



# STUDENT GROWTH PERCENTILES CALCULATIONS

RE-EVALUATING ANNUAL CENSUS TESTING

NOVEMBER 2022

ADAM VAN IWAARDEN LESLIE KENG EMMA KLUGMAN, HARVARD UNIVERSITY



National Center for the Improvement of Educational Assessment Dover, NH



# STUDENT GROWTH PERCENTILES CALCULATIONS

RE-EVALUATING ANNUAL CENSUS TESTING

#### Submitted to:

Bill Gates and the Walton Family The Gates-Walton Foundations November 2022

#### Author(s):

Adam Van Iwaarden, Leslie Keng and Emma Klugman

#### Acknowledgements:

We want to thank the State A Department of Education for their help in obtaining the data for this report, the **Gates-Walton Foundations** for their ongoing support and the Walmart greeter who told us to never (EVER) use Windows OS.

#### Suggested Citation:

The National Center for the Improvement of Educational Assessment (2022). Student Growth Percentiles Calculations. Submitted to The Gates-Walton Foundations, Anywhere, U.S.A..



# **REPORT CONTENTS**

Executive summary	1
Data cleaning and preparation	2
Load packages and custom functions	2
General data setup and cleaning	2
Additional variables for aggregated results	3
Summary and notes	3
Student Growth Percentiles Analysis	5
Load SGP package and modify sgPstateData	5
Simulation Condition 0	5
Simulation Condition 1b	7
Simulation Condition 1c	9
Simulation Condition 2	10
Simulation Condition 3	12
Growth and Achievement Aggregations	14
Condition specific summary tables	15
Achievement Improvement Aggregations	17
School level aggregations by demographics	18
References	23



## **EXECUTIVE SUMMARY**

This appendix contains the code used to prepare and format the data required for each condition of the "Re-evaluating the Efficiency and Efficacy of Annual Census Standardized Testing for Accountability Purposes" study. Each condition requires the data to be formatted in a consistent manner, and addition variables to be created. With the properly formatted and amended data, we then proceed with the Student Growth Percentiles (SGP) analyses and results aggregations.



## DATA CLEANING AND PREPARATION

For this simulation analysis we will be using the <code>sgpData\_LONG\_COVID</code> data from the <code>SGPData</code> package. It includes 7 years of annual assessment data in two content areas (ELA and Mathematics). As this data is typically used for testing and research purposes with the SGP (Betebenner et al. 2022) package, much of the data cleaning and formatting has already been done.

This section of the appendix assumes the user is operating with their working directory set to the state level directory (e.g., "./State\_A/".

```
# setwd("./State_A")
```

## LOAD PACKAGES AND CUSTOM FUNCTIONS.

The following R packages are required for the data source, cleaning and augmentation.

```
require(SGPdata)
require(data.table)
```

## **GENERAL DATA SETUP AND CLEANING**

This example dataset comes with a "built-in" impact in 2021 related to the pandemic as well as an unperturbed version - <code>SCALE\_SCORE\_without\_COVID\_IMPACT</code>. Here we will first subset the data to include only those years needed for the study, and then remove the perturbed score version and use the original scale score.

```
# First load and rename/remove SCALE_SCORE* variables included in the data
State_A_Data_LONG <- copy(SGPdata::sgpData_LONG_COVID)[YEAR < 2020]
State_A_Data_LONG[, SCALE_SCORE := NULL]
setnames(State_A_Data_LONG, "SCALE_SCORE_without_COVID_IMPACT", "SCALE_SCORE")</pre>
```

#### **NOTE TO LESLIE & EMMA**

We will need to either come to an agreement on the longitudinal data naming or rename according to the SGP package conventions. Here I rename the demographic variables to match the "analysis specification" documents and remove some of the variables we will not be looking at or using.

```
setnames(
  State_A_Data_LONG,
  c("ETHNICITY", "FREE_REDUCED_LUNCH_STATUS", "ELL_STATUS", "IEP_STATUS"),
  c("Race", "EconDis", "EL", "SWD")
)
```



```
State_A_Data_LONG[, Race := as.character(Race)]
State_A_Data_LONG[Race == "African American", Race := "Black"]
State_A_Data_LONG[, EconDis := gsub("Free Reduced Lunch", "FRL", EconDis)]
State_A_Data_LONG[,
    c("GENDER", "DISTRICT_NUMBER", "DISTRICT_NAME", "SCHOOL_NAME") := NULL
]
```

## ADDITIONAL VARIABLES FOR AGGREGATED RESULTS

A standardized score variable and an achievement proficiency indicator are required for school level aggregations, final analyses and results comparisons. The standardized scale score variable is scaled by each **year by subject by grade** test mean and standard deviation<sup>1</sup>.

