Coursera: Data Science: Foundations using R Specialization

Course 3: Getting and Cleaning Data

① Plyr Pkg → arrange( )

② dplyr pkg

/// select: return a subset of the columns of a data frame
/// filter: extract a subset of rows from a data frame based on logical conditions.
/// arrange: reorder rows of a data frame
/// rename: rename variables in a data frame
/// mutate: add new variables / columns or transform existing variables
/// summarise / summarize: generate summary statistics of different variables in the data frame, possibly within strata.

③ merge( ), Plyr: join( )
↳ Default left join

④ text variables:

1. fixing character vectors: /// tolower( ), toupper( )

Example: names(cameraData)

"address" "direction" "street" "crossStreet" "intersection" "location.1"

2. fixing character vectors: strsplit( )

/// strsplit(names(cameraData), "\\.")[[5]] → "intersection"

A: A[[b]] → "Location" "1"

3. fixing character vectors : sub( )

Example: "solution_id" → sub("_", " ") → "solutionid"

4 fixing character vectors: gsub( )

Example: A <- "this_is_a_test"

① sub("_", " ", A) → "this is_a_test"

② gsub("_", " ", A) → "this is a test"

5. finding values : grep( ) , grepl( )

Others: Pkg: stringr ① substr( ), ② str_trim( )

① Regular expression

/// character classes with []

/// *^[0-9][a-zA-Z] will match number & letters (order does not matter!)

/// "." is used to refer to any character

/// "|" OR

/// (.*) where * means "any number, including none, of the item." + means at least one of them.

* Example: [0-9]+(.*)[0-9]+

is "greedy" always matches the longest possible string

? fixes it

But ^s(.*?)s$