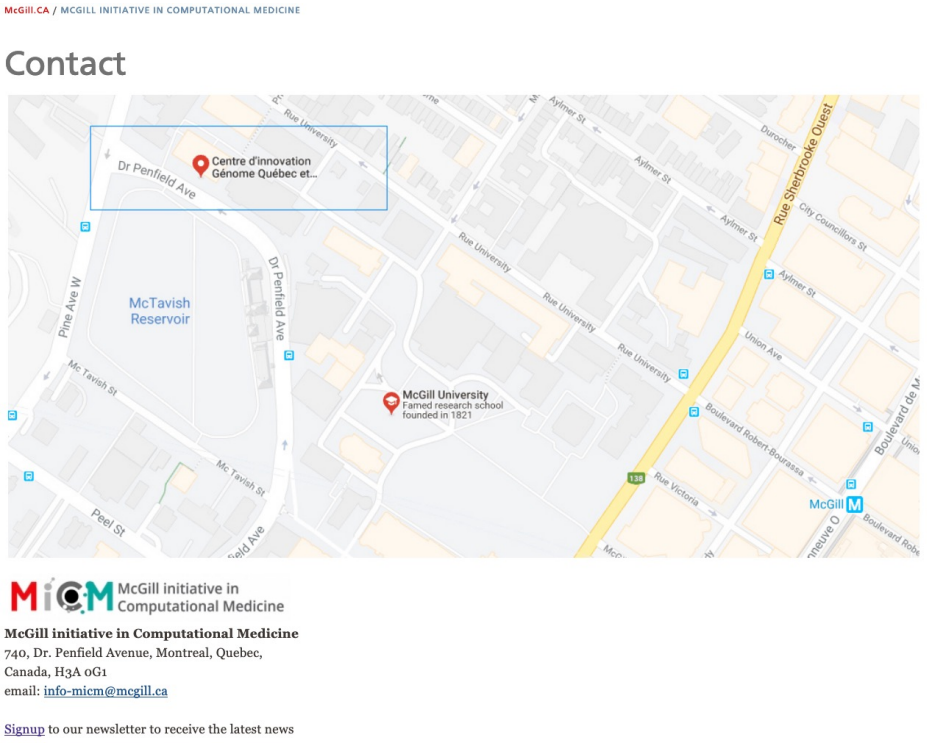
Intro to programming in R

Instructor: Larisa M. Soto
Computational and Statistical Genomics Laboratory
September 14 and 15, 2022

Mission : aims to deliver inter-disciplinary research programs and empower the use of data in health research and health care delivery



https://www.mcgill.ca/micm

Workshop outline – *Day 1*

1 The language
- History
- Foundation
- Syntax
- Logical ops
- Help
- Packages

2 Data types
- Vectors
- Factors
- Lists
- Data Frames
- Arrays
- ***Hands on***

3 Basic data manipulation
- Read
- Write
- Subset
- Split
- Join
- ***Hands on***

Workshop outline – *Day 2*

4 Control Structures
Functions
If statement
for loop
Real-life hands on

5 Advanced data Manipulation
dplyr
tidyr
plyr
DataTable
Hands on

6 Generating Outputs
Graphics
ggplot2
RMarkdown
Templates

7 Software development
Good coding practices
Documentation standards
Debugging

The R programming language

Learning objectives

- Become familiar with the language and the logic behind it
- Create a project in R studio
- Configure the working directory
- Create your first R script
- Get fluent in R using the console
- Compute arithmetic operations
- Use logical operators on variables
- Learn how to ask for help
- Get comfortable installing packages

- GNU project of *free software*
- Users have the freedom to:
  1) Run the program
  2) View and modify the source code
  3) Redistribute copies and
  4) Distribute their modifications

- Integrated suite for data manipulation, analysis, and graphical visualization
- Environment where statistical tests can be performed
- Its functionality can be easily extended with *packages*

https://www.r-project.org/about.html

R facts

- Object-oriented
- No spaces allowed in variable names
- Case sensitive
- 1-based indexing
- Allows user-defined functions
- Works with environments

Arithmetic operators

Addition	+
Subtraction	-
Division	/
Power	^
Scalar multiplication	*
Matrix multiplication	%*%

Syntax operators

Comment line	#
Assignation	<-
Access content	$
Equal	=

Logical operators

Equal	`==`
Not equal	`!=`
Greater than	`>`
Greater than or equal to	`>=`
Less than	`<`
Less than or equal to	`<=`
contains	`%in%`
x AND y	`x & y`
x OR y	`x | y`
NOT x	`!x`

Data types and data structures

Learning objectives

- Understand the differences between classes, objects and data types in R
- Create objects of different types
- Subset and index objects
- Learn and use vectorized operations

Atomic Classes

Also called **data types**

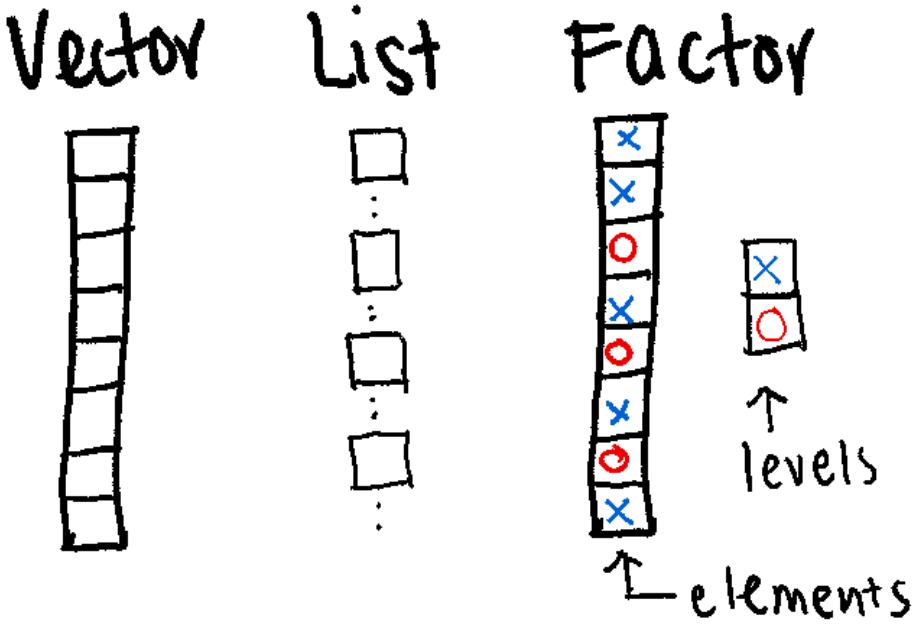
Character	`A,b,c,d,e,..`
Numeric (real numbers)	`1.00,2.00,…` `Inf, NaN`
Integer	`1,2,3,4,….`
Complex	`2i`
Logical (True/False)	`TRUE,FALSE`