*NOTE*: I am doing this here, but it could easily be done before the aggregation/summarization step. It is NOT required as any part of the growth analyses.

```
## Standardize SCALE_SCORE by CONTENT_AREA and GRADE using 2019 norms
State_A_Data_LONG[,
    Z_SCORE := scale(SCALE_SCORE),
    by = c("YEAR", "CONTENT_AREA", "GRADE")
]
```

A simple '1/0' binary indicator for proficiency will allow us to compute descriptive statistics (e.g., percent proficient) easily and consistently across all states included in the report.

```
## Proficient/Not (1/0) binary indicator.
State_A_Data_LONG[,
    PROFICIENCY := fcase(
        ACHIEVEMENT_LEVEL %in% c("Partially Proficient", "Unsatisfactory"), OL,
        ACHIEVEMENT_LEVEL %in% c("Advanced", "Proficient"), 1L
)

State_A_Data_LONG[,
    Z_PROFICIENCY := scale(PROFICIENCY),
    by = c("YEAR", "CONTENT_AREA", "GRADE")
]
```

 $<sup>^{1}</sup>$  The original  ${\tt SCALE\_SCORE}$  variable is used in the SGP calculations.



## **SUMMARY AND NOTES**

- "State A" uses the 2016 to 2019 subset of the <code>sgpData\_LONG\_covid</code> dataset from the <code>sgpData\_package</code>.
  - The "original", unperturbed version of the scaled score is retained.
- A standardized scale score variable is added (scaled by unique grade, content area and annual assessment).
- A binary indicator variable for proficiency status is added.



## STUDENT GROWTH PERCENTILES ANALYSIS

This section presents and explains the code used to conduct the Student Growth Percentiles (SGP) analyses. Each simulated testing condition is applied via the R code to the same set of data, thus only producing growth measures for the appropriate grades, content areas and years. At the end of each condition-specific analysis, the SGP variable is renamed to indicate the simulated condition before proceeding to the next SGP analysis step. Only cohort-referenced SGPs are calculated (SGP projections and targets are omitted). The goal of this step is simply to create growth percentiles and merge them into the longitudinal data before aggregation and investigation of the impact non-census testing has on school accountability measures.

## LOAD SGP PACKAGE AND MODIFY SGPSTATEDATA

The SGP package is required for all growth percentile analyses.

```
require(SGP)
```

We will use the assessment meta-data from the "Demonstration\_COVID" (abbreviated "DEMO\_COVID") dataset stored in the SGPstateData object. This meta-data is required to use various functions in the SGP package.

```
SGPstateData[["State_A"]] <- SGPstateData[["DEMO_COVID"]]
SGPstateData[["State_A"]][["Growth"]][["Levels"]] <-
SGPstateData[["State_A"]][["Growth"]][["Cutscores"]] <-
SGPstateData[["State_A"]][["SGP_Configuration"]][["percentile.cuts"]] <-
NULL
```

## SIMULATION CONDITION 0

In this simulation condition, we want to replicate the base condition of typical census-level testing with the base data set. Growth analyses will include grades 4 to 8, with consecutive-year assessment patterns. Students with a valid score from the previous year and grade level in their historical data will be included in the growth calculations and receive a SGP. Up to two prior scores will be used as available in the data.

#### LOAD AND COMBINE SGP CONFIG SCRIPTS

The growth calculation functions of the SGP software package allow users to manually specify which test progressions to run. That is, we can define the unique **year-by-grade-by-content area** cohorts of students included in each analysis.

As an example, the 2019 ELA analyses/cohorts are specified with this code:



All configurations are housed in condition specific R code scripts. Here we read these in and combine them into a single list object, state.a.config, that will be supplied to the abcSGP function.

### **CALCULATE CONDITION 0 SGPS**

We use the abcSGP function from the SGP package to produce 2018 and 2019 student growth percentiles. We provide the function with the longitudinal data that was previously cleaned and formatted, as well as the list of analysis configurations and other relevant arguments to tailor the analyses to our specifications.

The SGP analysis section of the appendix assumes the user is operating with their working directory set to "./Condition O".

```
setwd("./Condition_0")
State_A_SGP <-
   abcSGP(
      sgp_object = State_A_Data_LONG,
      state = "State_A",
      steps = c("prepareSGP", "analyzeSGP", "combineSGP"),
      sgp.config = state.a.config,
      sgp.percentiles = TRUE,
      sgp.projections = FALSE,
      sgp.projections.lagged = FALSE,
      sgp.percentiles.baseline = FALSE,</pre>
```



#### RE-NAME AND REMOVE THE SGP VARIABLES AS NECESSARY

In order to keep all growth results in the same longitudinal dataset, we will add a Cnd\_0 tag to growth related variables of interest. Extraneous variables will be removed as well before moving on to the next simulation condition.

