A simple repository for my Diploma Thesis

- **Topic** - Ability bias in the returns to schooling: How large it is and why it matters
- **Author** - Bc. Petr Čala
- **Year of defense** - 2024
- **Supervisor** - doc. PhDr. Zuzana Havránková Ph.D.

Prerequisites:

1. Make sure that your working directory contains the following files:
    - `<NAME_OF_YOUR_DATA_FRAME>.csv` (modifiable below, default `data_set_master_thesis_cala.csv`)
    - `endo_kink_master_thesis_cala.R`
    - `main_master_thesis_cala.R`
    - `maive_master_thesis_cala.R`
    - `pretty_output_master_thesis_cala.R`
    - `selection_model_master_thesis_cala.R`
    - `source_master_thesis_cala.R`
    - `stem_method_master_thesis_cala.R`
    - `<NAME_OF_YOUR_VARIABLE_INFORMATION_FILE>.csv` (modifiable below, default `var_list_master_thesis_cala.csv`)
2. Make sure your data frame (`<NAME_OF_YOUR_DATA_FRAME>.csv`) contains **no missing values**. If there are any, the script **will not run**
3. The data frame must contain these columns (named exactly as listed below):

- **study_name** - Name of the study, such as *Einstein et al. (1935)*.
- **effect** - The main effect/estimate values. Ideally it should be a transformed effect, such as the partial correlation coefficient.
- **se** - standard error of the effect
- **t_stat** - t-statistic of the main effect. Can be calculated as a ratio of the effect and its standard error.
- **n_obs** - Number of observations associated with this estimate.
- **study_size** - Size of the study that the estimate comes from. In Excel, this can be easily computed as `=COUNTIF(<COL>:<COL>,<CELL>)` , where `<COL>` is the column with study names or study id's, and `<CELL>` is the cell in that column on the same row you want to calculate the study size on. Example: `=COUNTIF(B:B,B2)` . This calculates the study size of the study located in cell B2, assuming that the column `B` contains the study information.

4. In the file `<NAME_OF_YOUR_VARIABLE_INFORMATION_FILE>.csv` , input the list of variables you are using in your data frame, along with these parameters:

- **var_name** - Name of the variable exactly as it appears in the data frame columns. Must not include spaces and various special characters. Underscores are allowed.
- **var_name_verbose** - A descriptive form of the variable name. Needs not to limit to any subset of characters.
- **data_type** - Type of the data this variable holds. Can be only one type. Can be one of:
    - *int* - Integer. Any integer.
    - *category* - Categorical variable. Any string.
    - *float* - Float. Any number.
    - *dummy* - Dummy. Either 0 or 1.
    - *perc* - Percentage. Any value between 0 and 1, inclusive.
- **group_category** - Group of the variable. Group similar together, otherwise make a new group. Examples - dummies, gender, urban vs. rural, short-run vs. long-run
- **na_handling** - Specify how missing values should be handled for the variable. Can be one of:
    - *stop* - Do not allow missing values. Throw an error in case there is a missing value.
    - *mean* - Interpolate with the mean of the existing data.
    - *median* - Interpolate with the median of the existing data.
    - *equal* - Allow missing values. Use **only** for variables which whose values will be filled in automatically during preprocessing, meaning for which you can guarantee no missing values.
- **variable_summary** - Boolean. If `TRUE` , this variable will appear in the summary statistics table.
- **effect_sum_stats** - Boolean. If `TRUE` , this variable will appear in the effect summary statistics table.
- **equal** - Float. If set to any value, the effect summary statistics table will print out the statistics for the main effect of the data when subsetted to this variable equal to the specified value. If set to any value, can not set the `gtlt` column value.
- **gtlt** - One of "*median*", "*mean*", float. Similar to "equal", but if set to *median*/*mean*, will print out the statistics for the effect of the data when subsetted to values above/below the median value of this variable. If set to float, the subsetting breakpoint will be that value instead.
- **bma** - Boolean. If `TRUE` , this variable will be used in the Bayesian model averaging. Do NOT set all values of one variable group to `TRUE` . This would create a dummy trap.
- **to_log_for_bma** - Boolean. If `TRUE` , this variable will be converted to logarithm during the Bayesian model averaging.
- **bpe** - If set to any value, this value will be used when evaluating the best practice estimate.

How to Run

To run the code, follow these steps:

1. If you do not want to parametrize anything, simply open the file `pretty_output_master_thesis_cala.R` and run it. Note that you **must** keep the `.csv` file names under the same names they are available in the DSPACE folder, if you wish to run the script in this way (unless you parametrize it as instructed below). The original file names are `data_set_master_thesis_cala.csv` for the data frame and `var_list_master_thesis_cala.csv` for the master thesis.
2. If you wish to see into the code a bit more, and maybe parametrize several parts, such as which tests should run, and which not, then open the script `main_master_thesis_cala.R` . This script is split into *two parts*:

- **Customizable part**: Here you define the custom name of your data files, which parts of the script you want to run, and with which parameters.
- **Technical part**: The actual code, which should run without any problems, and all at once, if you specify the parameters correctly.

3. Go to the customizable part, and set the correct names for your data files. These should follow the sctructure outlined in the *Prerequisites* section.
4. Choose which parts of the code you want to run. Use `T` to indicate that a part should be run, and `F` to indicate that it should not.
5. Adjust the parameters with which to run the script. Find the `adjustable_parameters` vector, and inside, feel free to adjust the various numeric or boolean parameters as you see fit.
6. Run the code **ALL AT ONCE**, and see the results in the console, and in the *Plots* section.
7. If you wish to look under the hood of the code, see the file `source_master_thesis_cala.R` , which contains all the technical functions, preprocessing, and

validation, that is hidden in the main file.