Objects

Also called **data structures**

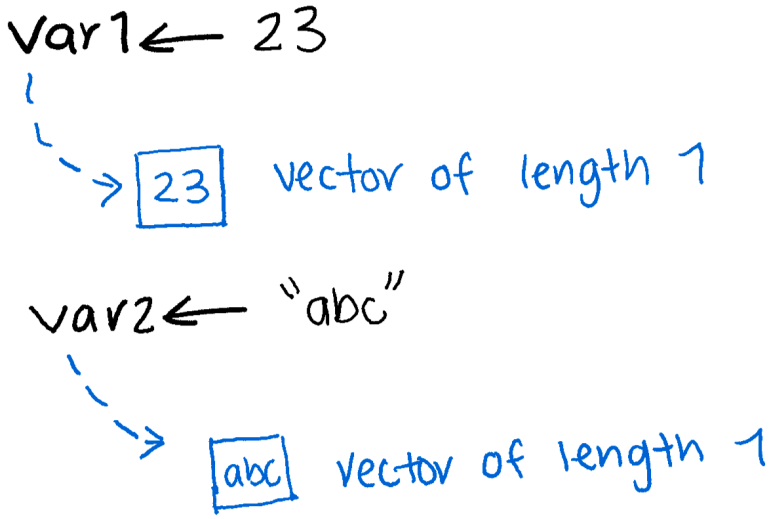
Vector	Only elements of the same class
List	Elements of any class
Factor	Categorical data
Matrix	Elements of the same class in 2D
Data frame	Elements of multiple classes in 2D

One dimension



Vectors

- Can only contain objects of the **same class**
- Most basic type of R object
- Variables are vectors

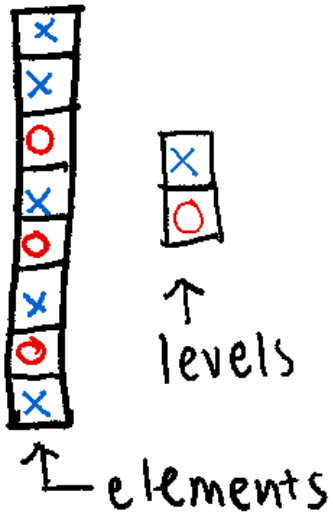


Lists

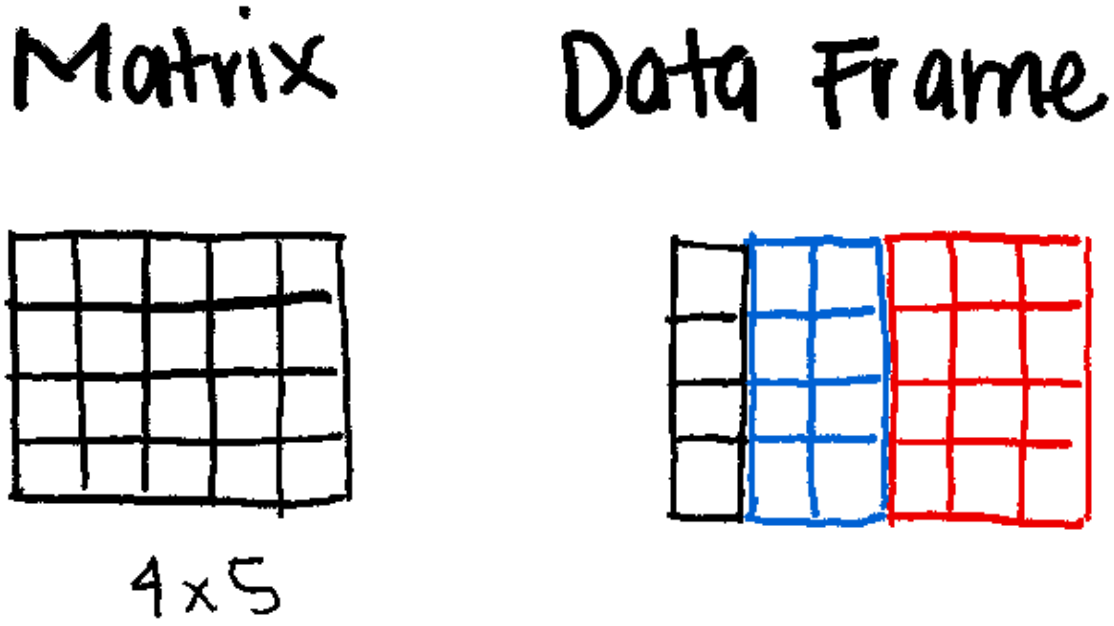
- Can contain objects of multiple classes
- Very important data type in R
- Extremely powerful when combined with some built-in functions