## **SIMULATION CONDITION 1B**

In this condition, students test twice per grade span (elementary and middle grades) in both subjects. Tests are administered every year in 3rd, 5th, 6th and 8th grades. Subsequently, all growth analyses will use a single prior score, and can be done with a either consecutive- or skipped-year assessment patterns.

#### LOAD AND COMBINE SGP CONFIG SCRIPTS

In order to avoid errors in specification of our analysis configurations, we first remove all previous configuration related objects before reading in the code for condition 1b and proceeding as before. Unlike the other simulation conditions, 1b requires *both* consecutive- and skipped-year configuration scripts.

The 2019 ELA configuration code is provided here as an example and for comparison with the code provided above for condition 0:

```
ELA_2019.config <- list(
    ELA.SKIP.2019 = list(
    sgp.content.areas = rep("ELA", 2),</pre>
```



```
rm(list = grep(".config", ls(), value = TRUE))
source("SGP_CONFIG/Condition_lb.R")

state.a.config <-
    c(ELA_2019.config,
        MATHEMATICS_2019.config,
        ELA_2018.config,
        MATHEMATICS_2018.config
)</pre>
```

#### **CALCULATE CONDITION 1B SGPS**

We again use the abcSGP function to compute the student growth percentiles for this simulation condition. Here we use the data with results from condition 0. The updated list of analysis configurations is now provided, and all other relevant arguments remain the same.

```
setwd("./Condition 1b")
State A SGP <-
   abcSGP(
        sgp object = State A Data LONG,
        state = "State A",
        steps = c("prepareSGP", "analyzeSGP", "combineSGP"),
        sqp.config = state.a.config,
        sgp.percentiles = TRUE,
        sgp.projections = FALSE,
        sgp.projections.lagged = FALSE,
        sqp.percentiles.baseline = FALSE,
        sgp.projections.baseline = FALSE,
        sgp.projections.lagged.baseline = FALSE,
        simulate.sgps = FALSE,
        parallel.config = list(
            BACKEND = "PARALLEL",
            WORKERS = parallel::detectCores(logical = FALSE)
```



## **SIMULATION CONDITION 1C**

In this condition, students alternate testing in each subject across grade levels. In this simulation, students in grades 3, 5, and 7 take ELA and students in grades 4, 6, 7 take mathematics each year. As with condition 1b, all growth analyses will be conditioned on a single prior score, but only skipped-year assessment patterns can be analyzed.

#### LOAD AND COMBINE SGP CONFIG SCRIPTS

We again remove all previous configuration related objects before reading in the condition 1c course progression code. The 2019 ELA configurations are once again provided here for comparison with other simulation conditions.

The mathematics configurations are nearly identical to the ELA code, with the exception of the sgp.grade.sequences element, which specifies the grades 4 to 6 and grades 6 to 8 progressions. Note that this particular testing pattern means traditional elementary schools will only have growth measures for grade 5 ELA, while traditional middle schools will have growth indicators in all three grades and both content areas. The only contribution mathematics makes to a school's accountability calculation is through grade 4 proficiency (status).

```
rm(list = grep(".config", ls(), value = TRUE))
source("SGP_CONFIG/Condition_1c.R")
```



```
state.a.config <-
    c(ELA_2019.config,
        MATHEMATICS_2019.config,
        ELA_2018.config,
        MATHEMATICS_2018.config
)</pre>
```

#### **CALCULATE CONDITION 1C SGPS**

The call to theabcSGP function here is identical to that made for conditions 1b and 2. The data object State\_A\_Data\_LONG now includes the results from conditions 0 and 1b, and the configurations have been updated.

```
setwd("./Condition 1c")
State A SGP <-
    abcSGP(
        sgp object = State A Data LONG,
        state = "State A",
        steps = c("prepareSGP", "analyzeSGP", "combineSGP"),
        sgp.config = state.a.config,
        sqp.percentiles = TRUE,
        sgp.projections = FALSE,
        sgp.projections.lagged = FALSE,
        sgp.percentiles.baseline = FALSE,
        sgp.projections.baseline = FALSE,
        sgp.projections.lagged.baseline = FALSE,
        simulate.sgps = FALSE,
        parallel.config = list(
            BACKEND = "PARALLEL",
            WORKERS = parallel::detectCores(logical = FALSE)
    )
setwd("..")
rm(State A Data LONG)
State A Data LONG <- copy(State A SGP@Data)
setnames(x = State A Data LONG, old = "SGP", new = "SGP Cnd 1c")
State A Data LONG[,
    c("SGP NORM GROUP", "SGP NORM GROUP SCALE SCORES",
      "SCALE SCORE PRIOR", "SCALE SCORE PRIOR STANDARDIZED"
   ) := NULL
]
```

## **SIMULATION CONDITION 2**

In this condition, all students are tested every two years in each grade and subject on the state's assessments. There are two instances of this condition to simulate:

- Testing only occurs in even years (e.g., 2016, 2018, etc.)
- Testing only occurs in even years (e.g., 2017, 2019, etc.)

