Data Transformation with dplyr::

list(suffix for column names = ~ list(fn(.x. fn args))

A guide to 37* different behaviours applied to one tibble (tb1) © R Data Berlin **MODIFY AGGREGATE** COUNT action "= (ren.+sel.+transm.+ column-related mut.+arr.+fil.+ summ. row-related **COLUMNS ROWS** +gr_by) * (_+_all+_at+_if)+ tallv() summarize function name (part) (ungr.+tal.+t_n+t_frac+sl.) function name (full) • group_by is effective on all functions but arrange, **POSITIONS VALUES GROUPINGS POSITIONS** NAMES unless using arrange_suffix(..., .by_group = TRUE) select_helper = starts_with(), ends_with(), contains(), matches(), num_range(), one_of(), group_by* everything(), or last_col() [from pkg tidyselect] keep keep keep keep keep keep all focal focal all all focal ungroup(` based on based on based on based on rank cut-off select test percentile cut-off transmute mutate rename arrange* their positions on values on values on values † %>% group_by(x) WHICH SUFFIX? %>% tallv() %>% ungroup() = filter top_frac() slice(top_n() %>% count(x) specific action common action on each focal on all focal **SYNTAX:** tbl %>% verb_suffix(., ..., .args) column columns [font style varies to tease appart placeholders from true R_commands] tbl %>% rename(new_name_columnX = old_name_columnX) focal focal tbl %>% select(name_columnX, new_name_columnY = old_name_columnY, select_helper ("text")) focal = all identified by identified by tbl %>% transmute(name_columnZ = fn_scalar_or_vectorized*(name_columnX)) positions test on values tbl %>% mutate(name_columnZ = fn_scalar_or_vectorized (name_columnX)) tlb %>% tallv()[†]: tlb %>% tallv(name numeric column, wt = weights)[†] tbl %>% arrange(name_columnX, desc(name_columnY)) tbl %>% filter(fn_test_vectorized(name_columnX), fn_test_vectorized(name_columnY)) if() at() tbl %>% top_n(n = number. name_columnX); tbl %>% top_frac(n = fraction, name_columnX) tbl %>% slice(row_indices) tbl %>% summarize(name_columnZ = fn_scalar (name_columnX)) ■ tbl %>% group_by(name_columnX, name_columnY) %>% verb_suffix() & tbl %>% ungroup() if the function fn does not return a scalar or a nrow-long vector, tbl %>% verb_all(.funs = fn fn fn_arg_name1 = fn_arg1, fn_arg_name2 = fn_arg2) # included below use list(fn()) to nest the output ** for multiple columns dispatch tbl %>% verb_at(.vars = indices_for_focal_columns OR .vars = vars(select_helper ("text")), use list(suffix_for_column_names = fn) or in case of *+ **

tbl %>% verb if(.predicate = logicals for each column.