Factors

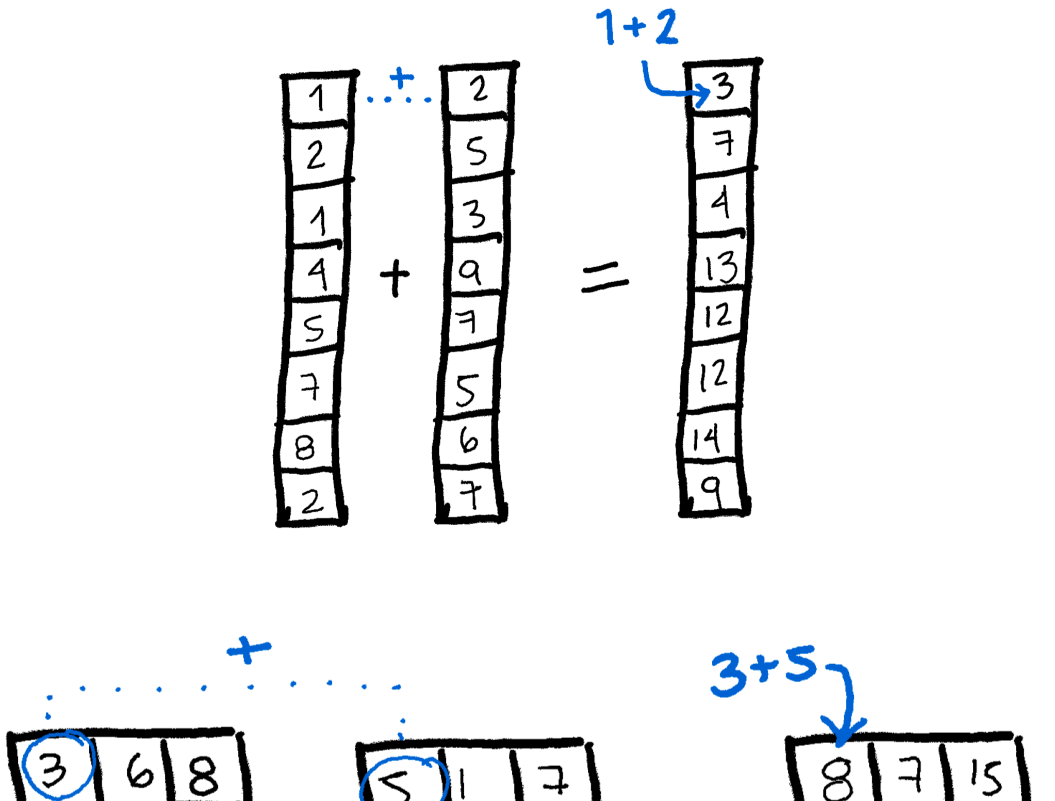
- Useful when for categorical data
- Can have implicit order, if needed
- Each **element** has a label or **level**
- They are important in statistical modelling and plotting with ggplot
- Some operations behave differently on factors