In both instances, in a year that testing occurs, all students are tested in every grade and subject. As with condition 1c, all growth analyses will be conditioned on a single prior score with skipped-year patterns.

#### LOAD AND COMBINE SGP CONFIG SCRIPTS

We again remove all previous configuration related objects before reading in the condition 2 course progression code. The 2019 ELA configurations are once again provided here for comparison with other simulation conditions.

```
rm(list = grep(".config", ls(), value = TRUE))
source("SGP_CONFIG/Condition_2.R")

state.a.config <-
    c(ELA_2019.config,
        MATHEMATICS_2019.config,
        ELA_2018.config,
        MATHEMATICS_2018.config
)</pre>
```

#### **CALCULATE CONDITION 2 SGPS**

The call to theabcSGP function here is identical to that made for conditions 1b and 1c. The data object State\_A\_Data\_LONG now includes the results from conditions 0 through 1c, and the configuration object, state.a.config, has been updated.



```
setwd("Condition 2")
State A SGP <-
    abcSGP(
        sgp object = State A Data LONG,
        state = "State A",
        steps = c("prepareSGP", "analyzeSGP", "combineSGP"),
        sgp.config = state.a.config,
        sgp.percentiles = TRUE,
        sgp.projections = FALSE,
        sqp.projections.lagged = FALSE,
        sgp.percentiles.baseline = FALSE,
        sgp.projections.baseline = FALSE,
        sgp.projections.lagged.baseline = FALSE,
        simulate.sqps = FALSE,
        parallel.config = list(
            BACKEND = "PARALLEL",
            WORKERS = parallel::detectCores(logical = FALSE)
    )
setwd("..")
rm(State A Data LONG)
State A Data LONG <- copy(State A SGP@Data)
setnames (x = State A Data LONG, old = "SGP", new = "SGP Cnd 2")
State A Data LONG[,
    c ("SGP NORM GROUP", "SGP NORM GROUP SCALE SCORES",
      "SCALE SCORE PRIOR", "SCALE SCORE PRIOR STANDARDIZED"
   ) := NULL
1
```

## **SIMULATION CONDITION 3**

In this condition, all students are tested every two years at specific grade and subject on the state's assessments. As with Condition 2, there are two instances of this condition to simulate:

- Testing only occurs in even years (e.g., 2016, 2018, etc.)
- Testing only occurs in even years (e.g., 2017, 2019, etc.)

In both instances, when testing occurs, students are tested specific grades in both subject areas. As with condition 1c, all growth analyses will be conditioned on a single prior score with skipped-year patterns. ### SGP config scripts

The pattern of testing for this condition is identical to that of condition 1b, with the exception of skipping years. This means that we have already calculated the SGPs for these patterns and do not need to reanalyze the data to get these results. Instead we can simply copy the results from



simulation condition 1b that use the skipped-year progressions (i.e. results for grades 5 and 8, but not 6th grade).

For the sake of completeness, however, the 2019 ELA configurations for this condition would be a subset of the condition 1b code, such as this:

#### **USE CONDITION 1B GROWTH FOR CONDITION 3**

Here we will simply copy the results from condition 1b to a new variable for condition 3. The grade 6 SGPs, which were consecutive-year (grade 5 to grade 6) will be omitted.

```
State_A_Data_LONG[
    GRADE %in% c(5, 8),
    SGP_Cnd_3 := SGP_Cnd_1b
]

if (!dir.exists("Data")) dir.create("Data")
save("State_A_Data_LONG", file = "Data/State_A_Data_LONG.rda")
```

## **GROWTH AND ACHIEVEMENT AGGREGATIONS**

To simplify the analysis and enable comparisons of results across participating states, we plan to simulate a standard "prototype" accountability model with the following features.

#### Reporting

- The minimum n-count for computing scores for schools and disaggregated student groups is varied depending on the simulated condition.
- The disaggregated student groups should include economically disadvantaged students, students from racial and ethnic groups, children with disabilities, and English learners, as long as they meet the minimum n-count threshold in the simulated condition.

