R Seminar Series Session 2

Data-manipulation in R

Maki IKEGAMI (Bio-Protection, Lincoln university)

Makihiko.Ikegami@lincoln.ac.nz

Targets of this course

- **Get along with R**
- **Understand data types and structures in R**
- **Use a sample dataset “iris” to practice “data.frame”**
- **(Import/export a file to R)**
- **(Understand importance of data preparation)**

Outline

- Before handling data in R… just play it!
- Data types in R
    - numeric, character, logical
    - factor, date, NA/NaN/Inf, complex
- Data structures
    - vector, matrix, data.frame, list
- (How to read/write a file in R / data preparation)
- Handling data frame
    - Subset data in R
    - split(), order(), merge()

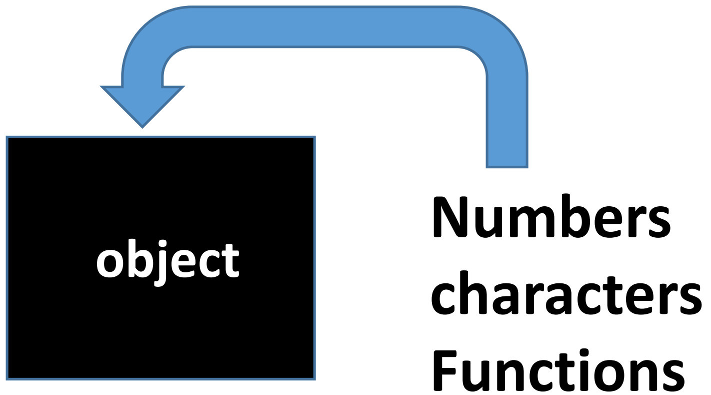
start

Please open R-Studio and create a new project.

And then open R file, "2_data_manipulation_in_R.r" from your local directory

Object (L50: see line 50 in sample code)

- **An object is a container where you can store "something" in R. You can store,**
  - **a number, a character, set of numbers/characters, text, functions (programs).**
  - **Some characters are predefined, or "reserved", by R**



Objects, data type, functions

- **In R, you can use numbers, characters, and some signs (+, -, *, / and so on)**
- **Numbers are numbers, characters can be "object" or "function"**
- **Most characters/texts are not used (empty), but some of them are pre-defined (reserved words)**
- **Object must not start with numbers/signs**

Now let's see how we can use object

Data types (L150)

You can put "data" into "object", data can be…

- **Numeric**
    - **1, 2, 3, 4…**
    - **0.1, -0.00001….**
- **Characters: with double quotations**
    - **"a", "b", "c",**
    - **"Pinus", "Acer", "Poa"**
    - **"1", "2", "3"**
- **Logical**
    - **TRUR/FALSE**

Data types –special cases-

You can put “data” into “object”, data can be…

- **Factor (numeric) (we will see this later)**
    - **Group of “character” with levels**
    - **For statistical analysis: categorical data**
    - **Be careful for converting factors to numeric data**
- **Date (numeric)**
    - **as.Date**
    - **"2014-09-15“, "2012/09/15"**
- **Complex numbers**
    - **1 +1i**
- **NULL/NA/NaN/Inf**

Data structures (L270)

R can store data in objects, and data can have different structures

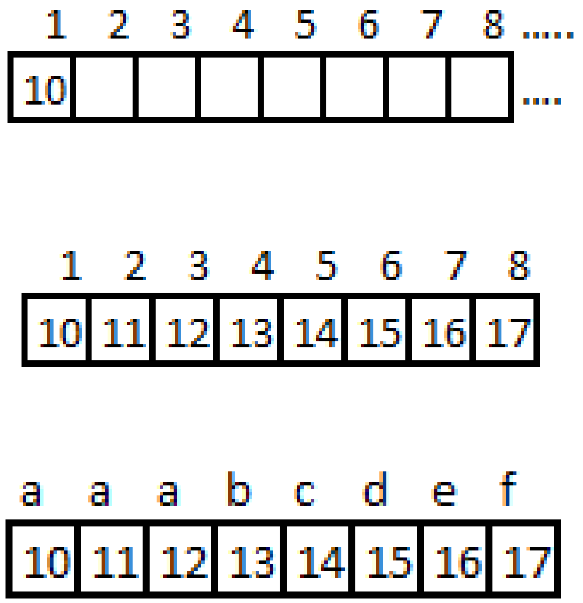
- **Vector**
- **(Matrix)**
- **Data.frame**
- **list**

Vectors –all about R-

Vector is a basic data structure in R.

