Education Research Naming Convention

1. *data_cleaning_component*: replication .ipynb files to obtain state-level .csv files in the *final_data_component* folder
    - .ipynb file naming convention: data_cleaning_**<outcome>**_**<state>**.ipynb

2. *descriptive_analysis*: (1) replication .ipynb file to obtain state-level .csv files used for descriptive analysis; and (2) state-level .csv files used for descriptive analysis
    - .ipynb file naming convention: descriptive_analysis_replication.ipynb
    - .csv file naming convention: **<outcome>**_by_**<interaction_variable>**.csv

3. *figure*: (1) replication .R file to obtain all included figures; and (2) .pdf figures
    - .R file naming convention: figure_replication.R
    - .pdf file naming convention: figure_**<number>**.pdf

4. *final_data_all_state*: (1) replication .ipynb file to obtain all-state combined. csv files used for regression analysis; and (2) all-state combined .csv files used for regression analysis
    - .ipynb file naming convention: final_data_all_state_replication.ipynb
    - .csv file naming convention: **<outcome>**_all_state.csv

5. *final_data_component*: state-level .csv files used to obtain all-state combined .csv files in the *final_data_all_state* folder
    - .csv file naming convention: **<outcome>**_**<state>**.csv

6. *python_regression*: replication .ipynb file to obtain regression analysis results in Python
    - .ipynb file naming convention: pyreg_**<regression type>**_**<outcome>**.ipynb

7. *stata_regression*: replication .do file to obtain regression analysis results in Stata
    - .do file naming convention: stata_**<regression type>**_**<outcome>**.do

8. *table*: (1) replication .R file to obtain all included tables; and (2) .pdf tables
    - .R file naming convention: table_replication.R
    - .pdf file naming convention: table_**<number>**.pdf

<outcome>: [mathpass, elapass, dropout]

<state>: [ca, il, tx, nyc]

<interaction_variable>: [year, black, hispanic, race, income, schooltype]

<number>: [1, 2, …]

<regression type>: [wls, logit]

Test Data File:

"schoolcode": code + <state_name> (e.g., 1007031california) / datatype: str

"districtcode": code + <state_name> (e.g., 21314california) / datatype: str

"countycode": code + <state_name> (e.g., 1313california) / datatype: str

"state": <state_name> (e.g., california) / datatype: str

"charter": 1 for charter school and 0 for public school / datatype: int

"year": [2015, 2016, 2017, 2018, 2019, 2021] – depending on which state / datatype: str

“mathpass/elapass”: 100 for 100% pass and 0 for 0% pass / datatype: float (2 decimals)

“hybridper”: 1 for 100% hybrid and 0 for 0% hybrid / datatype: float (4 decimals)

“virtualper”: 1 for 100% virtual and 0 for 0% virtual / datatype: float (4 decimals)

“schoolmode”: 1 for 100% virtual and 0 for 100% in-person / datatype: float (4 decimals)

“totaltested”: number of students taking annual state-level assessments / datatype: int

“white”: 100 for 100% white and 0 for 0% white / datatype: float (2 decimals)

“black”: 100 for 100% black and 0 for 0% black / datatype: float (2 decimals)

“hispanic”: 100 for 100% hispanic and 0 for 0% hispanic / datatype: float (2 decimals)

“lowincome”: 100 for 100% low-income and 0 for 0% low-income / datatype: float (2 decimals)

Drop-out Data File:

“schoolcode”: code + <state_name> (e.g., 1007031california) / datatype: str

“districtcode”: code + <state_name> (e.g., 21314california) / datatype: str

“countycode”: code + <state_name> (e.g., 1313california) / datatype: str

“state”: <state_name> (e.g., california) / datatype: str

“charter”: 1 for charter school and 0 for public school / datatype: int

“year”: [2015, 2016, 2017, 2018, 2019, 2021] – depending on which state / datatype: str

“dropout”: 100 for 100% drop-out and 0 for 0% drop-out / datatype: float (2 decimals)

“hybridper”: 1 for 100% hybrid and 0 for 0% hybrid / datatype: float (4 decimals)

“virtualper”: 1 for 100% virtual and 0 for 0% virtual / datatype: float (4 decimals)

“schoolmode”: 1 for 100% virtual and 0 for 100% in-person / datatype: float (4 decimals)

“totalenrolled”: number of enrolled students from grade 9 through 12 / datatype: int

“white”: 100 for 100% white and 0 for 0% white / datatype: float (2 decimals)

“black”: 100 for 100% black and 0 for 0% black / datatype: float (2 decimals)

“hispanic”: 100 for 100% hispanic and 0 for 0% hispanic / datatype: float (2 decimals)

“lowincome”: 100 for 100% low-income and 0 for 0% low-income / datatype: float (2 decimals)