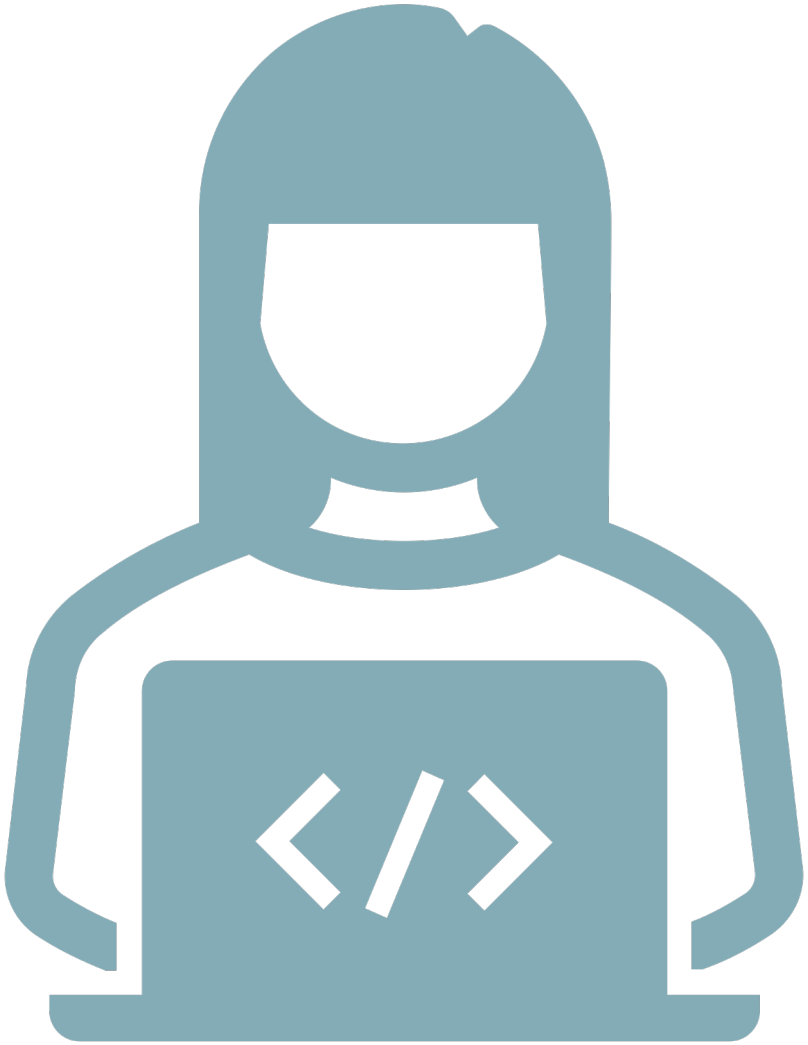


Multiple dimensions



Vectorized operations

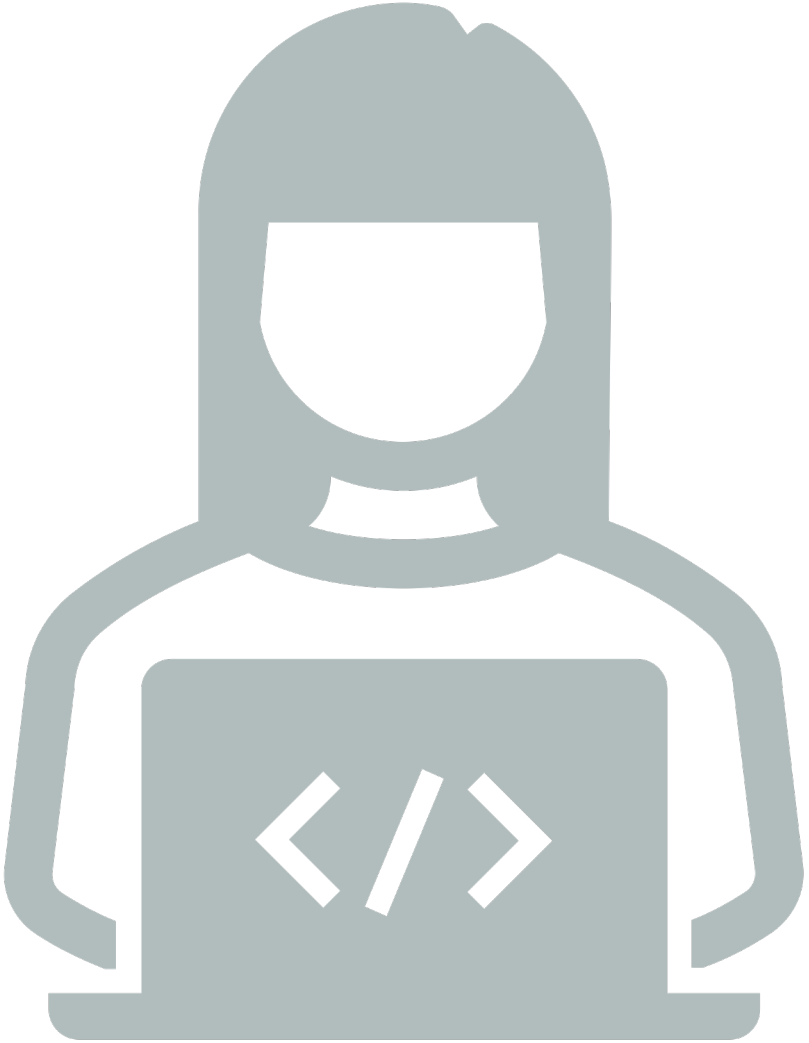




Basic data manipulation

Learning objectives:

- Learn how to read/write data to/from files with different formats (.tsv, .csv)
- Familiarize with basic operations of data frames
- Index and subset data frames using base R functions
- Manipulate specific data frame columns
- Joining by columns and rows



Control structures and functions

Learning objectives:

- Understand the concept of environments in R
- Create new functions
- Implement conditional statements
- Implement a for loop to iterate over a list of files

Conditional statements

- When we want a set of actions to be executed only if certain conditions are met

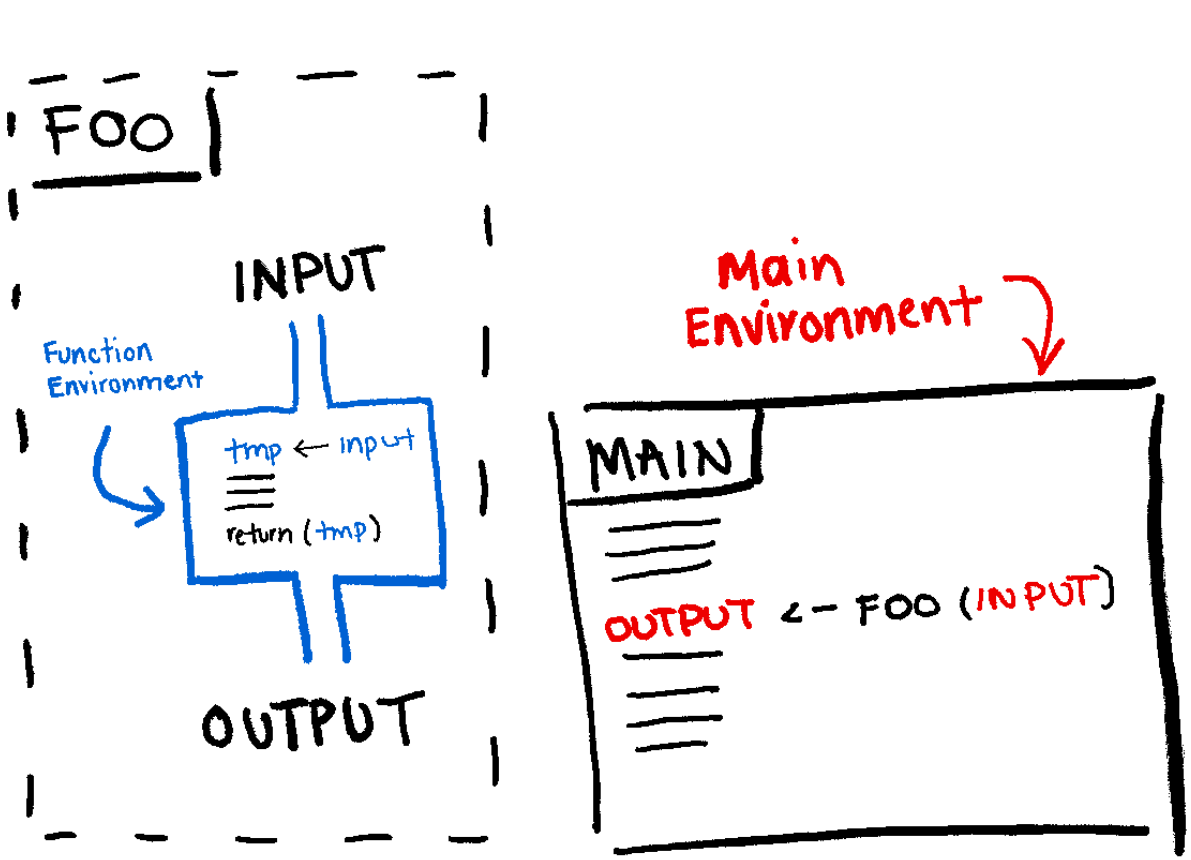
```
# if
if (condition is true) {
  perform action
}

# if ... else
if (condition is true) {
  perform action
} else {  # that is, if the condition is false,
  perform alternative action
}
```

