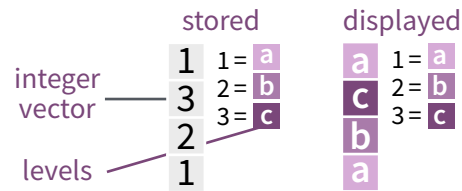


# Factors with forcats : : CHEAT SHEET

The **forcats** package provides tools for working with factors, which are R's data structure for categorical data.

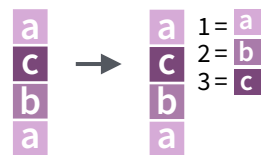
## Factors

R represents categorical data with factors. A **factor** is an integer vector with a **levels** attribute that stores a set of mappings between integers and categorical values. When you view a factor, R displays not the integers, but the levels associated with them.



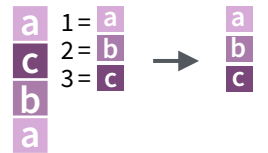
Create a factor with `factor()`

**factor**(x = character(), levels, labels = levels, exclude = NA, ordered = is.ordered(x), nmax = NA) Convert a vector to a factor. Also **as\_factor()**.  
**f** <- **factor**(c("a", "c", "b", "a"), levels = c("a", "b", "c"))



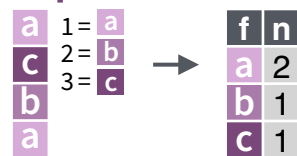
Return its levels with `levels()`

**levels**(x) Return/set the levels of a factor. `levels(f)`; `levels(f) <- c("x", "y", "z")`



Use `unclass()` to see its structure

## Inspect Factors



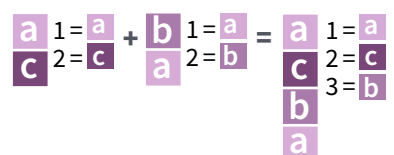
**fct\_count**(f, sort = FALSE, prop = FALSE) Count the number of values with each level. `fct_count(f)`



**fct\_match**(f, lvls) Check for lvls in f. `fct_match(f, "a")`

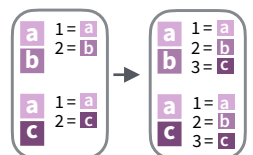
**fct\_unique**(f) Return the unique values, removing duplicates. `fct_unique(f)`

## Combine Factors



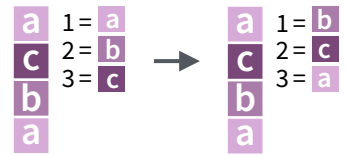
**fct\_c**(...) Combine factors with different levels. Also **fct\_cross**().

**f1** <- **factor**(c("a", "c"))  
**f2** <- **factor**(c("b", "a"))  
**fct\_c**(f1, f2)



**fct\_unify**(fs, levels = lvls\_union(fs)) Standardize levels across a list of factors. `fct_unify(list(f2, f1))`

## Change the order of levels



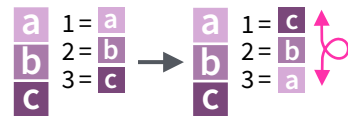
**fct\_relevel**(f, ..., after = 0L)  
Manually reorder factor levels.  
`fct_relevel(f, c("b", "c", "a"))`



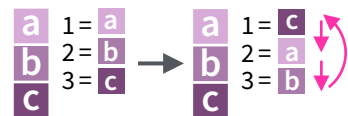
**fct\_infreq**(f, ordered = NA) Reorder levels by the frequency in which they appear in the data (highest frequency first). Also **fct\_inseq**().  
**f3** <- **factor**(c("c", "c", "a"))  
`fct_infreq(f3)`



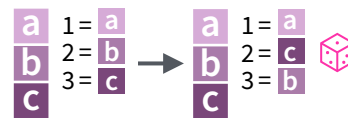
**fct\_inorder**(f, ordered = NA)  
Reorder levels by order in which they appear in the data.  
`fct_inorder(f2)`



