Clean ecological data collected from quadrats with quadcleanR

CHEAT SHEET & PACKAGE BY DOMINIQUE MAUCIERI

Clean

data

name1	name2	name3

labelset

from	to
name1	new1
name2	new2
name3	new3

new1	new2	new3

change_names(data, labelset, from, to) Using a new data frame of labels, change column names in one function. Helpful if column names are shorthands or contain spaces and characters that are not supported in column names in R.

change_names(data, labelset, "from", "to")

data

col1	col2
a	one
b	two
a	three
b	two
a	one

col1	col2
a	1
b	2
a	3
b	2
a	1

change_values(data, column, from, to) Using two vectors, change the values in one column to a new set of values. Helpful if you need to change many values at once, like updating changes to site names or taxonomy.

change_values(data, "col2", c("one", "two", "three"), c("1", "2", "3"))

data

col1	col2
a	1_1
b	1_2
a	2_2
b	1_2
a	2_1

col1	col2
b	1_2
a	2_2

keep_rm(data, values, select, keep = TRUE, drop_levels = TRUE, exact = TRUE, colname) Using a character, or part of character select rows or columns of the data frame to either keep or remove. A more customizable way to subset your data as you can keep or remove based on partial matches, or cells containing select characters.

keep_rm(data, "_1" , select = "row", keep = FALSE, drop_levels = TRUE, exact = FALSE, "col2")

data

col1	col2
one.jpg	
two.jpg	
three.jpg	
four.jpg	
five.jpg	

col1	col2
one	
two	
three	
four	
five	

rm_chr(data, rm, full_selection = TRUE, cols) Parts of characters can be removed based on a vector of removal characters. When working with images, this can be helpful to remove extra characters from image IDs, or anywhere else where you want to remove specific characters from your data.

rm_chr(data, rm = ".jpg", full_selection = FALSE, cols = "col1")

data

col1	col2	col3
4	7	5
0	6	4

col1_2	col3
11	5
6	4

sum_cols(data, from, to) Select columns and attach a vector of their new names, then columns with matching names will have each row summed. This is helpful to simplify your data quickly, like simplifying at a higher taxonomic group.

sum_cols(data, from = c("col1", "col2", "col3"), to = c("col1_2", "col1_2", "col3"))

Assess

data

Year	Site	Quadrat
2017	1	1
2017	1	2
2017	1	3
2017	2	1
2017	2	2
2018	1	1
2018	1	2
2018	2	1
2018	2	2

Site	2017	2018
1	3	2
2	2	2

sample_size(data, dim_1, dim_2, count) Specify which columns to use to produce a table with sample sizes. Helpful to visualize number of samples in your data.

sample_size(data, dim_1 = "Year", dim_2 = "Site", count = "Quadrat")

Add

data

ID	count
2017	5
2019	2
2017	5
2018	1

add

ID	Temp
2019	12
2018	14
2017	17

ID	Avg_Temp	count
2017	17	5
2019	12	2
2017	17	5
2018	14	1

add_data(data, add, cols, data_id, add_id, number = FALSE) Using key identifying columns, add additional columns to an existing data frame. Helpful for adding environmental or taxonomic data to your quadrat data

add_data(data, add, cols = "Temp", data_id = "ID", add_id = "Avg_Temp", number = 2)

data

ID	count
May	5
June	2
July	5
August	1

ID	Summer	count
May	No	5
June	Yes	2
July	Yes	5
August	Yes	1

categorize(data, column, values, name, binary = TRUE, exact = TRUE, categories) Using a column within the data frame, categorize rows in a binary of yes or no, or customize with a set of category names. Especially useful for turning ID tags into useful categories for analysis such as morphology, taxonomy etc.

categorize(data, column = "ID", values = c("Ju", "Au"), name = "Summer", binary = TRUE, exact = FALSE)

Process

data

ID	SP1	SP2
2017	4	5
2019	0	2
2017	4	5
2018	9	1

ID	SP1	SP2	total_pts
2017	0.444444	0.555556	9
2019	0	1.000000	2
2017	0.444444	0.555556	9
2018	0.900000	0.100000	10

cover_calc(data, spp, prop = TRUE, total = FALSE) Convert the number of observations for each species or non-species to proportion or percent cover within each row based on the total number of observations in each row. Useful for quadrats with varying numbers of observations to calculate each row's percent cover all at once.

cover_calc(data, spp = c("SP1", "SP2"), prop = TRUE, total = TRUE)

data

Year	Row	Column
2017	100	37
2017	34	100
2017	88	28
2017	93	25
2017	78	16
2018	45	100
2018	100	2
2018	27	73
2018	93	03

Year	Row	Column
2017	88	28
2017	78	16
2018	27	73

crop_area(data, row, column, id, dim, obs_rm = FALSE, obs_range, res = FALSE, res_dim_x, res_dim_y) Using the location of annotated points within quadrats and the size of the quadrat, crop quadrat data to a smaller area, while maintaining the spatial relationships between points. Useful for making different sized quadrat data comparable.

crop_area(data, row = "Row", column = "Column", id = "Year", dim = c(0.9, 0.9))

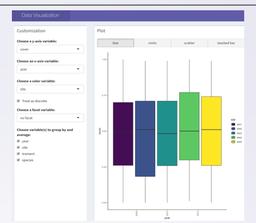
data

ID	SP1	SP2	Blurry	Unclear
2017	4	5	7	3
2019	0	2	3	5
2017	4	5	1	4
2018	9	1	6	3

ID	SP1	SP2	unusable
2019	0	2	8
2017	4	5	5
2018	9	1	9

usable_obs(data, unusable, max = FALSE, cutoff, print_max = FALSE, rm_unusable = TRUE) Sum columns containing unusable observations and remove rows that contain more than the specified cutoff number of unusable points. Helpful if there are annotations that were unidentifiable and you want to remove them from the total usable observations, and you can remove quadrats with too many unusable observations.

usable_obs(data, c("Blurry", "Unclear"), max = TRUE, cutoff = 9, print_max = TRUE, rm_unusable = TRUE)



visualize_app(data, xaxis, yaxis) Using an interactive shiny app, visualize and explore cleaned quadrat data.

visualize_app(data = coral, yaxis= c("year", "site", "transect", "species"), xaxis= "cover")