**Recommended data cleaning steps (last updated 190728)**

**Pre-R cleaning steps:**  
\_ Rename data file for read\_csv, must end w/ "\_usable.csv".  
\_ Replace (Ctrl+H in Excel) "\*intergenic\*" with "intergenic" - more convenient for pipeline.  
\_ Rename cols (not all funcs have these cols): "gene", "population", "details" (from "Annotations", etc.)(if there are no "details" cols, copy the "gene" column & make that the "details" - sometimes "intergenic" info is in the "gene" col itself), "frequency", & "selective\_pressure" (all lowercase, snake case). All timepoint col names must be numeric. Too many non-numeric timepoint columns? Check below.

|  |  |
| --- | --- |
| **Data type** | **Steps** |
| **multiple\_wide** | - read\_csv(here("data\_in", "original & usable",..., "\_usable.csv")) - clean\_names(): Resulting names are unique & consist only of "\_", numbers, & letters. Argument: case = "snake" (this\_is\_snake\_case).  - tolower() - Change all col names to lowercase, in case someone forgot to do so beforehand. - grep() in the "gene" & "details" cols to get index of rows containing indicators of inappropriate entries.  - If there are any such values (length > 0), remove those rows with [-(index name), ]. - (Custom): names(dataset) <- gsub(pattern, ", names(dataset)): If there are too many non-numeric timepoints to rename manually, gsub() for the naming pattern of a particular dataset, replacing the non-numbers with " (no space). - transmute/select(): Pick out cols to keep for analyses. Use transmute() if you need to collapse several cols into 1 or rename cols. For multiple\_wide: gene, population, & gen cols.  - write\_csv(data\_name, here("data\_in", "for\_func", "data\_name.csv")) - multiple\_wide (template in R script). |
| **single\_wide** | 190728: Function under construction. |
| **multiple\_long** | - read\_csv(here("data\_in", "original & usable",..., "\_usable.csv")) - clean\_names(): Resulting names are unique & consist only of "\_", numbers, & letters. Argument: case = "snake" (this\_is\_snake\_case).  - tolower() - Change all col names to lowercase, in case someone forgot to do so beforehand. - grep() in the "gene" & "details" cols to get index of rows containing indicators of inappropriate entries.  - If there are any such values (length > 0), remove those rows with [-(index name), ]. - (Custom) If dataset is wide, use gather() to convert it to long. - transmute/select(): Pick out cols to keep for analyses. Use transmute() if you need to collapse several cols into 1 or rename cols. For multiple\_long: gene, population, selective\_pressure, frequency.  - write\_csv(data\_name, here("data\_in", "for\_func", "data\_name.csv")) - multiple\_long (template in R script). |
| **single\_long** | - read\_csv(here("data\_in", "original & usable",..., "\_usable.csv")) - clean\_names(): Resulting names are unique & consist only of "\_", numbers, & letters. Argument: case = "snake" (this\_is\_snake\_case).  - tolower() - Change all col names to lowercase, in case someone forgot to do so beforehand. - grep() in the "gene" & "details" cols to get index of rows containing indicators of inappropriate entries.  - If there are any such values (length > 0), remove those rows with [-(index name), ]. - If dataset is wide & single\_wide() doesn't work, use gather() to convert it to long. - transmute/select(): Pick out cols to keep for analyses. Use transmute() if you need to collapse several cols into 1 or rename cols. For single\_wide: gene, population, frequency.  - write\_csv(data\_name, here("data\_in", "for\_func", "data\_name.csv")) - multiple\_long (template in R script). |