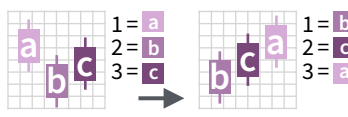
**fct\_rev**(f) Reverse level order.  
**f4** <- **factor**(c("a", "b", "c"))  
`fct_rev(f4)`



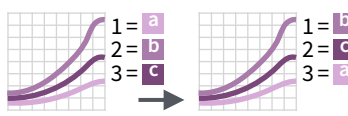
**fct\_shift**(f) Shift levels to left or right, wrapping around end. `fct_shift(f4)`



**fct\_shuffle**(f, n = 1L) Randomly permute order of factor levels. `fct_shuffle(f4)`

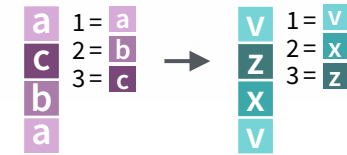


**fct\_reorder**(f, .x, .fun = median, ..., .desc = FALSE) Reorder levels by their relationship with another variable.  
`boxplot(PlantGrowth, weight ~ fct_reorder(group, weight))`

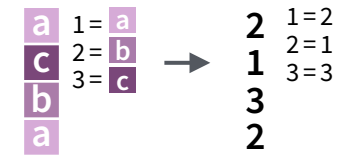


**fct\_reorder2**(f, .x, .y, .fun = last2, ..., .desc = TRUE) Reorder levels by their final values when plotted with two other variables.  
`ggplot(diamonds, aes(carat, price, color = fct_reorder2(color, carat, price))) + geom_smooth()`

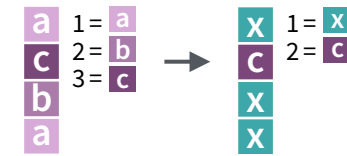
## Change the value of levels



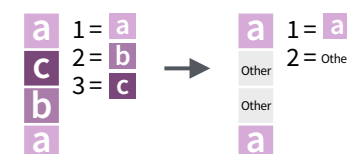
**fct\_recode**(f, ...) Manually change levels. Also **fct\_relabel**() which obeys purrr::map syntax to apply a function or expression to each level.  
`fct_recode(f, v = "a", x = "b", z = "c")`  
`fct_relabel(f, ~ paste0("x", .x))`



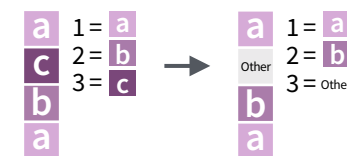
**fct\_anon**(f, prefix = "")  
Anonymize levels with random integers.  
`fct_anon(f)`



**fct\_collapse**(f, ..., other\_level = NULL)  
Collapse levels into manually defined groups.  
`fct_collapse(f, x = c("a", "b"))`



**fct\_lump\_min**(f, min, w = NULL, other\_level = "Other") Lumps together factors that appear fewer than min times. Also **fct\_lump\_n**(), **fct\_lump\_prop**(), and **fct\_lump\_lowfreq**().  
`fct_lump_min(f, min = 2)`



**fct\_other**(f, keep, drop, other\_level = "Other") Replace levels with "other".  
`fct_other(f, keep = c("a", "b"))`

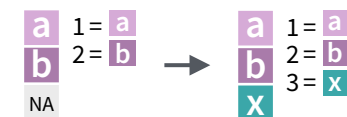
## Add or drop levels



**fct\_drop**(f, only) Drop unused levels.  
**f5** <- **factor**(c("a", "b"), c("a", "b", "x"))  
**f6** <- **fct\_drop**(f5)



**fct\_expand**(f, ...) Add levels to a factor.  
`fct_expand(f6, "x")`



**fct\_na\_value\_to\_level**(f, level = "(Missing)") Assigns a level to NAs to ensure they appear in plots, etc.  
**f7** <- **factor**(c("a", "b", NA))  
`fct_na_value_to_level(f7, level = "(Missing)")`