#### **Indicators**

- **Academic achievement** is the percentage of students in the school meeting the proficiency in ELA and mathematics (as defined by the 'Proficient' cut score on the statewide assessment).
- The computation of the ELA and math proficiency rates are adjusted if a school or student group does not have at least 95% participation.
- For the other academic indicator, we apply the following rules:
  - If student-level academic growth (consecutive-year or skip-year) can be computed, then we will use it for this indicator. For consistency, we will calculate student growth percentiles (SGPs) using the student-level assessment data.
  - If student-level academic growth (consecutive-year or skip-year) cannot be computed, then we will use an improvement measure, defined as the change in average scale scores for each grade-level subject area test between administrations for the school or student group.

#### **Summative Rating Computation**

All indicator scores are standardized by transforming into z-scores. Use the following means and standard deviations (SD) for the z-score computations (of all schools and student groups):

- Academic achievement
  - Mean: mean student-level proficiency rate for the focus year
  - SD: student-level proficiency rate SD for the focus year
  - standardized by year, subject and grade.
- Other academic indicator **growth** 
  - SGPs, being percentiles, can be converted directly to a standardized metric<sup>2</sup>.
- Other academic indicator **improvement** 
  - Mean: mean student-level scale score changes for the focus year, calculated separately for each grade level and subject area

<sup>&</sup>lt;sup>2</sup> Ex. in R: qnorm (c(1, 10, 25, 50, 75, 90, 99)/100) gives the z-score for the 1<sup>st</sup>, 10<sup>th</sup>, 25<sup>th</sup>, ... etc., percentiles. For more on mapping percentiles on to the standard-normal distribution, see this site



- SD: SD of student-level scale scores for the focus year, calculated separately for each grade level and subject area
- standardized by year, subject and grade.
- Graduation rates, progress in ELP, and SQSS
  - Mean: mean school-level indicator scores for the focus year
  - SD: SD of school-level indicator scores for the focus year

## **CONDITION SPECIFIC SUMMARY TABLES**

**NOTE TO LESLIE & EMMA** These tables and aggregations (as well as my variable additions such as Z\_PROFICIENCY and Z\_SCORE) are my first attempts to both interpret and implement what I've read in the "Analysis Specification" document. Since I have some extensive experience in aggregating growth and achievement data, this is how I would approach it at this early stage...

The following is an example of a preliminary school-level aggregation table for condition 0. Each condition will have a similar table, generally with only the appropriate SGP variable substituted for SGP Cnd 0 and changes to the inclusion criteria (YEAR, GRADE and sometimes CONTENT AREA).

```
sch_summary_cnd_0 <-
State_A_Data_LONG[
    YEAR %in% c(2018, 2019) &
    GRADE %in% 3:8,
    .(TotalN = .N,
        ProfN = sum(PROFICIENCY==1L),
        GrowthN = sum(!is.na(SGP_Cnd_0)),
        MGP = round(mean(SGP_Cnd_0, na.rm = TRUE), 1),
        Mean_Score = round(mean(Z_SCORE, na.rm = TRUE), 2),
        Pcnt_Prof = round(mean(PROFICIENCY, na.rm = TRUE), 3)*100,
        Z_Status = round(mean(Z_PROFICIENCY, na.rm = TRUE), 3),
        Z_Growth = round(mean(qnorm(SGP_Cnd_0/100), na.rm = TRUE), 3)
),
        keyby = c("YEAR", "CONTENT_AREA", "SCHOOL_NUMBER")
]</pre>
```

You may notice that there are more summary calculations than what will be used (e.g., percent proficient and mean standardized scale scores). Those are included for our review - so we can easily see what a z-score, of for example 0.5, corresponds to in the actual percent proficient or mean SGP. Here are two schools from the condition 0 table:

```
sch_summary_cnd_0[SCHOOL_NUMBER %in% c(1001, 3801)] |>
   setkey(SCHOOL_NUMBER) |> print()
```

## YEAR CONTENT AREA SCHOOL NUMBER TotalN ProfN GrowthN MGP Mean Score Pcnt P



##	2018	ELA								
			1001	120	109	73	49.7	0.62	90.8	0.511
##	2018	MATHEMATICS	1001	120	107	73	61.5	0.79	89.2	0.553
##	2019	ELA	1001	162	146	90	54.0	0.66	90.1	0.501
##	2019	MATHEMATICS	1001	162	145	90	55.1	0.77	89.5	0.565
##	2018	ELA	3801	170	44	103	43.4	-0.89	25.9	-0.870
##	2018	MATHEMATICS	3801	170	31	103	40.7	-0.91	18.2	-0.855
##	2019	ELA	3801	151	67	101	53.4	-0.61	44.4	-0.467
##	2019	MATHEMATICS	3801	151	52	101	54.3	-0.66	34.4	-0.531

