb$height_log10 <- log10(df_rcomb$height)
2 df_rcomb
3 ##      id height weight height_log10
4 ## 1    1   120    44    2.079181
5 ## 2    2   150    56    2.176091
6 ## 3    3   132    49    2.120574
7 ## 4    4   122    45    2.086360
8 ## 5    5   119    39    2.075547
9 ## 6    6   110    35    2.041393
```

Converting type of a column

```
1 # convert to a factor
2 df_rcomb$Fid <- factor(df_rcomb$id)
3 df_rcomb
4 ##   id height weight height_log10 Fid
5 ## 1  1    120    44    2.079181   1
6 ## 2  2    150    56    2.176091   2
7 ## 3  3    132    49    2.120574   3
8 ## 4  4    122    45    2.086360   4
9 ## 5  5    119    39    2.075547   5
10 ## 6  6    110    35    2.041393   6
11 str(df_rcomb)
12 ## 'data.frame':    6 obs. of  5 variables:
13 ##  $ id          : int  1 2 3 4 5 6
14 ##  $ height      : num  120 150 132 122 119 110
15 ##  $ weight      : num  44 56 49 45 39 35
16 ##  $ height_log10: num  2.08 2.18 2.12 2.09 2.08 ...
17 ##  $ Fid         : Factor w/ 6 levels "1","2","3","4",...: 1 2 3 4 5 6
```

Merging data frames

- Suppose we have one data frame with information about some common rocky shore invertebrates, called `taxa`
- And we have another data frame with information about where these invertebrates are usually found, called `zone`
- Can we combine these into one data frame with all information about the invertebrates?

Merging data frames

```
1 taxa <- data.frame(  
2   GENUS = c("Patella", "Littorina", "Halichondria", "Semibalanus"),  
3   species = c("vulgata", "littoria", "panacea", "balanoides"),  
4   family = c("patellidae", "Littorinidae", "Halichondriidae", "  
   Archaeobalanidae"))  
5  
6 taxa  
7 ##      GENUS      species      family  
8 ## 1    Patella    vulgata    patellidae  
9 ## 2    Littorina  littoria    Littorinidae  
10 ## 3 Halichondria  panacea    Halichondriidae  
11 ## 4 Semibalanus  balanoides  Archaeobalanidae  
12  
13 zone <- data.frame(  
14   genus = c("Laminaria", "Halichondria", "Xanthoria", "Littorina",  
15   "Semibalanus", "Fucus"),  
16   species = c("digitata", "panacea", "parietina", "littoria",  
17   "balanoides", "serratus"),  
18   zone = c("v_low", "low", "v_high", "low_mid", "high", "low_mid"))  
19  
20 zone  
21 ##      genus      species      zone  
22 ## 1  Laminaria  digitata  v_low  
23 ## 2 Halichondria  panacea    low  
24 ## 3  Xanthoria  parietina  v_high  
25 ## 4  Littorina  littoria  low_mid  
26 ## 5 Semibalanus  balanoides    high  
27 ## 6      Fucus    serratus  low_mid
```

Merging data frames

```
1 taxa_zone <- merge(x = taxa, y = zone)
2 taxa_zone
3 ##      species      GENUS      family      genus      zone
4 ## 1 balanoides Semibalanus Archaeobalanidae Semibalanus high
5 ## 2 littoria   Littorina   Littorinidae   Littorina low_mid
6 ## 3 panacea   Halichondria Halichondriidae Halichondria low
```

- Because the two data frames have a column name in common, R assumes you want to join on that column.
- The joined data frame has both GENUS and genus because they are spelled differently.
- The joined data frame has only those rows with information in BOTH original data frames.

Merging data frames

```
1 taxa_zone <- merge(x = taxa, y = zone, all = TRUE)
2 taxa_zone
3 ##      species      GENUS      family      genus      zone
4 ## 1 balanoides Semibalanus Archaeobalanidae Semibalanus high
5 ## 2 digitata    <NA>      <NA>      Laminaria v_low
6 ## 3 littoria    Littorina Littorinidae Littorina low_mid
7 ## 4 panacea    Halichondria Halichondriidae Halichondria low
8 ## 5 parietina    <NA>      <NA>      Xanthoria v_high
9 ## 6 serratus    <NA>      <NA>      Fucus low_mid
10 ## 7 vulgata     Patella    patellidae    <NA>    <NA>
```

- To include ALL data from BOTH frames use `all = TRUE`
- NAs will be substituted for missing data.

Merging data frames

- Use `by.x` and `by.y` if the names are different

```
1 taxa_zone <- merge(x = taxa, y = zone,
2                     by.x = "GENUS",
3                     by.y = "genus",
4                     all = TRUE)
5 taxa_zone
6 ##      GENUS  species.x      family  species.y  zone
7 ## 1      Fucus    <NA>    <NA>    serratus low_mid
8 ## 2 Halichondria panacea Halichondriidae panacea low
9 ## 3      Laminaria    <NA>    <NA>    digitata v_low
10 ## 4      Littorina littoria Littorinidae littoria low_mid
11 ## 5      Patella  vulgata  patellidae    <NA>    <NA>
12 ## 6 Semibalanus balanoides Archaeobalanidae balanoides high
13 ## 7      Xanthoria    <NA>    <NA>    parietina v_high
```

Merging data frames

- Can use multiple columns

```
1 taxa_zone <- merge(x = taxa, y = zone,
2                     by.x = c("species", "GENUS"),
3                     by.y = c("species", "genus"),
4                     all = TRUE)
5 taxa_zone
6 ##      species      GENUS      family      zone
7 ## 1 balanoides Semibalanus Archaeobalanidae    high
8 ## 2 digitata   Laminaria      <NA>      v_low
9 ## 3 littoria   Littorina    Littorinidae low_mid
10 ## 4 panacea   Halichondria Halichondriidae    low
11 ## 5 parietina Xanthoria      <NA>      v_high
12 ## 6 serratus   Fucus        <NA>      low_mid
13 ## 7 vulgata    Patella      patellidae    <NA>
```