

Demo 2

R for statistical analysis

Juulia T. Suvilehto D.Sc.(tech)

Data structures

Data types and data structures

- In the previous demo we encountered the following data types
 - character
 - double
 - integer
 - logical
- In the previous demo we encountered the following data structure
 - vector
- Can you remember what they all are and how they're different from one another?

More data structures

Vector

1
2
3

- 1 dimension
- One data type throughout
- Created with `c()`
- Indexing e.g. `[1]`

Matrix

1	2	3
2	1	4
3	4	1

- 2 dimensional extension of vector
- Created with `matrix()`
- Indexing e.g. `[1,1]` or `[1,]`

Array

- 3 or more dimensions
- Created with `array()`
- Indexing e.g. `[1,1,1]` or `[1,,1]`

Data frame

x	y	z
1	a	male
2	b	female
3	c	female

- 2 dimensions
- Each column can have only one data type (+ header)
- Created with `data.frame()` or when importing data
- Indexing commonly with `$`, but can also use `[1,1]`

List

1	2	3
2	1	4
3	4	1

"a long
string of
text"

$3+2i$

- 1 dimension
- Can combine data types & structures
- Created with `list()`
- Indexing of list members with `[[]]`

Factors

- A special data type for categorical variables (like gender) which can only get a limited number of values ('levels')
- Sometimes automatically created by R when importing data (you can control this)
- Helps manage categorical variables and prevents accidentally running nonsensical analyses on them
- Warning! While converting to factor is easy, converting away from factors does not always work as you would expect. If you convert from factor to another data type, remember to double check that the values are as you would expect!

NA ('not available')

- In data analysis it is important to differentiate between a missing value and a data with known value 0
- R has a special symbol, NA, to indicate data which are not available
- NA can be present in any data structure regardless of the data type of the other values
- Different analyses handle NAs differently
 - you may need to explicitly state `na.rm=True` (i.e. remove NA values) if you want to run your analysis excluding NAs
 - you may need to run some analyses on a subset data with no missing values

Working with data (finally!)

General plan for data analysis with R

- Load the data
- Inspect the data
 - Are the missing values coded appropriately?
 - Are there any outliers that are physiologically impossible (e.g. height >3m, age < 0 years)
 - Are categorical variables coded as factors and continuous variables coded as numeric etc.?
- Are the data organized in a tidy manner
- Modify the data as necessary
- Run analyses/build plots
- Save the outcome

General plan for data analysis with R

- Load the data
- Inspect the data
 - Are the missing values coded appropriately?
 - Are there any outliers that are physiologically impossible (e.g. height >3m, age < 0 years)
 - Are categorical variables coded as factors and continuous variables coded as numeric etc.?
- Are the data organized in a tidy manner
- Modify the data as necessary
- Run analyses/build plots
- Save the outcome

Well-behaved data demo

Conditional execution of code

If – else

- Sometimes you want to control the flow of the script
 - Depending on a value from an earlier step in the analysis
 - Depending on the type of data you're working on
 - Depending on a parameter you've set
- Arguably the simplest way to do this is the if – else statement
- The syntax is simple (see next slide), but you need to be careful about which order you place your conditions in to make the flow work the way you want it to

If – else

- Generic format is

```
if (condition) {  
    do something  
} else if (another condition) {  
    do something else  
} else {  
    do a third thing  
}
```

If – else

- Generic format is

```
if (condition) {  
    do something  
} else if (another condition) {  
    do something else  
} else {  
    do a third thing  
}
```

This is the only bit that **has** to be included. For example, if your data includes NAs (remember those?), do something to make the data suitable for the following analysis

If – else

- Generic format is

```
if (condition) {  
    do something  
} else if (another condition) {  
    do something else  
} else {  
    do a third thing  
}
```

If you only have two things you're choosing between (e.g. value is 0 or is anything other than 0 and depending on the situation, you do a different calculation), you only need the if and the else

If – else

- Generic format is

```
if (condition) {  
    do something  
}  
else if (another condition) {  
    do something else  
}  
else {  
    do a third thing  
}
```

You can have as many else if statements as you need. They always need to be positioned between the if and the else.