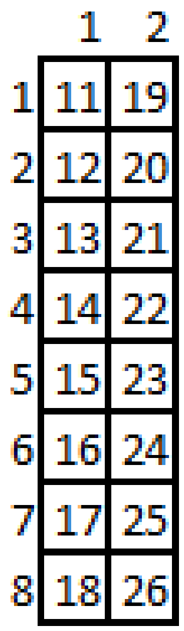
- **A vector has a set of “data” with order.**
- **You can store any “data” (number, character, logic, date) in a vector, but only one data type is allowed in one vector.**
- **You can perform algebra (numeric or sets) on vectors**
- **You can “name” each data in a vector**

Vectors – access data in a vector (L340)



A vector has an order, so you can specify “location” with number (order), or name (but if there are same name, then one of them (earlier order) will be chosen.

Matrix –vector with dimensions- (L440)



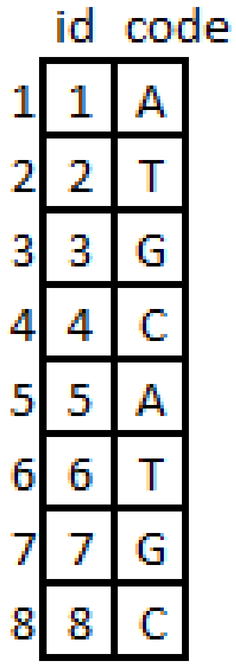
	1	2
1	11	19
2	12	20
3	13	21
4	14	22
5	15	23
6	16	24
7	17	25
8	18	26

Matrix is a special type of vectors; it has a "dimension", but it is a single vector

For this reason, you can not mix different types of data.

Because matrix (vector) is easy to process mathematically (thus for computer) many functions need "matrix (vector)" data to process. So you have to understand the nature of matrix (vector).

Data frame –key for data analysis- (L480)



	id	code
1	1	A
2	2	T
3	3	G
4	4	C
5	5	A
6	6	T
7	7	G
8	8	C

Data.frame is a data structure that allows you to store vectors into one object.

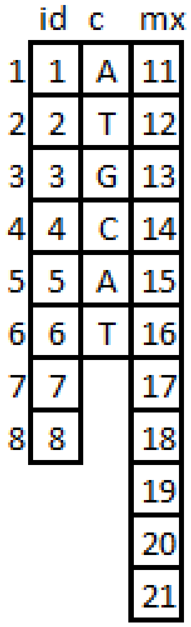
- **Each column is a vector.**
- **Each column can have a different data type.**
- **Each column must have same length.**
- **Each column must have a DIFFERENT name.**

Factors (L540)

Data with levels:

- **Factor is group of data based on same values (e.g. species, plot, country, same numbers)**
- **Each group has level information (numeric)**
- **useful and important for statistical analysis**
- **When you import your data in R using read.csv, any data with characters (text) are automatically read as factor**
  - **problems may occur!**

List – you will need in future – (L570)

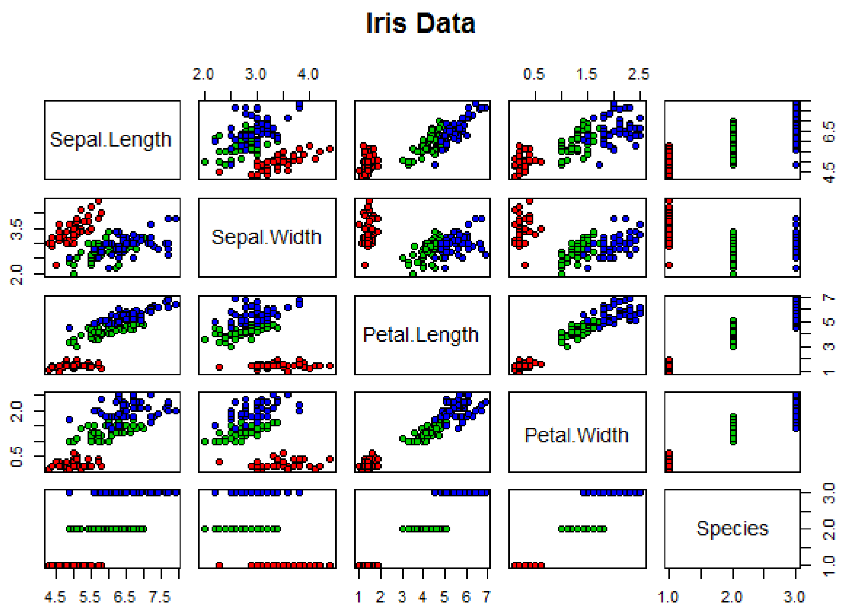


List is an expanded version of data.frame

- **Unlike data.frame, each vector can have different lengths.**
- **outputs from some R functions are in list format**
- **Accessing each vector is similar to data.frame**