At some point we will probably want to combine the condition aggregations into a single table so that we can do direct condition comparisons. We can also clean up some of the extra descriptive statistics, re-order the columns or anything else.

```
# Combine all condition-specific tables into one:
composite summary <-
   sch summary cnd 0[
   sch summary cnd 1b][
   sch summary cnd 1c][
   sch summary cnd 2][
   sch summary cnd 3]
# Remove extraneous aggregations:
composite summary[,
   grep("Pcnt Prof|Mean ", names(composite summary), value = TRUE) :=
     NULL
1
# School No. '1001' - changes in Growth N count
composite summary[SCHOOL NUMBER == 1001,
 c("YEAR", "CONTENT AREA",
   grep("GrowthN", names(composite summary), value = TRUE)
 ), with = FALSE
```

```
## YEAR CONTENT AREA GrowthN GrowthN 1b GrowthN 1c GrowthN 2 GrowthN 3
## 2018
               ELA
                       73
                                  40
                                           40
                                                    40
                                                              40
## 2018 MATHEMATICS
                        73
                                  40
                                            0
                                                     40
                                                              40
## 2019
                       90
                                  38
                                           38
                                                     38
                                                              38
               ELA
                                           0
## 2019 MATHEMATICS
                       90
                                 38
                                                     38
                                                              38
```



```
grep("Z_Growth", names(composite_summary), value = TRUE) # just z-growth
    ), with = FALSE
]
```

```
## YEAR CONTENT AREA Z Growth Z Growth 1b Z Growth 1c Z Growth 2 Z Growth 3
## 2018
                ELA -0.045
                                0.214
                                           0.214
                                                     0.214
                                                                0.214
## 2018 MATHEMATICS
                     0.386
                                 0.459
                                              NaN
                                                      0.459
                                                                0.459
## 2019
                      0.151
                                -0.128
                                            -0.128
                                                      -0.128
                                                                -0.128
                ELA
## 2019 MATHEMATICS
                     0.188
                                 0.130
                                              NaN
                                                      0.130
                                                                 0.130
```

```
YEAR CONTENT AREA Z Status Z Status 1b Z Status 1c Z Status 2 Z Status 3
                     0.511
                ELA
## 2018
                                0.533
                                             0.533
                                                       0.511
## 2018 MATHEMATICS
                      0.553
                                  0.513
                                             0.657
                                                       0.553
                                                                 0.513
## 2019
                      0.501
                                  0.493
                                             0.493
                                                       0.501
                                                                 0.493
## 2019 MATHEMATICS 0.565
                                  0.536
                                             0.622
                                                       0.565
                                                                 0.536
```

## **ACHIEVEMENT IMPROVEMENT AGGREGATIONS**

The simulation condition 1a does not allow for growth calculations and will instead use an indicator of status improvement. This **improvement** measure is defined as the change in average scale scores for each grade-level content area test between administrations for the school or student group.

For this aggregation we will create status summaries in a similar way as the other conditions, but include all available years. Lagged values are then created and the change scores calculated.

```
sch_summary_cnd_1a <-
State_A_Data_LONG[
    GRADE %in% c(5, 8),
    .(TotalN = .N,
         Mean_Score = round(mean(Z_SCORE, na.rm = TRUE), 2),
         Z_Status = round(mean(Z_PROFICIENCY, na.rm = TRUE), 3)
    ),
        keyby = c("YEAR", "CONTENT_AREA", "GRADE", "SCHOOL_NUMBER")
]</pre>
# Create lagged variables (1 year lag):
```



```
sch summary cnd 1a,
 c("SCHOOL NUMBER", "CONTENT AREA", "YEAR", "GRADE")
cfaTools::getShiftedValues(
   sch summary cnd 1a,
    shift group = c("SCHOOL NUMBER", "CONTENT AREA"),
    shift variable = c("TotalN", "Mean Score", "Z Status"),
    shift amount = 1L
)
# Subset the data for the two focus years:
sch summary cnd 1a <-
    sch summary cnd 1a[YEAR %in% c(2018, 2019)]
# Calculate changes (current year minus 1 year lag)
sch summary cnd 1a[,
    TotalN Change := TotalN - TotalN LAG 1
   Mean Score Change := Mean Score - Mean Score LAG 1
][,
    Z Status Change := Z Status - Z Status LAG 1
]
```

Here is our example school's improvement numbers

```
sch_summary_cnd_1a[SCHOOL_NUMBER == 1001,
    c(key(sch_summary_cnd_1a)[-1],
        "TotalN_Change", "Mean_Score_Change", "Z_Status_Change"
    ), with = FALSE
]
```

```
## CONTENT AREA YEAR GRADE TotalN Change Mean Score Change Z Status Change
         ELA 2018 5
##
                               24
                                            0.56
                                                        0.307
          ELA 2019
                   5
                               -3
##
                                           -0.20
                                                       -0.113
## MATHEMATICS 2018
                   5
                               24
                                            0.54
                                                        0.059
## MATHEMATICS 2019 5
                               -3
                                           -0.41
                                                        0.164
```