For loop

- Repeat a set of operations a certain number of times

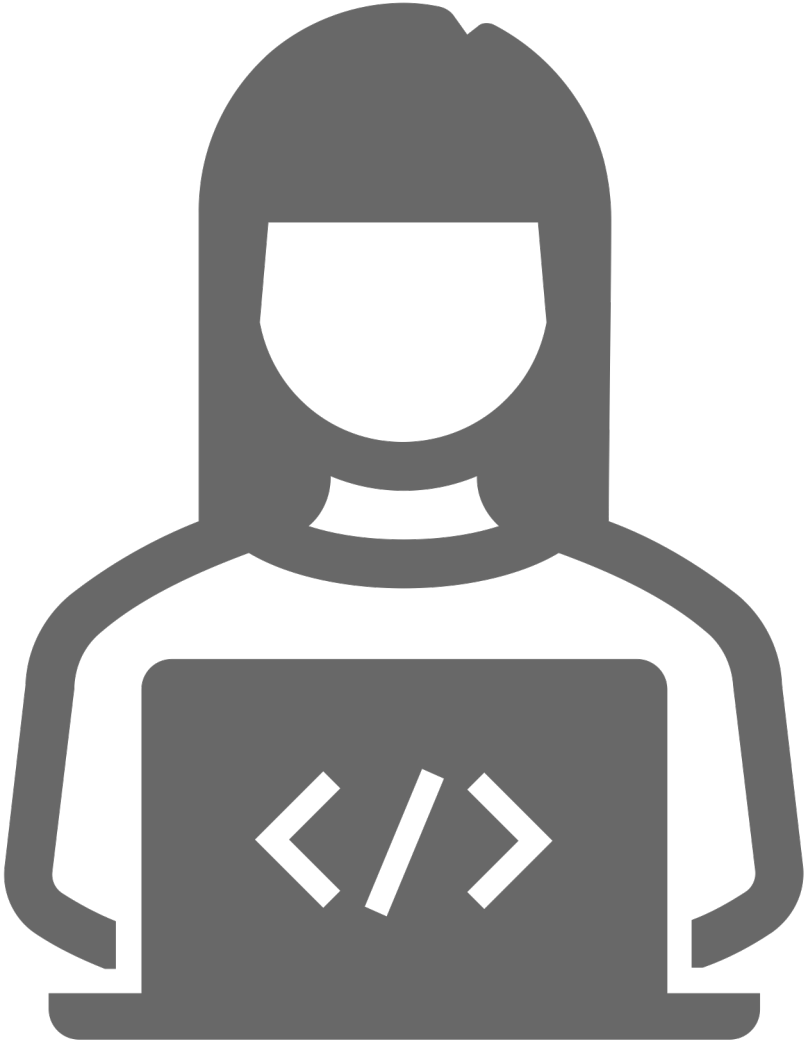
```
for (iterator in set of values) {
  do a thing
}
```

Functions and environments



Pass by value and scope

- When we pass an object to a function, a copy of it is created internally
- The changes made inside the function won't modify the original object we passed to it
- Any variables created inside the function will only exist during the function's execution time

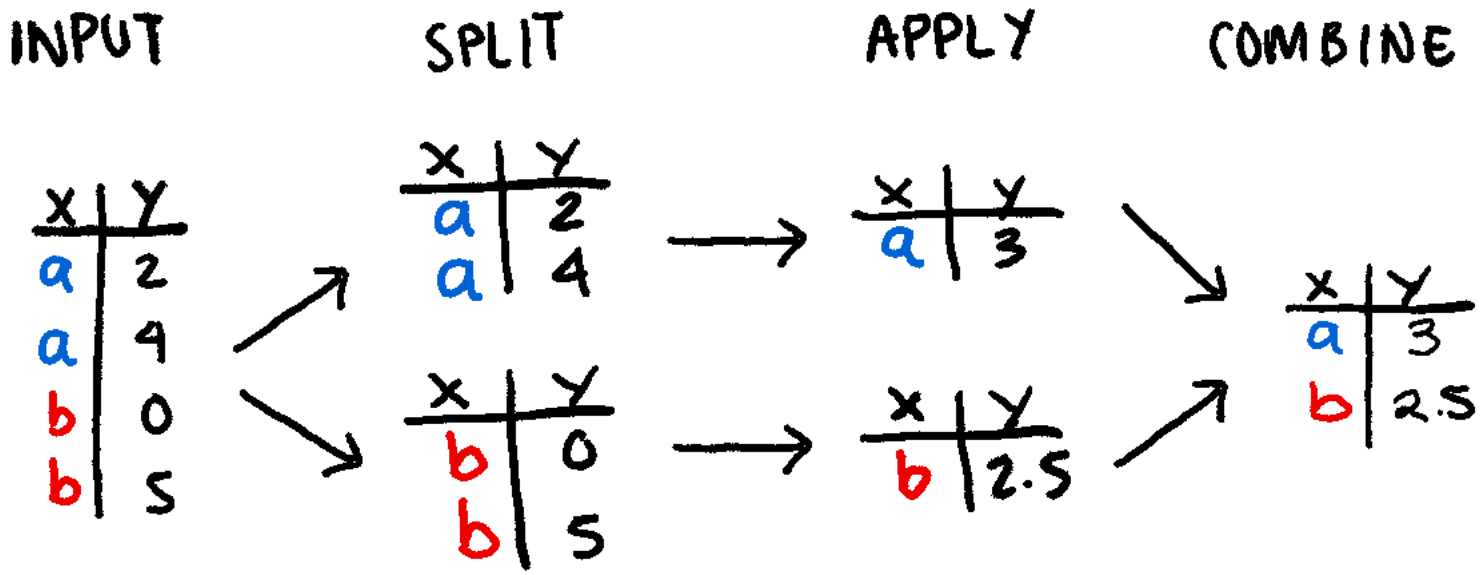


Advanced data manipulation

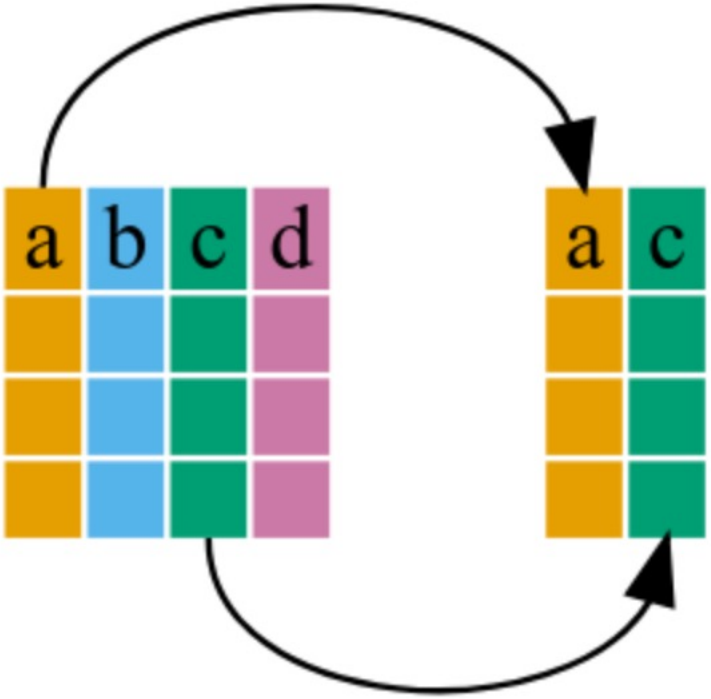
Learning objectives:

