

1 R Programming

List

- `[[idx]]`: get element in a list
- `str(ls)`: get **structure** of a list (similar to summary)
- `saveRDS` and `loadRDS`

Recycling Rule

- shorter vectors are recycled until they match the length of the longest vector
- the length of the longest vector must be a multiple of the shorter vector in arithmetic operations!

Useful functions

- `sample(x, size, replace, prob)`
 - `size`: length of output vector
 - `replace`: if TRUE, then sampling is with replacement
 - `prob`: a vector of probability weights
- `rep(x, times, length.out)`
- `table()`
- `args(func)`: list the arguments of a function
- `seq(from, to, by, length)`
- `paste(v1, v2, sep)`: concatenate vectors after converting them to characters
 - `sep`: separator between elements of `v1` and `v2`
 - The recycling rule applies when `length(v1) != length(v2)`
- apply function family: apply function to each row (1) or column (2)
 - `apply(X, margin, func, ...)`
 - * Note that `X` must be a matrix in `apply`
 - `sapply` returns a vector or a matrix
 - `lapply` returns a list, useful when the output of the function may not be all of the same length/type

Function debugging

- `cat("...")`: used to print statements
- `browser()`: debugging with breakpoint

Important classes

Strings

- Start by importing `tidyverse` and `stringr`
- Library functions
 - `str_length`: returns vector of string lengths
 - `str_c(..., sep)`: concatenate strings with optional separator

- `str_sub(string, start, end)`: returns vector of substrings
- Regular expressions (`str_view()` to test out regex), [*Tidyverse Article*](#)
 - to match an **a** at the beginning of a string
`str_view(x, "^a")`
 - to match an **a** at the end of a string
`str_view(x, "a$")`
 - to match an **a** or **e** at the end of a string
`str_view(x, "[ae]$")`
 - to match a string of 3 chars with **a** in the middle
`str_view(x, ".a.")`
- `str_detect(vec, regex)`: returns a boolean vector
- `str_extract(vec, regex)`: returns a vector of strings, particularly helpful for ".a." regex

Factors

`factor(vec, levels=c(...))`: convert `vec` to factors with fixed levels

Date

- `as.Date(x, format)`: convert string `x` to Date object
e.g. `as.Date("2014/02/22", "%Y/%m/%d")`
- `months(d)`: what month of the year is the date in?
- `weekdays(d)`: what day of the week is the date on?
- `Sys.Date()`
- `cut(x, breaks, labels)`: usually used to group dates that fall into a month/week/quarter
 - `breaks`: numeric vector/string ("month", "week")
 - `labels`: if TRUE, return a label vector

Basic Plotting

plot()

- `pch`: abbr. for plotting character
- `col`: use string or `rgb(..., alpha=?)` to change colour
- `cex`: abbr. for character expansion
- use `points()` or `lines()` to add more stuff to an existing plot

barplot()

2 Importing Data