## SCHOOL LEVEL AGGREGATIONS BY DEMOGRAPHICS

Adding in the demographic variables is a simple addition of the variable of interest into the keyby argument of the data.table aggregation. Since we are going to be doing this numerous times, it might be smart to create a custom function to create these tables, rather than copying the code for each use case.



```
schoolAggrGator =
 function(
   data table,
   growth.var,
   groups = c("YEAR", "CONTENT AREA", "SCHOOL NUMBER")
   data table[,
      # the list of summaries can be reduced/increased/amended as needed:
      .(TotalN = .N,
        ProfN = sum(PROFICIENCY==1L),
        GrowthN = sum(!is.na(get(growth.var))),
       MGP = round(mean(get(growth.var), na.rm = TRUE), 1),
       Mean Score = round(mean(Z SCORE, na.rm = TRUE), 2),
        # Pcnt Prof = round(mean(PROFICIENCY, na.rm = TRUE), 3)*100,
        Z Status = round(mean(Z PROFICIENCY, na.rm = TRUE), 3),
        Z Growth = round(mean(qnorm(get(growth.var)/100), na.rm = TRUE), 3)
     ),
     keyby = groups
   ][]
 }
```

Our original "base" condition table can be reproduced now with this call to our function:

```
schoolAggrGator(
  data_table =
    State_A_Data_LONG[YEAR %in% c(2018, 2019) & GRADE %in% 3:8,],
  growth.var = "SGP_Cnd_0",
)
```

Our function used for demographics (Economic Disadvantage):

```
schoolAggrGator(
   data_table =
      State_A_Data_LONG[YEAR %in% c(2018, 2019) & GRADE %in% 3:8,],
   growth.var = "SGP_Cnd_0",
   groups = c("YEAR", "CONTENT_AREA", "SCHOOL_NUMBER", "EconDis")
)
```

In order to do all the demographic summaries at once, we can combine calls to the function (rather than creating separate tables and THEN combining):

```
demog_cond_0 <-
   rbindlist(
   list(</pre>
```



```
data table =
            State A Data LONG[YEAR %in% c(2018, 2019) & GRADE %in% 3:8,],
          growth.var = "SGP Cnd 0",
          groups = c("YEAR", "CONTENT AREA", "SCHOOL NUMBER", "Race")
        ) |> setnames("Race", "Group"),
        schoolAggrGator(
          data table =
            State A Data LONG[YEAR %in% c(2018, 2019) & GRADE %in% 3:8,],
          growth.var = "SGP Cnd 0",
          groups = c("YEAR", "CONTENT AREA", "SCHOOL NUMBER", "EconDis")
        ) |> setnames("EconDis", "Group"),
        schoolAggrGator(
          data table =
            State A Data LONG[YEAR %in% c(2018, 2019) & GRADE %in% 3:8,],
          growth.var = "SGP Cnd 0",
          groups = c("YEAR", "CONTENT AREA", "SCHOOL NUMBER", "EL")
        ) |> setnames("EL", "Group"),
        schoolAggrGator(
          data table =
            State A Data LONG[YEAR %in% c(2018, 2019) & GRADE %in% 3:8,],
          growth.var = "SGP Cnd 0",
          groups = c("YEAR", "CONTENT AREA", "SCHOOL NUMBER", "SWD")
        ) |> setnames("SWD", "Group")
      )
    )
```