- Become familiar with the dplyr syntax
- Create pipes with the operator *%>%*
- Perform operations on data frames using dplyr and tidyr functions
- Implement functions from other external packages

Split-Apply-Combine problem

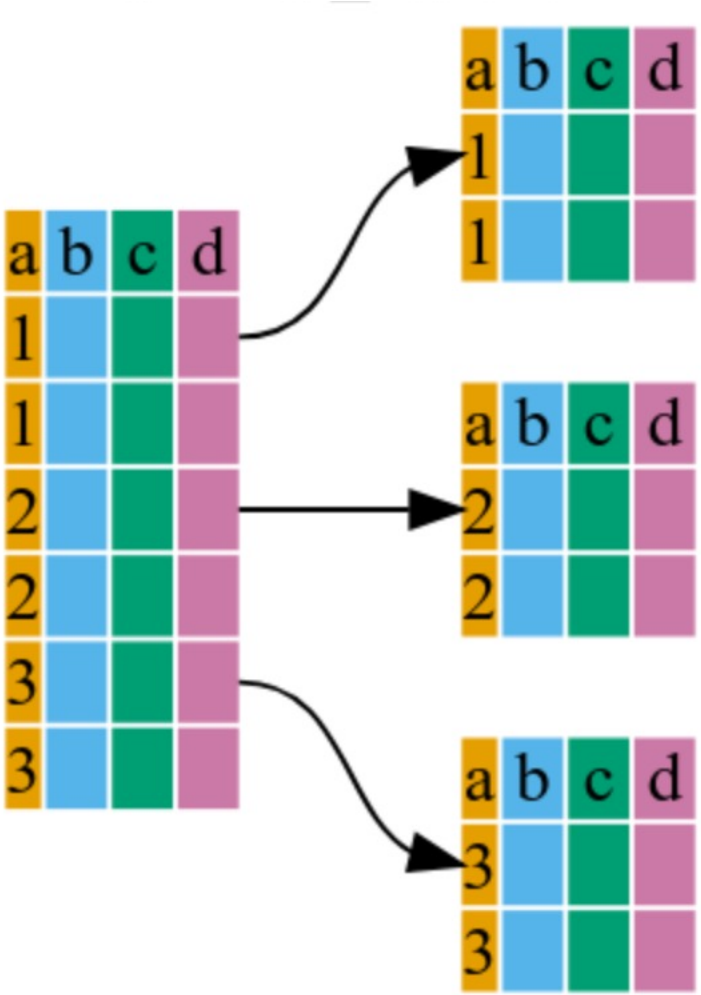


Select



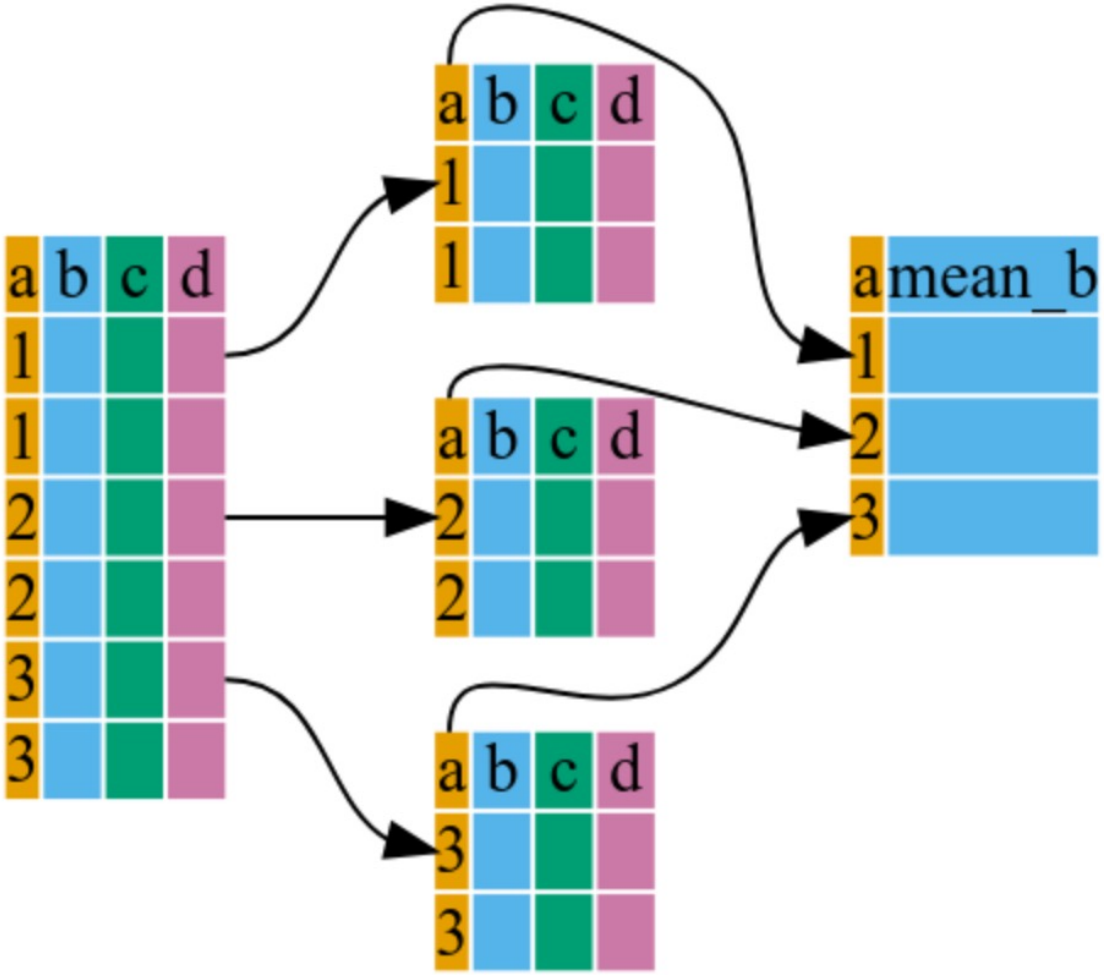
https://swcarpentry.github.io/r-novice-gapminder/13-dplyr/index.html

