Estadística Computacional Guía

Indice:

* Introduction to R P1
* Introduction to dplyr P2
* Efficient Programming P3
* Parallel Computing P4
* GPU Computing
* Testing P5
* Statistical Analysis: Basic Concepts
* Data cleaning
* Sampling
* Hypothesis Testing
* Introduction to Bayes
* Bayesian Inference

## Introduction to R

R packages:

* Ggplot2 Visualization and graphics
* Dplyr & tidyr Data transformation
* Lubridate Date Management
* Stringr Wrappers for string ops
* Shiny To create user interfaces
* Mass, h20, kernlab, glmnet, caret, rpart, randomforest, gmb ML

Advantages:

* Free, open source, state of the art, active user community

Disadvantages:

* No commercial support, easy to make mistakes without knowing, unclear relevant package choice.

Cheatsheet for code https://www.rstudio.com/resources/cheatsheets/

Plots:

* Dotplots and lines:

plot(x) and plot(x, type = ‘1’)

* Scatterplots:

plot(x,y, xlab=’x-axis’, ylab=’y-axis’, main= ‘Plot X v. Y’)

* Histograms:

Hist(x, main= ‘Histogram title’, xlab= ‘x-axis’)

* Boxplots:

Boxplot(x, main= ‘boxplot title’, ylab= ‘y-axis’)

Tools for data wrangling:

* Aggregate and rapply functions
* Tidyr and dplyr libraries

Variable types: Numeric (Continuous/Discrete), Factors (Categorical), Char(Text)

GGPlot2 Cheatsheet: https://www.maths.usyd.edu.au/u/UG/SM/STAT3022/r/current/Misc/data-visualization-2.1.pdf

Conditionals:

* if
* else if
* ifelse() Use this when a conditional needs to be applied to each member of a vector
* switch() Handy for deciding between options, never used
* && (AND) || (OR)

For further technical details, check the code guide:

For() vs. while():

For is better when the number of times to repeat is clear in advance.

While is better when you can recognise a condition to stop iterations, even if you can’t guess to begin.

While is more general. Every for could be replaced with a while, but not viceversa. REMEMBER TO USE VECTORISATION WHENEVER POSSIBLE.

Why use functions?

* Data structures tie values to a single object
* Functions tie commands into one object
* Easier to understand and work with

Function structure:

* Inputs (Arguments), Body (Code that’s executed), Output (Return value)

Function side effects:

* Printing something out to the console
* Plotting something on the display
* Save a data file

What’s an R environment?

* A collection of objects: Functions, Variables, etc…
* Each function has its own env.
* Names here override names in the global env.
* Internal env start with name arguments.
* Assignments inside the function only change the internal environment.

Top-down function design:

* Start with the big picture view of the task
* Divide and conquer
* Join them together

## Introduction to tidyverse

## Efficient Programming

## Parallel Computing

## GPU Computing

## Testing

## Statistical Analysis

## Data cleaning

## Sampling

## Hypothesis Testing

## Introduction to Bayes

## Bayesian Interference