Here a subset of the output from the example school:

```
demog_cond_0[
   SCHOOL_NUMBER == 1001 & YEAR == 2019
][,
   c("YEAR", "SCHOOL_NUMBER", "MGP", "Mean_Score") := NULL
][]
```

```
CONTENT AREA
             Group TotalN ProfN GrowthN Z Status Z Growth
##
         ELA
              Asian
                     30
                           27
                                 21 0.499 0.225
##
         ELA
              Black
                          10
                                     0.357 -0.274
##
         ELA Hispanic
                     30
                           24
                                18
                                     0.287
                                          -0.046
         ELA Other
                     10
                           9
                                 7 0.499 0.559
##
                     80 76
##
         ELA White
                                36 0.604
                                          0.222
##
             Asian
                                21
  MATHEMATICS
                     30
                         23
                                   0.316
                                            0.064
  MATHEMATICS
             Black
                     12 10
                                8 0.445 -0.117
                     30
                                18
##
   MATHEMATICS Hispanic
                         24
                                    0.379 -0.082
  MATHEMATICS Other
                     10
                          10
                                 7 0.800 0.400
## MATHEMATICS White
                     80 78
                                36 0.717 0.423
         ELA FRL: No
                                83
##
                     146 133
                                   0.522
                                           0.140
         ELA FRL: Yes
                     16
                          13
                                 7 0.312 0.280
  MATHEMATICS FRL: No
##
                     146
                         132
                                83
                                     0.586
                                            0.166
```



```
7 0.377 0.455
   MATHEMATICS FRL: Yes 16 13
         ELA ELL: No 159
                         143
##
                                89 0.497 0.143
         ELA ELL: Yes 3 3
##
                                1 0.709 0.878
  MATHEMATICS ELL: No 159 142
##
                               89 0.562 0.199
##
  MATHEMATICS ELL: Yes
                    3 3
                                1 0.753 -0.739
        ELA IEP: No 152 146
                               83 0.627 0.215
         ELA IEP: Yes 10 0
                                7 -1.408 -0.608
##
## MATHEMATICS IEP: No 152 143
                                  0.660 0.265
                               83
  MATHEMATICS IEP: Yes 10
                                7
                           2
                                    -0.869 -0.725
## CONTENT AREA Group TotalN ProfN GrowthN Z Status Z Growth
```

Another example of how to combine the aggregations along with output from a different school:

```
demog_cond0 <-
lapply(
    c("Race", "EconDis", "EL", "SWD"),
    \( (f) \) {
    schoolAggrGator(
        data_table =
            State_A_Data_LONG[YEAR %in% c(2018, 2019) & GRADE %in% 3:8, ],
            growth.var = "SGP_Cnd_O",
                 groups = c("YEAR", "CONTENT_AREA", "SCHOOL_NUMBER", f)
        ) |> setnames(f, "Group")
    }
    ) |> rbindlist()

demog_cond0[
    SCHOOL_NUMBER == 3801 & YEAR == 2019
][,
    c("YEAR", "SCHOOL_NUMBER", "MGP", "Mean_Score") := NULL
][]
```

##	CONTENT_AREA	Group	TotalN	ProfN	GrowthN	Z_Status	Z_Growth
##	ELA	Asian	13	5	12	-0.590	0.111
##	ELA	Black	4	2	2	-0.350	-2.054
##	ELA	Hispanic	36	23	22	-0.053	0.870
##	ELA	Other	8	4	6	-0.347	1.012
##	ELA	White	90	33	59	-0.630	-0.137
##	MATHEMATICS	Asian	13	4	12	-0.567	0.184
##	MATHEMATICS	Black	4	2	2	-0.276	-1.555
##	MATHEMATICS	Hispanic	36	19	22	-0.138	0.489
##	MATHEMATICS	Other	8	3	6	-0.426	0.520
##	MATHEMATICS	White	90	24	59	-0.704	-0.033
##	ELA	FRL: No	19	13	9	0.043	0.751
##	ELA	FRL: Yes	132	54	92	-0.540	0.082
##	MATHEMATICS	FRL: No	19	11	9	-0.042	0.274
##	MATHEMATICS	FRL: Yes	132	41	92	-0.601	0.093
##	ELA	ELL: No	109	61	69	-0.221	0.134



```
##
           ELA ELL: Yes
                          42
                                        32 -1.104
                                                    0.159
                                6
##
    MATHEMATICS ELL: No
                          109
                                 46
                                        69
                                            -0.372
                                                      0.276
##
                          42
    MATHEMATICS ELL: Yes
                                        32 -0.944 -0.251
                                 6
           ELA IEP: No
                                             -0.379
                                                      0.322
##
                          134
                                 65
                                        86
##
           ELA IEP: Yes
                                 2
                                        15
                                             -1.158
                                                     -0.888
                          17
##
   MATHEMATICS IEP: No
                          134
                                 52
                                        86
                                             -0.439
                                                      0.213
##
    MATHEMATICS IEP: Yes
                          17
                                  0
                                        15
                                             -1.260
                                                     -0.485
   CONTENT AREA Group TotalN ProfN GrowthN Z Status Z Growth
```

Either of the demographic aggregation and combination code chunks above can be run for each of the SGP\_Cnd\* growth fields. At that point, we can then combine those objects in a wide format (similar to what was done for the composite\_summary object - this would require re-naming the aggregate variables), stacked into a long format (with an added "Condition" variable for each table - probably what I would do) or written to separate .csv files as described in the specification doc.



## **REFERENCES**

Betebenner, Damian W., Adam Vanlwaarden, Ben Domingue, and Yi Shang. 2022. SGP: Student Growth Percentiles & Percentile Growth Trajectories.sgp.io.