Group by

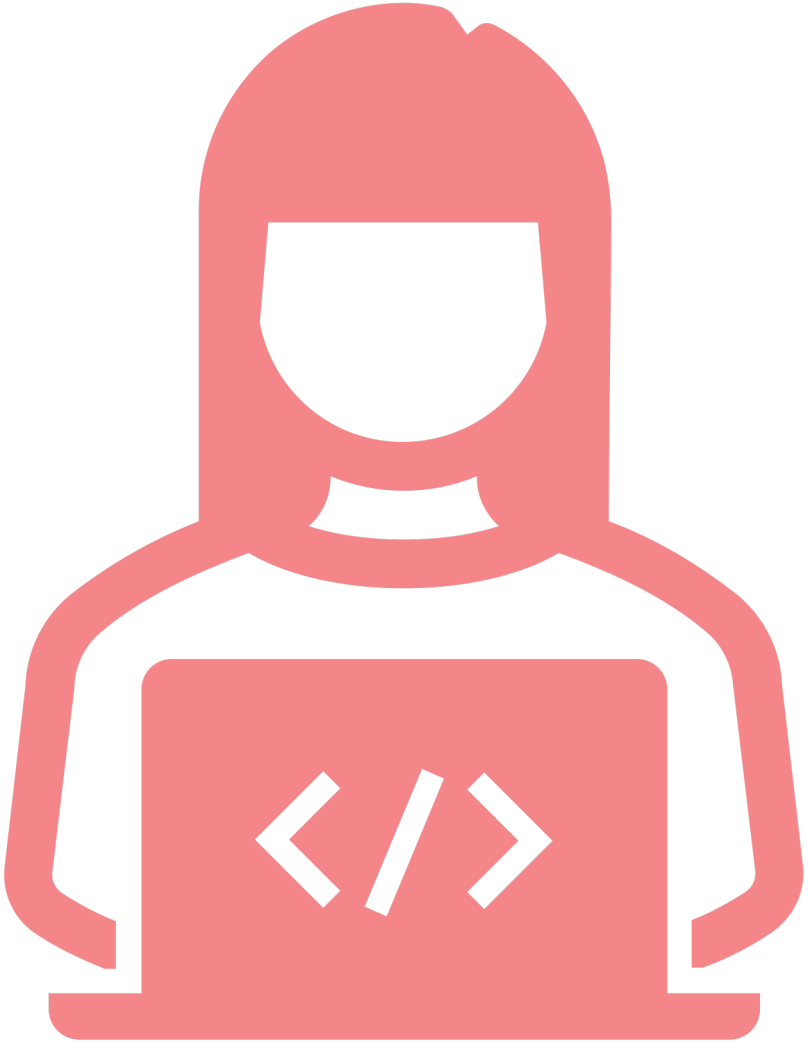


https://swcarpentry.github.io/r-novice-gapminder/13-dplyr/index.html

Summarize



https://swcarpentry.github.io/r-novice-gapminder/13-dplyr/index.html

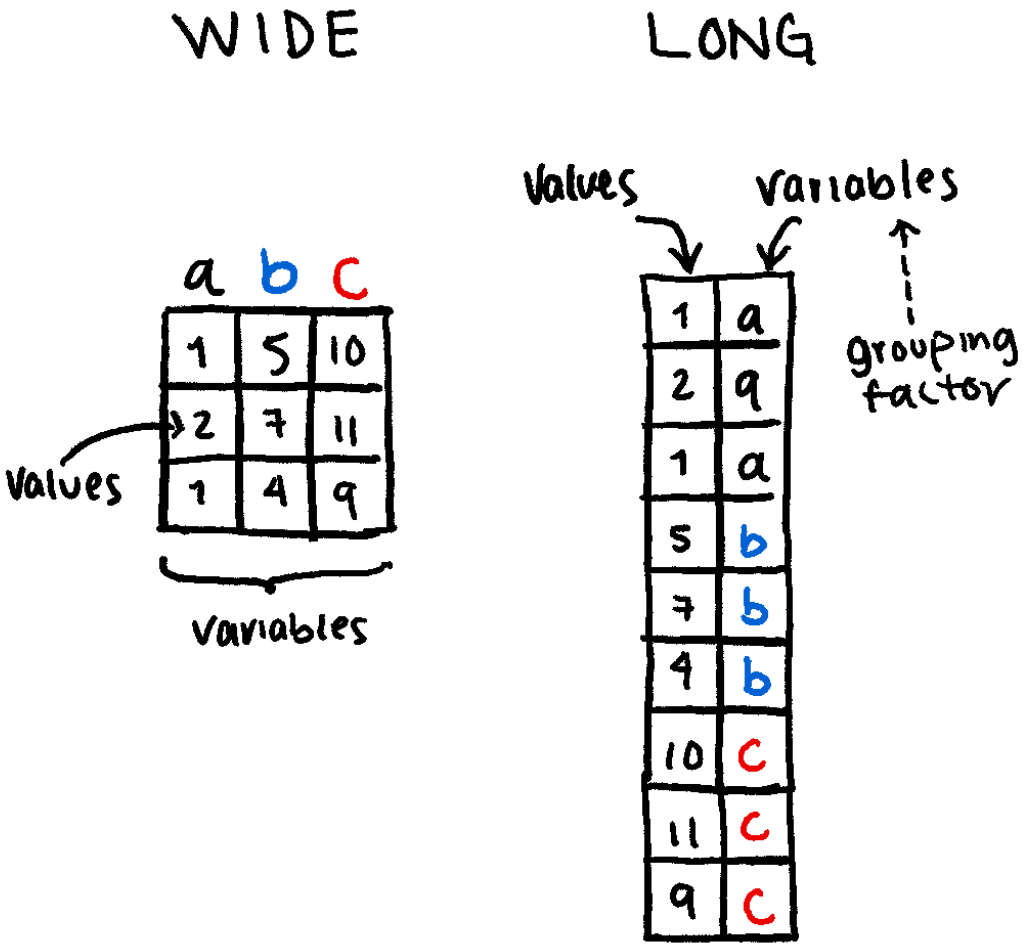


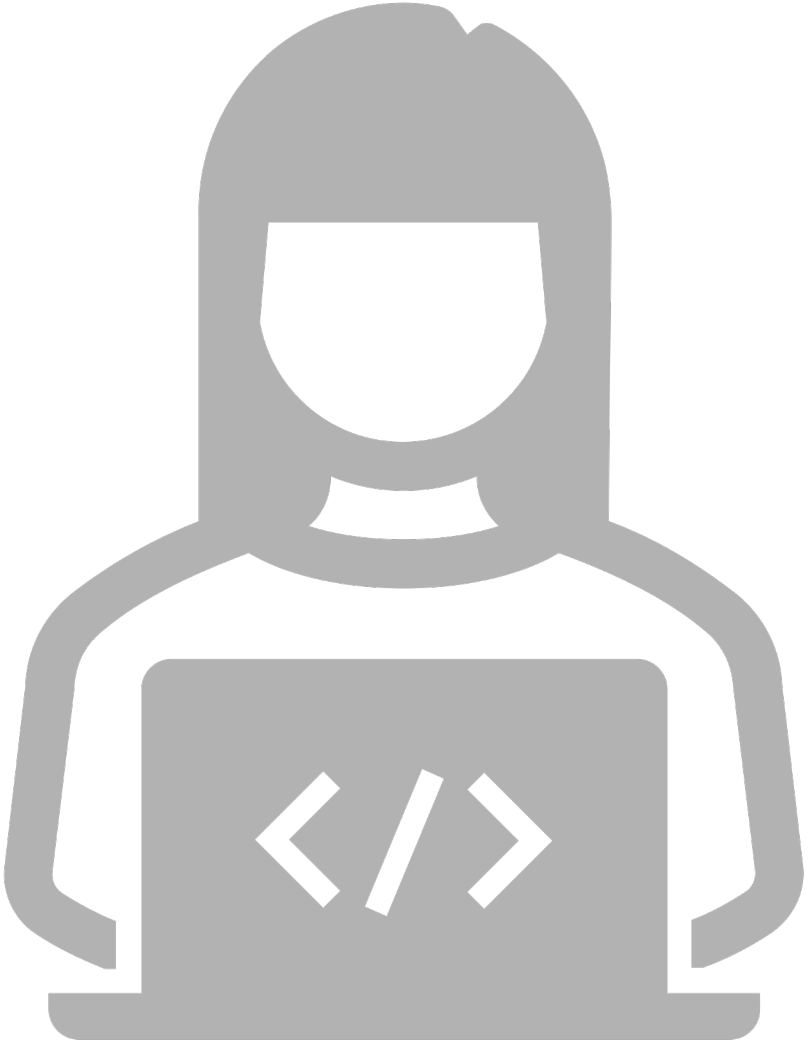
Generating visual outputs

Learning objectives:

- Create basic plots using base R functions
- Understand the connection between data frames and ggplot2
- Create basic graphs with ggplot2
- Use factors to customize graphics in ggplot2
- Learn about RMarkdown syntax to create reports
- Get familiar with existing RMarkdown templates

Formatting data for ggplot





Activity: Analyzing a medical data set

Learning objectives:

- Familiarize with a real-life use case of R
- Apply the knowledge from previous modules to create an analysis pipeline

COVID testing dataset

Details

Data on testing for SARS-CoV2 from days 4-107 of the COVID pandemic in **2020**. CHOP is a pediatric hospital in Philadelphia, Pennsylvania, USA. These data have been anonymized, time- shifted, and permuted.

Text taken from https://cran.r-project.org/web/packages/medicaldata/medicaldata.pdf

The dataset

Documentation

- Part of the medicaldata package
- https://htmlpreview.github.io/?https://github.com/higgi13425/medicaldata/blob/master/man/description_docs/covid_desc.html
- https://htmlpreview.github.io/?https://github.com/higgi13425/medicaldata/blob/master/man/codebooks/covid_testing_codebook.html

Format

A data frame with 15524 observations and 17 variables

subject_id id number for each subject; type: numeric

fake_first_name an auto-generated fake first name; type: character

fake_last_name an auto-generated fake last name; character

gender anonymized Gender, levels: female, male; type: character

pan_day day after start of pandemic; type: numeric

test_id test that was performed, levels: covid, xcvd1; type: character

clinic_name Clinic or ward where the specimen was collected, 88 levels; type: character

result result of test, levels: positive, negative, invalid; type: character

demo_group patient group, levels: patient, misc_adult, client, other adult, unidentified; type: character

