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.

About

The reason for having a GitHub repository

The reason for keeping this repository is to allow for easier version tracking, and portable work. The repository is *not meant for direct cloning*. I may clean up the structure in the future to allow for this, but the main idea is to distribute the `Dist/` folder that contains the important `.R`, `.csv` and `.yaml` files (listed below). You can download the folder itself using steps described below. The rest of the repository is intended purely for my own research.

Project structure

All necessary files for result reproduction can be found within the `Dist/` folder. To easily clone this folder onto your computer, simply download the folder using third party tools such as download-directory and inputting this link, or by manually downloading each individual file. I advise for the former approach.

The folder contains these files:

- `Diploma Thesis Cala Returns To Education.zip` -> Main text of the thesis in *LaTeX* form.
- `data_set_master_thesis_cala.csv` -> Main data frame. Contains data of 115 studies with over 40 variables. All numeric results are derived from this file.
- `var_list_master_thesis_cala.csv` -> Data frame with information about individual variables. The scripts rely on this data frame to identify variable types, their usage in various parts of the analysis, etc.
- `user_parameters.yaml` -> Script customizable parameters. Can be modified either directly using a text editor or from within the `script_runner_master_thesis_cala.R` file. Contains parameters with file names, parts of the script to run, and parameters with which those parts should be run.
- `main_master_thesis_cala.R` -> Main script. Using the `user_parameters.yaml` file, call the desired methods with the specified parameters. Automatically handle package installation, working directory handling, temporary file creation.
- `source_master_thesis_cala.R` -> Source script with all the functions. Virtaully any function called from the main script is located here. Every function (hopefully) has a docstring explaining its *functionality* (pun intended). Navigate the script using function names.
- `script_runner_master_thesis_cala.R` -> Script for running the code in an aesthetic way. Here you can modify the parameters without having to edit the `.yaml` file. Automatically calls the whole main script, but you can modify which parts of it should run within the parameters. With this, just run the *script runner* script as a whole and witness magic happen (after you handle all the bugs).
- `elliot_master_thesis_cala.R` -> Source code for the p-hacking tests developed by Eliott et al. (2022).
- `endo_kink_master_thesis_cala.R` -> Source code for the Endogenous Kink method (Bom & Rachinger, 2019).
- `maive_master_thesis_cala.r` -> Source code for the MAIVE estimator method (Irsova et al., 2023).
- `selection_model_master_thesis_cala.R` -> Source code for the Selection model (Andrew & Kasy, 2019). Rewritten from STATA, should be quite robust.
- `stem_method_master_thesis_cala.R` -> Source code for the STEM method (Furukawa, 2019).
- `README.md` -> This README file.
- `README.pdf` -> The README file in a presentable format.

Prerequisites:

1. Make sure that your working directory contains all the files from the `Dist/` folder.
2. The scripts are set in a way that recognizes the file names just as they are distributed. However, if you wish to customize the file names, you may do so. In that case, make sure to change the file names either directly in the `user_parameters.yaml` using a text editor, or using the `script_runner_master_thesis_cala.R`, in which case you must run said script afterwards. This will automatically modify the `user_parameters.yaml` file, which is then loaded into the main script. **Do not to modify the names of the user parameter file and the script runner**.
3. Try to eliminate as many missing values in your data frame as you can. The script will automatically use interpolation for missing data, so that model averaging can run, but in case of many missing values, the results may be unstable.
4. The data frame must contain these columns (named exactly as listed below):

- **study_name** - Name of the study, such as *Einstein et al. (1935)*.
- **study_id** - ID of the study. Should be numeric and unique for each study.
- **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.
- **reg_df** - Degrees of freedom associated with this estimate.

5. In the file `var_list_master_thesis_cala.csv` (or your renamed version), 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.
    - *allow* - 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 `script_runner_master_thesis_cala.R` and run it. Note that for the script to run without error, you must keep the original file names, have a version of R that is compatible with the packages I am using, and keep the data structure outlined in the *Prerequisites* section.
2. If you wish to see into the code a bit more, or run it only in parts, then open the script `main_master_thesis_cala.R`. The script automatically loads the `user_parameters.yaml` file, so it is assumed you have modified the parameters to your desired form. Afterwards, you can run the script as usual either at once, or by parts.
3. 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.

Miscellaneous

I can not guarantee the code will run perfectly in case you attempt a replication using a different data set. In case you use the data sets provided, the only caveat might be package installation, otherwise the code should run smoothly. In case of using a custom data set, various combinations of data might break some methods. As such, use the project with caution, and I hope it will be useful in any way possible.