Data frame –where you start- (L610)

- **Introducing IRIS data**
  - **A built-in data set in R**
  - **Useful dataset for different analysis and data handling**
- **First, recapture how to read and write a file in R**



Read/write a file (L610)

- where your working directory is : getwd()
- set your working directory: setwd()
- Store your data in csv file and use “read.csv” is simplistic but problematic sometimes.
- When you prepare data
    - One column, one data type
    - Do not mix “character” in a “numeric” column (except names)

One column, one data type

- **All data must have one column, one data type**
- **Same data from different categories (species, plot, country) should be store in a same column, and add a category column**

✗

	setosa				virginica				versicolor			
sample	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	5.1	3.5	1.4	0.2	7	3.2	4.7	1.4	6.3	3.3	6	2.5
2	4.9	3	1.4	0.2	6.4	3.2	4.5	1.5	5.8	2.7	5.1	1.9
3	4.7	3.2	1.3	0.2	6.9	3.1	4.9	1.5	7.1	3	5.9	2.1
4	4.6	3.1	1.5	0.2	5.5	2.3	4	1.3	6.3	2.9	5.6	1.8
5	5	3.6	1.4	0.2	6.5	2.8	4.6	1.5	6.5	3	5.8	2.2

○

id	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5	3.6	1.4	0.2	setosa
51	7	3.2	4.7	1.4	versicolor
52	6.4	3.2	4.5	1.5	versicolor
53	6.9	3.1	4.9	1.5	versicolor
54	5.5	2.3	4	1.3	versicolor
55	6.5	2.8	4.6	1.5	versicolor
101	6.3	3.3	6	2.5	virginica
102	5.8	2.7	5.1	1.9	virginica
103	7.1	3	5.9	2.1	virginica
104	6.3	2.9	5.6	1.8	virginica
105	6.5	3	5.8	2.2	virginica

Data frame –inspect your data- (L660)

data.frame can contain huge number of data, so you need to use some functions to inspect them

- **See parts of data? head(), tail(), str()**
- **Dimensional data? dim(), nrow(), ncol()**
- **Names of columns/rows? colnames() rownames()**
- **Summarise? summary()**
- **Plot them? pairs()**

Data frame –access column data- (L700)

Depending on how you access data.frame you will have different data outputs.

- **sampledata[1] -> data.frame**
- **sampledata[,1] -> vector**
- **sampledata["Sepal.Length"] -> data.frame**
- **sampledata[,"Sepal.Length"] -> vector**
- **sampledata$Sepal.Length -> vector**
- **sampledata[1,] -> vector**
- **Different functions require different data input!!**

Data frame –way to access row data- (L760)

Outputs from row data are data.frame

- **sampledata[1,] -> data.frame**
- **This is because row data contains different data.type and attributes.**

Subset -selecting data- (L800)

Subset is extracting some parts of data for analysis. There are different ways to do..

- **Specify rows you need:**
  - **e.g. sampledata[c(1:50),]**
- **Use subset function:**
  - **e.g. subset(sampledata, sampledata$Species == "setosa")**
- **Subscripting:**
  - **e.g. sampledata[sampledata$Species == "setosa",]**

Subset -logical expression-

For logical operations:

- **==: equals**
- **!=: not equal**
- **>: greater than (>= equal to or greater)**
- **<: smaller than (<= equal to or smaller)**
- **&: and**
- **| :or**

Combine/merge data frames(L880)

- **Combining different data can be nightmare but R has good options**
- **rbind()/cbind() can combine two data.frames (data objects) by rows (rbind) and columns (cbind).**
    - **Be careful what you are going to combine.**
- **For two data.frame with a same ID column (unique values for each row), you can use “merge()” functions.**
- **This is very useful for data based on individual plots, countries, regions, or species.**

tips

You can get information on books, internet or wherever.

- **just do exactly what they told you.**
- **change the codes little by little on the code.**
- **use different columns, change numbers.**
- **Split the codes into smaller parts.**
  - **pairs(iris, main = "Iris Data", pch = 21, bg = c("red", "green3", "blue")[unclass(iris$Species)])**
    - **c("red", "green3", "blue")[unclass(iris$Species)]**
      - **unclass(iris$Species)**
        - **iris$Species**