age Age of subject at time of specimen collection (Anonymized), units = years; type: numeric

drive_thru_ind Whether the specimen was collected via a drive-thru site, levels: 1: Collected at drive-thru site; 0: Not collected at drive-thru site; type: numeric

ct_result Cycle at which threshold reached during PCR, range: 14.05-45; type: numeric

orderset Whether an order set was used for test order, levels: 1: Collected via orderset; 0: Not collected via orderset; numeric

payor_group Payor associated with order, levels: commercial, government, unassigned, medical assistance, self pay, charity care, other; type: character

patient_class Disposition of subject at time of collection, levels: inpatient, emergency, observation, recurring outpatient, outpatient, not applicable, day surgery, admit after surgery-obs, admit after surgery-ip; type: character

col_rec_tat Time elapsed between collect time and receive time, range: 0 - 61370.2, units = hours; type: numeric

rec_ver_tat Time elapsed between receive time and verification time, range: -18.6 - 218.2, units = hours; type: numeric ...

Software development concepts

Learning objectives:

- Familiarize with general good coding practices
- Learn about documentation standards
- Things to avoid when programming in R
- Learn how to debug and troubleshoot

Look back at all we learned in this workshop:

1 The language
- History
- Foundation
- Syntax
- Logical ops
- Help
- Packages

2 Data types
- Vectors
- Factors
- Lists
- Data Frames
- Arrays
- ***Hands on***

3 Basic data manipulation
- Read
- Write
- Subset
- Split
- Join
- ***Hands on***

4 Control Structures
Functions
If statement
for loop
Real-life hands on

5 Advanced data Manipulation
dplyr
tidyr
plyr
DataTable
Hands on

6 Generating Outputs
Graphics
ggplot2
RMarkdown
Templates

7 Software development
Good coding practices
Documentation standards
Debugging

Big thanks to:

RISA

- Andreas Heilmann
- Jannika Finger

MiCM team

- Prof. Guillaume Bourque
- Prof. Celia Greenwood
- Adrien Osakwe
- Georgette Femerling
- Kevin Liang

Our sponsors

Roche

McGill UNIVERSITY

Useful links

- R software project
- RStudio Cheatsheet
- R ggplot2 Cheatsheet
- R dplyr Cheatsheet
- More resources