Data Transformation with dplyr::

A guide to 35 different behaviours applied to one tibble (tb1) © R Data Berlin **MODIFY AGGREGATE** (rename+select+transmute+ action mutate+arrange+filter+ column-related row-related summarize+group_by) * **COLUMNS ROWS** function name (part) summarize (_+_all+_at+_if)+ function name (full) $(top_n+top_frac+slice) = 35$ group_by is effective on all functions but arrange, **POSITIONS GROUPINGS VALUES 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 based on based on based on based on rank cut-off select test percentile cut-off transmute mutate arrange* rename their positions on values on values on values WHICH SUFFIX? filter top_frac() slice(top_n() 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 positions test on values tbl %>% transmute(name_columnZ = fn_scalar_or_vectorized*(name_columnX)) tbl %>% mutate(name_columnZ = fn_scalar_or_vectorized*(name_columnX)) 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)) if the function fn does not return a scalar or a nrow-long vector, tbl %>% group_by(name_columnX, name_columnY) %>% verb_suffix() tbl %>% verb_all(.funs = fn + fn_arg_name1 = fn_arg1,fn_arg_name2 = fn_arg2) # included below use list(fn()) to nest the output for multiple columns dispatch use list(suffix_for_column_names = fn) or in case of • + • • tbl %>% verb_at(.vars = indices_for_focal_columns OR .vars = vars(select_helper ("text")), list(suffix for column names = ~ list(fn(.x. fn args)) tbl %>% verb_if(.predicate = logicals_for_each_column,)