**Data management statements**

1. Data movement or transport: from file->data frame , from data frame🡪file
   1. **import** dataset from <file url> //synonyms: extract/read
   2. **export** dataset to <file url> //synonyms :load/write
   3. let <dataframe name> **point** to <dataset object> //synonyms: assignment to variable/macro
   4. convert to tbl representation of dataframe – tbl\_df()
2. Single dataset manipulation
   1. Re-arrange/re-order dataset
      1. **Arrange** a dataset by sorting rows in **descending/ascending numerical/lexical order** of <column> [, then <column>]\*
      2. **Arrange** a dataset by ordering in provided sequence of **column names**
   2. Sub-setting dataset
      1. **Select** some columns of a dataset **by keeping/dropping** **named columns**
      2. **Filter** dataset by **keeping/dropping rows whose values satisfy a given condition**
   3. Extending dataset
      1. **Mutate**/alter dataset by **generati**ng a new column of values using expression/calculation/formula that includes existing columns
      2. **Transmutate** a table by **replacing** the values of an existing variable with the result of an expression
      3. **Regex: Find/replace/extract values from column- regex**
   4. Grouping dataset:
      1. **Group**/split dataset **by** values of of <column> [, then <column>]\*
      2. All commands e.g. mutate maybe used after grouping to create a grouped variant of the command i.e. **split** then **apply** a command to each subgroup then **append** results
   5. **Summarize** a dataset by **collapsing** column(s) of values to single value
   6. Change structure/layout of dataset: long & narrow / short & wide, stacked columns/unstacked columns, gathered columns/spread columns
3. Other:
   1. Dataset construction ab-initio// a data frame (list of vectors of same length , each vector is collection of values of same basic data type)
   2. \*Rename an existing variable as <new name>
   3. \*Code- Encode, recode,decode
   4. \*Label- define/apply
4. Binary-dataset manipulations
5. Display/browse vs transform data frame [update vs resize (expand vs collapse) vs reshape]
6. Mutating join: **Merg**e/match/**Join** tables and keep left+ matching /right+ matching /**inne**r i.e. matching / all- matching + non-matching rows
   1. Join/Merge table1 and table2 by a common variable and keep **inner**/matching rows
   2. Join/Merge table1 and table2 by a common variable and keep all rows in **left**\_table and matching rows in right table
   3. Join/Merge table1 and table2 by a common variable and keep all rows in **right** table and matching rows in right table
   4. Join/Merge table1 and table2 by a common variable and keep all rows in both tables
   5. Append/Bind table1 to table2
7. Filtering join: Semi/anti join- use one table to keep/drop matching rows in another table